

Digital Mobility Platforms and Ecosystems

State of the Art Report

Digital Mobility Platforms and Ecosystems

State of the Art Report

Editors:

Anne Faber, Florian Matthes, Felix Michel

Authors:

Project Consortium TUM Living Lab Connected Mobility

DOI: [10.14459/2016md1324021](https://doi.org/10.14459/2016md1324021)

Technical University of Munich
Arcisstraße 21
D-80333 Munich
Germany

Copyright

Copyright © 2016 Technische Universität München, sebis, Germany. All rights reserved.

No part of this publication may be reproduced, stored, archived, or transmitted in any form by any means (electrical, mechanical, by photocopying, or otherwise) without the prior printed permission of the publisher, Technische Universität München, sebis. The information contained and opinions expressed herein may be changed without prior notice.

Trademarks

All trademarks or registered trademarks are the property of their respective owners.

About TUM Living Lab Connected Mobility

To support the digital transformation in the area of Smart Mobility and Smart City, the TUM Living Lab Connected Mobility (TUM LLCM) research project was initiated, funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

The project bundles the relevant research, implementation, and innovation skills of the Technical University of Munich in the fields of informatic and transport research. The research project contributes to the design and implementation of open, provider-independent digital mobility platforms. The actual commercial implementation of these platforms is carried out by leading digital providers based on the market requirements of customer-oriented mobility solutions.

Another significant achievement of the project is the networking of already established and currently arising mobility providers, service providers, developers and users on a personal, organizational and technical level. Thus, the project contributes to the establishment of a mobility ecosystem, which is necessary for the success of the mobility platform. Thereby, smaller companies and start-ups are enabled to develop their own digital mobility services with reduced financial, organizational and technical effort.

The TUM Living Lab Connected Mobility thus simplifies and accelerates the exchange regarding the development of digital mobility services between university, industry and end-users. The university contributes to this digital ecosystem with current research findings from key areas of digital mobility platforms such as data analysis, app development, service monitoring, platform governance and efficient and legally secure integration of other partners. It draws on the established cooperation between TUM, the local industry, but also the local start-up scene to account for practical demands in the field of digital mobility platforms from the beginning.

Furthermore, the dialogue with local and regional institutions for traffic management and operations (administrations, associations, system and service operators) places a significant role in the development processes of the Living Lab Connected Mobility.



**Bayerisches Staatsministerium für
Wirtschaft und Medien, Energie
und Technologie**

ZD.B ZENTRUM
DIGITALISIERUNG.
BAYERN

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

TUM Living Lab Connected Mobility Consortium

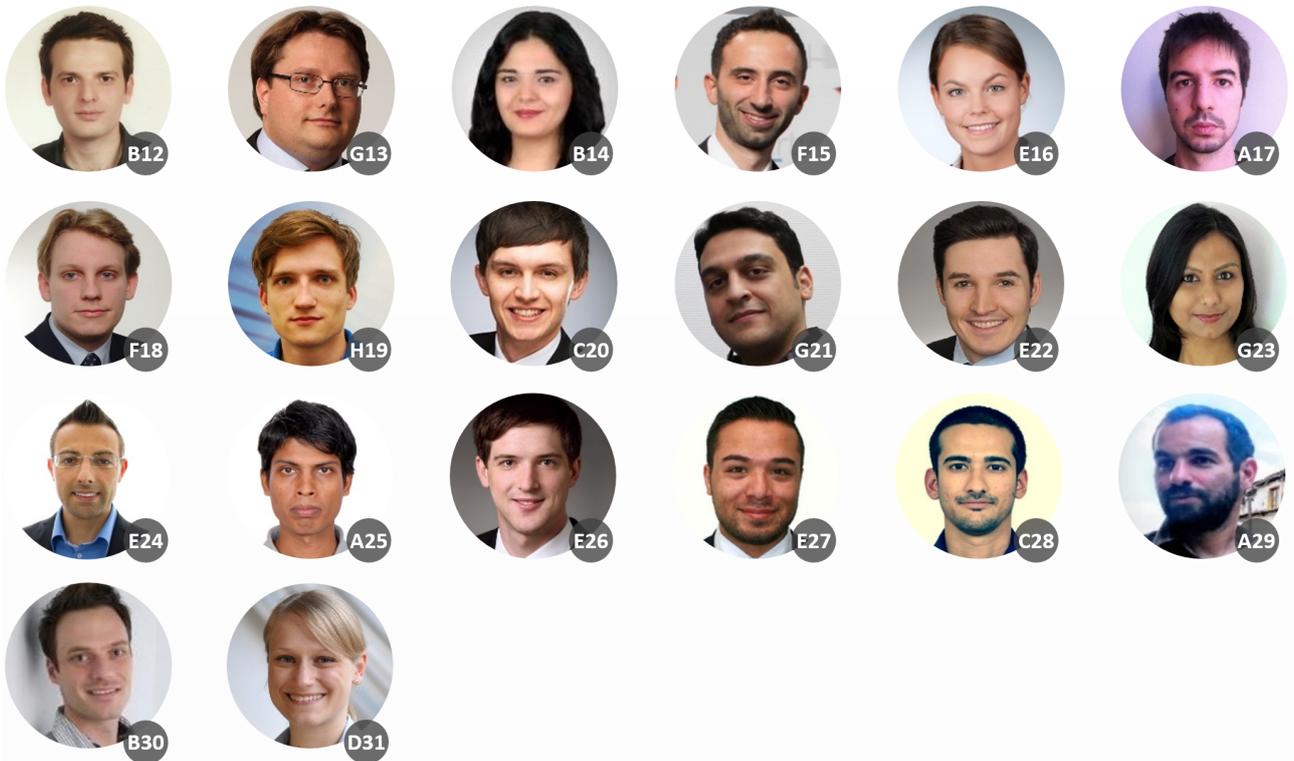
TUM is one of Europe's top universities. It is committed to excellence in research and teaching, interdisciplinary education and the active promotion of promising young scientists. The university also forges strong links with companies and scientific institutions across the world. TUM was one of the first universities in Germany to be named a University of Excellence.

The project consortium consists of seven informatic and one traffic engineering chairs of TUM.

The chairs contribute their relevant R& D competencies and results to the TUM Living Lab Connected Mobility to address the challenging open problems of mobility service integration. They do this in cooperation with industrial platform providers and platform users. Furthermore, the participating chairs activate their already established networks (project partners from industry and research, as well as graduates) for the establishment of an ecosystem, surrounding a digital mobility platform.



^{A1}Prof. Dr. Dr. h.c. Manfred Broy, ^{B2}Prof. Dr. Fritz Busch, ^{C3}Prof. Alfons Kemper, Ph.D., ^{D4}Prof. Dr. Helmut Krcmar, ^{E5}Prof. Dr. Florian Matthes, ^{F6}Prof. Dr. Jörg Ott, ^{A7}Dr. habil. Christian Prehofer, ^{G8}Prof. Dr. Alexander Pretschner, ^{H9}Prof. Dr. Johann Schlichter, ^{B10}Dr. Antonios Tsakarestos, ^{H11}Dr. Wolfgang Würndl



^{B12}Sasan Amini, ^{G13}Dr. Kristian Beckers, ^{B14}Nihan Celikkaya, ^{F15}Vittorio Cozzolino, ^{E16}Anne Faber, ^{F17}Ilias Gerostathopoulos, ^{F18}Michael Haus, ^{H19}Daniel Herzog, ^{G20}Amjad Ibrahim, ^{C21}Andreas Kipf, ^{E22}Martin Kleehaus, ^{G23}Dr. Prachi Kumari, ^{E24}Jörg Landthaler, ^{A25}Tanmaya Mahapatra, ^{E26}Felix Michel, ^{E27}Ömer Uludag, ^{C28}Varun Pandey, ^{A29}Georgios Pipelidis, ^{B30}Eftychios Papapanagiotou, ^{D31}Aenne Schweiger

^ASoftware- and Systems Engineering Research Group, ^BChair of Traffic Engineering and Control, ^CChair for Database Systems, ^DChair for Information Systems, ^EChair Software Engineering for Business Information Systems (sebis), ^FThe BMW-endowed Chair of Connected Mobility, ^GChair Software Engineering, ^HChair of Applied Informatics – Cooperative Systems

Structure of this Document

The project consortium has identified the following six key research areas for connected mobility platforms, which also provide the structure for the fifteen work packages and the structure of this report:

Platform and Ecosystem Governance. The ecosystem of platform-based service marketplaces is highly dynamic, which requires high competencies of the marketplace providers regarding governance to ensure continuous success. A general challenge for the platform operator is on the one hand, to keep sufficient control to secure the integrity of the platform, and on the other hand to provide enough freedom to enable innovation through the developers of the platform modules (Tiwana et al. 2010, 683). Examples for practice-related questions are possibilities of co-determination and mediation between end user, third-party developers and platform operators, as well as a methodology and a catalogue of measures in order to systematically increase the confidence of relevant stakeholders in the platform.

Platform Requirements, Business Models and Value Chains. There is a high demand for research to systematize and analyse existing business models and platform types in order to be able to deduce methodically and structured the demands on business models, technical architectures, technical components, management processes, interfaces, contract design and the tool support of a platform. Furthermore, there exists little secured knowledge about the necessary and beneficial metrics and measures regarding quality management and partner management.

Platform Architecture and Core Services. A central technical goal is to design a federated platform for local and geo-referenced mobility services. The architecture has to ensure the optimal interaction of the system components based on different quality criteria. These are among other things the clear division of responsibilities and decoupling, but also the guarantee of performance, efficient development and maintenance. A central design objective is safety and data protection in accordance with German and European standards and guidelines. Only selected end-user generated (sensor-) data of the platform should be made accessible and the users should have the control over their data to restrict the use of the data for specific purposes or specific mechanisms. The core services describe the differentiating value-added services of the platform, which are available for all partners through predefined interfaces and using standardized processes. They can be distinguished into generic domain-specific services and horizontal services, that are in principle relevant for all services.

Use Cases. For the success of a newly established service-platform, it is essential for the platform to provide attractive applications for the intended users and differentiated applications for the competition from the beginning. While the public space is already largely digitally charted, in the area of indoor maps it still provides a conceptual, obvious extension of the routing planning and guidance concept. So far, this is just rudimentarily implemented because of juridical and technical challenges. From a municipal or communal perspective, use cases, in which road users are specifically influenced, are desirable for environment-sensitive traffic management or risk-minimizing traffic management at major events or catastrophes to achieve a higher level goal. The users are not only individual, passive service users, it is also about the social context – for example with friends, family members and colleagues.

Geospatial-Temporal Analytics. For the purpose of the analysis and optimization of marketing, business models, user interfaces, quality of the software and data, user data is collected for longer periods and is temporally analysed with BI tools. In particularly with mobility services, the geographical reference of events and the recognition of movement patterns in the analysis is of great importance. Present algorithms and database architectures reach their limits with very large datasets, so that current worldwide database research takes places under the heading of geospatial big data exploration.

Table of Contents

Platform and Ecosystem Governance	1
Platform Requirements, Business Models and Value Chains	25
Partner On- and Offboarding	25
Crowd Sourcing and Crowd Innovation	36
Service Mashups and Developer Support	48
Platform Business Models	66
Platform Architecture and Core Services	78
Accountability	78
Multi-Layer Monitoring and Visualization	90
Sensing on Demand	111
Privacy-Preserving Proximity Services	125
Data-Driven Continuous Architecture Engineering	139
Use Cases	154
Models and Tools for Indoor-Maps	154
Eco-Sensitive Traffic Management	172
Traffic Management for Major Events	187
Collaborative and Social Mobility Services	198
Geospatial-Temporal Analytics	207
An Integration Platform for Temporal Geospatial Data	207
Geospatial Big Data Exploration	212

Platform and Ecosystem Governance

Ömer Uludağ, Stefan Hefele and Florian Matthes

Department of Informatics, Technical University of Munich, Munich
{oemer.uludag; stefan.hefele; matthes}@tum.de

Abstract

Transportation is reaching its limits in urban areas today, while world population growth and urbanization are further accelerating the difficulties caused by the wide adoption of individual motorized mobility. Public and private actors are seeking ways to enable smart solutions for future personal mobility, supported by digitalization which opens up a wide array of new possibilities. In Munich, the "TUM Living Lab Connected Mobility" (TUM LLCM) tries to research and develop a mobility platform and establish an ecosystem around it. The goal of this work is to present governance principles necessary for the establishment of such a platform ecosystem. The main results are a set of alternative platform ecosystem governance options for the strategic establishment and growth of the TUM LLCM mobility ecosystem from a governance perspective. A literature review in the areas of platform ecosystems and IT governance is conducted to define the vocabulary of platform ecosystems, their players, and interconnections. A framework is subsequently derived from literature in order to describe and analyze platform ecosystem governance in a structured manner. With the help of this framework, different successful platforms and ecosystems are analyzed and successful strategies extracted and compared. In a third step, these strategies are synthesized in order to provide two alternative platform ecosystem governance options for the platform ecosystem governance of the TUM LLCM project.

Keywords

Platform Ecosystem; Platform Governance

1. Introduction

In today's increasingly global and interconnected world, profound changes in world population are propelled by the rapid development of urbanization. The continuously spreading urbanization phenomenon has major implications on the evolution of mobility. This circumstance is exacerbated by the fact that the global urban population is projected to increase by 2.5 billion urban dwellers between 2014 and 2050. Moreover, 66% of the world's population is expected to be urban by 2050 [1, 2]. Given that the population increasingly resides in urban areas, the vehicle miles traveled (VMT) will progressively accrue in urban areas. This case is also evidenced by the development of VMT in urban areas in the USA. Beginning from 1980 with 0.86 trillion of VMT, in 2007 the VMT passed nearly the 2 trillion mark, a growth of 233.2% within 27 years [3]. In fact, this incline affects urban areas enormously, in ways such as losing productivity and competitiveness, deteriorating air quality, and increasing traffic volume and congestion [4]. The latter causes accelerating delay times of commuters in traffic and to growing fuel waste. Rising congestion is accompanied by increasing costs. In 1982, the annual costs of congestion in the USA amounted to 20.6 billion U.S. dollars, and are expected to be 175 billion U.S. dollars in 2020 [5]. Unsurprisingly, Munich is also affected by the phenomenon of rising urbanization which impedes the mobility in Munich significantly. Munich's population will grow by 15.4% until 2030 as compared to 2013, which will further increase the

number of trips per day within the city, already at about 7 million in 2015 [6]. In Munich as well, motorized individual transportation has a very high share in these trips (about 44% in 2015 [7]), leading to more and more overloaded roads. Against the backdrop of financial reasons, but also in particular due to lack of space, the growth of traffic can not be "built afterwards". Therefore, the intelligent use of data provides a welcome opportunity to improve the mobility situation in Munich noticeably. In today's digital world, there exist other ways to improve the mobility based on a better knowledge of the environment and traffic situations. This is possible by the crowdsourcing of data, by the availability of accurate maps, by the possibility of rapid and comprehensive data processing, and by the individual accessibility of road users enabled by mobile communication technologies. Such technologies have an enormous potential of providing solutions for the challenges of modern cities, especially, in terms of reducing congestion, better use of existing road networks and transportation capacities, and reduction of emissions. The respective relevant basic technologies are now available. Now, it is necessary to put them together into a platform that collects data and provides the basis for an efficient, safe, and comfortable mobility. The inter-disciplinary **TUM Living Lab Connected Mobility (TUM LLCM)** project [8] aims to exactly achieve this. Unlike a pure research project, it should provide a useful third party laboratory environment to encourage innovation based on an open platform to support and provide new mobility concepts. This work is part of the

platform and ecosystem governance sub-project of the TUM LLCM project. It aims to investigate governance processes and methods which support the subsequent operation of the mobility platform and the controlled evolution of the ecosystem. The objective of this work¹ is to perform a state of the art analysis of practice and literature by elaborating necessary governance principles and providing a broad analysis and description of platform ecosystem governance options for governing a mobility platform and ecosystem. This leads to following research questions:

1. How can platforms, their players, and interconnections be characterized according to existing literature?
2. What have been factors for the successful establishment of platform businesses in the past?
3. Which design and governance options do exist to successfully establish a mobility platform and ecosystem?

In order to answer those research questions, the remainder of this work is organized as follows: Section 2 summarizes related work in the field of platform ecosystem and provides the first answer for the first research question by clarifying the terms of "platform", "platform ecosystems", and related terms such as "components" and "actors". Section 3 presents a framework for the analysis of platform and platform ecosystem governance and thus completing the answer for the first research question. Section 4 answers the second research question by utilizing the platform governance framework, by analyzing mobility-related platform ecosystems, and by comparing the results of the analysis for extracting successful strategies and patterns. Section 5 combines successful strategies from practice to recommend promising avenues for the sustainable establishment of the TUM LLCM mobility platform ecosystem and thus answering the third research question. Last but not least, Section 6 summarizes the key findings of this work, reveals its limitations, and discusses themes for further research.

2. State of the Art Literature Review

This section comprises the state of the art analysis on literature and research streams concerning platforms and ecosystems. Accordingly, Subsection 2.1 reveals the underlying design of the literature review. Following this, Subsection 2.2 provides the results of the literature review. Finally, Subsection 2.3 defines relevant terminologies for the remainder of this work.

2.1 Design of Literature Review

In this work, we looked for publications that (a) focus on the platform ecosystem as unit of analysis and (b) derive insights on how to govern platform ecosystems. We screened relevant outlets on the guidelines by Webster and Watson [10]. Accordingly, we use three steps to identify relevant literature:

¹This work comprises state of the art analysis of platform and ecosystem governance in practice and literature (for more detailed information see [9]).

1. identifying major contributions in leading journals, backward search, and forward search,
2. uncovering other relevant literature by analyzing the sources of major contributions found in journal databases and tables of content, and
3. analyzing citations of the literature found in other literature.

Departing from the first research question, the following terms were initially used for identifying relevant literature: (1) Platform, (2) Platform Ecosystem, (3) Complementary Software Application, (4) (Platform) Provider, (5) (Platform) User, (6) Platform Architecture, and (7) Platform Governance. In the initial step of the search process, several databases and search services were consulted: (1) Google Scholar², (2) AIS Electronic Library³, (3) Business Source Premier⁴, (4) Web of Knowledge⁵, (5) ScienceDirect⁶, (6) ACM Digital Library⁷, and (7) IEEE Xplore⁸. After gaining an initial overview over the topic, the search terms were set to the following: **platform, software(-based) platform, platform ecosystem, platform governance, two-sided market, and IT value co(-)creation**. The last two search terms were added after reading the first few research papers revealing that many authors consider software-based platforms or ecosystems as being two-sided markets or leading to the "cocreation" of IT value. The results of the first research phase were analyzed and added to the body of literature if their content seemed to be relevant for the aim of this review. Relevance was determined by reading titles, abstracts, and summaries of potential relevant results. Subsequently, the included contributions were analyzed and classified into a scheme that comprises basic information of the contributions such as type of research, key words, content, and definitions for the terms mentioned above. If analyzed contributions did not provide own definitions but cited those of other scholars, then they were taken as a starting point for the backward search. Contributions with definitions by the authors themselves served as basis for the forward search. Finally, this resulted in a set of definitions for each term. These were analyzed about their differences and commonalities and then merged into a final definition (see Subsection 2.3). Table 1 presents the number of analyzed contributions. Plotting the data over time reveals that the number of studies on platforms has increased over the last decade (see Figure 1). By summarizing the insights along the search terms, we can carve out the focal points of existing research and build a state of the art understanding of platform ecosystem governance.

²<http://scholar.google.de>

³<http://aisel.aisnet.org/do/search/advanced>

⁴<http://search.ebscohost.com>

⁵<http://apps.webofknowledge.com>

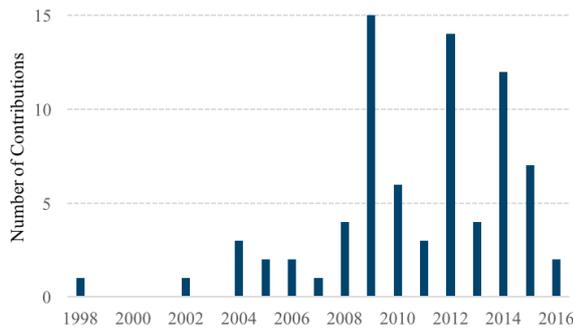
⁶<http://www.sciencedirect.com>

⁷<http://dl.acm.org/advsearch.cfm>

⁸<http://ieeexplore.ieee.org/Xplore/home.jsp>

Table 1. Number of Analyzed Contributions

Type of contribution	Analyzed
Books / book chapters	18
Conference articles	11
Journal articles	30
Other	2
Total	61

**Figure 1.** Number of Studies on Platforms Over Time

2.2 Research on Platform Ecosystem

According to Schreieck et al. [11], studies have a different understanding of the platform as unit of analysis. Therefore, there are different perspectives and research streams on platform ecosystems. Baldwin and Woodard [12] attempt to define the term "platform". They identify three major research streams, which are: product development, technology strategy, and industrial economy. Also, Ghazawneh and Henfridsson [13] identify product development as a research stream. Parker and Van Alstyne [14] include all three research streams in their definition of a platform.

Baldwin and Woodard's [12] perspective on platforms

The first research stream brought up the notion of product platforms. In this context, the platform is the basis from which different products could be derived by modifying features. An exemplary definition is provided by Wheelwright and Clark [15], who state that platform products are products that "meet the needs of a core group of customers but [are designed] for easy modification into derivatives through the addition, substitution, or removal of features". For instance, consider Intel's 80486 microprocessor. It introduced a number of performance improvements and provided an easy migration path for existing customers. Over the life of the 486 platform, Intel has introduced a host of derivative products, each offering some variation in speed, cost, and performance and each able to leverage the process and product innovations of the original platform [15]. In contrast, Meyer and Lehnerd [16] define product platform as "a set of subsystems and interfaces that form a common structure from which a stream of derivative products can be efficiently developed and produced". Thus, this concept focuses on reducing product development costs by creating a common product basis from which product in-

stances can be derived leading to the combination of "scale economics and product differentiation at the same time" [13]. The second research stream defines platforms as "valuable points of control (and rent extraction) in an industry" [12]. Scholars of this stream examine platforms that are at the center of whole industries (like the computer industry [17]), or parts of it, like web browsers or chipsets ([18, 19, 20]). These platforms are no longer internal to one company, but are "developed by one or several firms, and [...] serve as foundations upon which other firms can build complementary products, services or technologies" [21].

Finally, the third stream of research by industrial economists extends this view of industry platforms putting the emphasis on network effects that arise on such platforms with two or more groups of agents, making them "multi-sided" ([12, 22]). Network effects arise when the platform mediates transactions between those user groups that would otherwise not have been possible or at least very expensive. Additionally, the presence of one group of users makes the platform more valuable to the other side, and vice versa. This is caused by positive indirect network effects, also called "positive feedback", which consists of cross-group network effects [23]. Hagiu and Wright [24] define cross-group network effect as follows: "a cross-group network effect arises if the benefit to users in at least one group (side A) depends on the number of other users in the other group (side B). An indirect network effect arises if there are cross-group network effects in both directions (from A to B and from B to A)". Indirect network effects can be also negative. Furthermore, there are also same-side (direct network effects), which can either be positive or negative. For instance, the more users a road has, the less useful it gets to each of them [25]. Rochet and Tirole [26] define the characteristics of multi-sided platforms as "products, services, firms or institutions that mediate transactions between two or more groups of agents". However, a market (used synonymously for platform here) is only defined as two sided, "if the platform can affect the volume of transactions by charging more to one side of the market and reducing the price paid by the other side by an equal amount; in other words, the price structure matters, and platforms must design it so as to bring both sides on board" [22]. Dimensions other than pricing have to be considered as well, such as regulating terms of transactions between users, control of users in other ways, and monitoring of intra-side competition [22]. Table 2 classifies the analyzed literature in a concept matrix according to the platform research stream it represents. Some authors, in a first attempt to consolidate existing definitions, utilize aspects of two or even all three streams to define their understanding of a platform⁹.

Gawer's [46] perspective on platforms

In contrast to the previous perspective, Gawer [46] uses instead two categories to classify the literature on platforms, namely engineering design and economics. Gawer argues that platforms have been viewed as technological architectures

⁹The classification is carried out according to Baldwin and Woodard's [12] perspective on platforms

Table 2. Different Streams of Platform Research

Article	Platform research streams		
	Product Development	Technology Strategy	Industrial Economy
Bakos and Katsamakas [27]			X
Baldwin and Woodard [12]	X	X	X
Basole and Karla [28]		X	
Boudreau [29]	X		
Boudreau and Haigu [30]		X	X
Ceccagnoli et al. [31]	X		
Cusumano and Gawer [18]		X	
Cusumano [32]		X	X
Economides and Katsamakas [33]			X
Eisenmann et al. [34]			X
Eisenmann et al. [35]		X	X
Evans [25]			X
Evans and Schmalensee [36]			X
Gawer [21]		X	
Gawer and Cusumano [20]		X	
Greenstein [17]		X	
Hidding et al. [37]		X	X
Le Masson et al. [38]		X	
Parker and Alstyne [14]	X	X	X
Rochet and Tirole [22]			X
Rochet and Tirole [39]			X
Scholten and Scholten [40]		X	
Suarez and Cusumano [41]	X		
Suarez and Kirtley [42]		X	X
Tatsumoto et al. [43]	X		
Tiwana et al. [44]		X	
Tiwana [45]		X	X
Wheelwright and Clark [15]	X		

within the former, as markets in the latter research category, which comes with limitations in both streams. "Bridging" the differences between both, Gawer proposes a unified framework by defining platforms as "evolving organizations or meta-organizations that: (1) federate and coordinate constitutive agents who can innovate and compete, (2) create value by generating and harnessing economies of scope in supply or/and in demand, and (3) entail a modular technological architecture composed of a core and a periphery" [46]. Gawer also postulates that platforms can be sorted into a continuum of three types of platforms: internal platform, supply-chain platform, and industry platform (see Figure 2) [21].

Gawer and Cusumano's [47] perspective on platforms

In another paper, Gawer and Cusumano [47] propose a slightly different classification by dividing platform research into two categories, namely internal and external platforms. Internal platforms comprise what Baldwin and Woodard call "product platform", but also the special case of a supply chain platform that is not entirely internal, but serves for the production of a family of products of only one firm with the help of its

suppliers [47]. By contrast, external platforms are defined as "products, services, or technologies developed by one or more firms, which serve as foundations upon which a larger number of firms can build further complementary innovations and potentially generate network effects" [47]. This definition largely corresponds to an aggregation of the research streams "technology strategy" and "industrial economics". The difference between those two consists in the fact that not all technological industry platforms generate network effects whereas not all two-sided markets (with network effects) are technological platforms.

Manikas et al.'s [48] perspective on platforms

The search process reveals also another stream of research, namely the field of software ecosystems. Surprisingly, this stream remained nearly uncovered using the search terms and databases described above. However, taking the systematic literature review by Manikas and Hansen [48] as a starting point, it becomes clear that their understanding of (software-based) ecosystems is fundamentally the same. Therefore, this work relies on the work by Manikas and Hansen to integrate the

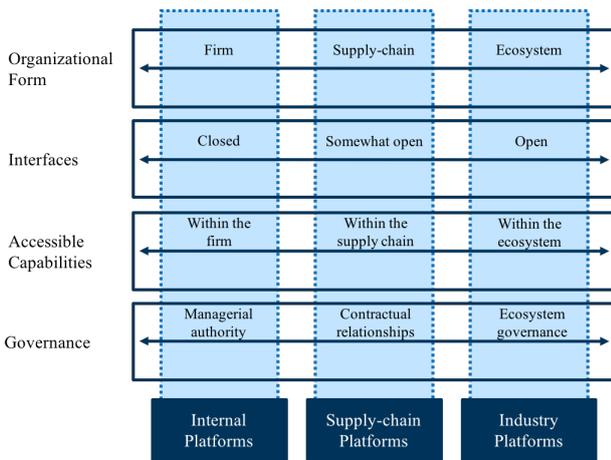


Figure 2. Organizational Continuum of Technological Platforms, according to Gawer [46]

software ecosystems stream, and does not conduct a second literature review on software ecosystems. Table 3 provides an overview of contributions which define software ecosystems and related terms. For the remainder of this work, we will use the consolidated definition of a software ecosystem by Manikas and Hansen [48] which is defined as follows: "we define a software ecosystem as the interaction of a set of actors on top of a common technological platform that results in a number of software solutions or services. Each actor is motivated by a set of interests or business models and connected to the rest of the actors and the ecosystem as a whole with symbiotic relationships, while, the technological platform is structured in a way that allows the involvement and contribution of the different actors". Figure 3 illustrates the different classifications of platforms that were mentioned above¹⁰.

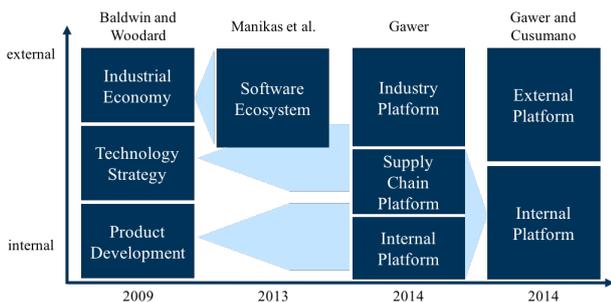


Figure 3. Comparison of Different Platform Classification Schemes and their Overappings, based on [12, 46, 47, 48]

Definitions of Platform Terms in Literature

Finally, the analyzed research streams also provide definitions of platform related terms. For instance, Bakos and

¹⁰The overlappings of definitions are illustrated by the light blue arrows. For instance, Gawer’s [46] perspective of supply chain platforms and internal platforms can be categorized as internal platforms in Gawer and Cusumano [47].

Katsamakos [27] state that "a two-sided Internet platform embodies a design, which defines the architecture of the services offered and the infrastructure that facilitates the interaction between the participating sides, and a set of rules, such as pricing terms and the rights and obligations of the participants". According to this definition, a platform consists of its architecture and its governance. This is also what Tiwana [45] suggests for characterizing the constituents of platforms. A platform ecosystem is mostly defined as consisting of the platform, secondary software applications, and the interacting actors. While Tiwana only names platform and secondary software applications, Scholten and Scholten [40] also integrate different actors into their definition: "the platform ecosystem embraces (a) the platform provider, operating the platform and core platform offerings as well as mediating between service consumers and platform providers; (b) the service ecosystem of complementary product and service providers enabling the 'whole' customized solution as offered to (c) the customers". While describing the Salesforce.com ecosystem, Baek et al. [55] tap into the same direction by stating that "the ecosystem consists of a platform provider (Salesforce.com), and the platform users. A platform user is categorized into a developer (i.e. a user engaged in the application development) and a customer (i.e. a user consuming the application created by developers)". Mostly, secondary software applications are not defined specifically. Thus, concrete definitions are only provided by Tiwana et al. [44] and Tiwana [45]: "an add-on software subsystem or service that connects to the platform to add functionality to it". Table 4 provides an overview of all sources which define platform and/or platform related terms.

Definition of Governance Terms in Literature

The term "governance" has many different and sometimes contradictory meanings [60]. In a broad sense, governance can be understood as the "establishment of policies, and continuous monitoring of their proper implementation, by the members of the governing body of an organization" [61]. Within the context of a company, the term "corporate governance" is defined as "the framework of rules and practices by which a board of directors ensures accountability, fairness, and transparency in a company’s relationship with its all stakeholders (financiers, customers, management, employees, government, and the community)" [62]. Finally, on the level of IT, Weill [63] provides a definition that is adopted also in a deep literature review of the IT governance field by Brown and Grant [64] by stating that IT governance represents "the framework for decision rights and accountabilities to encourage desirable behavior in the use of IT".

However, these definitions do not seem to be well suited in the context of a platform and its ecosystem as they focus on the "use of IT" within the boundaries of one company. This is not the case if the platform is intended to draw on network effects for platform and ecosystems. Therefore, a new term has to be defined drawing on existing literature. A first, very simplistic definition is provided by Tiwana et al. [44] who define platform governance as "who makes what

Table 3. Definitions of Software Ecosystems [48]

Software ecosystem definition	Corresponding platform terms					
	Platform	Complementary Software Application	(Platform) Provider	(Platform) User	Platform Architecture	Platform Governance
Messerschmitt and Szyperski [49]		X				
Jansen et al. [50]	X	X			X	X
Bosch 2009 [51]	X		X	X		
Bosch et al. [52, 53]	X	X	X	X		
Lungo et al. [54]		X				
Manikas et al. [48]	X	X	X	X	X	X

decisions about a platform”. This is further detailed by stating that “a platform’s governance design can be studied from three distinctive perspectives: (a) decision rights partitioning, (b) control, and (c) proprietary vs. shared ownership” [44]. Tiwana [45] refines the previous perspectives, now arguing that the dimensions of platform governance are: (a) decision rights partitioning, (b) control, and (c) pricing policies. The replacement of ownership structure with pricing is not further justified by Tiwana. However, the both dimensions ownership and pricing policies seem to play important roles for platform governance at first sight. Thus, the author concludes that an own definition of platform governance should not only cover a set union of the four dimensions mentioned, but also extend the pricing domain to other business related aspects. The definition by Tiwana forms the basis for many other scholars’ works (see [46, 13, 59]). For this reason, it provides a good foundation for the following framework. This assumption is further supported by the fact that Hein et al.’s [65] framework for platform and ecosystem governance shows a great conformity with Tiwana’s framework. Although Hein et al. categorize elements of governance into more domains than Tiwana’s framework, all factors, namely “governance structure”, “resources and documentation”, “accessibility and control”, “trust and perceived risk”, “pricing”, and “external relationships” of the former can be also represented in Tiwana’s framework.

2.3 Delineation of Related Terminologies

After describing the different research streams and providing an overview of existing definitions in the previous subsection, these definitions were analyzed about their differences and commonalities and then merged into a final definition for the

scope of this work, which are presented in the following:

- **Platform:** We define a software-based platform as the core of a digital multi-sided market. Within such markets, the volume of transactions characteristically not only depends on the overall platform fees, but also on the balance of their allocation to the different (market) sides [39]. The core functionality is extensible, reusable, and provides stable interfaces (architecture) and other rules for interaction (governance). A platform provider makes the platform available to secondary developers and customers (end-users).
- **Platform Ecosystem:** A platform ecosystem consists of the platform, secondary applications developed for it, the actors providing, extending, and using the platform and applications as well as their interactions and the effects of these interactions [40, 55].
- **Complementary Software Applications:** A complementary software application is a “software subsystem or service that connects to the platform to add functionality to it” [45].
- **(Platform) Provider:** The platform provider creates, configures, and makes the platform available to users. Configuring involves building the architecture and interfaces as well as implementing the governance [40, 45].
- **(Platform) User:** Users are ecosystem actors who are not directly involved in providing or sponsoring the platform. Developers are users who extend the platform’s core functionality by adding complementary software applications. Customers are end users who customize

Table 4. Definitions of Platform Terms in Literature

Articles and Authors	Terms defined					
	Platform	Platform Ecosystem	Secondary Developer	Customer	Platform Architecture	Platform Governance
Baek et al. [55]	X		X	X		
Bakos and Katsamakas [27]	X					
Baldwin and Woodard [12]	X				X	
Basole and Karla [56]	X					
Boudreau [29]	X					
Boudreau and Hagiu [30]	X					
Ceccagnoli et al. [31]	X	X				
Cusumano [32]	X	X	X			
Cusumano and Gawer [18]					X	
Eisenmann et al. [57]	X					X
Eisenmann et al. [35]	X		X	X		X
Evans [25]	X			X		
Evans and Schmalensee [36]	X					
Gawer [20]	X					
Gawer and Cusumano [47]	X		X			
Greenstein [17]	X					
Hidding et al. [37]	X					
Jansen and Cusumano [58]	X	X				X
Le Masson et al. [38]	X					
Manner et al. [59]						X
Parker and van Alstyne [14]	X					
Rochet and Tirole [22]	X					
Scholten and Scholten [40]	X	X				
Suarez and Cusumano [41]	X					
Tatsumoto et al. [43]	X					
Tiwana [45]	X	X	X	X	X	X
Tiwana et al. [44]	X	X			X	X

the platform and its complementary software applications by "mix-and-match" [45] to meet their specific needs. Developers can simultaneously act as customers and vice versa.

- **(Platform) Architecture:** A platform’s architecture is comprised of high-level design rules for the platform itself as well as interfaces that specify how secondary software applications can interact with it [45, 18].
- **(Platform) Governance:** Governance essentially defines who decides what about a platform (ecosystem) ([44], [45]). We distinguish between platform governance and platform ecosystem governance. Platform governance comprises two dimensions: decision rights partitioning and the internal structure of the platform provider. Governance of the platform ecosystem additionally requires control and pricing strategies aimed at customers and secondary developers.
- **(Ecosystem) Governance:** Ecosystem governance comprises governance structures and activities that try to exert influence or deal with actors and systems other

than the platform. The main difference between platform and ecosystem governance lies in the fact that secondary actors cannot be directly controlled by the platform owner via hierarchical power or authority [45].

Figure 4 illustrates the coherences of platform ecosystem components.

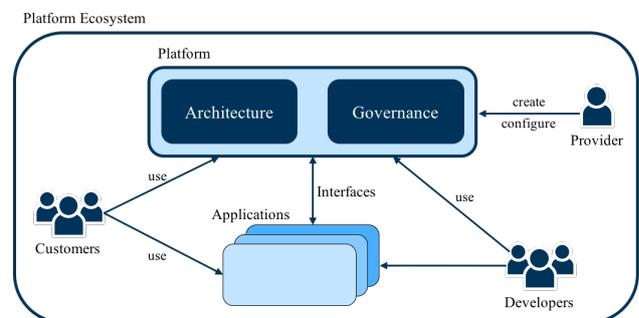


Figure 4. Visualization of a Platform Ecosystem with its Actors

3. Platform and Ecosystem Governance Framework

This section presents a framework for the analysis of platform and ecosystem governance, which provides a tool for subsequent analysis of successful platform and platforms ecosystems. This framework is described in Subsection 3.1 in detail.

3.1 Framework

Based on the definitions provided by Tiwana [44, 45], the framework for analyzing existing platforms' and ecosystems' governance is described along the dimensions: "platform provider structure", "decision rights partitioning", "business", and "control" (see Figure 5) in the following.



Figure 5. Overview of Platform and Ecosystem Governance, based on [44, 45]

Platform Provider Structure

The structure of the platform provider is concerned with the platform provider's organization, i.e. whether the provider consists of a single company or several. Secondly, the general characteristics of the platform provider may have its impact: typically, a start-up, whose whole existence depends on the platform will act differently than an incumbent, for whom the platform is only one source of revenue out of many. Following Tiwana [45], the main characteristics of the provider structure are: age, size, number of employees, and whether the provider is a start-up or an incumbent.

Decision Rights Partitioning

The division of decision rights between platform provider and secondary developers consists of three areas of decision rights: the platform itself, the platforms interfaces, and the secondary applications. For each of those, decision rights can be assigned differently between the two stakeholders: either completely residing on one side or taking the place "somewhere in the middle", which means that both sides have some sort of influence on the decisions to be taken in the domain. Our framework adopts the "decision rights partitioning framework" by Tiwana [45] who introduces four classes of decision rights: platform and application decision rights, each divided into strategic and implementation-related rights.

Business

In a multi-sided market environment, pricing was long time considered as the only possible governance in an ecosystem [46]. Therefore, not only pricing and its structure (that is the balance of allocating the charges to the different sides of platform [39]) is considered here, but also other business

aspects, namely the general business model of the platform, the strategy to achieve it, the overall market structure the platform and its ecosystem are placed within, and the relations with the ecosystem's secondary developers. These relations cover incentives, API documentation, personal assistance, mailing lists, and support forums.

Control

The fourth domain of the framework contains control measures to be applied by the platform provider towards secondary developers. Control measures have been combined from different papers in literature: Tiwana [45] names "gatekeeping", "process control", "output control", and "social control". Gatekeeping refers to a sort of "access control" to the ecosystem. Thereby, the platform provider decides whether he wants to admit certain secondary developers to the ecosystem or not. As a result, some ecosystems are considered to be more "open" than others, e.g. Google's Android vs. Apple's iOS. Secondly, process control denotes whether the platform provider insists on the usage of certain methods, e.g. agile programming frameworks like Scrum, when creating complementary software applications. Thirdly, output control, or respectively quality control, means the measures taken by platform providers to ensure quality and desirable characteristics of secondary developer's outputs. Therefore, from the ecosystem's perspective, it can also be called "input control". Besides gatekeeping, this form of control largely contributes to the perception of an "open" or "closed" ecosystem. Finally, social control refers to an "informal control" that builds on common values and beliefs that platform provider and secondary developers share and that are used by the provider to influence and guide their behavior [44, 45]. Additional to these four types of control, two other types have been added by other authors: Rochet and Tirole [22] introduce the notion of "regulation of transactions" by the platform provider. Therewith, the provider exerts control over secondary developers interactions with customers. They provide three different measures to achieve this: by regulating prices, by acting as a licensing authority, and by acting as a competition authority. Finally, Scholten and Scholten [40] mention sanctional control, defined as the "coercive action up the exclusion of services or service providers".

Dependencies Between Domains

Having explained the different dimensions of the framework, dependencies between those have to be elaborated (see Figure 6). Firstly, the structure of the platform provider might influence the division of decision rights between the platform provider and the other ecosystem members. One could consider that a platform provider consisting of different entities would be more inclined to share decision rights also with secondary developers, as there exists a logic of collaboration already within the provider. Secondly, The structure of the provider might also have an influence on business domain of the framework. This is mainly because a single provider with a single business model might want to focus on different aspects,

e.g. types of users or secondary applications that especially suit his needs, whereas a group of actors forming the provider will decide for a more balanced approach, requiring some sort of alignment of their interests beforehand. Thirdly, the control portfolio established by the platform provider is dependent on the decision rights that provider and secondary developers have. The platform provider can only control areas where it can exert the corresponding decision rights. For instance, if the platform provider states that the internal architecture of secondary applications do not affect their admittance or refusal to the ecosystem, it will not be able to justify the refusal of applications for that reason. Transparency and consistency between those two domains is of great importance.

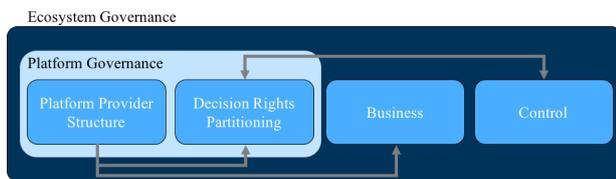


Figure 6. Dependencies Between Parts of the Governance Framework, based on [44, 45]

Template of the Governance Framework

The final framework with detailed dimensions is presented in Figure 7. While the platform provider structure and the business domain will be filled with textual content, decision rights partitioning and control can be easier displayed figuratively. Decision rights can either be centralized (arrows pointing to the middle) or decentralized (arrows pointing outwards), control can be used extensively (depicted by a filled Harvey Ball), to some extent (half-filled Harvey Ball), or not at all (empty Harvey Ball). This template serves as a foundation for analyzing the governance of mobility-related platforms and ecosystems in Section 4 and providing platform ecosystem governance options for the TUM LLCM project in Section 5.

4. Analysis of Mobility-Related Platforms and Ecosystems

In this section, existing mobility-related platforms and platform ecosystems will be analyzed regarding their governance. This will be accomplished by utilizing the framework elaborated in the previous section. The remainder of this chapter is organized as follows: Subsection 4.1 describes the underlying approach of the analysis. Subsequently, the results of the analysis of four different platforms and platform ecosystems are presented in four subsections: Waze (see Subsection 4.2), Moovit (see Subsection 4.3), Apple (see Subsection 4.4), and ITS Factory (see Subsection 4.5). Subsection 4.6 compares the results of the analysis and extracts successful strategies.

4.1 Approach

For the analysis of the platform ecosystems, relevant information was gathered by different types of sources, e.g. technical

literature, academic paper, press releases, and developer documentation. The selection criteria of potential relevant platforms for this work are based on two aspects: (1) platforms in the selection should have an ecosystem according to the definition in Subsection 2.3 and (2) platforms should operate in a mobility-related context. Table 5 provides an overview of the platforms considered before selection.

Alibaba is not selected for the later analysis since it is neither in a mobility-related context nor it provides a real ecosystem of secondary contributions around it. Furthermore, several other platforms are not in a mobility-related context which why only Apple iOS is selected as an appropriate candidate since it is the "picture perfect" example for an ecosystem. Of all public and city-related mobility platform projects, only ITS Factory is selected since it is the only one clearly building upon secondary developers from the beginning on. This is not the case, or respectively yet unclear, for other mobility-related ecosystems like Moovel, smile, Kansas City Living Lab, or Ubiquitous Mobility for Portland. ITS Factory, from now on ITS Finland, fulfills both criteria and is appropriate for further analysis. The remaining mobility-related platforms Moovit and Waze are selected since they provide detailed evidence on the topic of mobility platform. In the following subsections, Waze, Moovit, Apple, and ITS Factory are analyzed.

4.2 Waze

Waze¹¹ provides an app for smartphones that enables step-by-step navigation and real-time traffic information (RTTI). Traffic information is based on crowdsourced data. This data comprises movement data of Waze users in certain areas as well as their manually supplied additional data such as traffic density, road construction works, or police/radar controls. By utilizing this data, Waze detects congestions on the route and suggests alternatives. Waze users can also register themselves as map editors to add missing information and rectify altered design and lay-out of roads. Additionally, Waze provides traffic data to public entities and broadcasters.

Platform Provider Structure

Waze was founded in 2008 by Ehud Shabtai, Uri Levine, and Amir Shinar in Ra'anana, Israel. After raising \$67M in three rounds of funding [66], it was acquired by Google in 2013 for approximately \$966M [67]. However, it still continues to operate as a single entity today. Currently, Waze has about 120-200 employees in Ra'anana and Palo Alto [68, 69].

Decision Rights Partitioning

The decision rights about the Waze platform, its APIs, and Transport SDK remain completely with Waze. There are no possibilities for secondary developers to add own features to the platform. This is the reason why the platform and secondary applications are completely decoupled and secondary application developers possess the decision rights over their applications. Waze's privacy policy states that all information about the user is connected to his user name or device (if not

¹¹<http://www.waze.com/en>

Platform provider structure	<ul style="list-style-type: none"> Number of employees Age Size Type 	[textual information]
Decision rights partitioning	<ul style="list-style-type: none"> Platform strategic decision rights Platform implementation decision rights Application strategic decision rights Application implementation decision rights 	 or 
Business	<ul style="list-style-type: none"> Market structure Business plan Strategy Pricing structure Incentives for development Support for development 	[textual information]
Control	<ul style="list-style-type: none"> Gatekeeping Regulatory activities Process control Output control Social control Sanctional control 	  for each control 

Legend:

	centralized decision rights		No control
	decentralized decision rights		Some control
	divided/contested decision rights		Extensive control

Figure 7. Template of the Platform and Ecosystem Governance Analysis Framework

registered), aggregated, and can be utilized for many purposes, including sharing with other affiliated companies [70].

Business

The market for Waze’s mobile application is partly an existing one and partly a new one. The existing market is the one for route guidance systems with turn-by-turn directions. This market can be considered as mature. Thus, Waze cannot differentiate itself except for the aspect of crowdsourcing. Waze’s business plan relies on extracting revenue from cross-side network effects by selling hyperlocal advertisements to local businesses who can target users nearby [71]. Additionally to those users, there are relations to broadcasters and public entities, to whom traffic data is made available, e.g. broadcasters like radio and TV stations can use the data for their traffic service [72]. Information is also provided to public entities like cities or counties in exchange for other sources of real-time information not yet present on Waze [73, 74]. Waze aims to combine users and broadcasters into a virtuous cycle where more users lead to better data quality, which in turn attracts more users and broadcasters. Also, public entities are given the data for free. As a side effect, Waze establishes close relations to administrations and governments [74]. Waze prices only advertisers monetarily. All other sides “pay” with either their personal data (users) or data available to them (public entities), or by making the application known to a broader public (broadcasters). The exact pricing structure

towards the side of advertisers is not well-known. However, some characteristics are visible from Waze’s website [75]. Advertisers can define a budget that specifies how much they are willing to pay for each 1.000 impressions (cost-per-mile or cost-per-thousand, see [76, 77]). Additionally, they can define a monthly maximum budget which should not be exceeded. Advertisers pay only if their advertisement is displayed on the map; the minimum budget per month amounts \$50. Waze uses the level of “Game interface design patterns” [78] and enables users to collect points by many different mechanisms, e.g. by providing additional information such as reporting road-related information like traffic jams. Waze provides all forms of support to its developers, map editors, and users. There is a “Waze Help Center”, aimed primarily at users [79] as well as a “Community Wiki”, aimed at map editors. For developers, Waze provides for its Transport SDK a respective documentation. There is also a community forum where users and map editors can discuss their questions and receive announcements from Waze employees. Furthermore, there are tutorial videos and a “tryout” editor which enables to test new features without submitting the results into the production environment.

Control

As Waze is based on crowdsourcing, gatekeeping is as low as possible for prospective users. In order to use Waze, users only have to download the app, which is instantly usable with-

Table 5. Overview of Platform Businesses According to Selection Criteria

Platforms	Selection Criteria	
	Mobility-related	Platform Ecosystem
Alibaba		
Ally	X	X
Amazon		X
Apple iOS		X
Apple iTunes		X
eBay		X
Facebook		X
Future Urban Mobility (MIT/Singapore)	X	
Google Search Engine		X
ITS Factory (Tampere, Finland)	X	X
Kansas City Living Lab (Kansas, USA)	X	X
Microsoft Windows		X
Moovel	X	
Moovit	X	
Mozilla		X
SAP		X
smile (Vienna, Austria)	X	
Ubiquitous Mobility for Portland (Portland, USA)	X	X
Waze	X	

out any further steps necessary. For broadcasters, registration is more complex, as they have to fill out a "Partner Interest Form", accept the online agreement, after which Waze initiates personal contact with the prospective broadcasting partner [72]. Similarly, prospective advertisers have to fill out a form in order to get in contact with Waze.

Without possibilities for secondary developers of extending the platform, there is no need for regulatory activities. Regulatory activities towards advertisers are not known.

There is no process control in the sense that the development process of secondary applications is regulated due to the fact that platform and applications are completely decoupled. Thus, secondary applications have only to fulfill the specifications set by the API and the Transport SDK. The same reasoning applies to output control or metrics.

The element of social control is similarly not applied towards developers. However, map editors can be considered being under some sort of social or relational control. Map editors share the common vision of "outsmarting traffic" with Waze, and thus can be influenced in their behavior [44].

Finally, there is only "straight forward" sanctional control towards secondary developers and map editors. Technical abilities allowing to exclude certain secondary applications from connecting to the API or using the Transport SDK are presumably available at Waze. For map editors, a simple deletion or blocking of the user account is sufficient in order to prevent users from accessing the platform.

4.3 Moovit

Moovit¹² provides an app for smartphones which intends to improve the use of public transportation. Users can plan their travel with public transportation and other selected mobility services. Similar to Waze, Moovit relies on crowdsourced data to add real-time information about delays, cancellations, and other characteristics of individual trains or buses, e.g. crowdedness or cleanliness. Moovit also enables users to register as editors in order to alter or add lines, routes, and timetables. Moovit does not generate any revenue yet, but plans to do so in the future by location-based advertising and commissions from transit agencies [80].

Platform Provider Structure

Moovit was founded in 2012 by Nir Erez, Roy Bick, and Yaron Evron in Tel Aviv, Israel. It has raised \$81.5M in three stages and launched its app worldwide in 2013 [81]. The last round of financing included partners like Nokia Growth Partners, BRM Capital, but also BMW i Ventures [82, 83]. Additionally, it received an undisclosed amount of investment from Sound Ventures in 2015 [84]. It has not been acquired yet and thus remains a single entity. Currently, Moovit has about 60 employees in San Francisco and Tel Aviv [81].

Decision Rights Partitioning

Decisions rights between Moovit and secondary developers are restricted to the offered API. As developers have no influence on design and implementation of this interface, all strategically and implementation-related decision rights about the platform lie within Moovit. Moovit does not influence de-

¹²<http://moovitapp.com>

cisions of secondary application developers about strategy or implementation of their application as platform and secondary applications are decoupled.

Business

The market for applications that facilitate public transit planning used to be exclusive to the public transit providers, e.g. "Münchener Verkehrs- und Tarifverbund" (MVV) in Munich. Because of this, many applications are restricted to one specific city or region that corresponds to the region the provider is operating in. There are of course examples of country-wide applications, e.g. "DB Navigator" or "moovel". However, Moovit is the only one that highly relies on crowdsourcing in order to provide its information to users.

Moovit's business model is not clearly visible yet. According to interviews with its founders, Moovit has gathered "enough money for the next few years", so that the application can be developed further and expand into more countries and cities without having to be profitable [85, 80, 86]. In long-term, Moovit wants to generate profit from five business areas: integration of taxi services, cooperation with transit agencies in mobile ticketing, cooperation with carsharing, carpooling providers, and cooperation with other businesses.

Moovit's strategy is similar to Waze's. In the case of Moovit, areas without an active community and no schedule data available due to inability or reluctance of the transit provider can hardly be added to the list of supported areas. Thus, gaining more users around the world is vital for Moovit. In trying to attract users, it can rely on the same "same-side network effects" as Waze: more users in an area produce more crowd-sourced data, which in turn makes the app more useful for others. Having acquired a large user base, Moovit intends to acquire partners among taxi services, transit agencies, and carsharing providers, as well as sell advertisements and additional services to local businesses.

For users, gamification plays a large role also in Moovit. Users get points and derived ranks for reporting delays, crowdedness, and temperature inside the vehicle or friendliness of the bus driver, to name a few. Additional incentive for editors is the "Community Spotlight", where particularly active editors are presented to the community within Moovit's blog.

Support for secondary developers is as limited as the possibilities of integrating Moovit into own projects are: the only existing documentation components are the "Deeplinking Docs" [87] that explain how developers can link to Moovit's app by the methods described above. For users and map editors, there is a more detailed "Knowledge Base" [88] available that tries to answer all questions arising from using the application as well as editing lines and schedules. For registered editors, an additional "Community Wall" is provided in order to communicate with employees of Moovit and post questions or suggestions. Tutorial videos are also available for editors.

Control

Moovit does not employ any type of gatekeeping control towards its app users. As soon as the application is installed, it

can be used. Also, gatekeeping towards secondary developers is not visible. The relation towards transit agencies is twofold: on the one hand, Moovit does not deny access to its network to transit agencies, as this facilitates its business model. Transit agencies partnering with Moovit will usually provide their schedule data, which comes with cost and time savings for Moovit, as these plans are available faster and in better quality and reliability than if they were community-created. On the other hand, some transit agencies seem to be very reluctant in partnering with Moovit - thus, gatekeeping by transit agencies towards Moovit is very common. Moovit tries to overcome this gatekeeping with the help of its community of editors. Thus, gatekeeping towards editors is nearly not existent.

Just like in the case of Waze, process control measures are unknown. Similarly, no output control or metrics are installed. Social or relational control is not exerted over secondary application developers that connect their own applications to Moovit's API, but simply use it in order to provide additional service to the users of their application. On the contrary, social control over users and especially the community of editors can be substantial, as the action of becoming an editor already implies identification with Moovit's vision.

Finally, sanctional control can be achieved by deleting or blocking accounts.

4.4 Apple

Apple Inc.¹³ was founded in 1976 by Steve Jobs, Steve Wozniak, and Ronald Wayne [89]. Today, Apple develops and sells a range of mobile devices, laptops, and desktop computers. All mobile devices are running on the same operating system, "iOS", which contains several built-in standard applications, e.g. camera, calendar, and music [90].

Platform Provider Structure

Apple is the world's most valuable publicly listed company with a market capitalization of \$544bn [91] and employs about 110,000 people [92]. Thus, Apple is clearly not a start-up like Waze or Moovit, and already was a multi-national company when starting the platform ecosystem in 2008.

Decision Rights Partitioning

Decision rights are clearly unbalanced in the iOS ecosystem, with strong bias towards Apple. Platform decision rights are completely centralized, strategical as well as implementational decision rights. However, the past has shown that in some cases, Apple had to change some parts of its control policies due to public pressure. Eaton et al. call this process "distributed tuning of boundary resources" [93]. This has been the case both at strategical and implementational level. At strategical level, Apple allows the distribution of native applications inside the iOS ecosystem by providing the App Store. On an implementational level, APIs providing access to more of the hardware's resources in order to incentivize more developers to join the iOS ecosystem [13]. In the application domain, Apple does not have decision rights to

¹³<http://www.apple.com>

be exerted directly over the work of secondary developers. However, as every application can be refused by Apple, strategic decision rights over what application can be distributed within the ecosystem remain, at least partly, with Apple. On the implementational level, quality control imposes a long list of requirements that applications have to meet in order to be admitted to the App Store. Therefore, decision rights for applications are not decentralized.

Business

Apple's business model generates revenue from selling iOS devices to customers and retaining a 30% share from every secondary application sale on the App Store [90]. Apple's revenue mainly comes from the customers, as they are the ones purchasing applications in the App Store and thus financing developers' as well as Apple's share.

Apple's strategy consists of preserving the virtuous cycle of indirect network effects between iOS users and secondary application developers. Apple develops new iOS devices with improved hardware nearly every year and additionally tries to expand the market of iOS devices to other product categories. The main and probably most important incentive for secondary developers to join the iOS ecosystem is the direct access to millions of devices and its users. Another incentive is the higher willingness of iOS to pay higher prices, compared to Android users [94] and more possible ways of monetizing secondary applications [95].

Support for Apple's developers is vast, e.g. "Apple Developer Program" page [90], documentation web pages, personal technical support [90], and analytic tools [96].

Control

As the existing iOS ecosystem is very large, the controls are accordingly numerous, diverse, and oftentimes complex. Gatekeeping happens at two levels in the iOS ecosystem: secondary developers wishing to create applications for the iOS platform have to register for a developer account, which includes the payment of a yearly membership fee of \$99 per single developer and \$299 per company license. The second level of gatekeeping is applied to applications, which have to pass Apple's review (see the review guidelines [97]). This is where applications that do not fulfill Apple's criteria are sorted out and not admitted to the App Store.

Apple performs regulatory activities via gatekeeping. In the past, the company has been protecting the interests of business partners like AT&T, which was the reason for denying access to the App Store for VoIP (voice over internet protocol) applications in 2009 [93]. Apple also engages largely in regulatory activities protecting its own business, stating that applications doubling core, built-in iOS functionality, thus not being "useful", will not be admitted to the App Store [97]. However, as there is no balance of power in the iOS ecosystem, influence of developers remains marginal.

Apple organizes its extensive output control of application via the "App Store Review Process". As mentioned by Eaton et al. [93], this process has often been disputed, sometimes with

successful outcomes for developers.

Social control is not prevalent in the iOS ecosystem, as it is very obvious for secondary developers that Apple's only goal is to generate as much revenue as possible, and that extensive support for developers is only a means to achieve this goal. Developers hence will most likely adopt a similar approach and strive to be as commercially successful as possible.

As with all other formal controls, Apple retains a maximum of sanctional controls. This applies to applications, which can be excluded "for any reason", even when fulfilling the review guidelines [98], as well as developers, who can be suspended or excluded as a registered developer "at any time in Apple's sole discretion" [99].

4.5 ITS Factory

The ITS (Intelligent Transport Systems) Factory¹⁴ is a mobility platform project in Tampere, Finland [100]. The project was initiated in the years 2012 and 2013 and aims at opening up public and private traffic data in an "open data" approach [101] which should be available with "published interfaces and standard-based definitions" [102]. It is organized as a joint initiative of the city of Tampere, regional and national transit agencies, companies, service providers, and research institutions, like the two universities of Tampere [102]. The city and transit agencies aim to provide as much data as possible to everyone via public APIs - like real-time bus positions, traffic light circuits and more. ITS Factory thus represents a mobility platform that could evolve into an ecosystem, and is also mainly publicly financed as the TUM LLCM.

Platform Provider Structure

The structure of the joint initiative currently consists of 32 private companies and 11 public entities [103]. It is assumed that the project is mainly financed publicly, however, the exact financing structure is not disclosed publicly. Similarly, the internal organization of the initiative is not visible from the outside, but it can be assumed that decisions are taken collaboratively, as there are many different parties involved. Decisions will definitely be more balanced than with only one single company involved, as cooperative discussion oftentimes leads to changed points of view. The characteristics of the ITS Factory initiative are thus neither that of a start-up company, like Moovit or Waze, nor that of an incumbent company, like Apple, simply because it does not act as a company, but as a sort of public-private partnership for the purpose of promoting the development and use of open data concepts and developing new solutions with them.

Decision Rights Partitioning

Departing from the information provided by the developer wiki [104], developers play an active part in decisions about the platform or at least give feedback about certain aspects. For instance, in the newsletter 1/2015, developers were asked to name sources out of a list which they would find useful to be

¹⁴<http://www.hermiagroup.fi/its-factory>

integrated into the traffic light API [105]. Platform implementation decision rights seem to be rather concentrated with the platform provider, more specifically by the City of Tampere or its delegates, like the University of Tampere, who defines what standards will be used for data formats and how APIs will be designed. The definition and usage of data format standards is considered very important and treated accordingly [106]. Application strategic decision rights are completely with the provider, as the goal is to find new modes of transport and mobility services. On the implementation side of applications, the only restriction is the obligation to use the data formats imposed by the platform's APIs.

Business

As the platform provider is an association of private and public actors and publicly funded, ITS Factory's goal is yet not to establish a profitable business. Still, some aspects can be described for every sub-domain. The market of the platform is currently limited to the city of Tampere, as data is only available for this area.

The ITS Factory is organized as a non-profit association which aims to provide as much mobility data as possible to developers via public APIs. It does not expect any consideration for using the data, neither is there an obligation to return any of the data an application generates or gathers from users (like movement profiles or most used bus routes). The focus clearly lies on opening up public, but also private, traffic data sources. While the open data principle does not demand payment of any fees for using the data sources, developers can still monetize the secondary applications developed on their basis.

The strategy of the ITS Factory is to develop capabilities in the domain of data- and software- driven traffic and mobility solutions, and to "improve the awareness of Finnish ITS expertise" [107]. The intelligent transport solutions developed and tested within the mobility lab can then present opportunities for internationalization and export for the companies involved [108]. Thus, the project constitutes a measure of promotion of economic development for local and national companies in order to develop capabilities that can be exported into the rest of the world and strengthen the Finnish economy.

Incentives for secondary developers are mainly the access to data of quality and depth not available elsewhere and a testing environment to experiment with them. The access to freely available real-time and real-world data is a strong argument to start developing within the ITS Factory ecosystem. Resulting applications can be tested in so-called "sandbox" and with real users. If successful, developers have access to a large number of potential customers in Tampere.

Support for developers is mainly concentrated on documentation of existing APIs and information about future enhancements. This information is organized in the form of a "Wiki", such that registered members can add and edit it [104], as well as giving developers the possibility of connecting and interacting. There is also documentation about the different data format standards and justification of their use in the different APIs [106]. Additionally, there is support for searching

and finding financing for application ideas [105]. A common roadmap of all currently running projects provides developers an overview of the ecosystem and the intentions of other actors, pointing out possibilities for collaboration.

Control

Controls are purposefully very low for joining the ITS Factory ecosystem and using the platform. Developers should have access to all public data "against minimum bureaucracy and formalities" [102]. Thus, there is no gatekeeping, except the need for registration as an ITS Factory participant or developer. Additionally, access to some few APIs needs to be requested separately.

There is no process control or similar measures. Taking such measures would be contradictory to the goal of giving room for as much experimentation and creativity as possible to secondary developers.

Apart from some access limits for certain APIs, there are no output controls or metrics which secondary applications have to pass. This is only consequent, as the real goal of the project are not outputs of high quality, but the development of capabilities in a "learning-by-doing" approach.

Social control is possible to a certain extent, but economic considerations will outweigh them at some point.

Finally, there is no sanctional control envisaged within the ITS Factory ecosystem.

4.6 Cross-Platform and Ecosystem Comparison

Each type of investigated platform employs different strategies to achieve its goals: ecosystem strategy, crowdsourcing strategy, and open (public) data strategy. Apple's iOS turned out to be the only real ecosystem according to the definition developed from literature. Waze and Moovit do not provide developers with the possibility of really adding new services to their existing ones, and ITS Factory is not yet in the stage of being an ecosystem as applications are not distributed yet and mostly for testing purposes.

Apple's ecosystem is based on an innovative product, the iPhone. In its development process, Apple incurred high initial development costs, a typical process in establishing a platform, as Baek et al. noticed [55], to incentivize secondary developers to join the platform. The platform has two sides, customers and developers. Apple quickly managed to "get both sides on board", thus ensuring what Evans calls "catalyst ignition" [36] and could subsequently rely on positive indirect network effects where more iOS users make the platform more appealing to developers, which in turn leads to more available applications, attracting new users. Apple concentrates on developing a constant stream of improved devices and on maintaining the perceived high quality of available secondary applications, by applying mainly two control measures, gatekeeping and output control, both of which are also used for securing own interests, as many examples show [93]. The strategy employed by Waze and Moovit relies on crowdsourcing. Its basis is not a product, but the idea of an innovative and useful service. It uses smartphones to deliver the

service as well as collect data from its participants, thus in fact creating a platform on top of these smartphone platforms (like Apple's iOS). In order for these services to work properly, they have to acquire a large user base, which is why the strategy relies on a maximum of visibility and a gamification approach to foster user retention and incite them to share additional information. Business models of Waze and Moovit rely on hyper-local advertising. This leads to a pricing structure where profits are made on only one side of the platform (advertisers), whereas the service is free in a monetary sense for users, who agree to provide their data.

The third strategy observed, employed by the ITS Factory, is based on making data available to developers and other interested parties. Currently, this includes bus location, traffic flow, parking, weather data, and many others. The goal behind this open data strategy is in fact the political will to enhance domestic innovativeness in a special technological area with good future prospects by providing a "playground" for experimentation with data that will most likely become available in more and more cities and areas of the world in the near future. To provide open data in the first place, high investments from mostly public entities are necessary. In order to develop into a full ecosystem, two steps are required: Firstly, commercial usability of data has to be ensured, mainly by guaranteeing SLAs towards secondary applications. Only then can applications attract users and start a virtuous cycle of indirect network effects. Secondly, in order for the ecosystem to grow further after all public data has been made accessible, data of applications has to be "played back" to the platform to some extent, which further enlarges its data pool. Figure 8 provides an overview of the results. Based on the analysis of the platforms, the following success factors are elicited:

- **Using network effects:** Platform providers have to ensure "catalyst ignition" of their platform ecosystems [25], i.e. getting both sides on board of the platform quickly. This is possible in most cases by attracting one user group with an offer that is already useful on its own. With this user base on one side, the other side can easier be convinced to join the platform ecosystem.
- **Retaining strategic platform decision rights:** Platform ecosystem providers should retain strategic decision rights over the platform in order to direct its current and future development.
- **Finding the right balance for quality control:** Platform ecosystem providers have to balance quality control carefully to incite secondary developers to experiment with the possibilities of the platform and giving them real prospects of commercialization of the results. A careful combination of gatekeeping (rather low) and quality control of secondary applications (rather high) is necessary to ensure a sustainable ecosystem. High quality control has to be rewarded with access to a large number of customers with high willingness to pay.

- **Extensive developer support:** Developers should have access to high quality documentation and support in order to contribute with secondary applications to the platform ecosystems as easy as possible. This ensures higher quality of resulting applications, which in turn strengthens network effects within the ecosystem.

5. Platform Ecosystem Governance Options for TUM LLCM

In this section, success factors from the previous cases will be applied to the prospective mobility platform TUM LLCM, including a discussion of which strategies can be adapted and combined for establishing the mobility platform. The previously elicited success factors in Subsection 4.6 cannot be completely combined together in order to form a TUM LLCM strategy. The most obvious restriction lies in the fact that TUM LLCM will not develop any hand held smart device, like Apple did, to form the basis of its ecosystem. Additionally, the strategies are contradictory at some points (especially concerning the domains of decision rights and control), such that one of them has to be chosen for the mobility platform under development at TUM. Other success factors, namely crowdsourcing and open data, can be combined within the TUM LLCM ecosystem. There are two alternatives being discussed that have been named "Powered by TUM LLCM" (Subsection 5.1) and "TUM LLCM App" (Subsection 5.2).

5.1 Platform Ecosystem Governance Option "Powered by TUM LLCM"

The first alternative, "Powered by TUM LLCM" is presented in the following, guided by the platform ecosystem governance framework domains.

Platform Provider Structure

In the cases analyzed, providers were single, privately owned firms - with the exception of ITS Factory, which is a publicly funded research project. However, there are many platforms with other ownership structures like open source (Linux), open community (Java [109]), platforms where ownership is shared (Nissan Micra / Renault Clio common platform ([21]), or completely centralized (like Apple's iOS or the Alibaba ecosystem). Thus, we can conclude that the ownership structure of the platform provider is not decisive for its success. What is more important is the platform's evolvability in the long term and its flexibility in reacting to external threats and opportunities in the short term. As long as the internal decision making at the platform provider supports these issues, the structure of the provider does not influence success or failure.

Decision Rights Partitioning

Decision rights are derived mainly from ITS Factory. Concerning the platform strategy, the provider of TUM LLCM should be able to make decisions independently, that is hold these decision rights centrally. Yet, input from developers should be possible, and the process of getting developer's

	Waze	Moovit	Apple	ITS Factory
Platform provider structure	<ul style="list-style-type: none"> Single company < 10 years old < 200 employees Start-up 	<ul style="list-style-type: none"> Single company < 10 years old < 200 employees Start-up 	<ul style="list-style-type: none"> Single company 40 years old 10.000 employees Very large incumbent 	<ul style="list-style-type: none"> Large association < 10 years old No direct employees Publicly funded research project
Decision rights partitioning	<ul style="list-style-type: none"> Platform strategy Platform implementation Application strategy Application implementation 	<ul style="list-style-type: none"> Platform strategy Platform implementation Application strategy Application implementation 	<ul style="list-style-type: none"> Platform strategy Platform implementation Application strategy Application implementation 	<ul style="list-style-type: none"> Platform strategy Platform implementation Application strategy Application implementation
Business	<ul style="list-style-type: none"> Personal mobility market Based on crowdsourcing Revenue generation with advertising Gamification Documentation, wiki, and forums 	<ul style="list-style-type: none"> Personal mobility market Based on crowdsourcing Revenue generation with advertising Gamification Documentation, wiki, and forums 	<ul style="list-style-type: none"> Smartphone/Tablet market Based on strong ecosystem Revenue through devices and commissions Access to large ecosystem Documentation, extensive support, and full service for distribution 	<ul style="list-style-type: none"> Personal mobility market Based on open data No revenue Access to real data Documentation, Wiki, and forums
Control	<ul style="list-style-type: none"> Gatekeeping Regulatory act. Process control Output control Social control Sanctional control 	<ul style="list-style-type: none"> Gatekeeping Regulatory act. Process control Output control Social control Sanctional control 	<ul style="list-style-type: none"> Gatekeeping Regulatory act. Process control Output control Social control Sanctional control 	<ul style="list-style-type: none"> Gatekeeping Regulatory act. Process control Output control Social control Sanctional control

Legend:

- centralized decision rights
- decentralized decision rights
- divided/contested decision rights

Figure 8. Overview of Analyzed Platforms' Governance

feedback should be "institutionalized".

Platform implementation should be fully centralized in order to force the introduction and use of standards in data formats, exchange and data, and privacy protection. These standards have to be set according to clearly stated principles.

At the application level, strategic decision rights should remain completely with the developer to allow for as much experimentation and innovation as possible.

The implementation of applications should only be restricted in the sense that standards set by the platform regarding data are fulfilled. As long as this is the case, there should be no limits regarding programming frameworks or which mobile OS the developer uses.

Business

In a long-term perspective, only profitable solutions will most likely prevail. Therefore, the business domain of the prospective TUM LLCM presented here does not depend on perpetual public funding. However, on a smaller scale, such as cities or regions, public funding can be an option.

The market in which the TUM LLCM will operate does not have to meet any special criteria - except one: customers have to be equipped with a smart device. In a first time, conditions will certainly be more favorable in metropolitan regions where there are real alternatives in multi-modal travel planning, and enough customers available to attain a critical mass.

Business model and strategy are at the heart of the TUM LLCM if it is to be profitable in the long term. The proposed strategy would, just like ITS Factory, try to provide as much traffic-related data as possible as open data, publicly accessible to secondary developers. This will only be possible with the help of public funding, as it does not yield direct rewards for any of the involved parties. With this approach, the platform can attract first developers who provide innovative ideas for new applications, like ITS Factory successfully demonstrated. In a second step, data provisions has to be rendered commercially reliable by granting SLAs to secondary developers. Additionally, TUM LLCM could offer assistance for developers seeking investors, business angels, or other, non-monetary support. At this point, the platform would have to reach one very important agreement with the respective secondary developer: the developer would have to "play back" data that his or her application produces while being used by customers. Of course, many issues have to be solved before such two-way data exchange between platform and secondary applications can take place. To name the most important ones:

- Data protection and privacy concerns must be addressed carefully. Every application (and also the platform itself) has to adhere to developed principles before, it can access data from the platform and be published.
- Standards have to be defined by the platform to ensure the measures taken in the first step, and to promote standardization in order to prevent creating a fragmented market because of many isolated ecosystems.

- Each secondary application will have to undergo some review by the platform provider in order to determine the usefulness of the data it provides to the platform.

As a result, the platform's repository of data will start to grow and enable new uses for new developers. This could generate either direct (more applications lead to more data lead to more developers lead to more applications) and indirect (more applications lead to more customers lead to more developers lead to more applications) network effects and thus enlarge the ecosystem as a whole. Developers, who sell their applications via existing ways, like App Store, Play Store, and others, can then be charged for using the platform's data. In fact, TUM LLCM would become a platform on the existing smartphone platforms. In summary, this would lead to a business model where "Powered by TUM LLCM" would become a sign of quality of a mobile application, by signaling that (1) this application relies on the large and of high quality TUM LLCM data pool, and (2) the application was admitted to the TUM LLCM platform, which requires compliance with very strict protection of personal data. This would in some aspects resemble Intel's microprocessor strategy "Intel Inside", signaling to PC users that their device works with an Intel processor [110].

Pricing structure should concentrate on developers, as the platform would be visible to customers only via the "Powered by TUM LLCM" slogan. In pricing developers, TUM LLCM should be very careful in elaborating a structure that does not hinder the development of the ecosystem. For example, one-time access fees should be avoided, or at least be very moderate, such that the goal of attracting a large number of developers and inciting experimentation is not put in danger by these access fees. The usage fee could consist either in a percentage of application sales, or based on which and how much data is consumed compared to the data that is played back into the platform.

Incentives for developers would, in the first phase, be the same like for ITS Factory: access to real-world data and an environment for testing and experimentation for free. In a second phase, developers would have access to even more data, as the platform and its ecosystem grows.

Finally, support for developers would be a challenging task in TUM LLCM: not only because documenting all data sources and APIs is obligatory because of the newness of the data, but also because this helps in spreading the data format standards set by the platform provider. Moreover, extensive support is necessary in the application review process, as not only compliance with standards and quality control have to be checked, but also the application's contribution to the data "pool" and an individual usage fee derived from it have to be assessed. However, neither SDK nor similar have to be provided and documented, as developers will use already existing SDKs from mobile OS providers. Documentation thus is only necessary for the TUM LLCM's own APIs.

Control

The control structure of this approach combines Apple's and ITS Factory's control structure. Gatekeeping should be the first level of two access levels: initially, all developers can register themselves as developers at TUM LLCM. This is possible with a minimum of personal information by paying a preferably low, if any, registration fee. At this stage, developers are granted access to all API documentation, but only dummy test data access, ensuring that applications can be developed against the APIs. Once a developer has finished his or her secondary application, it will enter the review process, which constitutes the second level of access to the platform. As regulatory activities in the TUM LLCM ecosystem, TUM LLCM could admit all services fulfilling the standards set, which will lead to competition on the platform.

While process control should not be implemented to provide developers freedom in experimentation, output control is one very important pillar of control. In this step, compliance with standards of the platform and ecosystem is assessed, as well as the application's played back data valued, leading to individual prices for the application using the platform. Additionally, a formal quality control should be conducted in order to ensure a high overall quality in the ecosystem.

Social control will most likely be limited in an environment where actors try to earn money, whereas sanctional control exerted by the TUM LLCM provider should reserve the right to exclude developers and applications that change over time, resulting in conflicts with established standards, e.g. by exploiting personal data in ways not covered by the platform's privacy and data protection standards.

5.2 Platform Ecosystem Governance Option "TUM LLCM App"

The second alternative, "TUM LLCM App" is presented in a similar manner. As many dimension do not change significantly compared to the first platform ecosystem governance option, only deviations from the former are discussed.

Platform Provider Structure

The internal structure and other characteristics are not considered to be decisive. Thus, there are no other recommendations to provide here than for the option.

Decision Rights Partitioning

Platform decision rights could be organized like in the first platform ecosystem governance option. For application decision rights, the balance of power would incline more towards the platform's side, as developers are restricted in the amount and type of user data they can obtain. The "last word" in this case would always be spoken by the platform provider.

Business

Contrary to the first platform ecosystem governance option, TUM LLCM would develop and provide an own app in the second option. Secondary developers would then have to connect their services to this application, and be integrated into it. The application would integrate all kinds of services into

one user interface. Additionally, the TUM LLCM application could provide a service to the user that could be described with "my personal data belongs to me", making it very attractive to users compared to existing platform ecosystems and applications. Secondary developers would be granted access to only as much personal information as necessary, and as little as possible. For instance, it is not necessary to know the exact coordinates of a mobile device to show the user weather data, as this data is not available in such fine granularity anyway. Of course, users should be able to specify manually which data they want to share with which secondary service provider. As a result of this strategy, the TUM LLCM platform would not be able to grow its data pool as described in the first platform ecosystem governance option. Data sources would be those initially provided by public and private actors, and adding of new data sources would be independent of the number of secondary developers and services.

The pricing structure would attribute costs to the same side as in the first alternative, the developers. However, negotiations about the usage fee would not be individual, as no data would be played back to the platform. Finally, the application and the platform could be financed through advertising and a priced version of the application without advertising.

Incentives for developers to join the platform would remain the same concerning the access to new data made available by the platform provider in the first phase of platform establishment. On the contrary, the data protection concept of the TUM LLCM application can be a disincentive for developers, as they do not get access to personal data of their clients anymore, at least to a lesser extent than with an own application. Support for developers would have to stay on the same level concerning documentation of data format standards and APIs. Additional documentation would be necessary to explain the data protection mechanisms employed by the application and which data secondary services can request from it to provide their services. In return, the review process would not be as complex as in the first platform ecosystem governance option, as only data requirements from applications have to be critically reviewed and approved or rejected. This review process should also be documented extensively to allow for a maximum of transparency for both developers and users.

Control

Many of the controls could also remain the same for the "TUM LLCM App" scenario. There are two differences in regulatory activities of the provider and output control. Regulatory activities could again try to limit the number of secondary developers providing the same service on the platform. In the case of a TUM LLCM application, this becomes a more pressing issue, as it cannot integrate an unlimited amount of services in order to still be usable and user friendly.

The output control for secondary services connected to the TUM LLCM application would differ from above in that not the compliance with data protection and privacy standards would be checked in a review, but whether the amount and type of data requested by the application is really necessary

	“Powered by TUM LLCM”	“TUM LLCM App”
Platform provider structure	<ul style="list-style-type: none"> Structure not decisive 	<ul style="list-style-type: none"> Structure not decisive
Decision rights partitioning	<ul style="list-style-type: none"> Platform strategy Platform implementation Application strategy Application implementation 	<ul style="list-style-type: none"> Platform strategy Platform implementation Application strategy Application implementation
Business	<ul style="list-style-type: none"> Personal mobility market Based on open data Revenue generation with commissions or advertising Access to real data Documentation, wiki, forums, and extensive support 	<ul style="list-style-type: none"> Personal mobility market Based on open data Revenue generation with commissions or application sale Access to real data Documentation, wiki, forums, and extensive support
Control	<ul style="list-style-type: none"> Gatekeeping Regulatory act. Process control Output control Social control Sanctional control 	<ul style="list-style-type: none"> Gatekeeping Regulatory act. Process control Output control Social control Sanctional control

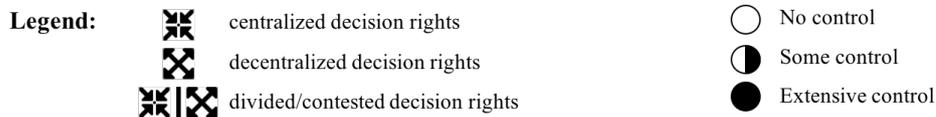


Figure 9. Overview of Recommendations for the TUM LLCM

to provide the promised service. Based on this review, data access rights can then be granted to the secondary service.

5.3 Overview of Platform Ecosystem Governance Options for TUM LLCM

Figure 9 provides an overview of the two previously described platform ecosystem governance options. Both platform ecosystem governance options place an emphasis on data and privacy protection. However, “Powered by TUM LLCM” also tries to use data created by applications and their customers to come to a data- and crowd-based ecosystem with network effects, whereas “TUM LLCM App” concentrates even more on the protection of personal data by not giving any personal data to secondary service providers and developers before thoroughly reviewing their data requests. Thus, it relies on the data made available in the first phase of platform establishment and network effects generated by the mutual attraction of developers and customers. Also, it integrates all services into one application.

6. Conclusion

Subsection 6.1 provides a summary of this work. Subsection 6.2 reveals the limitations of the work and provides a brief outlook of possible future investigations.

6.1 Summary

This work tried to shed light on the governance of a mobility platform and ecosystem to be developed within the research project TUM LLCM. The need for new concepts for personal mobility arises from two human megatrends, world population growth and urbanization, which both lead to larger and denser urban agglomerations. Within these agglomerations, transportation has reached its limits, already today. Mobility platforms, made possible through the digitalization of all areas of life, could be an element of the solution to these problems. All in all, the research questions were addressed accordingly:

- **First research question:** The goal was to gain an understanding about existing definitions of “platform”, “platform ecosystems” and other related terms. The

question was answered with the help of a literature review according to Webster and Watson [10]. The result presented in Section 2 was a comparison of existing definitions and classifications as well as a list of derived definitions for the scope of this work. Additionally, a platform ecosystem governance framework was derived from existing literature and presented in Section 3.

- **Second research question:** The goal was to analyze platforms, selected for their positioning in the mobility domain, their ecosystem characteristics, and their similarities to the TUM LLCM project. The analyzed platforms were described extensively guided by the framework's different domains. The different strategies and success factors were extracted and shown in the final part of Section 4.
- **Third research question:** The goal was to present platform ecosystem governance options for the implementation of TUM LLCM's governance. "Powered by TUM LLCM" would set strict standards for data formats and data and privacy protection while still trying to use the data generated by users and applications to enlarge an initially provided pool of data sources. "TUM LLCM App", on the contrary, would protect customer's data by forcing secondary developers to connect their services to an application controlled by TUM LLCM. Both platform ecosystem governance options would, based on the findings about success factors in the second research question, be suitable to "successfully establish a mobility platform and ecosystem", as demanded by the third research question.

6.2 Limitations and Future Research

The low number of only four platforms considered raises concerns about generalizability of the results. While the success strategies found most likely are valid, they may not be complete and their list not be exhaustive. Future research should thus try to analyze more relevant cases and validate not only the success factors and strategies found, but also the platform governance analysis framework itself. Furthermore, no negative examples of platforms or platform ecosystems were conducted within this research, as negative examples are hard to find. Although some research has already been done on failed platforms (see [37]), their examples were either very outdated (from before 1995) or unrelated to the topic of this work because of their very wide scope of the term "platform". If possible, future research should try to find failed platform ecosystem businesses and try to validate the success strategies by comparing their governance framework implementation with that of successful examples. With the establishment of more and more platform businesses, the search for failed examples will possibly also become easier in the future. Finally, the platform ecosystem governance framework did not consider any "external aspects", like regulatory interventions of governmental actors. Partly, this was included in some of the

domains, especially within the recommendations given about data standards and data and privacy protection, which should be "compliant" to existing laws. However, these external aspects are most likely not restricted to regulatory activities and laws and should be researched separately.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] United Nations, Department of Economic and Social Affairs, Population Division. World urbanization prospects: The 2014 revision. <https://esa.un.org/unpd/wup/Publications/Files/WUP2014-Highlights.pdf>, 2014. Accessed: 2016-06-18.
- [2] Susan Zielinski. New mobility: The next generation of sustainable urban transportation. In *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2006 Symposium*, page 107. National Academies Press, 2007.
- [3] U.S. Department of Transportation: Federal Highway Administration. Annual vehicle - miles of travel, 1980 - 2007 1/. http://www.fhwa.dot.gov/policyinformation/statistics/vm02_summary.cfm. Accessed: 2016-06-18.
- [4] World Business Council for Sustainable Development. world mobility at the end of the twentieth century and its sustainability. <http://web.mit.edu/aeroastro/sites/waitz/publications/WBCSD.report.pdf>, 2001. Accessed: 2016-06-18.
- [5] David Schrank, Bill Eisele, and Tim Lomax. Ti's 2011 urban mobility report. <http://www.pagregion.com/Portals/0/documents/HumanServices/2011MobilityReport.pdf>, 2011. Accessed: 2016-06-18.
- [6] Landeshauptstadt München. Demografiebericht münchen - teil 2, kleinräumige bevölkerungsprognose 2013 bis 2030 für die stadtbezirke. <https://www.ris-muenchen.de/RII/RII/DOK/SITZUNGSVORLAGE/2996152.pdf>, 2015. Accessed: 2016-06-18.
- [7] Landeshauptstadt München. Verkehrsentwicklungsplan. <https://www.muenchen.de/rathaus/dam/jcr:cb4ecc75-4d43-4a87-963c->

- 5f49186fb34b/vep06_kurz_de.pdf, 2006. Accessed: 2016-06-18.
- [8] Technische Universität München. Tum living lab connected mobility. <http://tum-llcm.de/>, 2016. Accessed: 2016-06-18.
- [9] Stefan Hefe. Living lab connected mobility - analysis and description of design options for the establishment of a sustainable mobility ecosystem. Master's thesis, Technische Universität München, 2016.
- [10] Jane Webster and Richard T. Watson. Analyzing the past to prepare for the future: Writing a. *Mis Quarterly*, 26(2), 2002.
- [11] Maximilian Schrieck, Manuel Wiesche, and Helmut Krcmar. Design and governance of platform ecosystems – key concepts and issues for future research. In *Twenty-Fourth European Conference on Information Systems (ECIS)*, İstanbul, Turkey, 2016.
- [12] Carliss Y. Baldwin and C. Jason Woodard. *The Architecture of Platforms: A Unified View*, book section 2, pages 19–44. Edward Elgar, Cheltenham, UK, 2009.
- [13] Ahmad Ghazawneh and Ola Henfridsson. Balancing platform control and external contribution in third-party development: the boundary resources model. *Information Systems Journal*, 23(2):173–192, 2013.
- [14] Geoffrey Parker and Marshall Van Alstyne. Managing platform ecosystems. *ICIS 2008 Proceedings*, 2008.
- [15] S. C. Wheelwright and K. B. Clark. Creating project plans to focus project development. *Harvard Business Review*, 70(2):67–83, 1992.
- [16] Marc H. Meyer and Alvin P. Lehnerd. *The power of product platforms*. Simon and Schuster, 1997.
- [17] Shane Greenstein. *Open platform development and the commercial Internet*, book section 9, pages 219–248. Edward Elgar, Cheltenham, UK, 2009.
- [18] Michael A. Cusumano and Annabelle Gawer. The elements of platform leadership. *MIT Sloan Management Review*, 43(3):51, 2002.
- [19] Annabelle Gawer and Michael A Cusumano. *Platform leadership: How Intel, Microsoft, and Cisco drive industry innovation*. Harvard Business School Press Boston, 2002.
- [20] Annabelle Gawer and Michael A. Cusumano. How companies become platform leaders. *MIT/Sloan Management Review*, 49(2), 2012.
- [21] Annabelle Gawer. Platform dynamics and strategies: from products to services. In Annabelle Gawer, editor, *Platforms, markets and innovation*, book section 3, pages 45–76. Edward Elgar, Cheltenham, UK, 2009.
- [22] Jean-Charles Rochet and Jean Tirole. Two-sided markets: a progress report. *The RAND Journal of Economics*, 37(3):645–667, 2006.
- [23] Carl Shapiro and Hal R. Varian. *Information rules : a strategic guide to the network economy*. Harvard Business Review Press, Boston, Massachusetts, 1999.
- [24] Andrei Hagiu and Julian Wright. Multi-sided platform. working paper no. 12024. harvard business school., 2011.
- [25] David S. Evans. *How catalysts ignite: the economics of platform-based start-ups*, book section 5, pages 99–130. Edward Elgar, Cheltenham, UK, 2009.
- [26] Jean-Charles Rochet and Jean Tirole. Platform competition in two-sided markets. *Journal of the European Economic Association*, 1(4):990–1029, 2003.
- [27] Yannis Bakos and Evangelos Katsamakas. Design and ownership of two-sided networks: Implications for internet platforms. *Journal of Management Information Systems*, 25(2):171–202, 2008.
- [28] Rahul C. Basole and Jürgen Karla. Entwicklung von mobile-platform-ecosystem-strukturen und -strategien. *WIRTSCHAFTSINFORMATIK*, 53(5):301–311, 2011.
- [29] Kevin Boudreau. Open platform strategies and innovation: Granting access vs. devolving control. *Management Science*, 56(10):1849–1872, 2010.
- [30] Kevin J. Boudreau and Andrei Hagiu. *Platform rules: multi-sided platforms as regulators.*, book section 7, pages 163–191. Edward Elgar, Cheltenham, UK, 2009.
- [31] Marco Ceccagnoli, Chris Forman, Peng Huang, and D. J. Wu. Cocreation of value in a platform ecosystem: The case of enterprise software. *MIS Quarterly*, 36(1):263–290, 2012.
- [32] Michael Cusumano. Technology strategy and management the evolution of platform thinking. *Communications of the ACM*, 53(1):32–34, 2010.
- [33] Nicholas Economides and Evangelos Katsamakas. Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. *Management Science*, 52(7):1057–1071, 2006.
- [34] Thomas R. Eisenmann, Geoffrey Parker, and Marshall Van Alstyne. *Opening Platforms: How, When and Why?*, book section 6, pages 131–162. Edward Elgar, Cheltenham, UK, 2009.
- [35] Thomas Eisenmann, Geoffrey Parker, and Marshall Van Alstyne. Platform envelopment. *Strategic Management Journal*, 32(12):1270–1285, 2011.
- [36] David S. Evans and Richard Schmalensee. The industrial organization of markets with two-sided platforms. Report, National Bureau of Economic Research, 2005.
- [37] Gezinus J. Hidding, Jeff Williams, and John J. Sviokla. How platform leaders win. *Journal of Business Strategy*, 32(2):29–37, 2011.

- [38] Pascal Le Masson, Benoit Weil, and Armand Hatchuel. *Platforms for the design of platforms: collaborating in the unknown*, book section 11, pages 273–305. Edward Elgar, Cheltenham, UK, 2011.
- [39] Jean-Charles Rochet and Jean Tirole. Two-sided markets: A progress report. *The RAND Journal of Economics*, 37(3):645–667, 2006.
- [40] Simone Scholten and Ulrich Scholten. Platform-based innovation management: Directing external innovative efforts in complex self-organizing platform ecosystems. In *Technology Management for Global Economic Growth (PICMET), 2010 Proceedings of PICMET'10*, pages 1–12. IEEE, 2010.
- [41] Fernando F. Suarez and Michael Cusumano. *The role of services in platform markets*, book section 4, pages 77–98. Edward Elgar, Cheltenham, UK, 2009.
- [42] Fernando F. Suarez and Jacqueline Kirtley. Dethroning an established platform. *MIT Sloan Management Review*, 53(4):35–41, 2012.
- [43] Hirofumi Tatsumoto, Koichi Ogawa, and Takahiro Fujimoto. *The effect of technological platforms on the international division of labor: a case study of Intel's platform business in the PC industry*, book section 14, pages 345–369. Edward Elgar, Cheltenham, UK, 2009.
- [44] Amrit Tiwana, Benn Konsynski, and Ashley A. Bush. Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research*, 21(4):675–687, 2010.
- [45] Amrit Tiwana. *Platform ecosystems*. Elsevier, Amsterdam, 2014.
- [46] Annabelle Gawer. Bridging differing perspectives on technological platforms: Toward an integrative framework. *Research Policy*, 43(7):1239–1249, 2014.
- [47] Annabelle Gawer and Michael A. Cusumano. Industry platforms and ecosystem innovation. *Journal of Product Innovation Management*, 31(3):417–433, 2014.
- [48] Konstantinos Manikas and Klaus Marius Hansen. Software ecosystems – a systematic literature review. *Journal of Systems and Software*, 86(5):1294–1306, 2013.
- [49] David G. Messerschmitt and Clemens Szyperski. Software ecosystem: understanding an indispensable technology and industry. *MIT Press Books*, 1, 2005.
- [50] Slinger Jansen, Anthony Finkelstein, and Sjaak Brinkkemper. A sense of community: A research agenda for software ecosystems. In *Software Engineering-Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on*, pages 187–190. IEEE, 2009.
- [51] Jan Bosch. From software product lines to software ecosystems. In *Proceedings of the 13th international software product line conference*, pages 111–119. Carnegie Mellon University, 2009.
- [52] Jan Bosch and Petra Bosch-Sijtsema. From integration to composition: On the impact of software product lines, global development and ecosystems. *Journal of Systems and Software*, 83(1):67–76, 2010.
- [53] Jan Bosch and Petra Bosch-Sijtsema. *Softwares product lines, global development and ecosystems: collaboration in software engineering*, pages 77–92. Springer Berlin Heidelberg, 2010.
- [54] Mircea Lungu, Michele Lanza, Tudor Girba, and Romain Robbes. The small project observatory: Visualizing software ecosystems. *Science of Computer Programming*, 75(4):264–275, 2010.
- [55] Sodam Baek, Kim Kibae, and Jörn Altmann. Role of platform providers in service networks: The case of salesforce.com app exchange. In *Business Informatics (CBI), 2014 IEEE 16th Conference on*, volume 1, pages 39–45, 2014.
- [56] Rahul C. Basole and Jürgen Karla. On the evolution of mobile platform ecosystem structure and strategy. *Business & Information Systems Engineering*, 3(5):313–322, 2011.
- [57] Thomas Eisenmann, Geoffrey Parker, and Marshall W Van Alstyne. Strategies for two-sided markets. *Harvard business review*, 84(10):92, 2006.
- [58] Slinger Jansen and Michael A. Cusumano. Defining software ecosystems: a survey of software platforms and business network governance. In Slinger Jansen, Jan Bosch, and Carina Alves, editors, *Fourth International Workshop on Software Ecosystems (IWSECO 2012)*, pages 40–58, 2012.
- [59] Julia Manner, David Nienaber, Michael Schermann, and Helmut Krcmar. Governance for mobile service platforms: A literature review and research agenda, 2012.
- [60] Rod Rhodes. Understanding governance: Ten years on. *Organization studies*, 28(8):1243–1264, 2007.
- [61] Business Dictionary. Governance. <http://www.businessdictionary.com/definition/governance.html>, 2016. Accessed: 2016-06-18.
- [62] Business Dictionary. Corporate governance. <http://www.businessdictionary.com/definition/corporate-governance.html>, 2016. Accessed: 2016-06-18.
- [63] Peter Weill. Don't just lead, govern: How top-performing firms govern it. *MIS Quarterly Executive*, 3(1):1–17, 2004.
- [64] Allen E. Brown and Gerald G. Grant. Framing the frameworks: A review of it governance research. *Communications of the Association for Information Systems*, 15(1):38, 2005.

- [65] Andreas Hein, Maximilian Schreieck, Manuel Wiesche, and Helmut Krcmar. Multiple-case analysis on governance mechanisms of multi-sided platforms. In *Multikonferenz Wirtschaftsinformatik*, Ilmenau, Germany, 2016.
- [66] Bloomberg. Company overview of waze mobile limited. <http://www.bloomberg.com/research/stocks/private/snapshot.asp?privcapId=58203778>, 2016. Accessed: 2016-06-18.
- [67] Dan Graziano. Google finally discloses waze acquisition price. <http://bgr.com/2013/07/26/google-waze-acquisition-price/>, accessed on May 9, 2016, 2013.
- [68] Quora. How many employees does waze have as of may 2013? <http://www.quora.com/How-many-employees-does-Waze-have-as-of-May-2013>, 2013. Accessed: 2016-06-18.
- [69] LinkedIn. Waze. <https://www.linkedin.com/company/waze>, 2016. Accessed: 2016-06-18.
- [70] Waze. Waze privacy policy. <https://www.waze.com/legal/privacy>, 2016. Accessed: 2016-06-18.
- [71] Ratko Vidakovic. How hyperlocal mobile advertising changes everything. <http://marketingland.com/hyperlocal-mobile-advertising-changes-everything-92979>, 2014. Accessed: 2016-06-18.
- [72] Waze. Waze for broadcasters. <https://www.waze.com/broadcasters>, 2016. Accessed: 2016-06-18.
- [73] Waze. Waze connected citizens. <https://www.waze.com/broadcasters>, 2016. Accessed: 2016-06-18.
- [74] Neal Ungerleider. Waze is driving into city hall. <http://www.fastcompany.com/3045080/waze-is-driving-into-city-hall>, 2015. Accessed: 2016-06-18.
- [75] Waze. Waze advertisers dashboard. <https://biz.world.waze.com/>, 2016. Accessed: 2016-06-18.
- [76] American Marketing Association. Cost-per-thousand. <https://www.ama.org/resources/Pages/Dictionary.aspx>, 2016. Accessed: 2016-06-18.
- [77] Waze. Waze ads for brands. <https://www.waze.com/brands>, 2016. Accessed: 2016-06-18.
- [78] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: defining "gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, pages 9–15, 2181040, 2011. ACM.
- [79] Waze. Waze help. <https://support.google.com/waze/?topic=6273402#topic=6273402>, 2016. Accessed: 2016-06-18.
- [80] Don Dahmann. Wir denken nicht über den verkauf unserer daten nach. <http://www.gruenderszene.de/allgemein/moovit-user-daten>, 2015. Accessed: 2016-06-18.
- [81] Moovit. Moovit at a glance. <http://moovitapp.com/wp-content/uploads/2014/07/Moovit-At-A-Glance-USA-4-14.pdf>, 2014. Accessed: 2016-06-18.
- [82] Moovit. Moovit embarks on global expansion, picks up \$50m series-c led by nokia growth partners (ngp). <http://moovitapp.com/wp-content/uploads/2015/03/press-release-moovit-series-c-2015.pdf>, January 14, 2015 2015. Accessed: 2016-06-18.
- [83] BMW Group. Bmw i ventures unterstützt weltweite verbreitung von moovit, der app für die schnellsten strecken in echtzeit., January 14, 2015.
- [84] Zirra.com. Moovit. <https://www.zirra.com/companies/moovit>, 2016. Accessed: 2016-06-18.
- [85] Felix Wadewitz. "jeden tag kommt eine neue stadt dazu". <https://www.impulse.de/gruendung/geschaeftsideen/moovit/2135800.html>, 2015. Accessed: 2016-06-18.
- [86] Max Biederbeck. "ich will ein imperium errichten, das die welt verändert", sagt der ceo der mobilitäts-app moovit. <https://www.wired.de/collection/latest/moovit-ceo-nir-erez-verrat-im-wired-interview-seine-ehrgeizigen-plane>, 2015. Accessed: 2016-06-18.
- [87] Moovit. Moovit for developers - deeplinking docs. <http://www.developers.moovitapp.com/#!deeplinking-docs/on53h>, 2016. Accessed: 2016-06-18.
- [88] Moovit. Moovit help center. <https://moovitapp.zendesk.com/hc/en-us>, 2016. Accessed: 2016-06-18.
- [89] Nik Rawlinson. History of apple, 1976-2016: The story of steve jobs and the company he founded. <http://www.macworld.co.uk/feature/apple/history-of-apple-steve-jobs-what-happened-mac-computer-3606104/>, 2016. Accessed: 2016-06-18.
- [90] Apple Inc. Apple developer program - program membership details. <https://developer.apple.com/programs/whats-included/>, 2016. Accessed: 2016-06-18.
- [91] Yahoo! Finance. Apple inc. (aapl). <https://de.finance.yahoo.com/q/co?s=AAPL>, 2016. Accessed: 2016-06-18.

- [92] Apple Inc. Form 10-k (annual report). Report, October 28, 2015. Accessed: 2016-06-18.
- [93] Ben Eaton, Silvia Elaluf-Calderwood, Carsten Sorensen, and Youngjin Yoo. Distributed tuning of boundary resources: the case of apple's ios service system. *Mis Quarterly*, 39(1):217–243, 2015.
- [94] Tobias Brockmann, Stefan Stieglitz, and Arne Cvetkovic. Prevalent business models for the apple app store. In *Wirtschaftsinformatik*, pages 1206–1221, 2015.
- [95] Peter Burrows. How apple feeds its army of app makers. <http://www.bloomberg.com/news/articles/2011-06-08/how-apple-feeds-its-army-of-app-makers>, 13.06. 2011. Accessed: 2016-06-18.
- [96] Apple Inc. App store - app analytics. <https://developer.apple.com/app-store/app-analytics/>, 2016. Accessed: 2016-06-18.
- [97] Apple Inc. App store review guidelines. <https://developer.apple.com/app-store/review/guidelines/>, 2016. Accessed: 2016-06-18.
- [98] Apple Inc. ios developer program license agreement. https://developer.apple.com/programs/terms/ios/standard/ios_program_standard_agreement_20140909.pdf, 2014. Accessed: 2016-06-18.
- [99] Apple Inc. Apple developer agreement. https://developer.apple.com/programs/terms/apple_developer_agreement.pdf, 2015. Accessed: 2016-06-18.
- [100] City of Tampere. Information on tampere. <http://www.tampere.fi/en/city-of-tampere/information-on-tampere.html>, 2016. Accessed: 2016-06-18.
- [101] ePSI Platform. 2012 epsi open transport data manifesto. <https://de.scribd.com/document/111890372/Helsinki-Open-Transport-Data-Manifesto>, 2012. Accessed: 2016-06-18.
- [102] Mika Kulmala and Aki Lumiaho. Open data as enabler for its factory. In *20th ITS World Congress*, 2013.
- [103] Hermia Group. Its factory members. <http://arkisto.hermiagroup.fi/its-factory/members/>, 2016. Accessed: 2016-06-18.
- [104] ITS Factory. Its factory developer wiki. http://wiki.itsfactory.fi/index.php/ITS_Factory_Developer_Wiki, 2016. Accessed: 2016-06-18.
- [105] ITS Factory. Its factory developer newsletter 1/2015. http://wiki.itsfactory.fi/index.php/ITS_Factory_Developer_Newsletter_1-2015, 2015. Accessed: 2016-06-18.
- [106] ITS Factory. Its factory wiki - its standards. http://wiki.itsfactory.fi/index.php/ITS_Standards, 2016. Accessed: 2016-06-18.
- [107] ITS Finland. Its finland - about us. <http://www.its-finland.fi/index.php/en/mita-on-its/tietoa-meista.html>, 2016. Accessed: 2016-06-18.
- [108] Hermia Group. Its factory - how it works. <http://arkisto.hermiagroup.fi/its-factory/how-it-works/>, 2016. Accessed: 2016-06-18.
- [109] Melissa A. Schilling. Protecting or diffusing a technology platforms: tradeoffs in appropriability, network externalities, and architectural control. In Annabelle Gawer, editor, *Platforms, markets and innovation*, book section 8, pages 192–218. Edward Elgar, Cheltenham, UK, 2009.
- [110] Peter Erdmeier. "Intel Inside": *Ökonomische Analyse einer mehrstufigen Marketingstrategie*. Diplom.de, 2001.

Partner On- and Offboarding

Felix Michel and Florian Matthes

Department of Informatics, Technical University of Munich, Munich
{felix.michel; matthes}@tum.de

Abstract

Today, we use several online services and digital eco-systems in our daily life. The number of consumed services increases rapidly, e.g. we continually sign up for new mobile apps. Comparable to these small services, large eco-system need a structured process to engage and onboard partner or new services. Onboarding processes have often a lack of documentation due to time issues and a lack of seamless integrated tool support. We conducted a literature review in several domains, that apply on- and offboarding processes such as client or customer onboarding, online communities and open source projects. Additionally, we compared existing tool solutions regarding on- and offboarding support. Finally, we present a brief roadmap to improve the tool support to provide a seamless integration of email communication tools and documentation tools.

Keywords

Adaptive Case Management (ACM); Process Documentation; Knowledge Retrieval; Partnership Lifecycle

Introduction

Nowadays, on- and offboarding processes are applied in several different domains. Human resource departments use onboarding processes during hiring new employees. Online communities try to engaged individuals to participate in an online community. Mobile app providers try to onboard as many new users as possible. Eco-system approaches try to attract new partners to engage them to participate in the existing eco-system. On- and offboarding processes are often not well documented, one reason is the lack of tool integration.

Simple documents, shared documents or wikis are widely used for documentation purposes. Simple documents are not accessible to all stakeholders and instead sent around via email. Shared documents are a better approach but they have often a lack of collaboration features. Wikis provide these needed collaboration features but tend to be outdated due to the fact that they are often not integrated in the organizational processes.

Within the last years many communication tools arise but there is no common agreement on a dedicated communication tool. Email based communication is still widely used. According to [1] in 2015 over 205 billion emails sent and received every day. In the next 4 years, the average annual growth rate is approximately 3 percent. In the year 2019, approximately 246 billion emails will be sent and received every day.

On- and offboarding best practices

The term onboarding was initially used in the human resource domain and is defined as follows:

The term onboarding comes from the field of human resources and the common practice of new hire orientation. In that context, the steps in the process are often referred to as accommodate, assimilate, and accelerate all of which apply quite nicely to how new users ought to be treated in order to bring them into the fold. [2, p. 70]

Several different domains apply partly or fully supported on- and offboarding processes. The onboarding process of on-line communities, clients, employees and open source projects is illustrated more detailed in the following sections.

Online Communities

E. Kraut and P. Resnick 2011 [3] studied the building of successful on-line communities and collected principles from literature. F. Michel et al. 2015 [4] presents social design principles for a task-centered interface for on-line collaboration in science. The collaboration tool section illustrates the Organic Data Science framework more detailed. The social design principles are grouped according to categories such as: starting communities, encouraging contribution through motivation, encouraging commitment, dealing with newcomers, best practices from polymath, lessons learned from encode.

The landing page describes clearly the science and technical objectives of the project, displays a summary of currently active tasks, and shows the leadership and major contributors. In geosciences, the models used in the project are important to anchor the work for newcomers, so they are also shown in the main page. The models and contributor lists are dynamically generated from the current content with a semantic wiki query, so they are always up to date. Everyone can see the content of the site, and therefore the process being followed by the whole community, and the tasks being undertaken by different subgroups are open and accessible. In order to edit the content, users have to become contributors by getting a login and undergoing training.

A separate site is set up to train new users. This training site also uses the Organic Data Science framework, so it has the same features as presented above. A new user is given a set of predefined training tasks, each for learning and practicing a different feature of the interface. The training tasks follow the structure of the documentation pages, and allow new users to practice by using the same interface as they will use in the main site. As they complete the tasks, users can see the task status changing. The training is divided in two phases. The first phase trains them to contribute to existing tasks. The second phase trains them to create new tasks and to manage them as owners. One person in the collaboration is always assigned to help new users with their training, and is available by email to answer questions. This appointment rotates as new members become more experienced and can contribute in this capacity.

In the last year, many refugees arriving in Munich led to arising organizational issues. In cooperation with bavarian relief organizations we analyzed the process of managing and coordinating volunteers and provided software support. The software supports the collaborative process of management, coordination, and assignment of volunteers, including any legal aspects and privacy policies. The organizations can plan aspiring events easily and announce the necessary personnel requirements efficiently and exploit. This does not only help the organizations, but also the volunteers to save time. Integrated mechanisms are used (eg, email messages) to notify volunteers when new events announced based on their availability. In addition, the organizations see the current state of applications and requirements for the respective events and can thus purposefully plan and intervene quickly where necessary. To maximize benefits, the time to operation is critical, that includes the onboarding process of organizations as well as volunteers. In the first 4 weeks of operation, we onboarded 13 organizations and approximately 400 users on our platform. First, we invited all coordinators of the participating organizations to demonstrate the volunteer platform and afterwards we onboarded all coordinators. The coordinators themselves onboarded their local team from each organization. Finally, volunteers who support the refugees during certain activities

are onboarded under consideration of german legal regulations. The ongoing regularly onboarding process is much simpler, so that coordinators, team members and volunteers can be simply added.

Open Source Projects

C. Casalnuovo et al. 2003 [5] studied the developer onboarding in GitHub with focus on the role of prior social links and language experiences. They exposed the sociotechnical relationship between developers in GitHub and how this influences their decisions to join new projects. The authors clearly stated that most of the developers prefer to join projects, where they share a prior working relationship. This knowledge might be helpful to recruit new developers.

C. Crumlish et al. 2009 [6] studied the motivation of open source developers and describe different roles of the onion model regarding open source projects. Within commercial projects, roles are well defined, however, in open source projects there is no clear separation between users and developers. Depending on the contribution in an open source project, a user may contribute to the project and become a developer. Common roles are: 1) passive users, that use the system, comparable to commercial projects, 2) readers, use the system and try to understand how it works, 3) bug reporters, reports existing bugs, 4) bug fixers, fix reported or self found bugs, 5) peripheral developers, provide irregularly contribution to functionality or features, 6) active developers, contribute on a regularly base bugfixes or features, 7) core members, responsible and lead in development, 8) project leads, responsible to communicate the vision of the project.

I. Steinmacher et al. 2014 [7] studied how to attract, onboard, and retain newcomer developers in open source software projects. Most open source projects depend on voluntary contribution from different developers. Involving and engaging newcomers is one critical success factor of open source software projects. A common representation of the joining process is the onion model, that consists out of several layers like core developers, active developers, bugfixers, bug reporters and coordinators.

I. Herraiz et al. 2006 [8] studied the processes of joining in global distributed software projects. First they analyzed mailing lists of the GNOME project to explore the joining patterns regarding several roles. Within the group of core developers two joining patterns occurred. Most volunteer developers joined according the onion model, most hired and paid developed did not. Additionally, they found that more than half of the developers committed a change before they created a bug report. The authors present a developer joining model, that extend the onion model with several forces. Motivation is the driving factor for developers to contribute, e.g. better job opportunities, enjoyment, improving programming skills are some of the factors. The project attractiveness ex-

presses the effort, that is made to foster contributions from newcomers. This includes, the projects license type, project state, number of downloads and the number of open tasks. Hindering factors describe forces, that prevent contribution from newcomers. Furthermore they analyzed emailing lists and found that quickly answered questions have positive impacts. Retention describes the effort that is made to transform newcomers into long time developers. Identifying potential longtime developers helps to support them better.

F. Fagerholm et al. 2013 [9] studied the onboarding in open source software projects and present a preliminary analysis. The need for companies to participate in open source projects is rising in order to benefit from open source eco-system. Many challenges arise, such as how to onboard newcomers in projects effectively. The study analyzed within 12 weeks several different open source projects regarding the onboarding effectiveness with and without mentoring. The activity is measured by the number of commutes, pull requests and other interactions. A sample of 20 randomly selected mentored developers has a significant higher activity, than the not mentored developers. Hackathons are mentioned as an alternative approach to get involved in the community.

G. Krogh et al. 2003 [10] present a case study of community, joining, and specialization in open source software innovation. The authors studied the development community of a peer to peer filesharing network named Freenet. The study analyzed strategies how new members join an existing developer community and how new members initially contribute code. In general, software development is a knowledge-intensive activity and requires often domain knowledge to understand how to contribute. In the course of time, the complexity of projects grows and only less members are involved over a long time period, which creates an additional barrier. Based on the findings of interviews and analysis of member interactions a joiner script is presented to improve the onboarding process. The joiner script contains activities to become a community developer and is also described as cost to join the community. Newcomers who follow the joining script, are more likely to become community members.

Client and Employee Onboarding

L. Hemphill et al. 2011 [11] analyzed onboarding challenges in new virtual teams. The case study observed the onboarding of the first remote member in 5 different virtual teams. The authors found, proactive and self monitored employees integrate much faster. Additionally, rich media helps to reduce information and social disparities between team members. Frequently communicating teams with a lot of interaction integrate new employees much faster. Teams with structured regularly activities such as scrum perform better in onboarding new members.

Bauer et al 2011 [12] studied the organizational socialization, the effective onboarding of new employees. An onboarding

process helps employees to learn the knowledge, skills, and the corporate culture. Employees change their job on average approximately 10 times within 20 years and the trend is still rising. Therefore, socialization that leads to a positive altitude is important, employees retain longer and less effort for recruiting is needed. The authors present a socialization model with three main impact factors: 1) new employee characteristics, such as proactive personality, openness, veteran employee, 2) new employees behavior, such as information seeking, relationship building, feedback seeking, 3) organizational efforts, such as formal orientations, socialization tactics. Depending on the occurrence of the impact factors a new employee socialize faster or slower, that impacts the employee's satisfaction, commitment and performance.

N. Shah et al. 2008 [13] present a new approach to effective onboarding and best practices for retaining new employees. New hired employees struggle or even leave the company due to a lack of established connections, personal relationships and adapt the company culture. Leading companies have specialized onboarding processes depending on different factors such as: age, role, job and department. Advantages of well designed onboarding processes are: cost savings, speed up time to new hire productivity, lower retention rates through better employee integration. Best practices are:

- A: *Measure and assess onboarding effectiveness.*
- B: *Forge social networks to help new employees become culturally acclimated to the company.*
- C: *Provide training to help new hires become productive members of the workforce.*
- D: *Maximize operational compliance by using technology.*
- E: *Focus on federal and state compliance issues during onboarding to reduce regulatory risks.*
- F: *Assign someone to own the onboarding process and oversee all departmental stakeholders.*
- G: *Start onboarding during the recruitment process.*
- H: *Pace the delivery of onboarding details to avoid information overload.*

D. Nederlandse 2013 [14] illustrates examples of user onboarding in app design. Four different approaches are presented: 1) explanation of the value proposition, e.g. guide the user with attractive screens through the first steps, 2) explanation of functionalities, e.g. when apps have a lot of hidden functionality, should be possible to skip at any time flexible, 3) interactive explanation, users fulfill small task to interact with the product and learn how it works, users receive feedback like a "great job" to keep them motivated, 4) onboarding during use, most of the functionality is explained during its

first use, no information overload just what is exactly needed, 5) blank slate, e.g. dropbox explains how to use the favorite function unless there is the first favorite. In general, there is no best practice, the best fitting approach regarding to the usecase is selected.

G. R. Padmanabhan 2015 [15] states client onboarding is most critical within the customer relationship life cycle. Onboarding represents the interaction between the enterprise and the customer and leads to a first impression. Financial institutions need to meet many different regulations and compliance requirements during the onboarding process. Challenges of the current onboarding processes are: 1) disjointed client interactions, e.g. different channels are used to collect necessary information from clients, paper derive process, lack of documentation capabilities, poor customer experience, 2) inadequate process automation, result in unnecessary time delays and multiple handoffs among different departments increase the probability of error, 3) ineffective integration of enterprise systems, data redundancy, pocket information, no single point of truth.

S. Kemsley 2015 [16] describes characteristics of employee onboarding from the business perspective. The complexity of the employee onboarding processes vary widely depending on specific domain but there are most common characteristics:

- A: The core of the process is a customer folder, collecting all of the content and history of the customer onboarding journey, and providing context to knowledge workers*
- B: Data must be collected, usually by completing forms.*
- C: Supporting documents must be collected.*
- D: Internal checklists and procedures are applied to guide the process.*
- E: External regulations, e.g., FATCA, dictate which activities must be performed and which information must be collected.*
- F: Information may need to be entered in multiple systems, e.g., enterprise resource planning (ERP) or human resources (HR).*
- G: Third parties may be involved, e.g., credit ratings services.*

S. Kemsley 2015 [17] illustrates customer onboarding from the technological perspective. Requirements to describe a complex onboarding process are: 1) standard predefined processes, 2) ad-hoc tasks and checklists, 3) rules for compliance and best practices, 4) informational context via content and analytics events from external systems and devices participants and personas, 5) internal and external collaboration, 6) metrics and analytics.

Existing collaboration tools

The number of web collaboration tools increases continuously. Specialized collaboration tools exist for alternative purposes. Table 1 lists and categorizes relevant tools. Two relevant approaches, that will be described more detailed are the open source Organic Data Science approach that uses semantic structures to organize content and the Darwin wiki that supports the adaptive case management approach. Additionally, M. A. Marin et al. 2015 [18] provides a more detailed evaluation of existing approaches for knowledge intensive processes.

Organic Data Science

The Organic Data Science approach [19, 20, 21, 22, 4, 23] enables an open task centered on-line collaboration process. Key principles to address challenges of the task-centered collaboration approach are 1) the self-organization of the community through task decomposition, 2) an on-line community support based on social design principles and best practices and 3) an open science process to enable unanticipated contributions.

The task-centered Organic Data Science framework approach is implemented based on the Semantic MediaWiki platform. The prototype implementation of the Organic Data Science framework is evaluated through a research project focused on the science question of modeling the age of water in an ecosystem. This project requires expertise in different research areas from multiple organizations in different time-zones. Different collaboration dimensions are evaluated such as how many different users access a task, how many different users are assigned to a task, how many different users edit the task metadata and how many different users edit the task content. The findings show that the framework supports the collaboration process.

In general the Organic Data Science framework is designed for helping scientists to collaborate to solve complex scientific research questions. The use of the Organic Data Science framework is not limited to scientific purposes, it helps to support complex knowledge intensive collaborative processes.

Figure 1 illustrates the Organic Data Science user interface. On the left the task navigation is shown to drill down to a certain tasks that requires action. On the right a task page is illustrated with the related context task, metadata and task content itself. A more detailed description of the core features is shown in Figure 2.

Darwin / SocioCortex

M. Hauder et al. 2015 [24] presents darwin, a wiki based task-centered tool with adaptive case management support. Darwin enables end users to structure knowledge-intensive processes easily without knowledge about the holistic process. Limited modelling capabilities are provided for modelling experts. Processes emerge during execution and model experts can create templates for repeatable processes afterwards. Several

concepts are used to structure content. Unstructured content is represented with simple wiki pages. Structured content is organized with attributes and tasks. Every wiki page contains key value pairs named attributes that describe the unstructured content more precisely. Several attributes can be assigned to a tasks to define an artefact creation process. The state of every task is visualized with pie chart icons. Additionally, a timeline of all task of a page represents the time dependencies of tasks. A concept of types is introduced to create templates for repeatable workflows. A CMNN Editor visualizes the workflow to support the modeler with the template creation or modification process. These templates are instantiated an adapted to individual needs of end users. Compared to the Organic Data science approach a wiki page describes an artefact creation process. Many tasks are assigned to one page instead of one task per page. Additionally, predefined templates are supported. Figure 3 illustrates the darwin user interface with structuring concepts and figure 4 shows the task centered features precisely.

Email knowledge extraction

In the recent years, the scientific community of email-mining and knowledge-retrieval focused on spam detection, email categorization, content analysis, network property analysis and visualizations as summarized in Table 2.

Y. Ye et al. 2003 [26] present a model to understand email usage and predict actions on a specific message. The authors conducted a literature review and classified the email activities into different categories: 1) project management, including task delegations and reminders, 2) information exchange, including information storage and retrieval, 3) scheduling and planing, e.g., organizing a meeting with external partners, 4) social communication, comparable to instant messaging within an enterprise or with family and friends. A survey with 124 participants from the Carnegie Mellon University has been conducted. The participants were split into approximately one third scientists, one third scientific programmer and one third graduates. In average, the age of participants was 30 years. In general, the mean of read messages per participant and day is 30 and the mean of wrote messages is 14. Regarding the number of emails left in the inbox, there is a huge distribution according to the job role of the participants (median of 105), in average 1336 emails left. E.g., 2.5% have more than 10.000 email in the inbox.

The study illustrated there are three common habit patterns found: 1) keep the inbox size small, 2) move messages into folders after they are read, 3) just leave messages in the inbox. Moreover, the study categorized the messages according their content into the dimensions: 1) action requests, 2) info requests, 3) info attachment, 4) status update, 5) scheduling, 6) reminder, 7) social, 8) other content. Not surprisingly, approximately one third of all messages contains an action

request. Furthermore, messages are classified according to possible actions. Classification categories are: 1) need reply (immediate reply, postponed reply), 2) no reply action.

The authors state, that the importance of messages is relevant to predict the most likely actions of messages. Several models are presented that use different combinations out of sender importance, message content and impact to predict the probability of reply.

C. Di Ciccio et al. 2011 [65] analyzed email messages for mining artful collaborative processes. Artful processes are typically executed by knowledge workers who work mentally and create an artful process on the fly. Email conversations are typically used to share information among knowledge workers. Conversation of email messages represent process traces. The authors present the mail of mine approach that extracted formal processes from collections of email messages. A formal representation is generated without effort of the knowledge workers. Possible applications such as personal information management, information warfare and enterprise engineering are named. The presented approach uses declarative workflows and regular grammar to repeat the extracted process. Email conversations are clustered with object matching algorithms to match activities and tasks. First emails are stored in a database and clustered to conversations. In the next step, footers and reply texts are removed from messages to extract the key parts. A similarity object matching algorithm is applied to combine different data sources like XML files and database tuples. Distance metrics are applied to compare message objects based on message id, subject, sender, reciever and body including the names of the attached files. This results in a cluster of messages. Based on additional regular expressions processes are formed.

Category	Publication
Spam detection	[27, 28, 29, 30, 31, 32, 33, 30, 34, 35, 36]
Categorization	[37, 38, 39, 40, 41, 42]
Content analysis	[43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53]
Network property analysis	[54, 55, 56, 57]
Visualization	[58, 59, 60, 61, 45]
Other Tasks	[62, 63, 64, 65]

Table 2. Discussed email mining approaches in research papers categorized according to accomplished tasks, adapted and extended from [66].

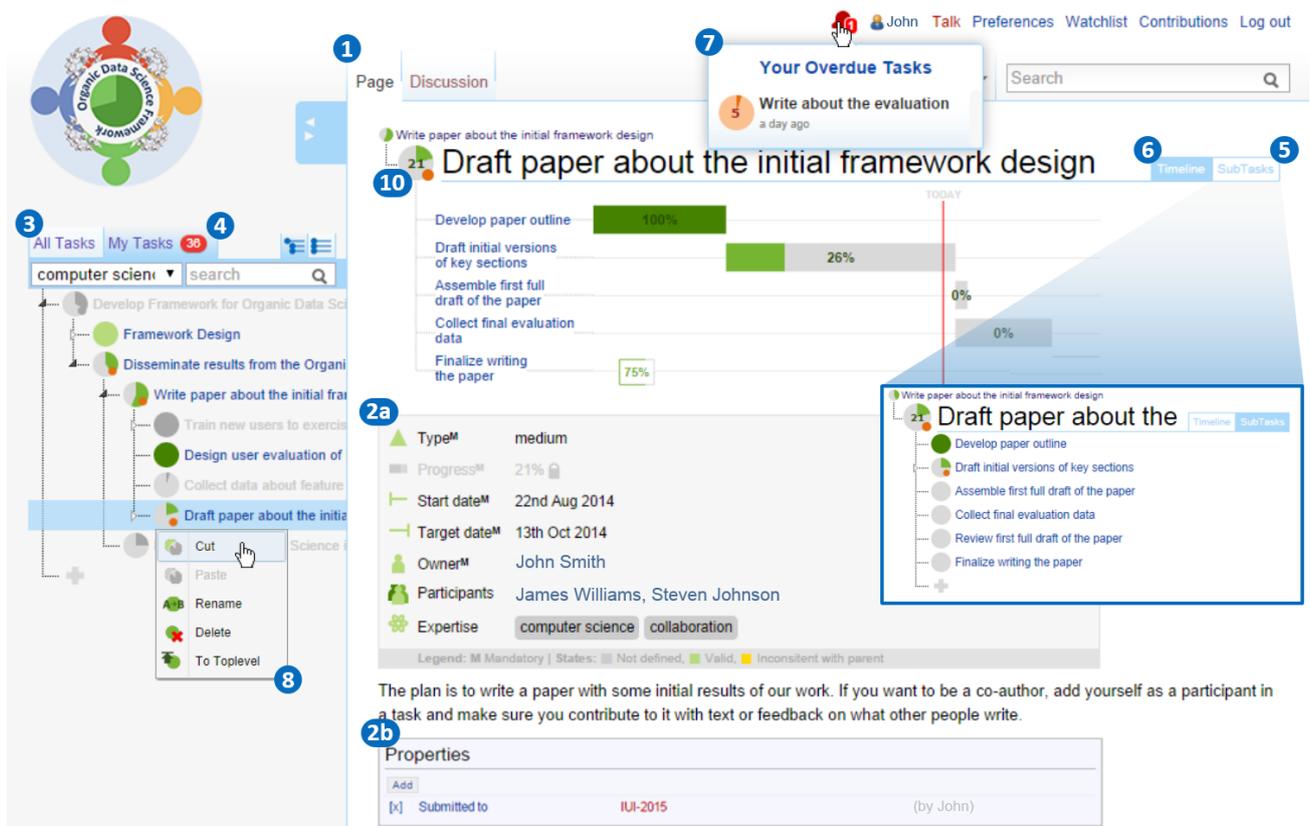


Figure 1. Organic Data Science user interface [4].

① **Welcome Page:** Describes clearly the science and technical project objectives, summarizes currently active tasks, and shows lead contributions (not shown).

② **Task Representation:** Tasks have a unique identifier (URL), and are organized in a hierarchical subtask decomposition structure.

③ **Task Metadata:** We distinguish between a) required metadata that is needed to progress a task and b) optional metadata that the users provides with structured properties. The structured properties can contain any key value pairs that are helpful to provide structured content.

④ **Task Navigation:** Tasks can expand until a leaf task is reached. Additionally users can search for task titles and apply an expertise filter.

⑤ **Personal Worklist:** The worklist contains the subset of tasks from the task navigation for which the user is owner or a participant. A red counter indicates the current number of tasks in the user's worklist.

⑥ **Subtask Navigation:** Subtasks of the currently opened task are presented. Filter and search options are not provided in this navigation.

⑦ **Timeline Navigation:** All subtasks are represented based on their start, target times, and completion status in a visualization based on a Gantt chart.

⑧ **Task Alert:** Signals when a task is not completed and the target date passed. A red counter next to the alert bell indicate the number of overdue tasks.

⑨ **Task Management:** The interface supports creating, renaming, moving and deleting tasks. For usability reasons, all these actions can be reversed.

⑩ **User Tasks and Expertise:** The interface allows users to easily see what others are working on or have done in the past. This creates a transparent process (not shown).

⑪ **Task State:** Small icons visualize the state of each task intuitively. Tasks with incomplete required metadata are represented with a cycle and tasks with completed required metadata are represented with a pie chart. The progress is indicated in green.

⑫ **Train New Members:** A separate site is used to train new users in a sandbox environment, where training tasks are explicit. The training is split into two parts: 1) Users who participate on tasks and 2) User who own tasks (not shown).

Figure 2. Organic Data Science core features, highlighted in figure 1 [4].

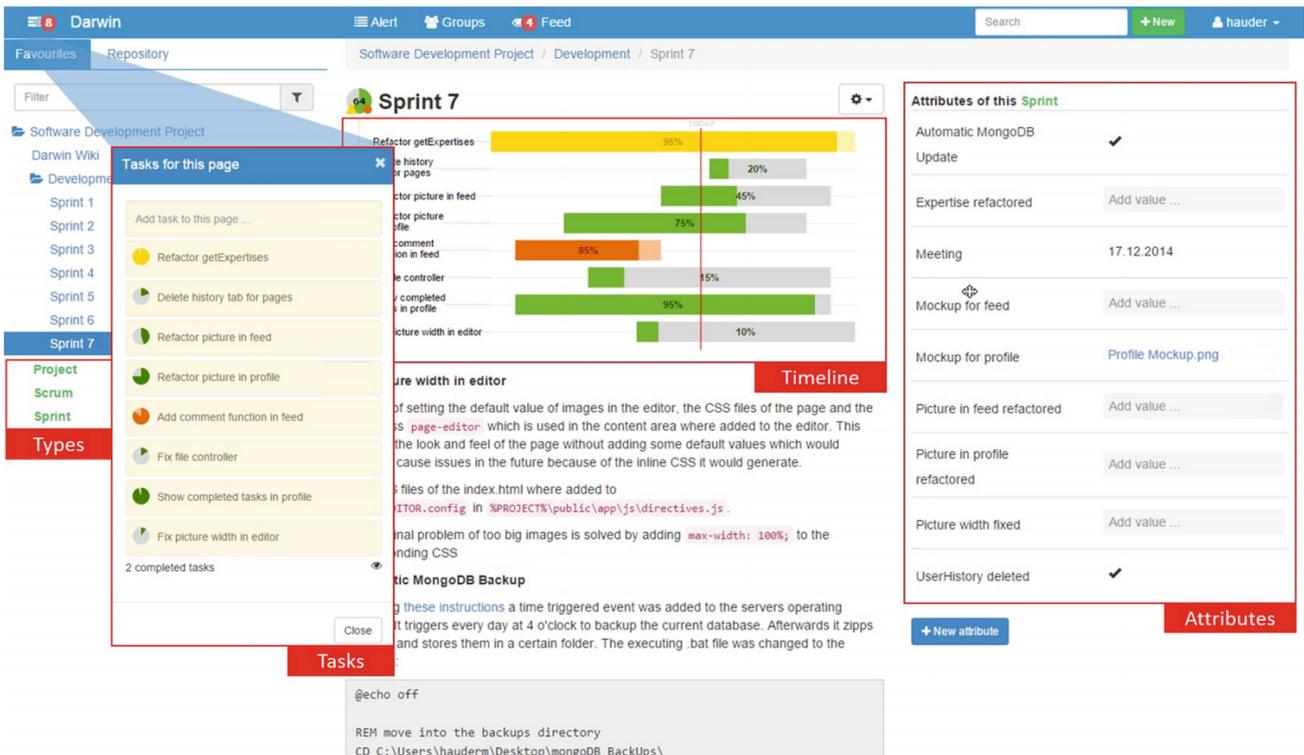


Figure 3. Darwin structuring concepts [24].

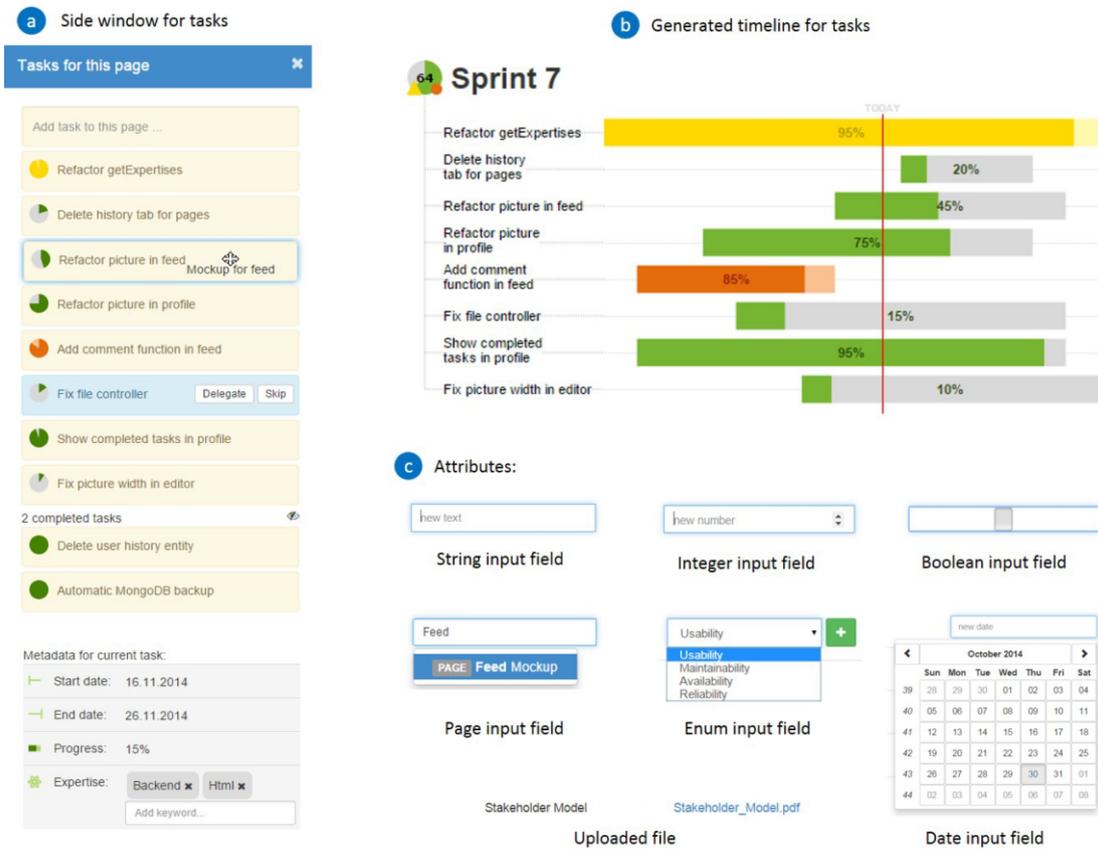


Figure 4. Darwin, task centered features [24].

Category	Tool	Extendable (open source, plug-ins, API)	Mainly task oriented	Meta model or semantic organization	3rd party contextual task information	Link (last accessed on June 2016)
Wikis	MediaWiki	✓	-	-		http://www.mediawiki.org/
	Semantic MediaWiki	✓	-	✓		https://semantic-mediawiki.org/
	Organic Data Science	✓	✓	✓		http://organicdatascience.org/ [19, 20, 21, 22, 4, 23]
Enterprise Knowledge Management	Confluence	✓	✓	-		https://de.atlassian.com/software/confluence
	Connections	✓	✓	-		http://www-03.ibm.com/software/products/en/conn
	Jive	✓	✓	-		http://de.jivesoftware.com/
	MS SharePoint	✓	✓	-		http://office.microsoft.com/en-us/sharepoint/
	Communote	✓	✓	-		http://www.communote.com/homepage/
	Yammer	✓	-	-		https://www.yammer.com/
	Redmine	✓	✓	-		http://www.redmine.org/
	JIRA	✓	✓	-		https://de.atlassian.com/software/jira
SocioCortex/Darwin	✓	✓	✓		http://sociocortex.com/ [24]	
ToDo Lists	Trello	✓	✓	-		https://trello.com/
	Keep	-	✓	-		https://keep.google.com/keep/
	Todoist	✓	✓	-		http://todoist.com/
	Wunderlist	✓	✓	-		https://www.wunderlist.com/de/
Email	GMail	✓	✓	-		https://mail.google.com/
	Apple Mail	✓	✓	-		https://www.apple.com/de/support/mac-apps/mail/
	Thunderbird	✓	✓	-		https://www.mozilla.org/de/thunderbird/
	Notes	✓	✓	-		http://www-03.ibm.com/software/products/de/ibmnotes
	Outlook	✓	✓	-		https://www.microsoft.com/de-de/outlook-com/
Messaging	Slack	✓	-	-		https://slack.com/
	Telegram	✓	-	-		https://telegram.org/
	Facebook at Work	✓	-	-		https://work.fb.com/ [25]

Table 1. Existing collaboration tools adapted from [21].

Conclusion and Outlook

Onboarding processes are applied in several different domains such as human resources, open source projects and service eco-systems. Several domains provide onboarding guidelines to structure processes. A email based communication strategy to share organizational process knowledge represents one of the best practices. The process documentation is mostly detached from the communication part. Our tool study indicates a need for seamless integration of process documentation and communication. In the next step we must identify the best subset of tools to demonstrate a prototypical seamless integration, e.g. as illustrated in figure 5. Every set of tools necessarily contains an email client or an email client plug-in and a knowledge base to provide context and process support for the email based communication.

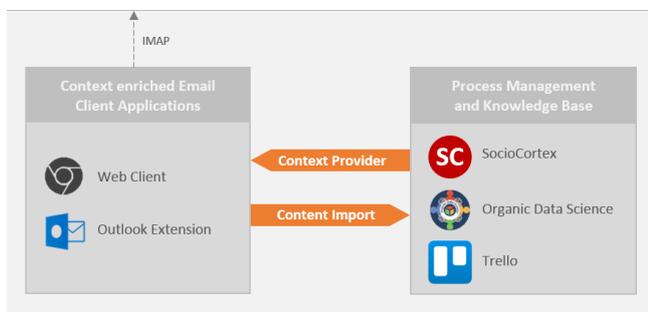


Figure 5. Context enriched email client to provide a seamless tool interaction, supporting on- and offboarding.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] Inc. The Radicati Group. Email statistics report, 2015-2019, 2016.
- [2] Christian Crumlish and Erin Malone. *Designing Social Interfaces - Principles, Patterns, and Practices for Improving the User Experience*. O'Reilly Media, February 2009.
- [3] E. Kraut and Paul Resnick. *Building Successful Online Communities: Evidence-Based Social Design*. MIT Press, February 2011.
- [4] Felix Michel, Yolanda Gil, Varun Ratnakar, and Matheus Hauder. A task-centered interface for on-line collaboration in science. In *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*, pages 45–48. ACM, 2015.
- [5] Casey Casalnuovo, Bogdan Vasilescu, Premkumar Devanbu, and Vladimir Filkov. Developer onboarding in github: the role of prior social links and language experience. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pages 817–828. ACM, 2015.
- [6] Yunwen Ye and Kouichi Kishida. Toward an understanding of the motivation of open source software developers. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 419–429. IEEE, 2003.
- [7] Igor Steinmacher, Marco Aurélio Gerosa, and D Redmiles. Attracting, onboarding, and retaining newcomer developers in open source software projects. In *Workshop on Global Software Development in a CSCW Perspective*, 2014.
- [8] Israel Herraiz, Gregorio Robles, Juan José Amor, Teófilo Romera, and Jesús M González Barahona. The processes of joining in global distributed software projects. In *Proceedings of the 2006 international workshop on Global software development for the practitioner*, pages 27–33. ACM, 2006.
- [9] Fabian Fagerholm, Peter Johnson, Alejandro Sanchez Guinea, Jay Borenstein, and Jurgen Munch. Onboarding in open source software projects: A preliminary analysis. In *Global Software Engineering Workshops (ICGSEW), 2013 IEEE 8th International Conference on*, pages 5–10. IEEE, 2013.
- [10] Georg Von Krogh, Sebastian Spaeth, and Karim R Lakhani. Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, 32(7):1217–1241, 2003.
- [11] Libby Hemphill and Andrew Begel. Not seen and not heard: Onboarding challenges in newly virtual teams. 2011.
- [12] Bauer, Talya N, and Berrin Erdogan. Organizational socialization: The effective onboarding of new employees. 2011.
- [13] Susan J. Leandri Christine Perovich Barbara Hower Nik Shah, Scott Pollak. Best practices for retaining new employees: New approaches to effective onboarding. 2008.
- [14] D. Nederlandse. Great examples of user onboarding in app design. <http://www.sodastudio.nl/kennis-ideeen/great-examples-of-user-onboarding-in-app-design>, 2013. Accessed: 2016-02-21.
- [15] Ganesh Raghavan Padmanabhan. Client onboarding: Digitize to optimize. <http://www.tcs.com/SiteCollectionDocuments/White-Papers/Client-Onboarding-Digitize-Optimize-0715-1.pdf>, 2015.

- [16] Sandy Kemsley. Customer onboarding with bpm: The business view, 2015.
- [17] Sandy Kemsley. Customer onboarding with bpm: The technology implementation, 2015.
- [18] Mike A Marin, Matheus Hauder, and Florian Matthes. Case management: An evaluation of existing approaches for knowledge-intensive processes. 2015.
- [19] Felix Michel, Yolanda Gil, and M Hauder. A virtual crowdsourcing community for open collaboration in science processes. In *Americas Conference on Information Systems (AMCIS)*, 2015.
- [20] Yolanda Gil, Felix Michel, Varun Ratnakar, Matheus Hauder, Christopher Duffy, Hilary Dugan, and Paul Hanson. A task-centered framework for computationally-grounded science collaborations. In *e-Science (e-Science), 2015 IEEE 11th International Conference on*, pages 352–361. IEEE, 2015.
- [21] Yolanda Gil, Felix Michel, Varun Ratnakar, and Matheus Hauder. A semantic, task-centered collaborative framework for science. In *The Semantic Web: ESWC 2015 Satellite Events*, pages 58–61. Springer, 2015.
- [22] Yolanda Gil, Felix Michel, Varun Ratnakar, Jordan Read, Matheus Hauder, Christopher Duffy, Paul Hanson, and Hilary Dugan. Supporting open collaboration in science through explicit and linked semantic description of processes. In *The Semantic Web. Latest Advances and New Domains*, pages 591–605. Springer, 2015.
- [23] Yolanda Gil, Varun Ratnakar, and Paul C Hanson. Organic data publishing: a novel approach to scientific data sharing. In *Proceedings of the 2nd international workshop on linked science*, volume 951, 2012.
- [24] Matheus Hauder, Rick Kazman, and Florian Matthes. Empowering end-users to collaboratively structure processes for knowledge work. In *Business Information Systems*, pages 207–219. Springer, 2015.
- [25] Ibrahim Evsan. Facebook at work will die kommunikation in unternehmen revolutionieren. <http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm>, 2016. Accessed: 2016-05-12.
- [26] Laura A Dabbish, Robert E Kraut, Susan Fussell, and Sara Kiesler. Understanding email use: predicting action on a message. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 691–700. ACM, 2005.
- [27] David Heckerman, Eric Horvitz, Mehran Sahami, and Susan Dumais. A bayesian approach to filtering junk e-mail. In *Proceeding of AAAI-98 Workshop on Learning for Text Categorization*, pages 55–62, 1998.
- [28] Ion Androutsopoulos, John Koutsias, Konstantinos V Chandrinou, and Constantine D Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167. ACM, 2000.
- [29] Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Georgios Sakkis, Constantine D Spyropoulos, and Panagiotis Stamatopoulos. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. *arXiv preprint cs/0009009*, 2000.
- [30] Minoru Sasaki and Hiroyuki Shinnou. Spam detection using text clustering. In *Cyberworlds, 2005. International Conference on*, pages 4–pp. IEEE, 2005.
- [31] Gordon Rios and Hongyuan Zha. Exploring support vector machines and random forests for spam detection. In *CEAS*, 2004.
- [32] Harris Drucker, Donghui Wu, and Vladimir N Vapnik. Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*, 10(5):1048–1054, 1999.
- [33] Salvatore J Stolfo, Shlomo Hershkop, Ke Wang, Olivier Nimeskern, and Chia-Wei Hu. A behavior-based approach to securing email systems. In *Computer Network Security*, pages 57–81. Springer, 2003.
- [34] Luíz Henrique Gomes, Fernando DO Castro, Virgílio AF Almeida, Jussara M Almeida, Rodrigo B Almeida, and Luis MA Bettencourt. Improving spam detection based on structural similarity. *SRUTI*, 5:12–12, 2005.
- [35] Jennifer Golbeck and James A Hendler. Reputation network analysis for email filtering. In *CEAS*, 2004.
- [36] Bradley Taylor. Sender reputation in a large webmail service. In *CEAS*, 2006.
- [37] Richard B Segal and Jeffrey O Kephart. Mailcat: an intelligent assistant for organizing e-mail. In *Proceedings of the third annual conference on Autonomous Agents*, pages 276–282. ACM, 1999.
- [38] William W Cohen. Learning rules that classify e-mail. In *AAAI spring symposium on machine learning in information access*, volume 18, page 25. California, 1996.
- [39] Jason Rennie. i le: An application of machine learning to e-mail filtering. 1998.
- [40] Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *Machine learning: ECML 2004*, pages 217–226. Springer, 2004.
- [41] Carman Neustaedter, AJ Brush, and Marc A Smith. Beyond from and received: Exploring the dynamics of email triage. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1977–1980. ACM, 2005.

- [42] Steve Whittaker and Candace Sidner. Email overload: exploring personal information management of email. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 276–283. ACM, 1996.
- [43] Malcolm W Corney, Alison M Anderson, George M Mohay, and Olivier de Vel. Identifying the authors of suspect email. 2001.
- [44] Vitor R Carvalho and William W Cohen. Ranking users for intelligent message addressing. In *Advances in Information Retrieval*, pages 321–333. Springer, 2008.
- [45] Joshua R Tyler, Dennis M Wilkinson, and Bernardo A Huberman. E-mail as spectroscopy: Automated discovery of community structure within organizations. *The Information Society*, 21(2):143–153, 2005.
- [46] Maayan Roth, Assaf Ben-David, David Deutscher, Guy Flysher, Ilan Horn, Ari Leichtberg, Naty Leiser, Yossi Matias, and Ron Merom. Suggesting friends using the implicit social graph. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 233–242. ACM, 2010.
- [47] Christopher S Campbell, Paul P Maglio, Alex Cozzi, and Byron Dom. Expertise identification using email communications. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 528–531. ACM, 2003.
- [48] Ralf Hölzer, Bradley Malin, and Latanya Sweeney. Email alias detection using social network analysis. In *Proceedings of the 3rd international workshop on Link discovery*, pages 52–57. ACM, 2005.
- [49] Lisa Johansen, Michael Rowell, Kevin RB Butler, and Patrick Drew McDaniel. Email communities of interest. In *CEAS*, 2007.
- [50] Ryan Rowe, German Creamer, Shlomo Hershkop, and Salvatore J Stolfo. Automated social hierarchy detection through email network analysis. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 109–117. ACM, 2007.
- [51] Parambir S Keila and David B Skillicorn. Structure in the enron email dataset. *Computational & Mathematical Organization Theory*, 11(3):183–199, 2005.
- [52] Marco Stuit and Hans Wortmann. Discovery and analysis of e-mail-driven business processes. *Information Systems*, 37(2):142–168, 2012.
- [53] Byron Dom, Iris Eiron, Alex Cozzi, and Yi Zhang. Graph-based ranking algorithms for e-mail expertise analysis. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 42–48. ACM, 2003.
- [54] Andrea Lockerd and Ted Selker. Driftcatcher: The implicit social context of email. In *INTERACT*, 2003.
- [55] Christian Bird, Alex Gourley, Prem Devanbu, Michael Gertz, and Anand Swaminathan. Mining email social networks. In *Proceedings of the 2006 international workshop on Mining software repositories*, pages 137–143. ACM, 2006.
- [56] Thomas Karagiannis and Milan Vojnovic. Behavioral profiles for advanced email features. In *Proceedings of the 18th international conference on World wide web*, pages 711–720. ACM, 2009.
- [57] Munmun De Choudhury, Winter A Mason, Jake M Hoffman, and Duncan J Watts. Inferring relevant social networks from interpersonal communication. In *Proceedings of the 19th international conference on World wide web*, pages 301–310. ACM, 2010.
- [58] Gina Danielle Venolia and Carman Neustaedter. Understanding sequence and reply relationships within email conversations: a mixed-model visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 361–368. ACM, 2003.
- [59] Steve Whittaker, Tara Matthews, Julian Cerruti, Hernan Badenes, and John Tang. Am i wasting my time organizing email? a study of email refinding. 2011.
- [60] Adam Perer and Marc A Smith. Contrasting portraits of email practices: visual approaches to reflection and analysis. In *Proceedings of the working conference on Advanced visual interfaces*, pages 389–395. ACM, 2006.
- [61] Fernanda B Viégas, Scott Golder, and Judith Donath. Visualizing email content: portraying relationships from conversational histories. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 979–988. ACM, 2006.
- [62] Steffen Bickel and Tobias Scheffer. Learning from message pairs for automatic email answering. In *Machine Learning: ECML 2004*, pages 87–98. Springer, 2004.
- [63] Tobias Scheffer. Email answering assistance by semi-supervised text classification. *Intelligent Data Analysis*, 8(5):481–493, 2004.
- [64] Yi Cui, Jian Pei, Guanting Tang, Wo-Shun Luk, Daxin Jiang, and Ming Hua. Finding email correspondents in online social networks. *World Wide Web*, 16(2):195–218, 2013.
- [65] Claudio Di Ciccio, Massimo Mecella, Monica Scannapieco, Diego Zardetto, and Tiziana Catarci. Mailofmine - analyzing mail messages for mining artful collaborative processes. In *Data-Driven Process Discovery and Analysis*, pages 55–81. Springer, 2011.
- [66] Guanting Tang, Jian Pei, and Wo-Shun Luk. Email mining: tasks, common techniques, and tools. *Knowledge and Information Systems*, 41(1):1–31, 2014.

Crowdsourcing and Crowdinnovation

Anne Faber and Florian Matthes

Department of Informatics, Technical University of Munich, Munich
{anne.faber; matthes}@tum.de

Abstract

For a successful establishment of a mobility ecosystem, the integration of relevant user groups is necessary. The attractiveness of the mobility ecosystem depends on a balanced participation of service users and services provided. In such a mobility ecosystem end-users are not only data evaluators as participants but also data sources, as they may contribute to the ecosystem by providing own traveling data and views regarding their mobility preferences. In this report crowdsourcing is defined and analysed based on a literature review. Furthermore, crowdsourcing initiatives in the mobility context are assessed for its relevance regarding the TUM Living Lab Connected Mobility.

Keywords

Crowdsourcing; Crowdinnovation; Mobility

Introduction

Crowdsourcing is a widely used umbrella term, used for a variety of procedures, which all include the involvement of a large group of people to gain from their resources. Jeff Howe [1] defined crowdsourcing in 2008 as

Crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call.

Thereby, *crowdsourcing* is a combination of *crowd* and *outsourcing* [2]. Thus, by making use of crowdsourcing the companies and businesses open up their boundaries to seek for new or additional workers and ideas.

Even though the term crowdsourcing was used for the first time in a blog post by Howe even two years earlier, the principle of crowdsourcing has been applied for centuries [3]. A very early example of the use of crowdsourcing is the “Longitude Prize” of £ 20,000, which was offered by the British Government in 1714 to anyone who can develop a reliable way to compute longitude [3]. Another example is Toyota, which ran a crowdsourcing initiative in 1936 to find a new logo design. The crowdsourcing initiative even led to the change in the brand name “TOYODA” to “TOYOTA” [4].

Web 2.0 applications have been an important enabler for crowdsourcing as broadcasting a problem statement over the internet allows to reach a large group of people. By making use of the internet, people all over the world can access the stated problem and apply their skills to solve the problem [2]. Since then, the usage of crowdsourcing has been widely applied and is still rising. In 2014, brands in the Fast Moving Consumer Good (FMCG) increased their investments by 46 percent compared to 2013, whereby PepsiCO increased it even

by 325 percent [5]. In 2015 Coca-Cola, Danone and Nestlé raised their usage of crowdsourcing even further compared to the previous year [6].

Research approach

Research question

To better understand the role of crowdsourcing and crowdinnovation within the TUM Living Lab Connected Mobility (TUM LLCM) project, the following research questions are addressed in this state-of-the-art report

1. How are the terms crowdsourcing and crowdinnovation defined and which characteristics define it?
2. What are existing crowdsourcing applications in the mobility context?

In order to answer these questions, we provide a general understanding of crowdsourcing and the aspects it consists of, followed by a more detailed evaluation regarding crowdsourcing in the mobility context.

Research process

The research questions were addressed using different digital libraries and index systems, such as IEEE Computer Social Digital Library (ieeexplore.ieee.org), ACM (dl.acm.org) and Scopus (scopus.com). The results by searching for “crowdsourcing” ranged from 1.246 results up to 4.861¹ results, where in most cases the first paper was published in 2008. Table 1 documents the precise search results.

Due to the amount of literature available, the literature review process proposed by Webster and Watson [7] was applied: first major contributions in leading journals were identified, second a backward search was conducted, and third

¹Research conducted in April 2016

Table 1. Overview of research results regarding crowdsourcing literature as of April 2016

Consulted database	Search term “crowdsourcing“
IEEE	1.246
ACM	2.657
Scopus	4.861
SpringerLink	4.491

a forward search contemplated the research.

This report is structured as follows: first, a general introduction on crowdsourcing covering an extended comprehensive definition, a crowdsourcing process, a taxonomy of crowdsourcing metrics, a description of crowdsourcing applications as a result of a systematic literature review, and the different crowdsourcing information systems is presented. Second the incentivisation of the crowd, as one key success factor of crowdsourcing initiatives, is described in detail, including the motivation method gamification, which is introduced and explained. After this crowdsourcing is analysed regarding its use in the mobility context, addressing the TUM LLCM framework. Thereby, crowdsourcing mobility applications coming from either a research or industrial background are presented.

Crowdsourcing

Comprehensive definition of crowdsourcing

Crowdsourcing procedures are applied in various fields, with different interpretations and intentions. Estelles-Arolas and Gonzalez-Ladron-de Guevara [8] addressed the missing comprehensive definition and categorization of crowdsourcing in 2012, and enlarged the definition of Howe to

Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, with variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage what the user has brought to the venture, whose form will depend on the type of activity undertaken.

This definition is based on 132 analysed documents, identifying three elements: crowd, initiator, and process, from which

they extracted eight characteristics:

Crowd:

- *Who forms the crowd* - Crowd is understood as a large group of individuals, whereby the optimum number of people depends on the crowdsourcing initiative. The same holds true for the heterogeneity and knowledge required from the crowd.
- *What has the crowd to do* - The aim of the crowdsourcing initiatives is to resolve a broadcasted problem or task, whereby its level of difficulty differ greatly, starting with an almost trivial task to expert addressing tasks, including creative and innovative tasks.
- *What does the crowd get in return* - The compensation the crowd receives for the resolution of the task vary depending on the crowdsourcer, but it should always compensate at least one of the individual needs. Maslow [9] described them as: economic reward, i.e. financial or token reward, social recognition, i.e. acknowledgment of the person’s contribution by a group, self-esteem or to develop individual skills, i.e. creative skills, or the improvement of a product. Thereby the return should address the motivation of the crowd to contribute to the crowdsourcing initiative. Because the motivation of the crowd is one key success factor, this topic is also addressed in a separate section.

Initiator:

- *Who is the initiator (crowdsourcer)* - Crowdsourcers range from companies and businesses, which is the largest initiator group, to public institutions, down to single persons, who for example are looking for investors via crowdfunding in private projects or start up initiatives. More and more crowdsourcing platforms are available on the internet, offering crowdsourcers the possibility to broadcast their problems.
- *What does the initiator get in return* - In an optimal case the crowdsourcer will get the solution of the published problem as a return of the crowdsourcing initiative. The input of the crowd varies from their knowledge, experience or even their money in case of crowdfunding. Further, the crowdsourcers, especially if they are large companies, get publicity as a return of their crowdsourcing initiative.

Process:

- *What type of process is it* - In literature the type of process addressed by crowdsourcing varies from problem-solving processes ([10], [11]), outsourcing processes ([12], [13]), or open innovation processes [14], to name just a few. For Estelles-Arolas and Gonzalez-Ladron-de Guevara [8] the overall common characteristic is that crowdsourcing is an online process, making use of the

internet. Nevertheless, there are also crowdsourcing initiatives using Short Messages Services, especially in countries with limited access to internet and smart-phones [15].

- *What type of call to use: open call* - Crowdsourcing is by definition the usage of an open call, whereby the execution of openness differs. Withla [16] identified three types of call, namely a truly open call where anyone who is interested in completing the task is invited to participate; limited to a particular community that is prescreened to have some particular knowledge or expertise or fit into a special target group; or a combination of both, so that for an example an open call is conducted first, and the participants are selected out of this large group of applicants.
- *What medium is used* - According to Estelles-Arolas and Gonzalez-Ladron-de Guevara [8] the medium used is without a doubt the internet, which enables the collaboration of the crowd. Nevertheless, in cases Short Message Services (SMS) are used, the medium used is the mobile phone with the according mobile operator [15]. Hosseini et al. [17] even consider crowdsourcing activities being performed in a real environment, not on-line, thus using no electrical communication medium.

According to the results which a crowdsourcer wants to achieve by applying crowdsourcing different characteristics will be chosen. The composition of the crowd, for example, can vary from a defined group of people all working for the same company to a truly open initiative with no restriction regarding the participation. The crowd can also vary in its qualification, which is needed or not. The qualification is in close relation to the purpose of integrating the crowd. Compared to microtasks, which will be described in detail later, the crowdsourced tasks offered on crowdsourcing platforms such as Freelancer (freelancer.com) required a certain qualification level of the crowd and qualified workers.

Crowdsourcing process

In case the crowdsourcer uses an intermediary crowdsourcing platform, the corresponding process consists of five major phases according to Mudhi et al. [18] and is visualised in Figure 1.

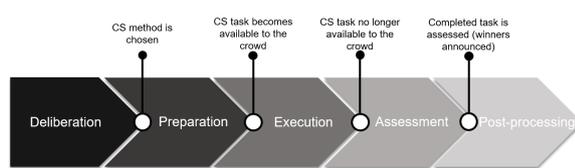


Figure 1. Five important phases of the intermediary mediated crowdsourcing (CS) process, based on [18]

In the first phase is *deliberation*, in which the crowdsourcer decides to use crowdsourcing to seek for external

workers and which crowdsourcing platform to use. Thereby the offers of the different platforms should be considered to achieve the expected results, including the community profile and the proposed incentives. The *preparation* phase covers all the necessary groundwork to accomplish before the execution starts [18]. Amongst others the crowdsourced task is concretised, if needed divided into minor subtasks and clearly formulated to avoid misunderstandings during the execution. According to Muhdi et al. [18] the clear formulation is the most challenging task. Further, the timing of publication and duration of the crowdsourcing initiative and the applied incentives, if not defined within the crowdsourcing platform, are determined. The *execution* phase begins when the task is available to the crowd. Members of the crowd individually or collaboratively process the task. The crowdsourcer can communicate with crowdsourced workers during the execution and is also able to assess first results. The phase ends when the processing time is over, and the tasks become unavailable for further actions. The submitted results are evaluated by the crowdsourcer in the *assessment* phase. In case a prize money for a winner solution was provided, this will be passed to the winner or the winning team. During the final *post-processing* phase the crowdsourcer should interpret and start implementing the results to achieve an optimal outcome of the initiative. Also, the crowdsourcing process as a whole should be assessed to build in future on lessons learned.

To achieve a clear picture of the different stakeholder and their interaction during the crowdsourcing process Zhao and Zhu [19] described the overall crowdsourcing system, which is visualised in Figure 2. It consists of three categories of components: (1) the crowdsourcer; (2) the individuals performing the tasks, the crowd member; and (3) a crowdsourcing platform which acts as an intermediate between the crowdsourcer and the crowd member [19].

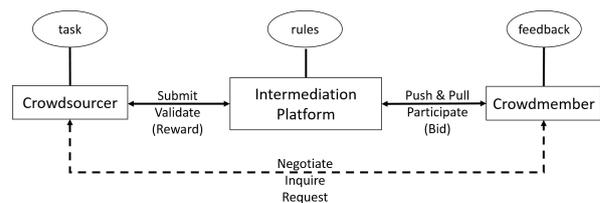


Figure 2. Components, processes and actions in crowdsourcing, based on [19]

As visualised in Figure 2, the crowdsourcer submits the open task to the crowdsourcing platform to broadcast it to the crowd. After completion of a task, the result is returned to the crowdsourcer, who will validate the solution and in some cases reward the crowdsourcing platform [19]. With push & pull all actions of the crowdsourcing platform are covered to attract, incent and sustain the crowd to use the platform [19]. By solving a broadcasted task, the crowd member actively participated in the crowdsourcing initiative. In case different crowd members solved the task they can offer their solution through bidding. Even though for microtasking it is not rel-

evant, the more complex an open task is, the more likely an exchange of information between the crowdsourcer and the crowd worker is necessary. The exchange can happen in the form of a request for further, concrete information, not covered by the original task description, an inquiry of the existing knowledge of the crowd member, which is needed to achieve a satisfactory solution or the negotiation of precise conditions.

A Preliminary Taxonomy of Crowdsourcing Metrics

Cullina and Morgan [20] addressed the gap of a missing operational crowdsourcing taxonomy in 2015. This taxonomy can be used by crowdsourcer as guidance on which characteristics of different crowdsourcing initiatives to use in practice. The authors aim was an overarching taxonomy, which addresses all different parts of the crowdsourcing process and their associated metrics. They based their examination of existing crowdsourcing metrics on the comprehensive definition of Estelles-Arolas et al. [8] and a substantial literature review they conducted. While the implication of relevant components of crowdsourcing is similar to the results of Estelles-Arolas et al., it differs in focusing on the mechanism which is used by the crowd to participate. Estelles-Arolas and Gonzalez-Ladron-de Guevara only consider the medium used, namely the internet. Cullina and Morgan [20] focus on the platform or participation architecture building the basis for the interaction of the crowd. Because crowdsourcing projects are complex, participation mechanisms are required to reduce the complexity and if possible increase efficiency. The resulting preliminary crowdsourcing metrics consisting of four categories, namely crowd membership, crowd platform, crowd incentivisation, and crowd interactions and outcomes, are documented in Table 2.

Analysing the crowd membership can assist an initiator in finding out where the success of his/her initiative is coming from and who is playing a major role in that success [20]. For example, a diverse crowd increases the potential for different types of solutions. The metrics covered in the category crowd platform address the operational level and are often simple numerical volume counts or percentages over time. Crowdsourcing platform provider can benefit from using these metrics during the implementation or improvement phase of their platform. Also, initiators searching for the right platform to broadcast their problem to the public can use these metrics as a requirement catalog. The type and amount of incentives are covered in the metrics analysing crowd incentivisation. The last category crowd interaction and outcome describe the mechanisms by which the crowd can participate and interact [20].

The proposed metric can also be applied to compare different existing crowdsourcing frameworks, in the case of this report crowdsourcing application in the mobility context.

Crowdsourcing Information Systems

Geiger et al. [21] introduced 2012 crowdsourcing information systems as a special case of information systems that produce information products and/or services for internal or

Table 2. Preliminary Crowdsourcing Metrics [20]

Metric	Indicator
I. Crowd Membership	<ul style="list-style-type: none"> - Crowd size - Age - Gender - Nationality/residency - Skill, knowledge, expertise - Individual vs. corporation - Identity - Internality or externality
II. Crowd Platform	<ul style="list-style-type: none"> - Cost - Reliability - Reach - Capacity and storage - Efficiency - Security - Complexity - Types of interaction method - Quality of experience
III. Crowd Incentivisation	<ul style="list-style-type: none"> - Types of incentive - Amount of incentive
IV. Crowd Interactions & Outcomes	<ul style="list-style-type: none"> - Tasks/Challenges created - Interactions - Time spent on platform - Time to complete tasks - Number of process cycles - Outcomes and outputs - Trust measurements

external customers by harnessing the crowds. In crowdsourcing information systems the essential work is performed by crowd members, as typical for crowdsourcing, who identify themselves as contributors. The purpose of the authors' work was to distinguish archetypes of crowdsourcing information systems based on their organizational function. They identified four archetypes of crowdsourcing information systems, by identifying how the system makes use of crowd contributions differentiating between two fundamental dimensions: (i) whether a system seeks homogeneous or heterogeneous contribution from the crowd and (ii) whether it seeks an emergent or a non-emergent value from these contributions. The four archetypes are visualised in Figure 3, including their organizational functions. According to Geiger et al. [21], in *crowd processing systems* often so called microtasks are being processed by a large crowd. In *crowd rating systems* the wisdom of the crowd is used to present votes on a given topic. Similar, the *crowd solving system* uses the wisdom of the crowd to solve a hard problem, for example, mathematical or algorithmic problems, or a soft problem, which do not have an optimal solution, as for crowdinnovation tasks. In *crowd creation systems* the main focus is the creation of a collective outcome contributed by a large heterogeneous crowd, as the knowledge

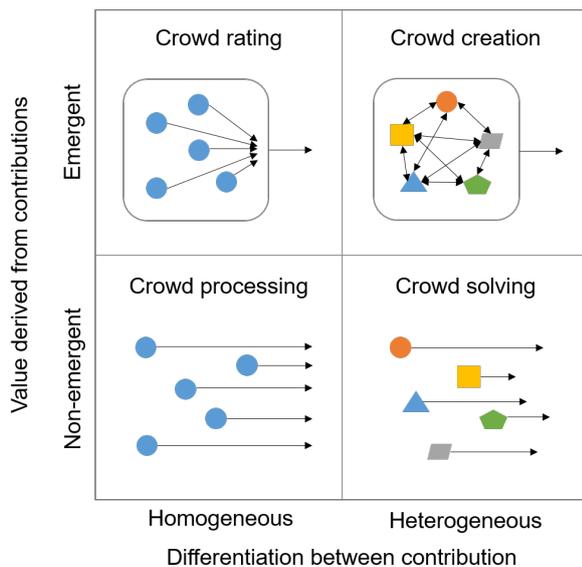


Figure 3. Four types of Crowdsourcing Information Systems, based on [21]

base Wikipedia (wikipedia.com) is one. Many crowdsourcing efforts combine some of these functions. For example, the user-generated platform YouTube (youtube.com), as part of crowd creation systems, uses collective votes as a quality indicator of the individual contributions.

These four types of crowdsourcing information systems can be used to achieve a specific goal of a crowdsourcing project. The authors also described the system components during the design of crowdsourcing information systems, which are (i) participants, similar to the crowd for Estelles-Arolas and Gonzalez-Ladron-de Guevara [8], capturing the role and nature of the crowd contributors; distinguishing between tasks everyone can perform, tasks for most people and expert tasks, due to the fact that the nature of contributors correlated strongly with characteristics of the performed tasks; (ii) information, defining the grade of information which is provided to the crowd; more specific in those systems that seek for individual contributions, than in those seeking for a collection of contributions; and (iii) technology, handling the way contributions are presented and collected.

Within the TUM LLCM context especially crowd creating and crowd rating systems are of interest to collect innovative ideas on how to address the urban mobility of the future or to participate as map contributor for indoor maps, to name just two.

Applications of crowdsourcing

Hossain and Kauranen [3] conducted a systematic crowdsourcing literature review in 2015 and documented the areas in which crowdsourcing is applied. They analyzed 346 articles in their study, which all contained “crowdsourcing” in the title, abstract or list of keywords. Crowdsourcing, which can be used besides other purposes for project planning, collection

of accurate and timely information in case of a natural disaster or collection of geographical data, is applied amongst others in the following areas:

- *Idea generation* - The usage of crowdsourcing to generate ideas is also known as open innovation or crowdinnovation. Due to the focus of this report, this application of crowdsourcing is described in detail in the following section.
- *Microtasking* - Microtasking is defined as a crowd solving system in which users can select and complete small tasks for monetary or non-monetary rewards. A well established intermediate platform is Amazon Mechanical Turk (mturk.com), which coordinates Human Intelligence Tasks (HIT) between a crowdsourcer and the crowd. HIT are microtasks in which the humans are still more efficient in completing given tasks than computers.
- *Open source software* - In open source software projects the crowd contributes to the development of software in the form of coding. Thereby, essential elements of software are accessible to the public for the purpose of collaborative improvements of the existing software [19] using crowd solving systems. Whereby for Rouse [2] crowdsourcing and open source are conflated, Zhao and Zhu [19] see three major differences between crowdsourcing and open source: (1) in a crowdsourcing initiative the call is not as open, as it is in open source projects. The ownership of the Intellectual Properties Rights (IPR) stays with the company, which acts as a crowdsourcer; (2) the motivation in open source projects is mainly intrinsic, the improvement of an existing solution. This is an insufficient motivation in most crowdsourcing initiatives and extrinsic motivation, such as monetary compensation is added; and (3) whereby in crowdsourcing initiatives besides the collaborative working also an independent contribution is possible, this is not the case for open source projects, where the solutions stay interdependent.
- *Public participation* - In a public participation initiative the crowd is included in decisions regarding the public development. Through the involvement of the crowd a broader acceptance of the planning and implementation of projects concerning the public life can be achieved, as for example for new residential areas or road expansions. Not only opinions of the crowd regarding an already proposed project are covered in public participation initiatives, but also the collection of new ideas such as the building of a new public swimming pool and the remodeling of the surrounding area [22]; Another example is gathering environmental observation for improving global land, such as Wikimapia (wikimapia.org) or OpenStreetMap (openstreetmap.org) analysed by Fritz et al. [23] to name just a few. In case

information systems are used for public participation, these are crowd creation and/or crowd rating systems.

- *Citizen science* - In this application of crowdsourcing the crowd is included in the act of solving real-world scientific problems [24]. According to Silvertown [25] a variety of scientific fields is addressed, such as ecology, geology, medicine, and environment. A well-known example of such a crowd solving and crowd creation system is Galaxy Zoo (galaxyzoo.org), where hobby astrologers can contribute in this field of research or the ARTigo (artigo.org) platform for the disciplines of art history and computer science. Thereby, citizen science crowdsourcing initiatives can be used for challenging tasks such as genomic sequencing, where the crowd could collect, synthesize, review, and analyze data [26].
- *Citizen journalism* - Citizen journalism as an alternative form of media arises when many members of the crowd contribute in the form of a citizen journalist. These are defined by Carpenter [27] as

A citizen journalist is an individual who intends to publish information online meant to benefit a community.

There are plenty of citizen journalism platforms, such as CNN iReport (edition.cnn.com), where the crowdsourcer is the established news media company CNN. If you consider the whole content created through citizen journalism, a crowd creating system is applied. Citizen journalism encourages the crowd to share stories with the public. Goodchild and Glennon [28] claims that the data quality is a major concern in citizen journalism because the assurance of traditional authoritative information in journalism is missing. Carpenter [27] claims that with the diversity of the contributors the accuracy can be increased.

- *Wikies* - Collaboration web platforms, which give users the opportunity to collaborate by reading, adding or removing content in crowd creation systems are called Wikies. Wikipedia is a well-known, public available example for a wiki, where knowledge is collected by the crowd. Beside public available wikies there are company internal wikies, which are used in almost every large organization to enable employees to work together on the same matter in an internal form to document, store and share business knowledge.

The areas of application of crowdsourcing are constantly growing, as new fields discover the potential of making use of the crowd. One example is crowdsensing, which uses crowd-sourced data actively and passively collected by mobile devices.

Crowdinnovation

As crowdinnovation is one application of crowdsourcing, all aspects presented in the previous section, such as the crowdsourcing process and the preliminary metrics, are also valid. Within crowdinnovation companies and businesses open up their innovation process, to commercialize both their own ideas as well as innovations coming from outside idea sources [29]. Thereby, the boundaries of the firms become porous to ideas generated outside, as visualised in Figure 4. This porosity also enables, in an optimal case, internal ideas, which are for example not useable within the firm, to become public available.

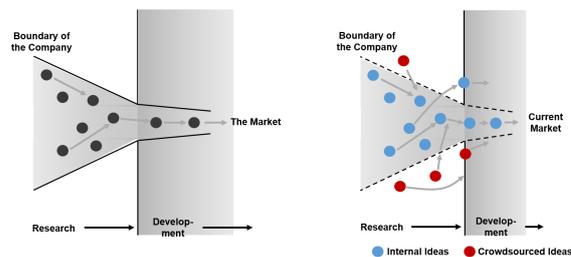


Figure 4. Open versus closed innovation, based on [29]

For Islam et al. [30], who conducted a comprehensive review of open innovation literature in 2016, the phenomenon of crowdinnovation is gaining a foothold worldwide. According to their research the concept of crowdinnovation, or precise open innovation, was coined in 2003 when Chesbrough [29] defined it as

Open innovation is a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market as the firms look to advance their technology.

Hossain and Kauranen [3] differentiated between two forms of crowdinnovation: idea competitions and ideation with collective intelligence. Within an idea competition, all submitted ideas regarding one broadcasted task are collected and a winning team is selected. For the ideation the selection procedure to achieve a ranking of submitted ideas is dismissed. The aim is rather in gathering valuable alternatives for internal ideation [3]. One example to use crowdinnovation within the context of the TUM LLCM is to collect ideas of citizens regarding possible improvements of urban mobility within Munich, which could be implemented within the project.

Motivating the Crowd

The success of all crowdsourcing initiatives depends highly on the user's willingness and motivation to engage and contribute to the corresponding activities [31].

A distinction between intrinsic and extrinsic motivation can be made. Leimeister et al. [32] describe extrinsic motivation as something activated by direct or indirect monetary or token

compensation, or recognition by others. Intrinsic motivation, such as altruism, personal achievement or enjoying a hobby, happens when no external incentives are given. By examining 128 experiments Deci et al. [33] showed that negative consequences can occur in case extrinsic motivation is applied whereas the motivation of the crowd is largely intrinsic. Thus, it is crucial to use the right incentives to achieve a high participation, which is a major challenge of every crowdsourcing initiative.

Gamification

One way to motivate the crowd in participating without additional payment is to use gamification elements. The mobility applications using crowdsourcing as their data collection method motivated the further analysis of gamification. Detering et al. [45] defined gamification in 2011 as

Gamification is the use of game design elements in non-game contexts.

Thereby game design elements, such as badges, levels, leaderboards, guidelines, and rules, are implemented in a non-game application to make the participation fun and rewarding and keep the crowd contributing over an extended period [46]. To emphasize more the goal of the using gamification Huotari and Hamari [47] adapted the definition of gamification in the following year to

Gamification refers to a process of enhancing a service with affordances for gameful experiences to support user's overall value creation.

Thereby, the authors stress that the enhancing service, including gamification aspects, supports a core service, and not the other way around.

According to Hamari et al. [48] who approached the research question *Does Gamification Work?* by conducting a literature review of empirical studies on gamification in 2014, the conceptualization of gamification leads to three main phases of gamification: 1) the implemented motivational affordance, 2) the resulting outcomes, and 3) the further behavioral outcomes (see also Figure 5).

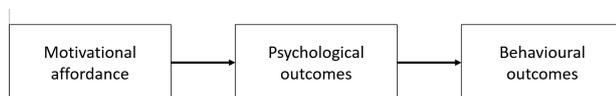


Figure 5. Main parts of gamification, based on [48]

The authors examined 24 empirical studies in their literature review. One interesting result is the collection of ten different motivational affordance categories, which are: points, leaderboards, achievements/badges, levels, story/theme, clear goals, feedback, rewards, progress, and challenge. The contexts of the analysed studies, in which gamification was used, was wide, but most of the studies were conducted in crowdsourcing systems. The context ranges from commerce, education/learning, health/exercise to innovation/creation and

data gathering, to name but a few [48]. The authors result after the analysis is that gamification does work, but some caveats do exist. Thus, the context being gamified is relevant, as services oriented towards strictly rational behavior might be hard to gamified. Further, the quality of users is crucial, as the same affordance is felt differently for different player types [48]. Most of the gamification elements, such as points, leaderboards, achievements/badges amongst others, are based on a social comparison and leads to a competitive dynamic among the users, which addresses the social need for achievement [49].

Crowdsourcing and Crowdinnovation in the mobility context

Crowds participating in crowdsourcing initiatives completing in most cases tasks which they can easily access and complete by using mobile devices [34]. This is especially applicable for crowdsourcing in the mobility context, such as

- navigation applications, providing the fastest, shortest or nicest driving route from point A to B, or the route with least emission, to name just a few,
- intermodal traffic recommendation applications, offering all possible routes from point A to point B providing combinations of different mobility services, such as public transportation, bike sharing or car sharing,
- mobility sharing applications, enabling two or more users to share a ride from point A to B,
- mapping applications, offering maps indoor and outdoor for special purposes and based on crowdsourced data.

The phenomenon of using mobile devices for crowdsourcing led to the expression of crowdsensing, where participant collect and aggregate data with their carry on smartphone everywhere they go [40].

Crowdsourcing in the mobility context

Mahmud [34] analysed in their state of the art review in 2015 all existing mobile crowdsourcing application, differentiating between research- and industry-based developers. Within their research, they identified 25 mobile crowdsourcing application, of which ten are crowdsourcing applications within the mobility context. Besides the three applications from a research community, which are Advanced Service Smart Parking, NaviTweets, and TeleEye, there were already seven mobile crowdsourcing applications for traffic and navigation purposes available until March 2015 provided by the industry. An extract from the complete overview is presented in Table 3, listing all mobility crowdsourcing applications. The table was extended with further interesting mobile crowdsourcing applications but is by far not comprehensive.

One example of the many available mobility applications

Table 3. (Mobile) Crowdsourcing applications addressing mobility, based on [34]

Research	Industry
Advances Services Smart Parking [35]	Placemeter (placemeter.com)
NaviTweets [36]	Waze (waze.com)
TeleEye [37]	Stereopublic (stereopublic.net)
Tiramisu [38]	BlaBlaCar (blablacar.com)
UnCrowdTPG [39]	Kamino (gokamino.com)
TrafficSense [40]	Sit or squat (sitorsquat.com)
TUMitfahrer (tumitfahrer.de)	OpenStreetMap (openstreetmap.org)
Next Stop Design [11]	Moovit (movit.com)
OurWay [41]	JustPark (justpark.com)
PocketParker [42]	UMapper (umapper.com)
Future Tram (Research & Industry) (futuretram.siemensinnovation.spigit.com)	Parkopedia (parkopedia.de)
Walkly (previously WalkSafe) [43]	Tamyca (amyca.de)
GAFU [44]	SmartTanken (smarttanken.de)
	DB Mitfahrer (bahn.de/wmedia/view/mdb/media/intern/mitfahrer-app/)
	BMW Mobility Experience Challenge (startnext.com/pages/bmw)

based on university research projects is Tiramisu, a crowd-powered transit information system to delivering real-time information about when the bus is coming and about the condition of this bus [50]. The research group around Steinfeld of the Carnegie Mellon University in Pittsburgh, Pennsylvania, USA, identified customers' and service providers' need to have accurate information about the actual arrival time and conditions of the next coming bus [38]. Their motive to implement the crowdsourcing approach was due to high automation costs to achieve the same outcome. For the implementation, they adopted universal design principles, conducted interviews with bus riders to examine riders preferences and evaluated different reporting methods. To motivate users, a crucial factor when using crowdsourced data, they encourage the development of a community. The launched application is now available in Pittsburgh, Pennsylvania, and New York City, New York.

A very successful crowdsourcing application coming from the industry is Waze, which is a social navigation system. Founded 2006 in Israel and purchased by Google in 2013 for around one billion dollars, Waze has currently an estimated user base of 55 million users worldwide [51]. Crowdsourced information are being used to provide better routing information to the local driver community. Besides the passive data collection during the ride regarding the speed and the location, the users have the option to report the current traffic situation, such as a policy control, a traffic jam or bad road conditions. Waze achieves revenue by placing hyper-local advertisement. Moovit is an Israeli start-up providing intermodal traffic recommendations for the usage of public transportation. It improves the service of simply reporting scheduled public transportation by offering real-time information about the arrival time, crowdedness, and cleanliness, amongst others, obtained

through crowdsourced data. Thus, as in the case of Waze, besides the passive data collection obtained through the users' location, an active contribution is possible.

Both applications rely heavily on the crowdsourced data, due to their added value compared to similar existing application. The more users the applications attract, the better the services gets, which will attract further users. This effect is called positive network effect [40].

To analyse the different crowdsourcing aspects of Waze and Moovit the crowdsourcing metric proposed by Cullina and Morgan [20] was applied and visualised in Table 4. As mentioned before, particularly the used incentives are relevant because they are one major success factor of the application. In the cases of Waze and Moovit, gamification elements are used in form of avatars and badges, which represent the maturity level the user reached through active and passive contributions. Also, community aspects, such as direct feedback from other users, is implemented. An avatar represents the current status, and in case of Waze also the users' mood, serving as an additional incentive. Waze and Moovit not only motivate user to use navigation if they are traveling unknown routes, what classical navigation systems are used for, but to use it on a daily basis for already known routes, for example, to get from home to work. Furthermore, both enable and achieve the active involvement in enlarging their information basis in form of map contribution (Waze) and public transport plans (Moovit). One reason behind this is the development of a community, in which every volunteer helper is involved, receiving recognition from others.

For their research based project TrafficSense, Heiskala et al. [40] compared their current and future result with Waze and Moovit. TrafficSense, a project of the Aalto University, Finland, will be a multimodal personal mobility assistant,

Table 4. Crowdsourcing Metrics applied to Waze and Moovit

Indicator	Waze	Moovit
I. Crowdmembership		
- Crowd size	55 mio active user	30 mio active user
- Age, Gender	No restriction	No restriction
- Nationality/residency	Available in 200 countries	60 countries & 800 cities
- Skill, knowledge, expertise	Usage of mobile devices	Usage of mobile devices
- Individual vs. corporation	Cooperation between users by sharing information	Cooperation between users by sharing information
- Identity	Pseudonimized identities	Pseudonimized identities
- Internality or externality	Externality, open to public	Externality, open to public
II. Crowd Platform		
- Cost	No cost for users/public entities/broadcast media; fee-based for advertisers	No cost for users
- Reliability	Depending on number of users in Germany: Medium	Depending on number of users in Germany: Low
- Reach	Available for all mobile devices with internet access in provided countries	
- Capacity and storage	No limitation	No limitation
- Efficiency	High	Medium (e.g. usage of old bus schedules)
- Security	Tampering with traffic data possible, manipulating traffic flow; High privacy concerns (data transfer to third business partners)	Tampering with traffic data possible
- Complexity	Intuitive handling, low complexity	Low-Medium complexity
- Types of interaction method	Two-sided information exchange	
- Quality of experience	4,6/5 Stars of 4.695.789 user feedbacks on Google Play (as of April 2016)	4,3/5 Stars of 364.760 user feedbacks on Google Play (as of April 2016)
III. Crowd Incentivisation		
- Types of incentive	- Intrinsic (Fastest way to get from A to B) - Recognition (symbolic honours) - Enjoyment (gamified design) - Socialisation (involvement in community)	- Intrinsic (Fastest way to get from A to B using public transportation) - Recognition (symbolic honours) - Enjoyment (gamified design) - Socialisation (involvement in community)
- Amount of incentive	High with a variety of incentives	Medium with a variety of incentives
IV. Crowd Interactions & Outcomes		
- Tasks/Challenges created	Map improvement tasks, no further challenges	No challenges used
- Interactions	Permanent exchange of location data and manually forwarded information	Permanent exchange of location data and manually forwarded information
- Time spent on platform	Sum of all car drives (in optimal cases)	Sum of traveling time with public transportation (in optimal cases)
- Time to complete tasks	Little time necessary	No tasks available
- Number of process cycles	No process cycles available	
- Outcomes and outputs	Real time traffic data/information, improved maps, location-based advertisement	Real time traffic data/information, crowdness and cleanliness of public transport
- Trust measurements	Not applicable	Not applicable

analysing users passive crowdsourced data to predict better traffic system level situation compared to just real-time observation [40]. In their work, the authors point out the challenges and possibilities of crowdsourcing based mobility applications. They aim at giving insides on why the existing services are successful and providing a first checklist for new mobility applications, such as TrafficSense. The research group has not yet decided on which incentives they will use but are already considering to copy success factors of Waze and Moovit, such as the usage of gamification and social interaction between users.

Crowdinnovation in the mobility context

Particularly in the context of public participation often crowdinnovation initiatives were conducted and analysed within research projects. Brabham [52] analysed the motivation for participation in the Next Stop Design contest, which was a cooperation between the University of North Carolina and the Utah Transit Authority (UTA). Participants were enabled to submit design ideas for a bus stop shelter for a Salt Lake City, Utah, within the four-month timeframe of the initiative. Registered participants could also comment and vote, using a 1-5 point vote, each submitted idea. Even though no form of compensation was offered, 3.187 participants registered, 260 ideas were submitted, and 11.058 votes placed. The author analysed the motivation of users, ranging from the opportunity to advance their career, to have fun, to express themselves. The conducted study shows that people are interested in contributing with their ideas and views to transit planning. One successful example of a crowdinnovation initiative in the mobility context is the “Future Tram - Straßenbahn der Zukunft” (futuretram.siemensinnovation.spigit.com), which was a cooperation between Siemens and the Institut für Schienenfahrzeuge und Transportsysteme, RWTH Aachen University, within the Center of Knowledge Interchange. The initiative was open for all RWTH students for almost two months in the first half of 2015, giving them the possibility to hand in ideas for the construction and overall appearance of the tram of the future in the areas tram and human, tram and city, and tram and technology. 150 students participated, and 63 ideas were submitted. The initiative collected requirements of potential future customers, who would have as citizens an interaction with the tram. During the initiative, the community was able to access, comment and vote for each submitted idea. The final voting was done by experts from Siemens and RWTH Aachen. As an incentive, the five best teams were invited to a several-day trip to Vienna to present their ideas in front of the jury, and the three winning teams received prize money. The transfer of such a crowdinnovation project within the TUM LLCM could collect ideas and impulses to discuss the future of urban mobility in Munich.

Conclusive Remarks and Future Work

The success of crowdsourcing applications addressing mobility topics shows that the involvement of the crowd is one major aspect to be considered during the development of the TUM LLCM ecosystem. Especially the aspect of a community seems to be one major key success factor in case mobility applications will be provided. A community gives the users the possibility to integrate their ideas and views, feel recognised and thus being willing to contribute with own data, which will enlarge the ecosystem.

Another aspect, which already emerged within the TUM LLCM, is the idea of a user-centered mobility ecosystem, owned by a user cooperative. This cooperative would ensure the appropriate protection of the users’ data and thereby differentiate themselves from already existing mobility services. For this idea crowdfunding, raising monetary contributions from a crowd will be considered for the initial seed capital. Within the project runtime, further mobility crowdsourcing applications will be identified, enlarging Table 3, analysed to benefit from lessons learned and examined whether these are applicable within the TUM LLCM.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] Jeff Howe. Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business. *UK: Business Books*, (December), 2008.
- [2] Anne C. Rouse. A Preliminary Taxonomy of Crowdsourcing. *ACIS 2010 Proceedings*, page 76, 2010.
- [3] Mokter Hossain and Ilkka Kauranen. Crowdsourcing: a comprehensive literature review. *Strategic Outsourcing: An International Journal*, 8(1):2–22, feb 2015.
- [4] Toyota showroom history. <http://www.toyota-global.com/showroom/emblem/history/>. Cited on 22.05.2016.
- [5] Yannig Roth and Joël Cére. Big Brands Increased Investment In Crowdsourcing By Nearly 50 % In 2014. *The State of Crowdsourcing in 2015Trend Report eYeka pressroom*, (December), 2014.
- [6] Yannig Roth, Francois Petavy, and Mario Braz de Matos. The State of Crowdsourcing in 2016. *eYeka*, 2016.
- [7] Jane Webster and Richard T Watson. Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, 26(2):xiii – xxiii, 2002.

- [8] E. Estelles-Arolas and F. Gonzalez-Ladron-de Guevara. Towards an integrated crowdsourcing definition. *Journal of Information Science*, 38(2):189–200, apr 2012.
- [9] Abraham Harold Maslow. A theory of human motivation. *Psychological Review*, 50:370–396, 1943.
- [10] Chrysaida-Aliki Papadopoulou and Maria Giaoutzi. Crowdsourcing as a Tool for Knowledge Acquisition in Spatial Planning. *Future Internet*, 6(1):109–125, 2014.
- [11] D. C. Brabham. Crowdsourcing as a Model for Problem Solving: An Introduction and Cases. *Convergence: The International Journal of Research into New Media Technologies*, 14(1):75–90, 2008.
- [12] Marion K. Poetz and Martin Schreier. The value of crowdsourcing: Can users really compete with professionals in generating new product ideas? *Journal of Product Innovation Management*, 29(2):245–256, 2012.
- [13] Paul Whitla. Crowdsourcing the Public Participation Process for Planning Projects. *Contemporary Management Research*, 5(1):15–28, aug 2009.
- [14] Valerie Chanal and Marie-Laurence Caron-Fasan. How to invent a new business model based on crowdsourcing: the Crowdspirit R case To cite this version: How to invent a new business model based on crowdsourcing: the Crowdspirit ® case. *Conférence de l'Association Internationale de Management Stratégique*, 2010.
- [15] Dinesh Govindaraj, Naidu K.V.M, Animesh Nandi, Girija Narlikar, and Viswanath Poosala. MoneyBee: Towards Enabling a Ubiquitous, Efficient, and Easy-to-Use Mobile Crowdsourcing Service in the Emerging Market. *Bell Labs Technical Journal*, 18(4):3–17, 2014.
- [16] Paul Whitla. Crowdsourcing the Public Participation Process for Planning Projects. *Contemporary Management Research*, 5(1):15–28, aug 2009.
- [17] Mahmood Hosseini, Alimohammad Shahri, Keith Phalp, Jacqui Taylor, and Raian Ali. Crowdsourcing: A taxonomy and systematic mapping study. *Computer Science Review*, 17:43–69, 2015.
- [18] Louise Muhdi, Michael Daiber, Sascha Friesike, and Roman Boutellier. Crowdsourcing : an alternative idea generation approach in the early innovation process phase of innovation. *Int. J. Entrepreneurship and Innovation Management*, 14(4):315 – 332, 2011.
- [19] Yuxiang Zhao and Qinghua Zhu. Evaluation on crowdsourcing research: Current status and future direction. *Information Systems Frontiers*, 16(3):417–434, jul 2014.
- [20] Eoin Cullina and Lorraine Morgan. Measuring the Crowd – A Preliminary Taxonomy of Crowdsourcing Metrics Categories and Subject Descriptors. *OpenSym*, August 19:10, 2015.
- [21] David Geiger, Erwin Fieft, Michael Rosemann, and Martin Schader. Crowdsourcing Information Systems –Definition, Typology, and Design. *Proceedings of the 33rd International Conference on Information Systems. 2012. Association for Information Systems/AIS Electronic Library (AISeL)*, pages 1–11, 2012.
- [22] Alexandra Collm and Kuno Schedler. Crowd innovation: The role of uncertainty for opening up the innovation process in the public sector. *IRSPM Conference*, 2011.
- [23] Steffen Fritz, Ian McCallum, Christian Schill, Christoph Perger, Linda See, Dmitry Schepaschenko, Marijn van der Velde, Florian Kraxner, and Michael Obersteiner. Geo-Wiki: An online platform for improving global land cover. *Environmental Modelling & Software*, 31:110–123, may 2012.
- [24] Andrea Wiggins and Kevin Crowston. From conservation to crowdsourcing: A typology of citizen science. *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 1–10, 2011.
- [25] Jonathan Silvertown. A new dawn for citizen science. *Trends in ecology & evolution*, 24(9):467–71, 2009.
- [26] M Swan, K Hathaway, C Hogg, R McCauley, and A Vollrath. Citizen science genomics as a model for crowd-sourced preventive medicine research, 2010.
- [27] S. Carpenter. A study of content diversity in online citizen journalism and online newspaper articles. *New Media & Society*, 12(7):1064–1084, nov 2010.
- [28] Michael F. Goodchild and J. Alan Glennon. Crowdsourcing geographic information for disaster response: a research frontier. *International Journal of Digital Earth*, 3(3):231–241, sep 2010.
- [29] Henry W. Chesbrough. The Era of Open Innovation. *MIT Sloan Management Review*, 44(3):34–41, nov 2003.
- [30] Mokter Hossain, K.M. Zahidul Islam, Mohammad Abu Sayeed, and Ilkka Kauranen. A comprehensive review of open innovation literature. *Journal of Science and Technology Policy Management*, 7(1):2–25, 2016.
- [31] Marianna Sigala. *Gamification for Crowdsourcing Marketing Practices: Applications and Benefits in Tourism*. Springer International Publishing, Cham, 2015.
- [32] Jan Marco Leimeister, Michael Huber, Ulrich Bretschneider, and Helmut Krömer. Leveraging Crowdsourcing: Activation-Supporting Components for IT-Based Ideas Competition. *Journal of Management Information Systems*, 26(1):197–224, jul 2009.
- [33] Edward L. Deci, Richard M. Ryan, and Richard Koestner. A Meta-Analytic Review of Experiments Examining the Effects of Extrinsic Rewards on Intrinsic Motivation. *Psychological Bulletin*, 125, Nr. 6:627–668, 1999.
- [34] Farahidayah Mahmud. State of Mobile Crowdsourcing Applications : A Review. pages 27–32, 2015.

- [35] Alessandro Grazioli, Marco Picone, Francesco Zanichelli, and Michele Amoretti. Collaborative mobile application and advanced services for smart parking. *Proceedings - IEEE International Conference on Mobile Data Management*, 2:39–44, 2013.
- [36] Daehan Kwak, Daeyoung Kim, Ruilin Liu, Liviu Iftode, and Badri Nath. Tweeting Traffic Image Reports on the Road. *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*, pages 40–48, 2014.
- [37] Yao Chung Fan, Cheng Teng Iam, Gia Hao Syu, and Wei Hong Lee. TeleEye: Enabling real-time geospatial query answering with mobile crowd. *Proceedings - IEEE International Conference on Distributed Computing in Sensor Systems, DCoSS 2013*, 1(d):323–324, 2013.
- [38] Aaron Steinfeld, John Zimmerman, Anthony Tomasic, Daisy Yoo, and Rafae Aziz. Mobile Transit Information from Universal Design and Crowdsourcing. *Transportation Research Record: Journal of the Transportation Research Board*, 2217:95–102, dec 2011.
- [39] Mattia Gustarini, Jerome Marchanoff, Marios Fanourakis, Christiana Tsiourti, and Katarzyna Wac. UnCrowdTPG: Assuring the experience of public transportation users. In *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–7. IEEE, oct 2014.
- [40] Mikko Heiskala, Jani-Pekka Jokinen, and Markku Tinnilä. Crowdsensing-based transportation services — An analysis from business model and sustainability viewpoints. *Research in Transportation Business & Management*, 18:38–48, mar 2016.
- [41] Harald Holone, Gunnar Misund, and Håkon Holmstedt. Users are doing it for themselves: Pedestrian navigation with user generated content. *NGMAST 2007 - The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies, Proceedings*, (Ngmast):92–99, 2007.
- [42] Anandathirtha Nandugudi, Taeyeon Ki, Carl Nuessle, and Geoffrey Challen. PocketParker: pocketsourcing parking lot availability. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '14 Adjunct*, pages 963–973, 2014.
- [43] Tianyu Wang, Giuseppe Cardone, Antonio Corradi, Lorenzo Torresani, and Andrew T. Campbell. WalkSafe : A Pedestrian Safety App for Mobile Phone Users Who Walk and Talk While Crossing Roads Categories and Subject Descriptors. *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications - HotMobile '12*, page 1, 2012.
- [44] V. Corcoba Magana and M. Munoz-Organero. GAFU: Using a gamification tool to save fuel. *IEEE Intelligent Transportation Systems Magazine*, 7(2):58–70, 2015.
- [45] Sebastian Deterding, Rilla Khaled, Lennart Nacke, and Dan Dixon. Gamification: toward a definition. *Chi 2011*, pages 12–15, 2011.
- [46] Flavio A. de Franga, Adriana S. Vivacqua, and Maria Luiza M Campos. Designing a gamification mechanism to encourage contributions in a crowdsourcing system. In *2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 462–466. IEEE, may 2015.
- [47] Kai Huotari and Juho Hamari. Defining gamification. In *Proceeding of the 16th International Academic MindTrek Conference on - MindTrek '12*, page 17, New York, New York, USA, 2012. ACM Press.
- [48] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does Gamification Work? – A Literature Review of Empirical Studies on Gamification. In *2014 47th Hawaii International Conference on System Sciences*, pages 3025–3034. IEEE, jan 2014.
- [49] Sebastian Deterding. Situated motivational affordances of game elements: A conceptual model. *Chi 2011*, (Deterding, S. (2011). Situated motivational affordances of game elements: A conceptual model. *Chi 2011*, 3–6. <http://doi.org/ACM 978-1-4503-0268-5/11/05>):3–6, 2011.
- [50] Tiramisu the real-time bus tracker. <http://www.tiramisutransit.com/>. Cited on 31.05.2016.
- [51] Abdullah AIDwyish, Egemen Tanin, and Shanika Karunasekera. Location-based social networking for obtaining personalised driving advice. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '15*, volume 1, pages 1–4, New York, New York, USA, 2015. ACM Press.
- [52] Daren C. Brabham. Motivations for Participation in a Crowdsourcing Application to Improve Public Engagement in Transit Planning. *Journal of Applied Communication Research*, 40(3):307–328, aug 2012.

Service Mashups and Developer Support

Tanmaya Mahapatra and Christian Prehofer

Department of Informatics, Technical University of Munich, Munich
{tanmaya.mahapatra, christian.prehofer}@tum.de

Abstract

Physical mobility in major cities has become an ostentatious issue and connected mobility, an application of Internet-of-Things (IoT) technologies has been readily propounded to soothe the situation. The context of connected mobility, where applications generally have to be designed on an adhoc basis to meet the user requirements, has gradually shifted the art of programming from the realms of professional software developers to third party application developers(End-Developers) or possibly even novice end users. The concepts of web mashups can be leveraged here to create IoT applications. This paper discusses the concept of web mashups in details and the tool-kits which provide support for IoT application development. The domain of mashups is interesting but the challenges involved with mashup development in an IoT scenario are quite heavy. The developmental strategies followed by the tool-kits can be classified into either mashup based or model-based. The functionality of these tool-kits have been described in great detail to represent the current state-of-art in the context of IoT application development. These tool-kits have been compared with respect to one another, followed by a discussion on their strengths and weakness. The existing weaknesses signify the open research challenges.

Keywords

Mashups; Mashup Tools; Internet of Things

1. Introduction

At present, the rate at which data moves with the help of Internet technologies has increased considerably at the same time mobility of human beings in major urban areas has become a bit irksome. The population and number of cars is growing at an unprecedented rate while the space to develop new transportation infrastructures is just non-existent. With this alarming rate of growth of human population and cars it appears that the life in the big cities will definitely come to a halt. The frequency of daily congestion is increasing and jeopardizing the day to day life of people [1]. The traffic congestion woes can be reduced by optimizing the mode of transportation used by majority. Maximal usage of public transport can help solve these issues to some extent. Normally a wide range of transport options are available in mega cities. But there are certain limitations in the design of the public transport system which prevented from their wide spread adoption. For example people normally travel from one point of interest to another and not generally from one public halt to another. People are desirous of having real time information to facilitate change of transportation mode in case of some congestion occurs.

To facilitate this type of scenario, a vision of connected mobility is highly sought after. Connected mobility takes into account all available transport options, real time traffic information to facilitate hassle free transportation. In a sense connected mobility can be seen as an application of Internet of things (IoT) technologies.

IoT has been defined as the interconnection of ubiquitous computing devices for the realization of value to end users [2]. This includes data collection from the devices for analysis leading to better understanding of the contextual environment as well as automation of tasks for optimization of time and enhancing the quality of human life to the next level. IoT has already pierced into fields like health care, manufacturing, home automation etc. [3]. But to truly exploit the possibilities offered by IoT is to rapidly enhance the application landscape.

Unfortunately the development of applications for the IoT landscape is not a straightforward software development process. The developer needs to handle the communication protocol details of various devices, data mediation as well as develop the business logic. It is also noteworthy to mention here that most of the IoT applications need to be designed in an adhoc fashion typically by end users. Hence a tool-kit for application development is unavoidable. Having a toolkit to take care of these complicated stuff and allowing the developer to focus solely on the business logic would be the most ideal and desirable situation.

Mashup and model-based approaches have been used to build applications for the IoT. They differ in terms of expressiveness and modeling the data flow between various components [4]. Currently, there are a plethora of tool-kits aiming to ease the development process. However at present the IoT community lacks a toolkit that enables the inexperienced developers to develop IoT prototypes rapidly [5] i.e striking the right balance between simplicity and functionality. Mashups have traditionally been used to combine data collected from

different IoT devices to perform some interesting tasks.

We start by discussing mashups in detail in Section 2. Section 3 discusses the most important tools and platforms supporting application development in the context of IoT. Section 4 compares the tools with one another in a very broad manner taking the conceptual approach employed into consideration. Section 5 identifies the strengths and weaknesses of the existing tool-kits thereby highlighting some of the open research challenges. Section 6 discusses the possible work directions to achieve the requirements of the work package.

2. Mashups

A mashup is a composite application that integrates two or more existing components available on the web. These components can either be data, application logic, or user interfaces. The individual components are called “mashup component”; the gluing mechanism is called “mashup logic”. The mashup logic is the internal logic which defines how a mashup operates or how the mashup components have been orchestrated [6]. It specifies which components are selected, the control flow, the data flow and data mediation as well as data transformation between different components [7].

Mashups are quite broad and are generally classified based on their composition, domain and the environment. Composition of a mashup extensively deals with the kind of components that make it up. The application stack has been broadly classified into data, logic, and presentation (user interface) layer. The mashup created accordingly is called either a data, logic, or user interface mashup. Similarly, domain of a mashup explains the functionality of a mashup like social mashups or mobile mashups etc. Lastly, the environment explains the context where it is deployed. For instance, it can be web mashups or enterprise mashups. The difference between web and enterprise mashups is very subtle and it is not the area we are trying to focus here. But it would be sufficient to know that web mashups are generally targeted for end users on the Internet while enterprise mashups are specifically used in business contexts. These need to adhere additional security guidelines and other business specific requirements which the normal web mashups need not adhere to [7].

2.1 Mashup Components

Mashup components are the building blocks of a mashup. In practice, several technologies and standards are used in the development of mashup components. Simple Object Access Protocol (SOAP) web services [8], RESTful web services, Javascript APIs, Really Simple Syndication (RSS) [9], Comma-Separated Values (CSV) [10] etc. are some of the prominent ones. Depending on their functionality the mashup components have been broadly classified into three categories (Figure 1):

1. Logic components provide access to functionality in the form of reusable algorithms to achieve specific functions.

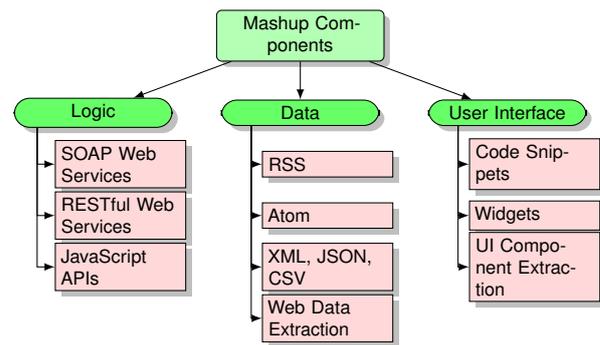


Figure 1. Classification of Mashup Components, following [7]

2. Data components provide access to data. They can be static like RSS feeds or dynamic like web services which can be queried with inputs.
3. User interface components provide standard component technologies for easy reuse and integration of user interfaces pieces fetched from third-party Web applications with in the existing user interface of the mashup application.

2.2 Mashup Tools

Mashup tools have been proposed as a simple way to develop mashups. This was supported by uniform communication protocols and APIs based on REST principles. Early mashup tools among others are Microsoft Popfly and Yahoo Pipes; for an overview we refer to [11]. In recent years, there has been a lot of interest in applying the same ideas to the IoT/WoT, also building on REST interfaces [12, 13, 14].

According to [15], mashup tools typically include data mediation. This involves converting, transforming, and combining the data elements from one or multiple services to meet the needs of the operations of another.

For connecting services, there are different concepts as discussed in [16]. The main, predominant one is modelling data flow. For others, mainly in the enterprise area, also centralized approaches with processing rules are considered. For communication, asynchronous messages are used, e.g. using REST-style communication. In general, orchestration can be described by data flow and/or workflow, or through a publish-subscribe model [16].

IoT/WoT mashup tools typically provide a graphical editor for the composition of services for one application. This models the message flow between the components. Components can be sensor nodes, processing or aggregation entities as well as external web-based services. Thus, mashup tools can also be seen as specific cases of end-user programming [17] but are however limited to the specific model of describing message flow. In addition, some mashup tools provide simulation tools and also interoperability for messaging between different platforms.

3. State of the Art in Mashup Tools

In this section, we detail on the most prominent mashup tools based on their striking features, usage, extensibility, user support, documentation availability and thriving community. We deliberately not distinguish between mashup and model-based tools as the distinction is many times artificial and/or driven by market needs. We use "mashup tool" as an umbrella term.

3.1 Node-RED

Node-RED is an open-source mashup tool developed by IBM and released under Apache 2 license. It is based on the server side JavaScript platform framework Node.js¹ (that is why the "Node" in its name). It uses an event-driven, non-blocking I/O model suited to data-intensive, real-time applications that run across distributed devices.

Node-RED provides a GUI where users drag-and-drop blocks that represent components of a larger system which can either be devices, software platforms or web services that are to be connected. These blocks are called nodes. A node is a visual representation of a block of JavaScript code designed to carry out a specific task. Additional blocks(nodes) can be placed in between these components to represent software functions that manipulate and transform the data during its passage [18].

Two nodes can be wired together. Nodes have a grey circle on their left edge, which is their input port, and a grey circle on their right edge represents their output port. To connect two nodes, a user has to link the output port of one node to the input port of the other node. After connecting many such nodes, the finished visual diagram is called a flow.

IoT solutions often need to wire different hardware devices, APIs, online web services in interesting ways. The amount of boilerplate code that the developer has to write to wire such different systems, e.g. to access the temperature data from a sensor connected to a device's serial port or to manage authentications using OAuth [19], is typically large. In contrast, to use a serial port using Node-RED, all a developer has to do is to drag on a node and specify the serial port details. Hence, with Node-RED the time and effort spent on writing boilerplate code is greatly reduced, and the developer can focus on the business parts of the application.

Node-RED flows are represented in JSON and can be serialized, in order to e.g. be imported anew to Node-RED or shared online. There is a new concept of "sub-flows" that is being introduced into the world of Node-RED. Sub-flows allow creating composite nodes encompassing complex logic represented by internal data flows.

Since in Node-RED nodes are blocks of JavaScript code, it is — technically — possible to wrap any kind of functionality and encapsulate that as a node in the platform. Indeed, new nodes for interacting with new hardware, software and web services are constantly being added, making Node-RED a very rich and easily extensible system. Lastly, note that the

learning curve to develop a new node for the platform is low for Node.js developers since a node is simply an encapsulation of Node.js code.

To make a device or a service compatible with Node-RED, a native Node.js library capable to talk to the particular device or service is required. However, with the growing acceptance of REST style in Web and IoT systems, more and more devices and services provide RESTful APIs that can be readily used from Node-RED.

3.2 glue.things

The objective of "glue.things" is to build a hub for rapid development of IoT applications. "glue.things" heavily employs open source technologies for easy device integration, service composition and deployment [20]. TVs, phones, and various other home/business tools can be hooked up to this platform through a wide range of protocols like Message Queue Telemetry Transport (MQTT) [22], Constrained Application Protocol (CoAP) [22] or REST APIs over HTTP.

The development of mashup applications in glue.things roughly goes through three stages [20].

Firstly, the devices are connected to the platform to make them web accessible using protocols like MQTT, CoAP or HTTP/TCP etc. Device registration and management is handled by the "Smart Object Manager" layer in the glue.things architecture as explained in Section 3.2.1. REST APIs provide communication capabilities and JSON data model is used for propagating device updates. These facilities are leveraged using the client libraries or for a more intuitive experience of device addition the web based dashboard can be used. The dashboard also features several templates for connecting devices and simplifying the tasks for the developer.

The second stage deals with creation of mashups. glue.things uses an improved version of Node-RED as a mashup tool to collect data streams from connected devices and combine them. This improved version supports multi-users, sessions and automatic detection of new registered device and makes them available on the panel. External web services like Twitter, Foursquare etc. can also be used during mashup composition. The "Smart Object Composer" layer in the glue.things architecture houses the mashup tool as explained in detail in section 3.2.1.

Lastly, the created mashups are deployed as Node-RED applications including various triggers, actions and authorization settings. These deployed mashup applications are accessible by RESTful API to the developers who may want to use them in their own custom web applications. To the normal end users, they can be browsed through a collection of mashup applications which can be used after suitable alterations to the connection settings and other environment specific values. Sharing of these mashup applications is also supported by the platform. This functionality is reflected in the "Smart Object Marketplace" layer in the architecture.

¹<https://nodejs.org/>

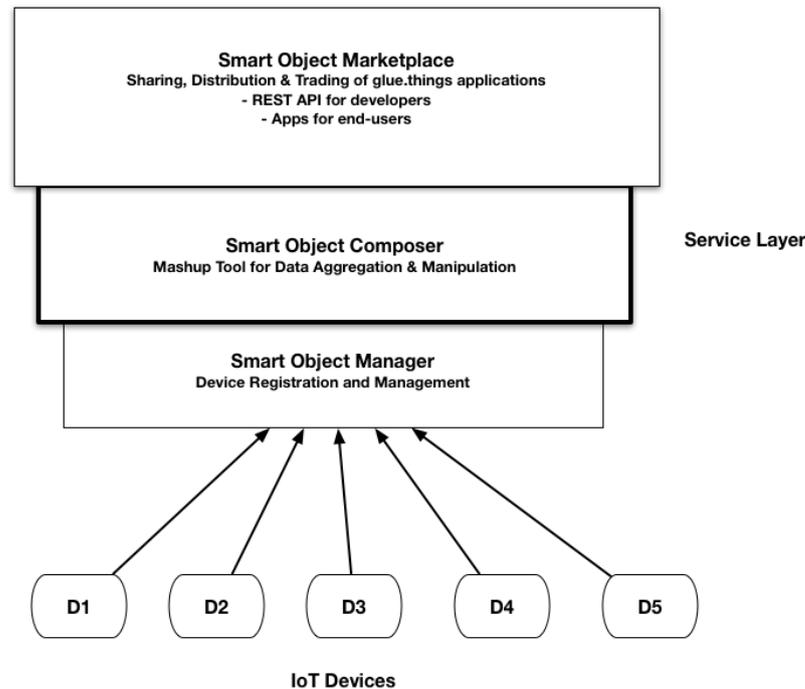


Figure 2. glue.things Architecture [20]

3.2.1 glue.things Architecture

Figure 2 shows the simplified architecture based on the detailed architecture of the platform. This can be segregated into three distinct layers, namely the Smart Object Manager, the Smart Object Composer and the Smart Object Marketplace.

Smart Object Manager This layer integrates real-time communication networks to easily access a large number of IoT devices. These networks support messaging with real-time web sockets via RPC, MQTT and CoAP. There is also a device directory to search and query for any device on the Internet. This layer is extensible, meaning any future real-time communication network/gateway can be integrated into the platform.

Smart Object Composer This layer provides mechanisms for data and device management. The mashup development environment is built on Node-RED and is used for service composition. Mashups are JSON objects in combination with a Node.js-based work-flow engine. This layer also has a virtualized device container for managing the registered devices.

Smart Object Marketplace This layer contains all the created and deployed applications. These applications can be shared, distributed or traded. Developers can access them via REST APIs to embed them in a new application. End users can access these as normal applications.

The application layer contains all the user interfaces for device registration, configuration and monitoring. A dashboard combines all these UI in a coherent front-end accessible by both users and developers alike.

3.3 WoTKit

WoT aims to leverage web protocols, and technologies to facilitate rapid construction of web applications exploiting real world objects. WoTKit, a lightweight mashup toolkit and platform provides a simple way for end-users to find, control, visualize and share data from a variety of things [21]. WoTKit aims for:

1. Easy integration of physical devices, virtual devices and the toolkit.
2. Easy visualization of data collected from different devices.

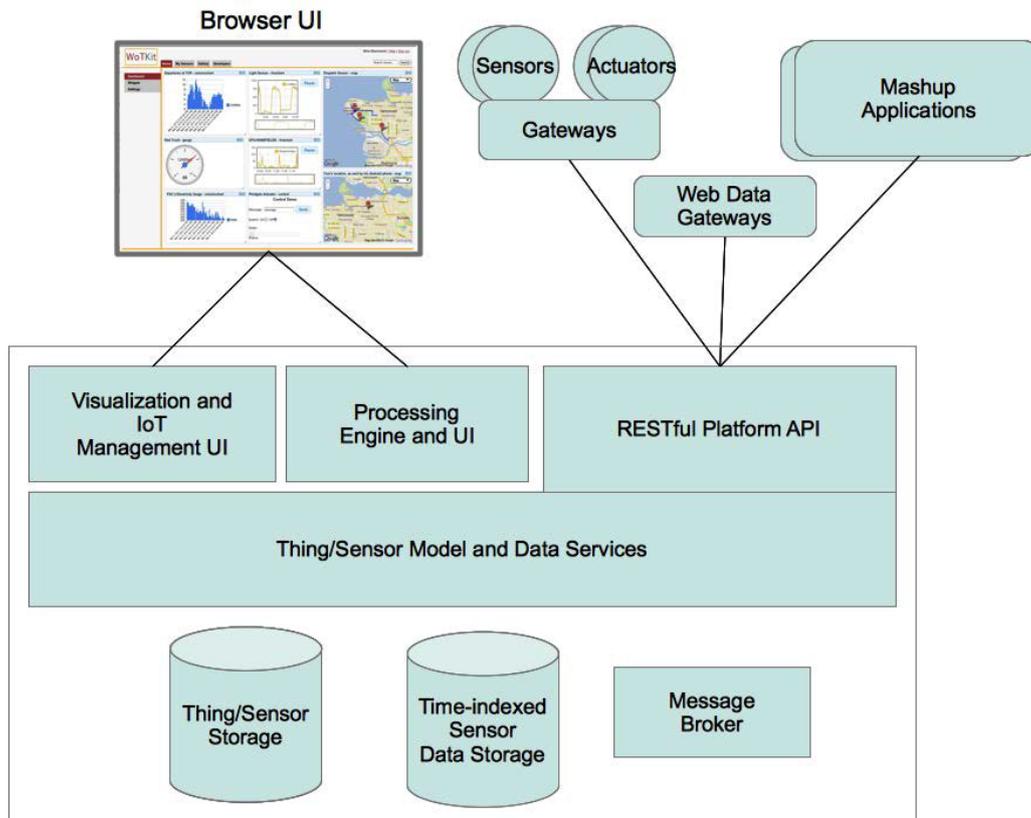


Figure 3. WoTKit Architecture, as in [21]

3. Smart and efficient information processing capability for converting low level data collected from devices to high level sensible data to be used in mashups.
4. Ability to quickly combine different data streams and apply various transformations, triggers i.e. easy service composition or mashup creation.
5. Easy sharing of created mashups and accessibility of features via APIs.

WoTKit in order to satisfy the integration requirements implements the gateways a bit differently. Here gateways are simple scripts that can gather data from the device, push the data into the system and also register the discovered devices. These gateways are web clients, and not web servers, thereby eliminating the problem of making a device available to the outside world due to firewall issues.

Similarly, for quick visualization of data collected from different devices, WoTKit uses a JavaScript-based dashboard, which supports the creation of user defined widgets. Every widget holds some specific set of data collected from devices and an associated visualization. The system comes with visualization plugins like Flot²; more visualization plugins can be

²Flot : Attractive JavaScript plotting for jQuery. <http://www.flotcharts.org>

hooked up into the dashboard at run-time.

WoTKit also contains an event-based data processing subsystem that processes the low-level data collected from devices and converts them into more sensible high-level data before they are fed into the system. It also features a visual programming environment(mashup tool) for mashing up different data sets. This is similar to the data flow model adopted by Yahoo Pipes. The mashup created using this environment is basically a pipe which consists of connected modules to generate new data from the input data sets. A pipe created is analogous to a flow created in Node-RED.

The toolkit supports end-user scripting to create new custom modules using Python and sharing of created pipes and devices registered in the system. It provides a RESTful API for interacting with the registered devices, thereby facilitating easy creation and integration of applications.

The high level architecture of WoTKit is depicted in Figure 3. WoTKit is essentially a Java based web application developed with the Spring Framework. The “UI” part provides the dashboard to interact with the system components graphically while the “RESTful Platform API” provides access to the created mashup applications and registered devices in the system (which obtain unique APIs). The “Thing/Sensor Storage” is the repository containing all the registered devices while the data fetched from devices and pushed into the sys-

tem are stored in the “Time-indexed Sensor Data Storage”. The data model consists of sensors and sensor data having a unique time-stamp attached to it. The “Message Broker” is used to deliver data between different components and has been implemented with the Apache ActiveMQ message broker³.

3.4 EcoDiF

EcoDiF is an IoT platform that integrates heterogeneous devices in order to provide real time data control, visualization, processing and storage. The platform supports integration of users, devices, applications to create an IoT ecosystem on top of which new applications can be built. It is designed to handle the key challenges of an IoT environment like: high degree of heterogeneity, environment dynamism and the massive amount of data exchange widely prevalent in a modern IoT setup [23]. The overall architecture of the system is depicted in Figure 4. EcoDiF has four different kinds of stakeholders:

Device Manufacturers Develop drivers to make their device compatible with EcoDiF openAPI. They also construct data profiles which is basically the metadata describing the type of data provided by their devices.

Data Providers Device owners who want to make the data produced by their devices available to the IoT ecosystem.

Application Developers Develop web applications using input data from devices or services available within EcoDiF or also from external web services.

Information Consumers Users that interact with the platform to search or use the information available in the ecosystem including data and applications.

The architecture has several components which together form the functionality of the platform. The “Devices Connection Module” is responsible for connection of physical devices to the EcoDiF platform and also to the Internet. Devices are configured as per EcoDiF’s specific API to facilitate easy integration with the platform. The connection between a device and EcoDiF is enabled by a customised driver specific to the device so that the same driver can be used by data providers to connect their device to the platform and make their data available. The data available from different devices is called feed and is represented using Extended Environments Markup Language (EEML) [24]. EEML is an XML based language which describes data obtained from devices in a specific context [23]. Acquired data from a device is sent to EcoDiF with the help of a HTTP PUT request (REST architectural style) so that it can be manipulated by users at real time using the “Data manipulation Component” of EcoDiF.

³The Apache ActiveMQ Message Broker : <http://activemq.apache.org/>

The Visualization and Management component provides a web interface to the end users to perform device management, create alerts, triggers or view historical data collected from the device. The Collaboration Module facilitates to search for devices and applications registered in the platform. The Applications module is the most interesting component in the entire ecosystem. It provides a model and environment for programming applications that can use the data feeds available within EcoDiF and generate new information. These applications are built as web mashups. The EEML is adopted for developing web mashup applications by integration of different data feeds available within the platform and also data feeds from external web services and databases. The Storage module stores data collected from devices in relational databases and application scripts in a file system. The module can connect to external cloud services for storage purposes and satisfying other constraints like security, availability and reliability.

The Applications module is the most interesting component in the entire ecosystem. It provides a model and environment for programming applications that can use the data feeds available within EcoDiF and generate new information. These applications are built as web mashups. The EEML is adopted for developing web mashup applications by the integration of different data feeds available within the platform and also data feeds from external web services and databases [23].

3.5 IoT-MAP

The mobile environment prevalent today has a number of smart objects around itself. These objects offer a diverse range of functionality. But the applications available on a smart phone are generally bound to a specific operational model as designed by the developers which does not adapt itself during its run-time thereby not exploiting the features and functionality available in its run-time context. IoT Mashup Application Platform (IoT-MAP) supports smart phone centric discovery, identification, installation, mashup and composition of the pervasive smart things. It specifically aims to eliminate the problem of inflexibility by aiding interoperability between mobile devices and surrounding smart things. The applications developed using this platform are called as IoT App [25].

IoT devices if tightly coupled to their offered functionality (i.e they do not offer a set of APIs to invoke their functionality) cannot be readily used in custom applications. This problem arises because the role of device manufacturers has not been differentiated with that of application developers. The IoT-MAP platform efficiently divides the segment of IoT devices and applications into three distinct actors as depicted in Figure 5 and provides support for each of them appropriately.

Application Developers The platform provides a set of APIs to build IoT apps easily. Concerned with the usage of various functionality of heterogeneous devices with

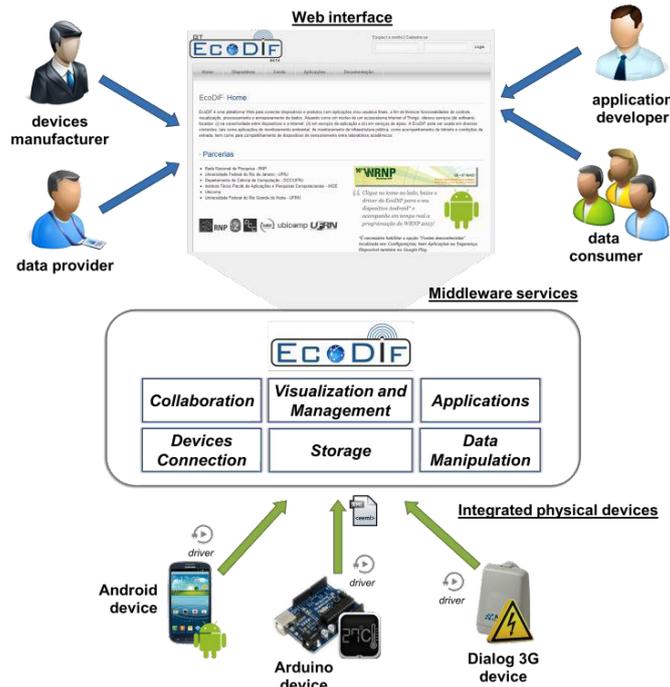


Figure 4. EcoDiF Architecture, as in [23]

IoT App API without caring for connectivity protocol (business logic).

Device Manufacturers Focus on providing device functionality by correct implementation of underlying connectivity protocols.

Users May want to create a custom application or enhance an existing one. Therefore they are provided with a GUI mashup tool.

The platform relies on the concept of model driven architecture [26] to achieve this segregation of different actors. The main idea is to extract a platform independent and domain-specific model from platform specific elements. The Platform Independent Model (PIM) layer provides generalized functional abstraction interfaces which can be accessed by application developers without caring for the underlying connectivity platforms. The Platform Specific Model (PSM) layer provides the device functionality as defined by the device manufacturers including implementation of all needed protocols and logic.

The platform architecture is well designed to easily build applications using the IoT-App API. This API utilizes the device’s functionality transparently if the abstract functionality of the device are available. Users can use an existing IoT App depending on various smart devices detected by the platform during the run-time or can compose a new one using Composition UI (GUI mashup tool). The app created is not

tightly bound to a vendor specific model and can interact with a range of smart devices depending on their availability.

The architecture (Figure 6) has the following layers [25]:

Connectivity Provider This layer abstracts and provides various connectivity protocols to the upper layers in the platform like Bluetooth or Universal Plug and Play (UPnP). Developers can use APIs to discover smart things using various connection protocols and are spared from handling the technical complexities of these protocols.

Object Abstraction Layer This core layer is responsible for abstracting real-world devices into a group of abstracted services and enables composition of those services in an IoT App.

Composition Layer This layer is utilized by a special application known as versatile App. This application is responsible for decoding of information from the mashups composed by the users in the mashup tool and can discover as well as connect to devices. While general IoT Apps integrate various smart things as defined in the business logic by the application developer but versatile App gathers devices from mashup information and can compose each software module based on that information. The authoring tool (mashup tool) used in this platform is a customized version of Node RED.

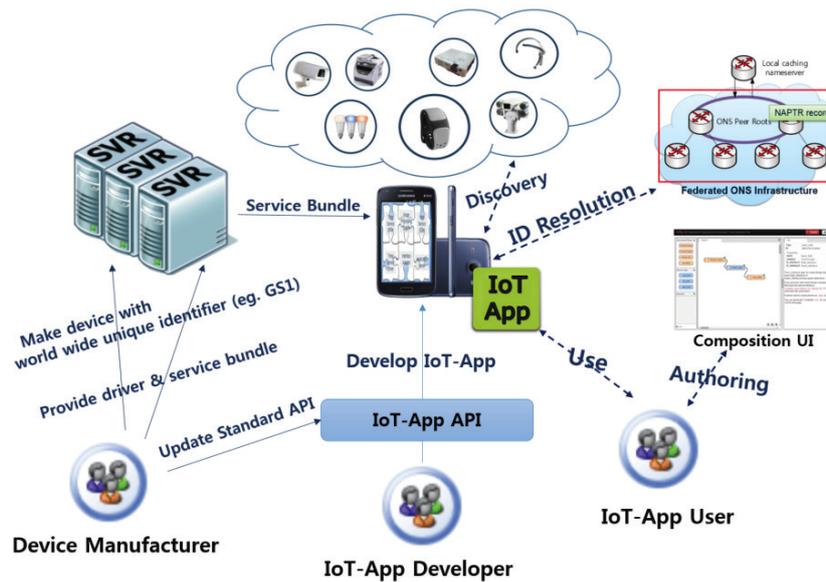


Figure 5. IoT-MAP Concept Diagram, as in [25]

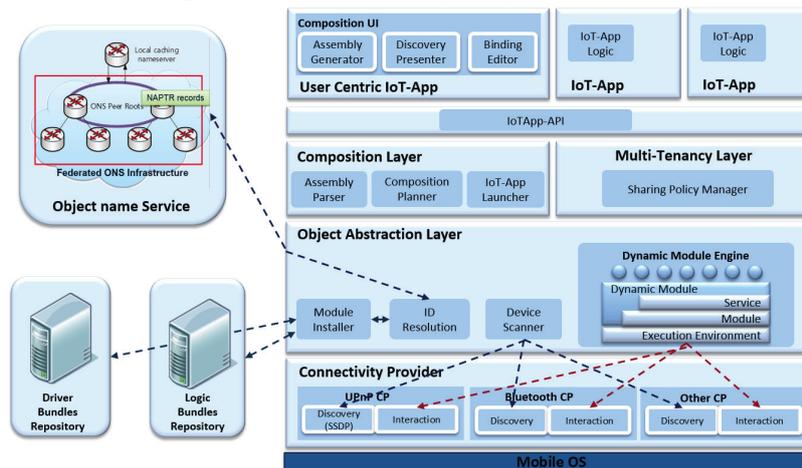


Figure 6. IoT-MAP Platform Architecture, as in [25]

3.6 OpenIoT

OpenIoT is an open service framework for the IoT which facilitates entrance into the IoT related mass market. It helps to setup a new IoT ecosystem with adoption of IoT devices and software. This takes place in phases and they also form the stakeholders of the platform [27]:

- Device developers produce IoT devices and register its platform’s APIs to an Open API portal.
- Software developers develop IoT apps for mobile devices, tablets which can fetch data from IoT devices, control them or transform the data fetched using the APIs. These can be registered on an App store.
- Service providers purchase IoT devices and register

them on the open IoT framework where they can be managed efficiently.

- Network operators focus on the mobile and wireless communication technologies.
- Consumers can find, connect and control using IoT devices searching service.

The main distinguishing feature is that this framework has support for B2C (business-to-consumer) and C2C (consumer-to-consumer) business model as well as B2B (business-to-business) and B2G (business-to-government) business models. The architecture is shown in Figure 7. OpenIoT consists of three server side platforms namely Planet Platform, Mashup Platform and Store Platform and one device side platform

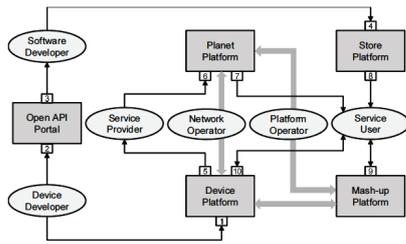


Figure 7. OpenIoT Architecture, as in [27]

called Device Platform. The function of the components are [27]:

Planet Platform A server side platform used for IoT device registration, management, monitoring, and searching.

Mashup Platform A service side platform for providing new integrated services based on mashup of data sets collected from IoT devices over the Internet.

Store Platform An App/Web store containing applications or links to Web address that provide user services - through interaction between IoT devices or Mashup Platforms.

3.7 ThingStore

ThingStore is an advanced app-store concept designed to facilitate collaboration on IoT applications development and a platform for deploying IoT applications. It is a platform which integrates smart devices called things, software and human users [28]. The platform aims to serve three kinds of users:

Thing Provider Things are smart devices and sensors which can be more intuitive through event detection software routines called smart services. A thing provider deploys his devices or things and announces smart services for them at the ThingStore marketplace.

Software Developers Develop IoT apps that query smart services using Event Query Language (EQL) quite similar to normal database applications developed over a database management system.

End User Subscribes to a particular app for notification and management stuff.

The overall architecture of ThingStore is depicted in Figure 8. Thing providers can be individual users or organizations who aim to deploy “things” to reach a wider audience. “Things” are treated as infrastructural assets in the platform. Applications can be developed on top of this. For example: a set of cameras located along a particular motor way can be used in an application. This sharing of “things” reduces the cost for software developers and also turns out to be profitable for thing providers. The life-cycle starts with deployment of “things”. These “things” produce data which can be consumed

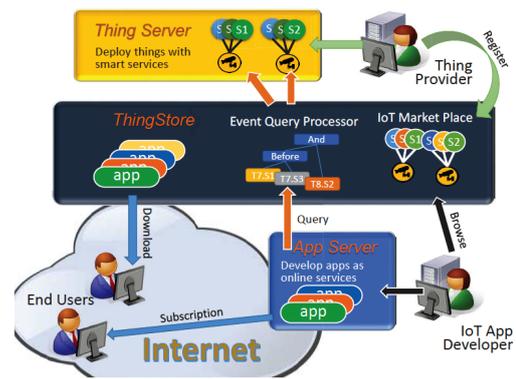


Figure 8. ThingStore Architecture, as in [28]

by applications. Things are said to be intelligent when they also have some associated software routines that automatically transform the raw data into something more meaningful.

There is a marketplace where “thing” providers can advertise about their things and services. To deal with device heterogeneity, ThingStore provides a service definition library which is used by the “thing” providers while defining their smart services. Software developers can develop applications by querying interested events like standard database applications interact with a standard database management system. ThingStore provides an SQL-like query language for applications to define and query events. Event computation and device management are handled by the platform so the developers only have to focus on business logic of the application. The IoT application developed atop this platform can be deployed here. End users can interact with the IoT environment through the developed and deployed applications which provide GUI [28].

3.8 IoTLink

IoTLink is a mashup development tool-kit based on a model driven approach which permits inexperienced developers to compose a mashup application through a graphical domain specific language [5]. It makes use of visual components to encapsulate graphical domain specific language which are then wired together to generate Java code. These visual components also act as points of abstraction for hiding the complexity involved in communication of different devices and services. The main idea is to help IoT application developers to easily handle technological challenges like heterogeneous network protocols, data format interoperability. The theoretical approach of the tool is to streamline the development process by defining the computation independent model (CIM) which is refined to platform-independent model (PIM) and which is detailed in a platform-specific model (PSM). Therefore it becomes easy for inexperienced developers to develop an IoT prototype as they just have to specify how different services are combined to form the final prototype. The resulting model can be subjected to transformation to generate complete stand-alone Java code.

IoT metamodels generally try to specify how physical objects should be represented by software services. Several European research projects like IoT-A after working on several aspects of IoT like standardization of IoT architecture have concluded that physical objects could be uniquely identifiable has physical qualities that can be observed with the help of sensors and has some capabilities that possibly can effect the environment. Physical objects are represented by virtual objects which are proxies to communicate with the actual device. Based on this concept, IoTLink’s platform-independent metamodel has four abstraction layers (Figure 9):

1. The first layer is responsible for abstracting the heterogeneous connections to physical sensors. This provides specific communication technologies and a uniform interface for other layers.
2. The second layer processes sensor data to determine the actual status of physical objects thereby treating noises in the data accumulated. It also encompasses complex algorithms needed to successfully fetch value from a particular kind of sensor.
3. The third layer abstracts the domain objects using an object oriented paradigm that represent the “Things” and their attributes.
4. The fourth layer exposes the domain objects to the application logic, distributed applications, as well as persistence storage.

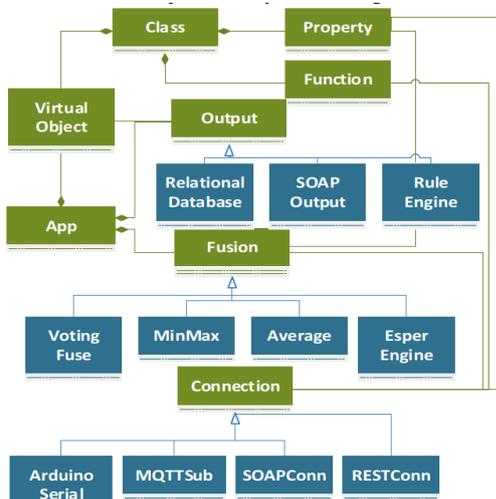


Figure 9. IoTLink meta-model (Logical view), as in [5]

IoTLink allows developers to define the applications in a platform-independent model through effective usage of visual notations which is then converted to platform specific model which in this case is Java. The platform has been developed as an Eclipse Plugin. Eclipse Modeling Framework (EMF) has been used to define the meta-model of the modeling language. Similarly, Eclipse Graphical Modeling framework (GMF) to

create a graphical editor and Extended Editing Framework (EEF) to create a property editor for the EMF elements. Aceleo has been used to create a model transformations from the EMF objects to Java code. The meta-model (Figure 9) has been implemented using a simplified UML called EMFCore (ECore). The high level architecture depicted in Figure 10 has been generated by GMF which is essentially the Graphical definition model, called “gmf-graph”. This defines the visual elements to be shown on the main canvas, properties, relations and constraints between diagrams etc. In addition to this, GMF creates a tooling definition model called “gmftool”, which defines the notations to be used on the palette menu. The gmfgraph and gmftools are then mapped in a mapping configuration to decide what notations are displayed on the screen when an item from the palette menu has been dragged and dropped to the main canvas. EEF plugin is used to create a property sheet for every diagram. The tool has several input components which allow a composition to interact with various devices for taking data streams as input like Arduino Serial devices, SOAP, REST, MQTT etc.

There is a concept of virtual object container, which allows the developers to define the physical object representations. This is of two types [5]:

StaticObject There is a stationary relation between physical objects and the sensors and actuators that monitor them. Example: a temperature sensor fixed to a wall.

MovingObject These objects have temporary relation to the sensors. Example: People moving from one location to another can be observed by the nearby sensors.

In addition to this there are output components which govern how the virtual objects are exposed to external applications like the object states can be stored to a RDBMS, exported as SOAP object, published to MQTT or exposed through REST etc. After the composition, IoTLink generates Java code based on the platform-independent model.

3.9 M3 Framework

Machine-to-Machine Measurement (M3) framework is a framework based on semantic web technologies that helps to build IoT applications, assists in sensor data interpretation and combines domains with each other [29]. Machine-to-Machine [30] is a part of Internet of Things to automate the communication between machines. Most of the IoT applications do not semantically interpret M2M data and the applications are not inter-operable with each other because they are domain specific [31, 32, 33]. The main objectives of M3 has been summarized in Figure 12.

Figure 11 shows the overall architecture of the framework. It has been split into several layers. The “perception layer” contains physical devices such as sensors, actuators and RFID tags. The “data acquisition layer” collects data from sensor devices in SenML format [34]. The data collected is also converted in a unified way (Resource Description Framework) as per M3 ontology. Resource Description Framework

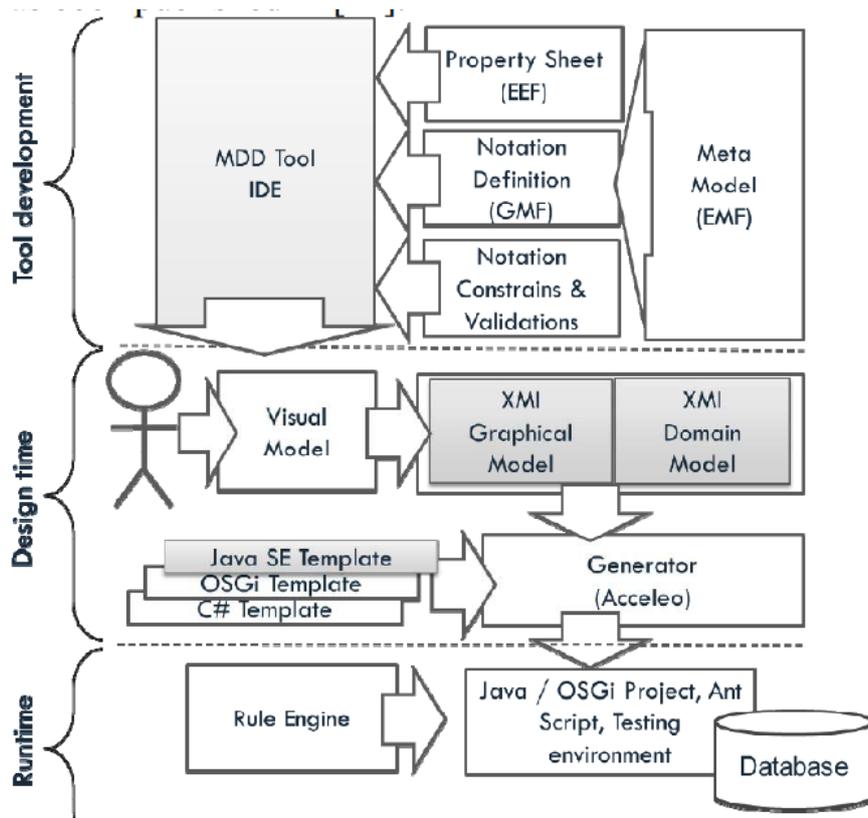


Figure 10. IoTLink High-level Architecture, as in [5]

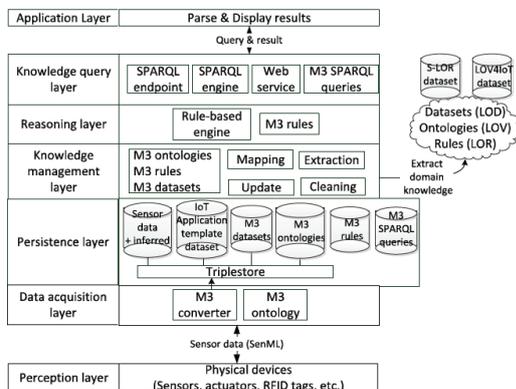


Figure 11. M3 Framework Architecture, as in [29]

(RDF) [35] is a basic semantic web language to describe triples composed of subject-predicate-object. The “persistence layer” stores M3 domain knowledge, semantic sensor data and inferred sensor data in triple store, a database to store semantic sensor data. It also contains necessary data sets to retrieve the domain knowledge to easily build an IoT application template. M3 rules and SPARQL queries are stored in files. SPARQL [36] is quite similar to SQL and is extensively used for querying semantic data. The “Knowledge management”

layer is responsible for finding, indexing, designing and combining domain-specific knowledge like datasets to update M2 domain ontologies. The “reasoning layer” infers high-level knowledge using reasoning engines and M3 rules extracted from Sensor-based Linked Open Rules(S-LOR) [37] M3 rules work with M3 ontology to infer new knowledge on the sensor data. The “knowledge query” layer executes SPARQL queries on inferred sensor data. The “application layer” has an application which parses and displays the results to users.

The Operation process of M3 is depicted in Figure 13.

3.10 Other Prominent Mashup Tools

There are several other tool-kits which are used in IoT landscape. Some of them are quite popular and they have been briefly described below:

3.10.1 ThingWorx

ThingWorx platform aims to build and run applications for the IoT landscape using a so-called model-driven approach [3]. It composes services, applications and sensors as data sources and interconnects these through a virtual bus. The framework supports a wide range of connection protocols for devices like CoAP, MQTT, REST/HTTP and Web Sockets. It can integrate with other cloud providers such as Xively and web services such as Twitter, Facebook or various weather services as data sources. Once data sources are connected to dashboards, they

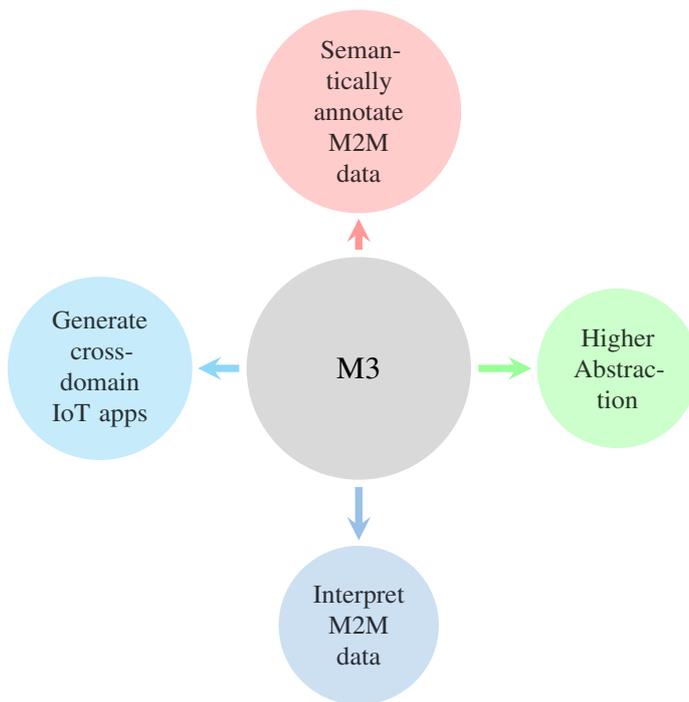


Figure 12. M3 Framework Objectives, following [29]

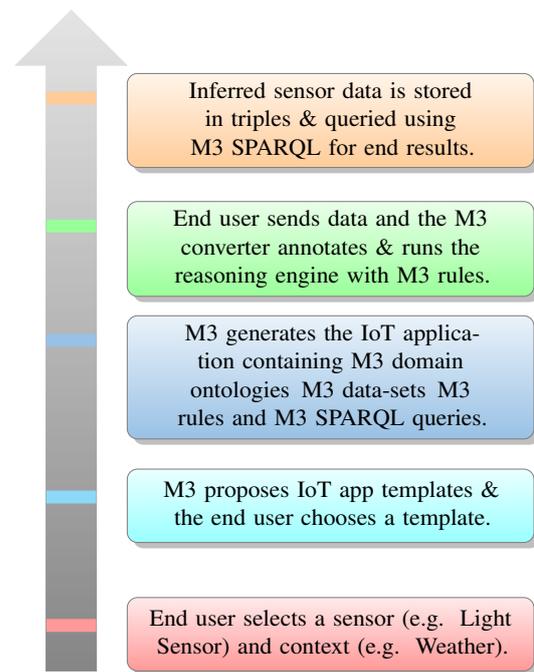


Figure 13. M3 Process, following [29]

can be used for data gathering and monitoring and can be mashed up to create mashup applications. The data can also be subjected to analytics.

3.10.2 Paraimpu

Paraimpu is a web-based platform which allows to add, use, share and interconnect real HTTP-enabled smart objects and “virtual” things like services on the Web and social networks [38]. User can easily create IoT applications to facilitate their devices to react to environmental changes and activities [20]. In order to have a unifying view on different devices, these devices are segregated based on their functionality. “Sensors” are devices/services capable of producing data in an acceptable format while “Actuators” are entities that can consume data and in the process of consumption generate some actions. Sensors and actuators communicate using the HTTP protocol and therefore it is easy to create hybrid mashups.

3.10.3 Xively

Xively is a cloud based IoT platform formerly known as Pachube. The architecture is depicted in Figure 14. The platform provides a central message bus to route messages between devices using different protocols. The message bus combined with the Xively API for MQTT, HTTP, and Web Sockets strives to provide an interoperability layer. Based on the client server model the configuration of devices is done in a centralized way where each device has a virtual presence and when a device comes online it uses its serial number and some form of mutual authentication to receive its configura-

tion parameters setup on the Xively server [3].

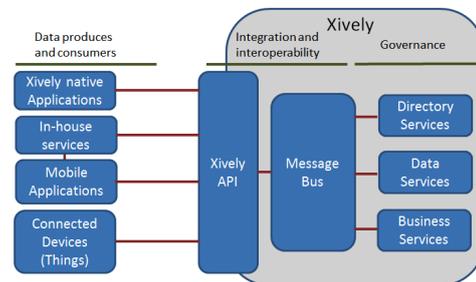


Figure 14. Xively Platform Architecture, as in [3]

3.10.4 PyoT

IoT applications exploiting the data produced by IoT devices are required to fully exploit the possibilities offered by the IoT landscape. In order to facilitate the widespread adoption of IoT the methodologies for application development needs to be simplified. One of the proposal is to use the concepts of macroprogramming [39, 40]. It enables the development of applications involving a large number of nodes while hiding the low level implementation details. PyoT is a macro-programming framework based on standard protocols like CoAP which aims to simplify the management of complex IoT networks and provides a convenient interface for application developers. It abstracts the IoT devices as resources which can be combined to perform useful and complex tasks. Networks, nodes, sensors and actuators are represented as

objects in a high-level scripting language [41].

4. Comparison of Mashup Tools

The mashup tools and platforms for IoT landscape have been described above from a high level with their key features. One of the common objective of these mashup tools is to reduce the development time of applications for the IoT landscape. It is quite interesting to mention a difference between IBM Node-RED and other tools described above. Node-RED is just a visual programming environment and not a complete platform by itself. A platform has support for both device and application management. For instance, it does not provide a device management layer, so we cannot explicitly register IoT devices to it but it supports a wide range of connection protocols enabling it to communicate to different devices. This limitation is eliminated in other tool-kits like glue.things which is a platform in itself. It provides support for device registration and management and uses an improved version of Node-RED as its mashup tool i.e Nod-RED is embedded with in this tool to provide a complete IoT platform functionality. Here, we briefly compare the tools with respect to some dimensions as indicated in the sub-sections below.

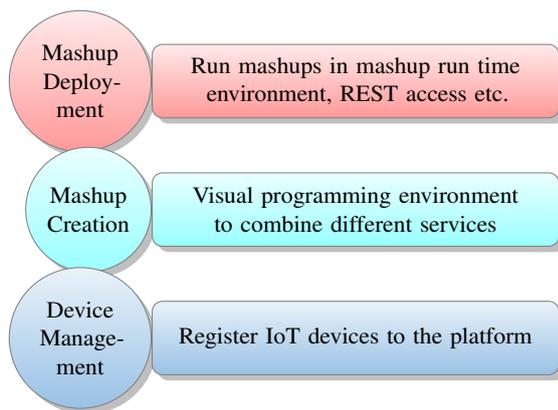


Figure 15. Conceptualization of Features Available in Mashup Tools

4.1 Terminological Differences

After careful observation of many exiting tool-kits, it is appropriate to say that they use different terminologies to denote similar concepts. Mashups are known by different names in different tool-kits but in essence they reflect the same conceptual approach. For example, in Node-RED a mashup is called as a flow while in WoTKit it is called a process. The created mashups are generally deployed in a mashup run-time environment. Here the name of the run-time environment differs. For example, it is called “Smart Object Marketplace” in glue.things while “RESTful Platform API module” in WoTKit.

Figure 15 summarizes the essential features provided by these tool-kits under the banner of different terminologies.

Difference arises in the features provided by these tool-kits in these three distinct layers of service. For example in “Device Management”, the protocols supported by a toolkit with which we can connect and register IoT devices vary. Almost all the tool-kits support common protocols like MQTT and CoAP. But glue.things also has support for extra protocols like PubNub (Real time publish/subscribe messaging API for web and mobile apps), Meshblu (Machine to machine instant messaging network and API) etc. Similarly, in “Mashup Creation”, Node-RED permits the user to embed JavaScript codes while WoTKit has support for Python scripting. In “Mashup Deployment” almost all tool-kits provide the same features which include sharing of created applications and accessing them by REST APIs.

4.2 Methodological Differences

Although the mashup tools vary in degree to which they strive to ease the development process but nevertheless the underlying concepts they adopt is the same. Almost all tools, e.g. WoTKit or Node-RED rely on the concepts of data flow for developing an IoT application. Different data streams from different devices are connected in a logical way and data transformation is applied during the transit of the data. ThingWorx advertises to heavily rely on model-based software development approach for creating IoT applications but nevertheless we believe that the underlying concepts used and features offered by the platform largely correspondent to other existing platforms. However IoTLink uses a model-driven approach to build applications from a graphical domain specific language.

5. Strengths and Weaknesses

IoT environment provides many beneficial services by connecting devices to the Internet. But simple data accumulation and processing of raw data does not convey much. Applications in IoT is unavoidable to fully leverage the offerings of this emerging world. The development of applications in IoT landscape requires a great amount of skills and expertise. It is also important to understand that most of these applications are to be developed in an adhoc fashion by end-users on top of smart devices, mostly by using the concepts of mashups [23].

Mashups can be readily applied in IoT environments if most of the components are available in the form of web services. But there are certain challenges faced by mashup tools in IoT environments like:

1. It is difficult to handle a large number of heterogeneous IoT devices.
2. The intermittent behavior of devices makes interactions with them unpredictable.
3. The life-cycle of data streams in an IoT environment is uncertain as the device can be unplugged by the owner any time.
4. The mashup tool has to deal with dynamic changes in the IoT environmental topology. Devices come and go

and their locations cannot be predetermined. Therefore the mechanisms to dynamically detect devices, data availability before offering the user an opportunity to mashup is a major challenge.

5. Mashup tools also have to deal with strict data privacy and security requirements.

5.1 Strengths

The main strength of the above described tools is that they definitely assist the user to develop an application, abstract the low-level complexity to some degree and are flexible and intuitive to a great extent. It is imperative that no tool can strike the right balance between functionality and simplicity. Some of the core strengths of these tools are (Figure 16 summarizes the key points):

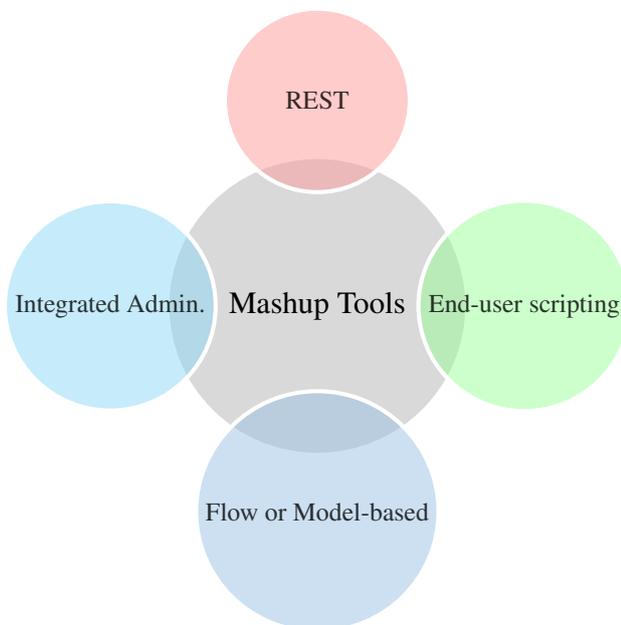


Figure 16. Summarized Strengths of Mashup Tools

5.1.1 REST Architectural Style

There are two widely used organizational styles for the web namely the Service-oriented architectures (SoAs) and the resource-oriented architectures (RoAs). SoAs are software architectures that make the service central to the web service design. The protocol used is SOAP which uses XML messages over HTTP. While RoA makes the resource central in the web service organization. They strongly emphasize the way a resource is identified. In the context of Machine-to-machine(M2M) communications, the main benefit of RoA is uniformity. REST is much more flexible than SOAP which employs XML over another application protocol, e.g., HTTP or SMTP limiting the M2M communication interoperability and is also a considerable communication overhead for the resource constrained devices found in usual IoT scenarios.

REST is also used in an application protocol for constrained devices, the Constrained Application Protocol (CoAP) over UDP. This makes REST an ideal choice for achieving the IoT vision of a totally-connected physical world. Additionally, the actual implementation of SOAP-based web services is often more complex than the REST services [42].

The main strength of the mashup tool-kits is that they support the usage of REST architectural style. With the usage of RESTful APIs data accumulation from different sources becomes easy within the tool-kit. The data providers (data generated from IoT devices) are assured that they are cross-compatible and can be mashed up upto a greater degree. With other protocols, the handshake between the tool-kit and device which is the prerequisite for device integration and data accumulation, becomes unnecessarily complex. This new protocol is extremely simple in design, adding minimal new rules to normal HTTP verb behaviors [43]. The tool-kits described above support REST architectural style for created applications even. The applications when deployed are also accessible by REST APIs. This enables the created applications to be re-used in a new application mashup, shared online or even traded.

5.1.2 Stakeholder Segregation

Tools like EcoDiF, IoT-MAP, ThingStore, IoTLink, etc. clearly segregate the IoT landscape into three distinct stakeholders namely device manufacturers, data producers and developers. Device manufacturers are only concerned to make the hardware functionality available with APIs following some guidelines. Data providers are actual device owners and they register their device with the tool-kit using a specific set of APIs. Application developers can focus solely on the business logic of the application without caring for connectivity protocol issues. This kind of segregation abstracts away the complexity, introduces pillars of interoperability thereby making the application development process innovative and intuitive.

5.1.3 Integrated Administration

One of the strongest feature of the tool-kits is that they help in device registration, management, creation and deployment of applications through a centralized interface where the user actually interacts to perform the tasks. This simplifies the IoT context as device are scattered, heterogeneous and use different connectivity protocol. The mashups created are deployed on a separate cloud based infrastructure. With such a setup things get complicated if the user has to login to separate interfaces to see the devices, deployed application or perform some administrative tasks. The tool-kits are cloud based i.e they provide the hosting platforms and application API for interacting between devices from applications running in the cloud. This is especially good for business platforms where a centralized application's presence is highly sought [3]. For example in a large scale factory, installing temperature sensors, gathering and analyzing data from them manually is tedious. But if there is a centralized IoT platform offering device registration and management services then implementation and

maintenance of an IoT scenario becomes relatively easy. The administrator need not remember the physical address of all the innumerable temperature sensors scattered throughout the factory, instead just login to the centralized platform to look how the devices are functioning, select some devices to check their data and even name the devices for easy reference and remembrance!

5.1.4 Developmental Methodology

The tool-kits employ either a mashup based approach (e.g., Node-RED, glue.things etc.) or a model-based approach (IoTLink etc.) to assist the user in application development. Mashup approaches are relatively simple and they model the flow of data between different components very efficiently. On the otherhand, model-based approaches rely on specifications using a domain specific language and then the specification is subjected to transformations to generate the application. The expressiveness to model complex situation is inherently high with this approach [4], but so is the complexity. The type of developmental methodology a tool employs solely depends on the level of users it is targeting to serve, the environmental context and the user requirements. If the users are completely novice to programming, the requirements are quite simple then mashups serve the need but in complex enterprise scenarios model-based approaches fit adequately into the graph.

5.1.5 End-User Scripting

The tools depending on the developmental approach allow the user to add some custom code to enhance the business logic of the IoT application. In flow based tool-kit like Node-RED the user can add Java Script codes while WoTKit has support for python code. ThingStore supports a SQL like programming language to insert some querying logic. The end-user scripting facilitates to add additional logic which remain normally inaccessible when solely relying on the GUI options of the tool-kits. For instance, in Node-RED while creating a mashup if a user wants to express a for-loop or may be check for an arithmetical error then the definition, solely with the usage of GUI components, becomes complex. It is here that the user feels the necessity of condition and logic specifications in the form of code snippets or possibly even pseudo-code. These code-snippets as well as the GUI components joined in the form of a diagram are translated by the tool-kits to generate the final code which forms the mashup application. The model-based tool-kits like IoTLink have the support to express almost anything within the dimensions of the domain specific language which is finally translated to generate the application code in a specific target language like Java or Python.

5.2 Weaknesses

Some of the weaknesses found in the existing IoT application development tool-kits and frameworks have been summarized below (Figure 17 summarizes the key points).

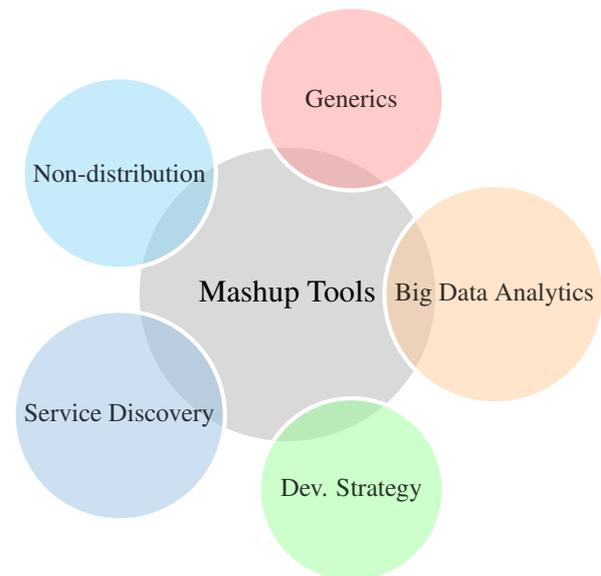


Figure 17. Summarized Weaknesses of Mashup Tools

5.2.1 Service Discovery

Depending on the context where an IoT infrastructure has been setup, e.g., connected mobility, the presence of IoT devices is uncertain. The devices join and leave the network at unprecedented events. The devices can also change the services offered. Dynamic detection of devices and their services is a challenging task. Some devices specify services in specially formatted files like XML while others have a dedicated service as a lookup point to gain information about services offered. Detection of service specification of different devices itself is cumbersome and difficult. Additionally parsing of the service information resulting in service recognition and listing of available services within the tool-kit is fairly an uphill task. After this review, it would be fairly justified to state that the existing tool-kits have very primitive level of device and service discovery support. This needs to be enhanced in order to realize the true potentials of IoT in a dynamic environment like connected mobility. Some guidelines can be standardized to effectively document services offered by a device to minimize the hassles involved in service detection and recognition.

5.2.2 Lack of Generics

In almost all the IoT application development tool-kits except M3, the application developed are already instantiated i.e they belong to the object level. To be more explicit, they are not generic applications. Since the application development is done by end-users, there is a high probability that the situation they are trying to accomplish matches to the business logic of an already existing application. If the same application can be reused just by proving new data sources and context information then the application development landscape would be benefited tremendously. For example, a mashup which is used to turn off the lights in a building can also be used to turn

off some other equipment in the same or any other building. This is only possible if the business logic and the context information are clearly demarcated i.e made generic. A number of strategies have been proposed on how to achieve this [44] within the IoT tool-kits but as of now most of the tool-kits do not support this concept. The generic support level claimed by the developers of M3 framework is not backed by sufficient statistical data and we are unsure of the level of support it actually offers to cater to this specific requirement of generics.

5.2.3 Developmental Strategy

The existing tool-kits either solely rely on mashup strategies or model-based approaches for IoT application development. This seldom strikes the right balance between functionality and simplicity. The result is that the tool is either extremely simple or extremely powerful (also complex). It would be nice if a tool-kit uses a combination of both these developmental strategies then it would strike the right balance. The idea is to allow mashing up of services and specification of complex scenarios with the usage of a domain specific language.

5.2.4 Lack of Distributed Deployment

The applications created in the existing tool-kits like glue-things are generally deployed locally on some specific cloud based infrastructures. In case of Node-RED the application is deployed locally on the device itself. This leads to certain challenges and problems. Because the deployed application itself is accessible by REST APIs, this means that this can be used as an input in a new mashup. During the execution of this new mashup if one of the constituent service (locally hosted on an IoT device) is inaccessible then the entire application fails to execute. This problem arises because the created applications are deployed locally and there is no concept of distributed deployment to provide a higher degree of fault tolerance and reliability to the IoT applications which is crucial for the success of IoT in domains of Connected Mobility (environmental dynamism). The concept of distributed data flow and application deployment has been worked upon to certain degree of success in the Fog computing model, realized by an implementation of Node-RED called “Distributed Node-RED (D-NR) but needs to be further investigated for precise conclusions [45].

5.2.5 Big Data Analytics

Connecting a large number of physical objects with sensors generates “big data”. The IoT paradigm relies on the concept of interconnected objects which communicate with each other, collect data about their context. After days of interaction, this situations tends to produce zeta bytes of data. Big data needs smart and efficient storage. The real market value of IoT can be exploited if big data analytics can be integrated in the realms of IoT. Thus IoT is a perfect prototypical example of Big Data. A great amount of effort has been directed to collect data from sensor devices and store them in a Big Data infrastructure and possibly perform analytics on the accumulated data to gain insights on the environmental context [46].

But no work has been done in the community to couple Big Data analytics with IoT mashup application creation. It would be great to have a mashup tool to create a mashup which can perform real time analytics. To have a mashup application which can intelligently suggest routes to users depending on live traffic conditions is an apt example of big data analytics and mashup coupling. The main challenge is how to model the analytics logic and sequence graphically within the context and dimensions of the mashup tool, which can then be mapped to equivalent code in a big-data environment to perform the actual analytics and return back the result to the application and govern the next course of the application.

6. Concluding Remarks

The report summarizes the IoT landscape and the needs for IoT application development. The task being a challenging one and generally accomplished by end-users calls for a tool-kit to provide good amount of abstraction thereby lowering the learning curve. The most prominent tool-kits supporting IoT application development have been summarized and have been compared adequately.

The report discusses the needs of an IoT context and how the tool-kits cater to those needs signifying their inherent strengths. However some complex issues where the tool-kits fail to deliver appropriately throws light on some of the existing open research challenges in the domain of IoT application development.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] S. Van Themsche. *The Advent of Unmanned Electric Vehicles*. Springer International Publishing, Cham, 2016.
- [2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787 – 2805, 2010.
- [3] H. Derhamy, J. Eliasson, J. Delsing, and P. Priller. A survey of commercial frameworks for the internet of things. In *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–8, Sept 2015.
- [4] Christian Prehofer and Luca Chiarabini. From Internet of Things Mashups to Model-Based Development. In *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, pages 499 – 504. IEEE, July 2015.
- [5] F. Pramudianto, C. A. Kamienski, E. Souto, F. Borelli, L. L. Gomes, D. Sadok, and M. Jarke. Iot link: An

- internet of things prototyping toolkit. In *Ubiquitous Intelligence and Computing, 2014 IEEE 11th Intl Conf on and IEEE 11th Intl Conf on and Autonomic and Trusted Computing, and IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UTC-ATC-ScalCom)*, pages 1–9, Dec 2014.
- [6] C. Peltz. Web services orchestration and choreography. *Computer*, 36(10):46–52, Oct 2003.
- [7] Florian Daniel and Maristella Matera. *Mashups: Concepts, Models and Architectures*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
- [8] Arthur Ryman. Simple object access protocol (soap) and web services. In *Proceedings of the 23rd International Conference on Software Engineering, ICSE '01*, pages 689–, Washington, DC, USA, 2001. IEEE Computer Society.
- [9] D. Ma. Offering rss feeds: Does it help to gain competitive advantage? In *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, pages 1–10, Jan 2009.
- [10] J. Wang, Z. Xu, and J. Zhang. Implementation strategies for csv fragment retrieval over http. In *2015 12th Web Information System and Application Conference (WISA)*, pages 223–228, Sept 2015.
- [11] Volker Hoyer and Marco Fischer. Market overview of enterprise mashup tools. In *Service-Oriented Computing-ICSOC 2008*, pages 708–721. Springer, 2008.
- [12] Michael Blackstock and Rodger Lea. Iot mashups with the WoTKit. In *Internet of Things (IOT), 2012 3rd International Conference on the*, pages 159–166. IEEE, 2012.
- [13] Antonio Pintus, Davide Carboni, and Andrea Piras. Paraimpu: a platform for a social web of things. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 401–404. ACM, 2012.
- [14] Dominique Guinard, Vlad Trifa, Friedemann Mattern, and Erik Wilde. From the internet of things to the web of things: Resource-oriented architecture and best practices. In *Architecting the Internet of Things*, pages 97–129. Springer, 2011.
- [15] E Michael Maximilien, Hernan Wilkinson, Nirmal Desai, and Stefan Tai. *A domain-specific language for web apis and services mashups*. Springer, 2007.
- [16] Jin Yu, Boualem Benatallah, Fabio Casati, and Florian Daniel. Understanding mashup development. *Internet Computing, IEEE*, 12(5):44–52, 2008.
- [17] Jeffrey Wong and Jason I Hong. Making mashups with marmite: towards end-user programming for the web. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1435–1444. ACM, 2007.
- [18] Nick Health. How ibm's node-red is hacking together the internet of things, March 2014. TechRepublic.com [Online; posted 13-March-2014].
- [19] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, and G. Ferrari. Iot-oas: An oauth-based authorization service architecture for secure services in iot scenarios. *IEEE Sensors Journal*, 15(2):1224–1234, Feb 2015.
- [20] Robert Kleinfeld, Stephan Steglich, Lukasz Radziwonowicz, and Charalampos Doukas. glue.things: A Mashup Platform for wiring the Internet of Things with the Internet of Services. In *Proceedings of the 5th International Workshop on Web of Things, WoT '14*, pages 16–21, New York, NY, USA, 2014. ACM.
- [21] Michael Blackstock and Rodger Lea. Wotkit: A lightweight toolkit for the web of things. In *Proceedings of the Third International Workshop on the Web of Things, WOT '12*, pages 3:1–3:6, New York, NY, USA, 2012. ACM.
- [22] D. Thangavel, X. Ma, A. Valera, H. X. Tan, and C. K. Y. Tan. Performance evaluation of mqtt and coap via a common middleware. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6, April 2014.
- [23] Flavia C. Delicato, Paulo F. Pires, Thais Batista, Everton Cavalcante, Bruno Costa, and Thomaz Barros. Towards an iot ecosystem. In *Proceedings of the First International Workshop on Software Engineering for Systems-of-Systems, SESoS '13*, pages 25–28, New York, NY, USA, 2013. ACM.
- [24] Eeml: Extended environments markup language.
- [25] S. Heo, S. Woo, J. Im, and D. Kim. Iot-map: Iot mashup application platform for the flexible iot ecosystem. In *Internet of Things (IOT), 2015 5th International Conference on the*, pages 163–170, Oct 2015.
- [26] N. Elleuch, A. Khalfallah, and S. B. Ahmed. Software architecture in model driven architecture, March 2007.
- [27] J. Kim and J. W. Lee. Openiot: An open service framework for the internet of things. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 89–93, March 2014.
- [28] Katalmics Akpınar, Kien A. Hua, and Kai Li. Thingstore: A platform for internet-of-things application development and deployment. In *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems, DEBS '15*, pages 162–173, New York, NY, USA, 2015. ACM.
- [29] A. Gyrard, S. K. Datta, C. Bonnet, and K. Boudaoud. Cross-domain internet of things application development: M3 framework and evaluation. In *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, pages 9–16, Aug 2015.
- [30] David Boswarthick, Omar Elloumi, and Olivier Hersent. *M2M Communications: A Systems Approach*. Wiley Publishing, 1st edition, 2012.

- [31] B. Manate, V. I. Munteanu, and T. F. Fortis. Towards a smarter internet of things: Semantic visions. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2014 Eighth International Conference on*, pages 582–587, July 2014.
- [32] Daniele Miorandi, Sabrina Sicari, F. De Pellegrini, and Imrich Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.
- [33] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang. A vision of iot: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things Journal*, 1(4):349–359, Aug 2014.
- [34] Cullen Jennings, Zach Shelby, and Jari Arkko. Media Types for Sensor Markup Language (SENML). Internet-Draft draft-jennings-senml-10, Internet Engineering Task Force, April 2013. Work in Progress.
- [35] O. Lassila and R. Swick. Resource description framework (RDF). model and syntax specification. Technical report, W3C, 1999. W3C Recommendation. <http://www.w3.org/TR/REC-rdf-syntax>.
- [36] Bob DuCharme. *Learning SPARQL*. 2011.
- [37] Amélie Gyrard, Christian Bonnet, and Karima Boudaoud. Helping IoT application developers with sensor-based linked open rules. In *SSN 2014, 7th International Workshop on Semantic Sensor Networks in conjunction with the 13th International Semantic Web Conference (ISWC 2014), 19-23 October 2014, Riva Del Garda, Italy, Riva Del Garda, ITALIE*, 10 2014.
- [38] Antonio Pintus, Davide Carboni, and Andrea Piras. Paraimpu: A platform for a social web of things. In *Proceedings of the 21st International Conference on World Wide Web, WWW '12 Companion*, pages 401–404, New York, NY, USA, 2012. ACM.
- [39] Luca Mottola and Gian Pietro Picco. Programming wireless sensor networks: Fundamental concepts and state of the art. *ACM Comput. Surv.*, 43(3):19:1–19:51, April 2011.
- [40] Salem Hadim and Nader Mohamed. Middleware: Middleware challenges and approaches for wireless sensor networks. *IEEE Distributed Systems Online*, 7(3), 2006.
- [41] Andrea Azzarà, Daniele Alessandrelli, Matteo Petracca, and Paolo Pagano. Demonstration abstract: Pyot, a macro-programming framework for the iot. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, IPSN '14*, pages 315–316, Piscataway, NJ, USA, 2014. IEEE Press.
- [42] M. Dissegna, R. Manfrin, M. Rotoloni, L. Vangelista, and M. Zorzi. Ral: a restful m2m communications framework for iot. In *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1096–1101, Aug 2015.
- [43] Charles Engelke and Craig Fitzgerald. Replacing legacy web services with restful services. In *Proceedings of the First International Workshop on RESTful Design, WS-REST '10*, pages 27–30, New York, NY, USA, 2010. ACM.
- [44] Christian Prehofer and Dominik Schinner. Generic Operations on RESTful Resources in Mashup Tools. pages 1–6. ACM Press, 2015.
- [45] N. K. Giang, M. Blackstock, R. Lea, and V. C. M. Leung. Developing iot applications in the fog: A distributed dataflow approach. In *Internet of Things (IOT), 2015 5th International Conference on the*, pages 155–162, Oct 2015.
- [46] C. Cecchinell, M. Jimenez, S. Mosser, and M. Riveill. An architecture to support the collection of big data in the internet of things. In *2014 IEEE World Congress on Services*, pages 442–449, June 2014.

Platform Business Models

Aenne Schweiger, Julius Nagel, Markus Böhm and Helmut Krcmar

Department of Informatics, Technical University of Munich, Munich
{aenne.schweiger; julius.nagel; markus.boehm; krcmar}@tum.de

Abstract

This State of the Art report is the first result of the subproject Business Model for Platform Provider within the Living Lab Connected Mobility project at Technical University Munich. The aim of the subproject is to conceptualize a configurable business model for a connected mobility platform. The purpose of this report is to first give an overview of the research in the field of platform business models and collect possible component of business models for further research. The three key findings of this report are: 1. There is a disagreement on a common definition for platforms and business models have been revealed. The kind of platform this report focusses is named Digital Platform, Two- or Multi-Sided Market or Network, as concluded from relevant articles. 2. Eleven components have been identified for platform business models. 3. Lifecycle and Competition have been identified as platform specific business model components, which should be further investigated, as guidelines for platform management can be derived. The key methods applied were a literature review and the creation of a concept matrix.

Keywords

Platforms; Two-Sided Markets; Multi-Sided Markets; Business Model Components

Introduction

The Rise of Platforms

Artificial Intelligence (AI) and Machine Learning are among the most hyped technologies at the moment and their applications are said to be completely changing the way we live in the near future. So when Google announced that it would open its own AI engine “TensorFlow” to external developers in 2015 [1], it left people wondering: Why would they open source, what is possibly their most important competitive edge at the moment? Only about a month later, Facebook followed, by open sourcing their own AI development [2] and in early 2016, a group of deep learning researchers (with the help of Tesla founder Elon Musk), brought “OpenAI” into being - an open innovation platform for the development of next level artificial intelligence applications [3]. The examples above display the significance of platforms for innovation and development. Same can be said for transaction platforms, where companies such as Airbnb and Uber have become very successful in a short period of time, by connecting multiple sides in a very efficient way. This development shows that even the biggest and most successful companies of our time have understood the power of platforms (the top 15 public platform companies already make up a market cap of 2.6 trillion US Dollar [4]). People are getting more and more connected through faster and better internet availability and the possibilities to store and process large amounts of data have become ubiquities. It is also why, in the light of an ongoing trend towards further connected, “smart” businesses and real time data analytics, platforms will become even more attractive and might emerge as the prevailing business model. This “Rise of the Platform”,

which has attracted lots of media attention recently [5] [6], has not been met by a thorough understanding about what a platform is and what parts of its business model are influencing the success [7]. We want to contribute to this understanding and aim at closing this research gap further, by conducting a comprehensive literature review on platform business models.

The Project: Living Lab Connected Mobility

The Living Lab Connected Mobility (LLCM) aims at researching a reference architecture for a connected mobility platform. The platform should offer interfaces for mobility services and tools for developers which they can use to either connect with a service of the platform or to develop and add a new service to the platform. Services, which address the first category are, for example, being researched within the subprojects Eco-Sensitive Traffic Management and Collaborative and Social Mobility Services. Tools concerning the second category are addressed, for example, by the subproject Service-Mashups and Developer Support.

All of these services and tools should be offered on one platform that like any specific service must be designed cost-efficiently and sustainably. Whereas some subprojects work on the technological concept, the subproject this report belongs to works on economic issues and aims at creating a reference model of a configurable Platform Business Model. As defined by the project brief, the business model must consider three conditions: It must be sustainable, configurable and integrate all potential partners from start-ups to large corporations. For giving this research a structure, we came down to the following three research questions: 1. What is the state of the art of platform business models? 2. Which

are the relevant elements of a platform business model in the context of connected mobility? 3. How can a configurable business model for a connected mobility platform look like? This research gives an overview of the research in the field of platform business models and will identify components for platform business models and academic voids. Therefore, it provides an answer to question 1 and an outlook on possible answers to question 2.

The remainder is structured as follows: In the next chapter theoretical background information on platforms is summarized, before the method of research is explained. Then we derive a definition of platform according to the results of our literature review, that we will apply for extracting relevant papers. Afterwards, we present four research levels of how platform business models have already been examined. Going through the literature of each of the four levels, we identify business model components which are consolidated in the chapter Platform Business Model Components. In the last two parts, we discuss our findings, show limitations and draw conclusions for further research.

Theoretical Background on Platforms

With the rise of the Windows operating system in the late 1990s platform ecosystems reached a high interest within the research community. As Schrieck et al. [8] summarize, “IS research tries to understand how successful platform ecosystems in the IT industry need to be designed and governed [9], [10], [11]. Researchers analysed the technical requirements of software platforms [12], characteristics of successful platforms [13], optimal pricing for platform-based businesses [14] and control mechanisms applied on platforms [15].” Authors such as Gawer and Cusumano, Eisenmann, Parker, Van Alstyne and Tiwana have contributed fundamental research and have been frequently cited in the papers we found. Their work considers the leadership of a platform depending on the establishment of standards [16], the positioning of a platform between vertically integrated firms, resellers or input suppliers [17], the transformation of platforms over time [18] and strategies concerning pricing and control [19].

Schrieck et al. [8] see two perspectives on platforms in research: technology-oriented and market-oriented. “According to the technology-oriented perspective, a platform is defined as ‘a set of stable components that supports variety and evolvability in a system by constraining the linkages among the other components’ [12], whereas from the market-oriented perspective ‘platform ecosystems can be seen as ‘markets, where users’ interactions with each other are subject to network effects and are facilitated by a common platform provided by one or more intermediaries’ [20]. In the context of LLCM, we will focus on the market orientation, as will be explained in depth in the second next chapter.

Methodology

In order to capture the state of the art in research of platform business models we conducted a literature review according to Vom Brocke et al. [21]. We searched EbscoHost, ScienceDirect and Scopus. In the EbscoHost search all databases have been included and we searched with the keywords “platform AND business AND model” in abstract and title, separately, and received altogether 227 results. In ScienceDirect, we searched the same keywords within abstract, title and keywords and received 341 results. The same search pattern was applied on Scopus, although his database was only limited to the Senior Scholar’s Basket of Eight which delivered another 16 results. We retrieved 584 results over all databases. In a second step the results were reduced to 366 by filtering of academic journals and removal of duplicates. The abstracts of all these results have been checked for relevance regarding the following leading questions: 1) Is the understanding of the term “platform” congruent to ours? 2) Is the view on the platforms in focus holistic or specific? 3) Are specific components of a platform business model in the focus of research? After classifying the articles, we had 76 articles left. The omitted articles were either focussing on details of platform operations, addressed technological issues of software ecosystems or examined platforms in a surrounding out of scope of this research, like for example supply chain information exchange platforms or oil platforms. Reading these articles in depth left 27 relevant articles. These articles have then been coded with platform business model components according to Webster and Watson [22]. After the coding we clustered the elements and received 11 Platform Business Model components.

Definition of Platform

Before we start analysing platform business models a clear understanding of the term “platform”, as we will use it, is needed. Hefele [23] has just recently completed a literature review concerning the definition of platform and related terms, which we will summarize briefly and comment regarding our own literature review.

Three Schools of Thought about Platforms

Hefele identified three views or rather thought schools on platforms according to Baldwin and Woodard [12]: 1) product development, 2) technology strategy and 3) industrial economy. The product development perspective sees a platform as a “basis from which different products, resulting in a product family, could be derived by modifying features” [23]. The advantage here is that due to easy modification in complete functions the same platform can be produced with economies of scale which results in cost reductions. A product platform is, for example, a technical base of a car which can be transformed to a customized product by combination with a range of complementary and individual parts. Worth mentioning is that Gawer [24] describes such platforms as “internal platforms” as they are usually proprietary goods and

either used within a company or within the supply chain of a manufacturing company. The technology strategy encompasses similar advantages as the product development but on a more information-technological level: Here a platform is “developed by one or several firms, and [...] serves as foundation upon which other firms can build complementary products, services or technologies” [23] [24]. Representatives in this area are operation systems (Windows), web browsers or smartphones (iPhone), which can both be expanded by the development of programs and applications. The access to this kind of platform granted to more than just one company shows clearly that this is not an “internal”, but rather an “external platform”. The third stream is called the industrial economy. Its understanding of platform still includes a technological base but focuses more on the interaction function that it provides. Analog expressions for platforms in this stream are two- or multi-sided markets or networks. “Two-sided platforms are specific multi-sided platforms that bring together two distinct but interdependent groups of customers. They create value as intermediaries by connecting these groups (Eisenmann, Parker, and Van Alstyne, 2006; Osterwalder, Pigneur, and Smith, 2010)” [25]. Two- or Multi-Sided Platforms are “social networking platforms such as Facebook which link networks of users with the providers of various services and applications, e-commerce websites such as Amazon or eBay, which bring together buyers and sellers, and search engine platforms such as Google, which connect advertisers and Web users” [26]. Our understanding of platform correlates the most with this third stream. Within the relevant articles of our literature review we found different platform terms, but with the same understanding, such as in Hagi et al. (Multi-Sided Platform), Cennamo et al. (intermediaries between the users and service providers in the markets characterized with the indirect network effects), Evens et al. [27] and Henten and Windekilde [28](Multi-Sided Platform), Rochet and Tirole (Two-Sided Platform) [29] and de Pablos et al. (Platform-mediated networks) [30].

Characteristics of Platforms and the connection to LLCM

Before we start with analysing the results of our literature review further, we see it useful to explain 1) essential characteristics of Multi-Sided Platforms and 2) why this type of platform definition suits our project LLCM best.

Although the three schools of thought of platforms can be distinguished as described, they share two features: All three platforms share a **base**, which can be connected to **verifying elements** (services, products, actors) to result in a more valuable service or product and all kinds of these platforms are reusable and reduce production or operating costs compared to having different products. This variability and cost-efficiency help platforms to become very successful business models in the first place.

However, Multi-Sided Platforms have to face **network externalities** and along with them, a so-called “**chicken-and-**

egg-problem”. There may be platforms that act as merchants and only sell products from one side to another as this was the case with Amazon.com before they opened their platform for other sellers [31]. Amazon.com uses a portal to sell products and is dependent on their customer base. But the customers were neither dependent on Amazon as the retailer nor on the other customers. But in two-sided markets network externalities are always present. Network externalities are a phenomenon that describes a rise in the value of a two-sided market for customers in case the customer base grows. There can be network externalities which develop within one side of actors on a platform (i.e. the more people use a voice-over-IP telephone service the more valuable it is) or cross-externalities which develop between two or more groups of actors on a platform (i.e. the more people have a video game console the more game developers have an interest in developing games for it; the more games available for the console, the more people will buy a PlayStation) [17]. Network externalities can be positive or negative. In the case of positive network effects, the value of the platform increases. Negative network effects reduce the value of a platform as when the usage of a mobile network grows to the point where no more calls can be processed and the network collapses.

For the success of a platform a growing customer base is crucial and so network effects are desirable. But the effect of network externalities is not stable over the time of platform development. If a video game console, such as PlayStation, has one customer and one game to offer, the value of the PlayStation is low for both the customer and the game developer. If a critical mass of customers can be attracted, it becomes easier to convince game developers to develop more games. And if a critical game variety has been reached, it is easier to convince potential customers to buy a PlayStation. However, the question is, which side can be attracted easier or must be attracted first to reach a critical mass in the shortest time possible? This dilemma is called the chicken-and-egg problem, which goes hand in hand with network externalities and therefore with Multi-Sided Platforms.

Now, why do Multi-Sided Platforms suit best as a platform understanding within the LLCM project? The LLCM project aims at a reference model of a connected mobility platform that offers both APIs for developers and comprehensive services to end customers, such as a traffic routing in case of emergency. Developers will be able to add their services or to create new services which operate on the platform. Customers can either use services as real customers, or contribute to services by providing data or information (Crowdsourcing). As can be seen from this description: There will be at least two different kinds of actors on the platform who can use the platform independent from each other or interact by working on a service together. In addition, cross-externalities will apply. Either with crowdsourcing services, where end customers can take the role of a pure customer or the role of an information provider, or with conventional mobility services where developers and partners demand a critical mass

of customers to provide their service on the LLCM platform. In consequence, the LLCM platform would encompass all characteristics that Multi-Sided Platforms own, so we go with the following definition:

We define platform as a Multi-Sided Market with a software-based core that enables two or more actors to interact with each other and that underlies the influence of network externalities. The software-based core of a Multi-Sided Market is extensible, reusable and provides stable interfaces (architecture) [23]. On the platform services for end users or for developers may be developed, added or changed and products and services may be sold.

Definition of Business Model

Finally, the term business model should be clarified, too. Similarly, to platform, the term business model comprises different meanings ranging from short descriptions of how a business generates its revenue to detailed constructs, which have many different aspects in focus influencing the relationships of value exchanges within a business. For our further research,

”a business model is defined as: a description of how a company or a set of companies intends to create and capture value with a product or service. A business model defines the architecture of the product or service, the roles and relations of the company, its customers, partners and suppliers, and the physical, virtual and financial flows between them” [32].

In this sense, “business model” will be understood as a framework, conceptualizing all relevant domains of a business and is thus similar to the business model canvas [33]. As defining business model was not in the center of this research we refrain from a deeper analysis here. However, for a more detailed overview of business models we recommend the article by Krcmar et al. [34].

After having created a fundamental understanding of platforms, in this case Multi-Sided Platforms and business models, we will continue to present our findings concerning the state of the art in research of platform business models.

Findings

This chapter is subdivided in two parts. In the first part we will describe the different research levels on platform business models we could identify within the relevant articles. In the second part we name and describe different business model components we could extract from the literature. In case, business model components already appear in the first part, we will highlight them with bold typing.

Research Levels on Platform Business Models

Within the relevant article set we could rarely find the term “Platform Business Model”. This can be explained by the fact, that business model and platform are no clear defined terms, but also by different levels of research. We could identify four different levels of research which differ in the granularity of the business model parts in focus and which we depicted in figure 1.

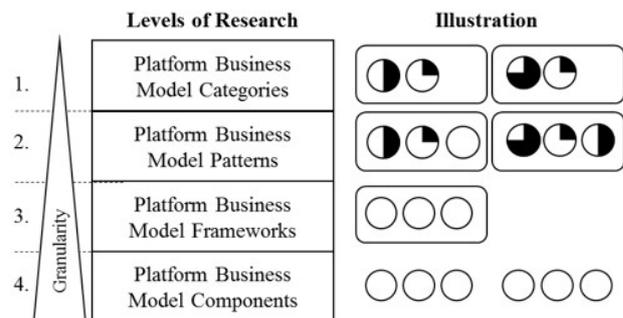


Figure 1. Research Levels on Business Model Components.

The smallest element of a business model analysis here is the Business Model Component (see level 4). It consists of a specific part of a business model, such as the revenue model or the governance structure. There may be many business model components each of which can be examined in detail separately. But not all of them are relevant in every research field. Especially for giving practitioners a guideline, which components are relevant in certain industries, many researchers try to define frameworks which represent the third research level in figure 1. If Business Model Frameworks are used with real companies, they provide a basis for comparison and if a comparison is conducted Business Model Patterns can be derived. Consequently, a Business Model Patterns is a bundle of few business model components with certain instances. In figure 1 the generic business model components are depicted as empty circles, and specific instances of components are depicted by Harvey Balls. Business Model Categories are very close to Business Model Patterns, but they comprise less component.

Platform Business Model Categories

On the level of Platform Business Model Categories we found an article by Boudreau and Lakhani [35]. They present three categories which focus on the **relations** between the platform owner, the customer and external innovators.

1. The Integrator Platform is located strategically between external innovators and the customers. This gives the platform owner a large amount of **control** over the goods and services that are traded on his platform and the way business is conducted. Apple’s App Store is one example of such an integrator platform. Apple is known for its strict specifications and requirements for external programmers in order to ensure that only products of high standard are offered on its platform.

2. Less control rights are granted in the Product Platform, where the platform owner offers a foundation for so-called external innovators to build upon. These innovators sell their products directly to the customers, which makes it difficult for the platform owner to guarantee certain standards in the products. Boudreau and Lakhani refer to Gore-Tex, the producer of a waterproof fabric, as such a platform owner: “Gore provides the core technology (and rules for its use), and the licensees innovate on that platform and sell their applications to customers.”

3. The third business model category is a Two-Sided Platform. Both, customers and external innovators are linked to the platform even though they conduct their business directly with each other. The platform owner thus only has limited control over the products that external innovators are offering, and can impose this control, for example by issuing certain rules and standards. The online auction exchange Ebay is one example of such a platform business model. As this categorization has been referred to by six of our 27 relevant articles, it seems useful for structuring research in the first place and reduce the scope of the article accordingly. With regards to the LLCM project, our platform will rather become a Two-Sided Platform since the project requires an open platform which will not interfere in the relation between customer and external innovators (i.e. developers). For business model components, we here identified the **relations between platform actors and control**.

Platform Business Model Patterns

On the level of *Platform Business Model Patterns*, we only found one article in our literature review, but, we know about other similar work, i.e. by Gassmann et al. [36]. Chen [37] researched the design of web 2.0 business models and proposes an overview of possible business model patterns for practitioners. His work is not explicitly about platforms, however we decided to include his work here as our understanding of platforms affects many web 2.0 business models. In his study, he refers to nine different types of web business models (Brokerage, Advertising Model, Infomediary Model, Merchant Model, Manufacturer Model, Affiliate Model, Community model, Subscription Model, Utility Model). These patterns are mainly described by aspects about the actors’ relationships and the revenue generation. Business Model Patterns are very useful in practice when new ideas for commercial strategies are needed. In terms of this research they can be used for a more detailed clustering of platform business models and therefore help to compare different platforms with each other. Again, the **actors’ relationships** occur as business model component, and additionally we identify **revenue generation**.

Platform Business Model Frameworks

Next to platform business model patterns we found several business model frameworks that have been applied to a platform context: The Business Model Circle, four strategic decisions for Multi-Sided Platforms, the Integrated Methodologi-

cal Framework and the Business Model Canvas.

Three papers reference the “Business Model Circle”, a framework initially developed by Braet and Ballon [38]. The Business Model Circle consists of four parameters: The first parameter, **organization design**, describes a business’ value network and thus includes the business actors, their roles and relationships. The **technology design** specifies the technological base of service provision. Service design describes how the customer **value is created** and finance design is the parameter concerned with **revenue and cost accounts**. These parameters can be used to analyze how value is created and how control is exercised within a business, and therefore we highlight them as business model components.

Buchinger et al. [39] stress that the Business Model Circle does not capture the specifics, that a platform business model comprises and thus needs to be modified in order to be applicable to multi-sided platforms. So, they integrate the Business Model Circle with Hagiu’s [40] framework of several strategic decisions that multi-sided platforms need to consider. A comparison of both models shows that except for the **governance rules**, all aspects of Hagiu have already been included under different naming. Buchinger et al. state, that “Governance rules apply for i) regulating the access to the platform; and ii) regulating the interactions on the platform and regulates the terms and conditions.” [39] The aim of governance is to keep the platform activities on a legally correct and value creating level. To foster this aim, governance needs to impose control mechanisms and penalties in case of rule violations. The aspects of both models are depicted in table ?? . The same model with different naming has also been used by Evens et al. for analyzing business models for mobile network operators entering sports broadcasting market [27]. As a further business model component, we derive **governance** from this part of the literature.

Poel and Tee [41] study business models for television platforms and focus especially on the interplay of policy instruments and platform business models. The authors propose the Integrated Methodological Framework by Ballon et al. [42] for analysis, that consists of four distinct domains: value proposition, value network, financial architecture and financial model. By having a closer look, we could relate the functional architecture to the technology design, the value network to the organizational design (complemented by aspects of the **customer relationship**), the financial architecture to the finance design and value proposition to service design of the Business Model Circle according to Braet and Ballon [38].

Other researchers used the Business Model Canvas (Osterwalder et al.) [33] as a tool for platform analysis, such as Muzellec et al. [25], Kohler [43] and Duval and Brasse [44]. Therefore, and because Osterwalder et al. find the Canvas suitable to analyse platform business models, too, we see the importance to consider the nine components here as relevant business components as well. The nine business model components according to Osterwalder are shown in figure 2.

The business model circle (Braet & Ballon, 2007)	Strategic decisions of Multisided Platforms (Hagiu, 2014)
Organization design Mobilizing resources and capabilities	Number of sides to bring on board (added: quantity and quality of partners)
Technology Design Products & Service Creation	-
Service Design Creating customer value	Platform Design possibilities Functionalities and features
Finance Design Creating shareholder value	Pricing Structures
-	Governance Rules Rules and regulations

Table 1. Comparison and merger of business model frameworks, according to [39]. Own illustration.

Business Model Components

On the fourth level of research perspective we found some articles which examined single business components. Compared to the business model components identified so far, which overlap in their scope, the following business model components are specific to platform business models and relevant in the LLCM project.

Becker et al. [45] developed a taxonomy of platform models for mobile service delivery. The two parameters for their taxonomy are openness of the platform system and the medium of interaction (portal vs. device). **Openness** refers to the degree of freedom a developer is being granted when using a platform. An open platform, on the one hand, imposes none or few restrictions according to the use of software, the users’ access requirements and other. A closed platform imposes, on the other hand, many restrictions. Each of these modes have advantages and disadvantages: an open platform can attract more users, probably even in a shorter time frame and therefore, especially in terms of innovations, has a greater pool of potential contributors. However, as the access is usually not limited, everyone is able to grasp information about innovations in the moment they are invented, so there is no real first mover advantage. De Pablos-Heredero et al. [30] also address openness and add that “open modes of innovation and development [33] present some managerial challenges for sponsoring firms which must soften their intellectual property policy in order to accommodate external parties with different business objectives and incentives.” For a closed platform

these arguments hold in reverse.

Nine papers claim business innovation due to changes in the business environment as an important success factor. Velu [46] studied business model innovation of electronic trading platforms and finds that “new firms adopting both incremental and radical business model innovations are more likely to survive longer than those adopting moderate business model innovations”. Duval and Brasse [44] argue that the platform under research should convert their business model from a free to freemium to premium business model over time. Although this is very specific research on a public collaborative workspace platform it shows that platforms might not stick with only one business model over time. Muzellec et al. [25] do agree upon this as they state that “a sustainable business model is a dynamic ecosystem which constantly changes as the business evolves and the relative positions of the multiple participants and the flow of resources shifts over time.” As a consequence, the **platform lifecycle** is another relevant parameter to the platform business model.

Customer Relationship (1) is an aspect we already mentioned and it describes the means of interaction which is offered in-between customers and developers. Here we discuss not only technical issues but also thoughts on long term business relationships. In addition to the struggle with the “chicken-and-egg” problem, there are other characteristics a platform must keep in mind to attract customers. Trust is mentioned by Enders et al. [47] as a driving factor. Especially in the information age we live in platforms always depend on personal data from their customers. If trust concerning the use of this data cannot be established and ensured over a long time, then customer attraction will not arise. From a different perspective, platform developers are also “customers”. Bergvall-Kareborn et al. investigate the working conditions for developers in platform environments [48]. They argue that they must cope with diversity of technology and according to that more and more knowledge is requested from the individual. In addition, the lack of structure in some situations (no precise contracts for a development job) puts pressure on the personal revenue situation, in case one developer finishes an app before the other. To conclude, a platform must not focus

Key Partners	Key Activities	Value Propositions	Customer Relationships	Customer Segments
	Key Resources		Channels	
Cost Structure		Revenue Streams		

Figure 2. Business Model Canvas according to Osterwalder et al. Own illustration.

on the end customer but also on its partners and find suitable ways to please both [49].

From Lai's [50] and Montgomerie et al.'s [51] perspective the mass of users on a platform also hold a potential for other business activities: Lai argues that in the start-up phase of a platform the critical mass can easier be attracted with only one specific value proposition and that probably only has one certain revenue model. However, when the critical mass is reached, a platform business should rethink the utilization of the user base, for example, selling the access to the users to other business for advertising. Montgomerie et al. go a step further by analyzing the Apple business model which they call "owning the consumer". This means that the consumer is loyal to Apple concerning all services and products along the whole supply chain, which gives Apple huge power over him and their suppliers and allows them "to translate its dedicated consumer base into meaningful revenue streams". Reconsidering the elements Channels and Customer Segmentation we conclude that both they are tools which have influence on or define the customer relationship in the broader sense. Customer segmentation is a necessary step to understand the customer and to select the appropriate means for advertising the service offered. The choice of channel to reach the customer also depends on the customer segmentation. As they are so tightly connected, we again drop the two components and incorporate them in the **Customer Relationship in the broader sense (2)**.

The literature finds three different perspectives on the interactions between platform actors which are tightly connected to the openness of a platform. First, there is value co-creation which means cooperative work on a platform for a common goal. Crowdsourcing is one example for value co-creation: "Crowd-based businesses enable organizations to harness the collective energy and creativity of a large number of contributors" [43]. Second, there is co-opetition. This term encompasses the phenomenon that competitive platforms open up their platforms for each other to reach a higher value creation together. It describes the coexistence of cooperation and competition, where "competitors on a project/product can be partners for the modular development of a different project/ product" [52], [31]. Third, there is platform **competition**. Visnjic and Cennamo [53] argue that competition arises when a platform incorporates functions of another platform in an adjacent market. As this is the easiest form of business innovation, this kind of competition will arise sooner or later in every platform market. The danger of such a business innovation lies in the possibility, that the other platform notices a business innovation for itself in the first platform's market, too, which will evoke a responding envelopment.

11 Platform Business Model Components

In the beginning of chapter Findings, we remarked that business model components will be typed bold when they are identified. Until here we identified the 22 components on the left side of figure 3.

By reading through this list, redundancies become clear, so we consolidated this list to 11 components following the arrows in figure 3. 1. Value Proposition and Service Design were the first two components we drew together in Value Creation, as they both describe the external view on the offering of the platform: What value does the service propose to the consumer? Service design [38] is a medal with two sides: One side is directed to the platform that creates the service or with other words, creates the value. This side directs at all the processes and "raw materials", i.e. software programs, that are needed for service creation. When considering this side of the medal the platform provider needs to focus on efficiency, quality and cost issues. The other side is directed to the customer who consumes the service and only sees it working, but sees nothing from the "back office". So in essence, this side represents the value proposition which is very important due to the increasing need of differentiation and instead of service design we include Value Creation and **Value Proposition** as components of a platform business model in our results. 2. Value Creation and Key Activities are two terms for the same component: They depict the internal view on the service design, i.e. the processes', actors' and resources' interplay for key activities. In the following we go with **Value Creation**. 3. Referring to the cash flows, we bundle Revenue Generation, Revenue Streams, Cost Accounts and Cost Structure to **Capital Structure**. This way, we stay flexible in also adding financing activities to this component, which we think of an important point, even though it could not be found in the relevant literature. 4. **Governance and Control** belong together, as Governance is a more general term to describe regulations concerning decision rights, access rights, permitted activities and penalties. The aim of governance is keeping processes under control in favor of business targets. 5. Under **Organizational Design** we summarize all components that describe the business value network and thus includes the business actors, their roles and relationships, except for the customer. 6. **Customer Relationship (1)** describes what interactions a customer relationship consists of and what kind of relationship is established. 7. **Customer Relationship in a broader sense (2)** defines how the customers are targeted and which communication and incentive tools are employed to reach the customers. Therefore, it integrates Channels and Customer Segments. 8. The remaining components stay separated and named as they are.

We applied these eleven components to a concept matrix, which is depicted in table ???. The matrix shows the relevant articles and their relation to each concept.

Discussion

Through our literature review we have identified 11 platform business model components summarized in Table 2. Most of them come from Business Model Frameworks, which is not surprising as they are generally applicable on every kind of business. The other components seem to address additional problem areas in establishing and maintaining a platform.

	Value creation	Value Proposition	Revenue Model	Governance / Control	Organizational Design	Technology Design	Openness	Customer Relationship 1	Lifecycle	Customer Relationship 2	Competition
Attour (2014) [54]		X		X			X		X		X
Baghdadi (2016) [55]					X	X					X
Becker et al. (2012) [45]						X	X				
Bergvall-Kareborn et al. (2014)[48]	X						X			X	X
Borello et al. (2015) [56]											X
Boudreau & Lahkani (2009) [35]	X			X			X		X	X	X
Buchinger et al. (2015) [39]		X	X	X	X	X					
Cennamo & Santaló (2015) [57]		X									
Chen, T. F. (2009) [37]	X										
de Pablos-Herederó et al. (2012) [30]				X			X		X		
Duval & Brasse (2014) [44]	X	X	X		X		X	X	X		
Enders et al. (2008) [47]		X	X					X		X	
Evens et al. (2011) [27]		X	X		X	X					
Ghezzi (2010) [52]		X	X				X				X
Hagiu & Wright (2015) [17]				X							X
Haile & Altmann (2014) [58]	X										
Henten & Windekilde (2015) [28]			X								X
Hsieh & Hsieh (2012) [49]			X							X	
Kohler (2015) [43]	X	X					X		X	X	X
Lai (2013) [50]									X	X	
Montgomerie & Roscoe (2013) [51]				X						X	
Muzellec et al. (2015) [25]	X	X	X		X			X	X	X	
Poel & Tee (2007) [41]		X							X		
Ritala et al. (2014) [31]	X										X
Rochet & Tirole (2003) [29]			X	X							
Velu (2015) [46]	X	X							X		
Visnjic & Cennamo (2013) [53]	X		X				X		X		X

Table 2. Concept matrix according to Webster and Watson [22] regarding Platform Business Model Components.

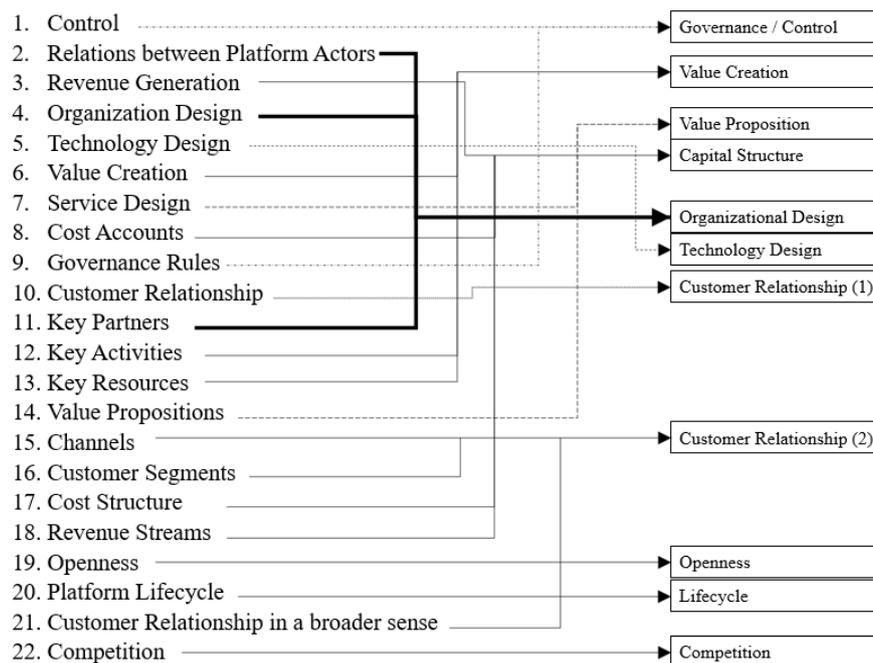


Figure 3. Consolidation of Business Model Components. Own illustration.

Most of the components mentioned are derived from generic or mobile services specific business model frameworks. All platforms have the ambition to at least cover their costs and consequently all will need a business model. Components of generic business models are applicable to all business models by definition. The components from the business model circle by Braet and Ballon [38] can be transferred to the platform business as well, as many platforms have risen from business innovations in the mobile services sector. Therefore, they all are qualified for being included in this concept for a platform business model. Comparing our results to Schreieck et al. we find that we have several components in common with their concept: revenue, openness, technical design, control and competitive strategy. Our understanding of openness, technical design and competitive strategy overlap. Differences occur due to differences in the level of research, but there are no contradictions. New with our concept are the lifecycle and the competition component. The business model lifecycle by Muzellec et al., depicted in figure 4, is a construct that shows a typical, rather ideal development of revenue of a successful two-sided market over time from an embryonic to a maturity stage.

We see an essential component in the lifecycle, because the creation of a business model for LLCM requires sustainability and configurability. To be sustainable a business needs to adjust to changes in its environment and that means 1. to understand the current and future environment, 2. to analyze the business' performance in both environments from the current point of view and 3. to derive necessary actions on the business model in order to address changes in the business

environment properly. The changes are addressed properly when it is clear for the business where they stand and where they want to go next, which can be supported by ranking the business in the business lifecycle. Muzellec et al. point out that "it does not seem sufficient, however, to evaluate the maturity of a business model solely on its revenues. What is more interesting is to analyse the evolving nature of each component of its business model at the different stages." We agree upon this and see it as necessary to examine business lifecycles according to the different business model components. The aim here is to derive guidelines for actions in different phases of a business model lifecycle and suitable for different environmental changes. Moreover, trends, which will turn into certain changes in the future should also be taken into account. Opening up a platform may be an option for many incumbent companies who want to innovate their business model. But entering a new business means that new rules apply and new competition risks arise, especially concerning competing platforms. Cases like Facebook vs. MySpace and iOS vs. Android show different outcomes of platform developments underlying network effects. Whereas Facebook outperformed MySpace by far, Android caught up despite a later market entry, but Apple and Android manage to cater the market simultaneously. Consequently, the consideration of the competitive environment is very important for business model design.

Limitations and Further Research

This research has some limitations to mention. We considered only three databases for our literature review which reduces

the number of resulting relevant articles. So other databases can be searched, particularly databases from other disciplines which also deal with platforms. Furthermore, other keywords can be included in the search, such as Multi-Sided Market, Two-Sided Market, Multi-Sided Network and Digital Platforms, for example. The components we identified represent a clustering of different business model aspects. It is object to further research to check upon the boundaries of each component and the interdependencies of the components. For example, choosing an open business model imposes restrictions directly on the amount and strictness of governance rules of the platform. Interdependencies, as well as further research on the lifecycle, also will shed light upon the sequence in which the components should be defined and changed during the evolution of a platform. The authors suggest to fill the components exemplary with information about one specific platform to create a proof of concept. Digital Platforms are currently growing everywhere and now is a point in time where historical data about the development of platforms is available. In context with the lifecycle component, which spans all of the other components, too, it would be very interesting to collect data over many different platforms in a comprehensive database and analyze this data according to patterns of platform lifecycles. Afterwards, it would be interesting for practitioners to identify measures that go with specific lifecycle phases and to derive guidelines for the management of platforms. In the same manner, competition between platforms should be investigated. We have already shown that platforms which seem to address the same market can either exist simultaneously or one outperforms the other. Researching the reasons behind these phenomena would provide insights about solid business models design. Together with the open business models, sharing economy and crowd-sourcing businesses the borders of companies, especially platforms, become blurred. Respectively, it would be interesting to analyze influencing variables on competition: How far can a platform go with value co-creation and co-opetition and when does the competition overwhelm the surplus? Finally, the focus of the next step should shift closer to the LLCM project and connected mobility aspects. The literature reviewed included only one article concerning urban mobility, so further research in this field is necessary.

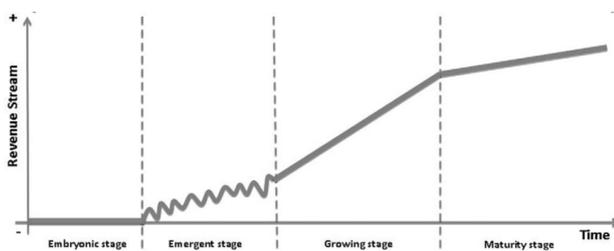


Figure 4. Platform Business Lifecycle according to Muzellec et al. [25].

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] Google just open sourced tensorflow. <http://www.wired.com/2015/11/google-open-sources-its-artificial-intelligence-engine/>. Accessed: 2016-06-30.
- [2] Inside openai, elon musk's wild plan to set artificial intelligence free. <https://code.facebook.com/posts/1687861518126048/facebook-to-open-source-ai-hardware-design/>. Accessed: 2016-06-30.
- [3] Facebook to open-source ai hardware design. <http://www.wired.com/2016/04/openai-elon-musk-sam-altman-plan-to-set-artificial-intelligence-free/>. Accessed: 2016-06-30.
- [4] Platform economy: Technology-driven business model innovation from the outside in. https://www.accenture.com/t20160125T111719_w_/us-en/_acnmedia/Accenture/Omobono/TechnologyVision/pdf/Platform-Economy-Technology-Vision-2016.pdf#zoom=50. Accessed: 2016-06-30.
- [5] The emporium strikes back. <http://www.economist.com/news/business/21699103-platforms-are-futurebut-not-everyone-emporium-strikes-back>. Accessed: 2016-06-30.
- [6] Are platform businesses eating the world? <http://www.forbes.com/sites/brookmanville/2016/02/14/are-platform-businesses-eating-the-world/#6441b44c621c>. Accessed: 2016-06-30.
- [7] Grünbuch - digitale plattformen. <http://www.bmwi.de/BMWi/Redaktion/PDF/G/gruenbuch-digitale-plattformen,property=pdf,bereich=bmwi2012,sprache=de,rwb=true.pdf>. Accessed: 2016-06-30.
- [8] Maximilian Schreieck, Manuel Wiesche, and Helmut Krmar. Design and governance of platform ecosystems - key concepts and issues for future research. In *Forthcoming: Twenty-Fourth European Conference on Information Systems (ECIS), Istanbul, Turkey, 2016*.
- [9] Yannis Bakos. The emerging role of electronic marketplaces on the internet. *Communications of the ACM*, 41(8):35–42, 1998.

- [10] David G Messerschmitt, Clemens Szyperski, et al. Software ecosystem: Understanding an indispensable technology and industry. *MIT Press Books*, 1, 2005.
- [11] Lisen Selander, Ola Henfridsson, and Fredrik Svahn. Capability search and redeem across digital ecosystems. *Journal of Information Technology*, 28(3):183–197, 2013.
- [12] Carliss Y Baldwin and C Jason Woodard. The architecture of platforms: a unified view. *Harvard Business School Finance Working Paper*, (09-034), 2008.
- [13] Barney Tan, Shan L Pan, Xianghua Lu, and Lihua Huang. The role of its capabilities in the development of multi-sided platforms: The digital ecosystem strategy of alibaba.com. *Journal of the Association for Information Systems*, 16(4):248, 2015.
- [14] Mei Lin, Shaojin Li, and Andrew B Whinston. Innovation and price competition in a two-sided market. *Journal of Management Information Systems*, 28(2):171–202, 2011.
- [15] Tobias Goldbach and Viktoria Kemper. Should i stay or should i go? the effects of control mechanisms on app developers' intention to stick with a platform. 2014.
- [16] Annabelle Gawer and Michael A Cusumano. How companies become platform leaders. *MIT Sloan management review*, 49(2):28, 2008.
- [17] Andrei Hagiu and Julian Wright. Multi-sided platforms. *International Journal of Industrial Organization*, 43:162–174, 2015.
- [18] Amrit Tiwana, Benn Konsynski, and Ashley A Bush. Research commentary-platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research*, 21(4):675–687, 2010.
- [19] Thomas Eisenmann, Geoffrey Parker, and Marshall W Van Alstyne. Strategies for two-sided markets. *Harvard business review*, 84(10):92, 2006.
- [20] Thomas Eisenmann, Geoffrey Parker, and Marshall W Van Alstyne. Platform envelopment. *Strategic Management Journal*, 32(12):1270–1285, 2011.
- [21] Jan Vom Brocke, Alexander Simons, Bjoern Niehaves, Kai Riemer, Ralf Plattfaut, Anne Cleven, et al. Reconstructing the giant: On the importance of rigour in documenting the literature search process. In *ECIS*, volume 9, pages 2206–2217, 2009.
- [22] Jane Webster and Richard T Watson. Analyzing the past to prepare for the future: Writing a. *MIS quarterly*, 26(2):13–23, 2002.
- [23] Stefan Hefe. Living lab connected mobility - analyse und beschreibung von gestaltungsoptionen für den aufbau eines nachhaltigen mobilitätsökosystems. Master's thesis, Technische Universität München, 2016.
- [24] Annabelle Gawer. Platform dynamics and strategies: from products to services. *Platforms, markets and innovation*, 45:57, 2009.
- [25] Laurent Muzellec, Sébastien Ronteau, and Mary Lambkin. Two-sided internet platforms: A business model lifecycle perspective. *Industrial Marketing Management*, 45:139–150, 2015.
- [26] Yannis Bakos and Evangelos Katsamakakos. Design and ownership of two-sided networks: Implications for internet platforms. *Journal of Management Information Systems*, 25(2):171–202, 2008.
- [27] Tom Evens, Katrien Lefever, Peggy Valcke, Dimitri Schuurman, and Lieven De Marez. Access to premium content on mobile television platforms: The case of mobile sports. *Telematics and Informatics*, 28(1):32–39, 2011.
- [28] Anders Hansen Henten and Iwona Maria Windekilde. Transaction costs and the sharing economy. *info*, 18(1):1–15, 2016.
- [29] Jean-Charles Rochet and Jean Tirole. Platform competition in two-sided markets. *Journal of the European Economic Association*, 1(4):990–1029, 2003.
- [30] Carmen de Pablos-Heredero, David López-Berzosa, and Gloria Sánchez-Gonzalez. Open business models and platform mediated networks: an application in the mobile industry. *Procedia Technology*, 5:122–132, 2012.
- [31] Paavo Ritala, Arash Golnam, and Alain Wegmann. Coopetition-based business models: The case of amazon.com. *Industrial Marketing Management*, 43(2):236–249, 2014.
- [32] Pieter Ballon, S Kern, M Poel, R Tee, and S De Munck. Best practices in business modelling for ict services. *TNO-ICT Report*, (33561), 2005.
- [33] Alexander Osterwalder and Yves Pigneur. *Business model generation: a handbook for visionaries, game changers, and challengers*. John Wiley & Sons, 2013.
- [34] Helmut Krcmar, Markus Böhm, Sascha Friesike, and Thomas Schildhauer. Innovation, society and business: Internet-based business models and their implications. *Prepared for Berlin Symposium on Internet and Society, Oct. 25–27, 2011*, 1:1–33, 2011.
- [35] Kevin Boudreau and Karim Lakhani. How to manage outside innovation. *MIT Sloan management review*, 50(4):69, 2009.
- [36] Oliver Gassmann, Karolin Frankenberger, and Michaela Csik. The st. gallen business model navigator. Master's thesis, Working Paper, University of St. Gallen, 2014.
- [37] Te Fu Chen. Building a platform of business model 2.0 to creating real business value with web 2.0 for web information services industry. *International Journal of Electronic Business Management*, (3):168, 2009.
- [38] Olivier Braet and Pieter Ballon. Business model scenarios for remote management. In *Project E-Society: Building Bricks*, pages 252–265. Springer, 2006.

- [39] Uschi Buchinger, Heritiana R Ranaivoson, and Pieter Ballon. Mobile wallets' business models: Refining strategic partnerships. *Organizacija*, 48(2):88–98, 2015.
- [40] Andrei Hagiu. Strategic decisions for multisided platforms. *MIT Sloan Management Review*, 55(2):71, 2014.
- [41] Martijn Poel and Richard Tee. Business model analysis as a tool for policy analysis digital tv platforms in the netherlands and france. Technical report, TNO Information and Communication Technology www.tno.nl, 2006.
- [42] P. Ballon, S. Limonard, R. Tee, and U. Wehn de Montalvo. Integrated methodological framework. investigating business models for broadband services: the case of idtv platforms. *Freeband B@Home deliverable D2.8*, 2006.
- [43] Thomas Kohler. Crowdsourcing-based business models. *California Management Review*, 57(4):63–84, 2015.
- [44] Amaury Duval and Valérie Brasse. How to ensure the economic viability of an open data platform. 2014.
- [45] Alexander Becker, A Mladenowa, Natalia Kryvinska, and Christine Strauss. Evolving taxonomy of business models for mobile service delivery platform. *Procedia Computer Science*, 10:650–657, 2012.
- [46] Chander Velu. Business model innovation and third-party alliance on the survival of new firms. *Technovation*, 35:1–11, 2015.
- [47] Albrecht Enders, Harald Hungenberg, Hans-Peter Denker, and Sebastian Mauch. The long tail of social networking.: Revenue models of social networking sites. *European Management Journal*, 26(3):199–211, 2008.
- [48] Birgitta Bergvall-Kåreborn and Debra Howcroft. Persistent problems and practices in information systems development: a study of mobile applications development and distribution. *Information Systems Journal*, 24(5):425–444, 2014.
- [49] Jung-Kuei Hsieh and Yi-Ching Hsieh. Appealing to internet-based freelance developers in smartphone application marketplaces. *International Journal of Information Management*, 33(2):308–317, 2013.
- [50] Puqing Lai. Utilizing the access value of customers. *Business Horizons*, 57(1):61–71, 2014.
- [51] Johnna Montgomerie and Samuel Roscoe. Owning the consumer—getting to the core of the apple business model. In *Accounting Forum*, volume 37, pages 290–299. Elsevier, 2013.
- [52] Antonio Ghezzi. Emerging business models and strategies for mobile middleware technology providers: A reference framework. In *ECIS*, pages 1915–1926, 2009.
- [53] Ivanka Visnjic and Carmelo Cennamo. Towards an integrated perspective on platform market competition. In *Academy of Management Proceedings*, volume 2013, page 16837. Academy of Management, 2013.
- [54] Amel Attour. Quel leader et business model ouvert pour les écosystèmes-plateformes de type nfc? *Management & Avenir*, (7):33–53, 2014.
- [55] Youcef Baghdadi. A framework for social commerce design. *Information Systems*, 60:95–113, 2016.
- [56] GIULIANA BORELLO, Veronica De Crescenzo, and FLAVIO PICHLER. The funding gap and the role of financial return crowdfunding: Some evidence from european platforms. *Journal of Internet Banking and Commerce*, 20(1), 2015.
- [57] Carmelo Cennamo and Juan Santalo. How to avoid platform traps. *MIT Sloan Management Review*, 57(1):12, 2015.
- [58] Netsanet Haile and Jörn Altmann. Value creation in software service platforms. *Future Generation Computer Systems*, 55:495–509, 2016.

How *Accountability* is Understood and Realized in Implementations – A Systematic Mapping Study

Kristian Beckers, Amjad Ibrahim, Prachi Kumari and Alexander Pretschner

Department of Informatics, Technical University of Munich, Munich
{beckersk; ibrahim; kumari; pretschn}@in.tum.de

Abstract

With the growing use of cyber-physical systems in complex socio-technical setups, we need mechanisms that enable us to hold specific entities accountable for safety and security incidents. For this, a clear understanding of accountability in the socio-technical context is needed in general for proposing a novel and innovative solution for TUM LLCM. Although there exist models that try to capture and formalize accountability concepts, many of these lack practical implementations. Hence, we know little about how accountability mechanisms should work in practice and how specific entities could be held responsible for incidents. As a step towards the practical implementation of providing accountability, this mapping study investigates the existing implementations of accountability concepts with the goal to (1) find definitions and understand what the term *accountability* means to researchers in different contexts, and (2) identify the general trend of practical research.

Keywords

accountability; tools; literature review; survey; systematic mapping study

Introduction

Traditionally, IT practitioners have aimed to avoid problems using preventive measures. In complex systems, however, it is often hard to enumerate and plan for possible contingencies. Besides, preventive measures require in general many additional resources and are thus expensive to implement (for examples from the security domain see [1]). This has shifted the focus of research towards alternate ideas like detective security [2] or root cause analysis [3]. Detective security, for example, is inspired by how law enforcement works in the real world [4]: Speeding violations are not prevented by technical means (e.g., by limiting the maximum speed of the car). If, however, someone exceeds the posted speed limit there is a good chance that they will be caught, held accountable and punished according to the law.

In our vision, accountability is a property of a Socio-Technical Systems (STS) that provides the ability to answer questions regarding why specific unwanted events happened. It, however, does not focus on the technical means, but wants to find out which person or organization is responsible for a given event or problem. Although unwanted events vary according to the desired properties of the system, a general understanding of them in specific contexts can be easily reached. E.g., brake failure: pressing the brakes does not lower the speed of the car. Accountability mechanisms in this case should be capable of answering why the brakes failed. Another example would be to answer the question how some party got access to some sensitive information. In this case the accountability mechanism should help us to trace the leak. The goal of an accountability mechanism is to answer such

questions end to end, from the unwanted event to the person liable for its occurrence. Bringing the idea of accountability to socio-technical systems (STS) is however challenging. To start with, there is no universally-accepted definition of accountability. We do not know

- (1) which events should be considered in different domains,
- (2) which systems should be monitored for violations of which properties,
- (3) which type of monitoring mechanisms should be in place,
- (4) which events should be monitored,
- (5) how the monitored events should be analyzed, or
- (6) how violations should be handled

In order to develop an initial understanding of these issues, we designed this mapping study with a focus on the term *accountability* in the context of privacy, safety, and security.

With the aforementioned understanding of accountability in mind, we look into the literature to identify the various accountability mechanisms that address violations of safety, security, and privacy requirements. Our focus is on the post-mortem analysis of unwanted events. Therefore, we do not distinguish among the unwanted events in cases of safety, security and privacy violations as once the unwanted event is known, the methods for analysis are similar.

We are aware of one survey by Xiao et al. [5] that investigated accountability in computer networks and distributed

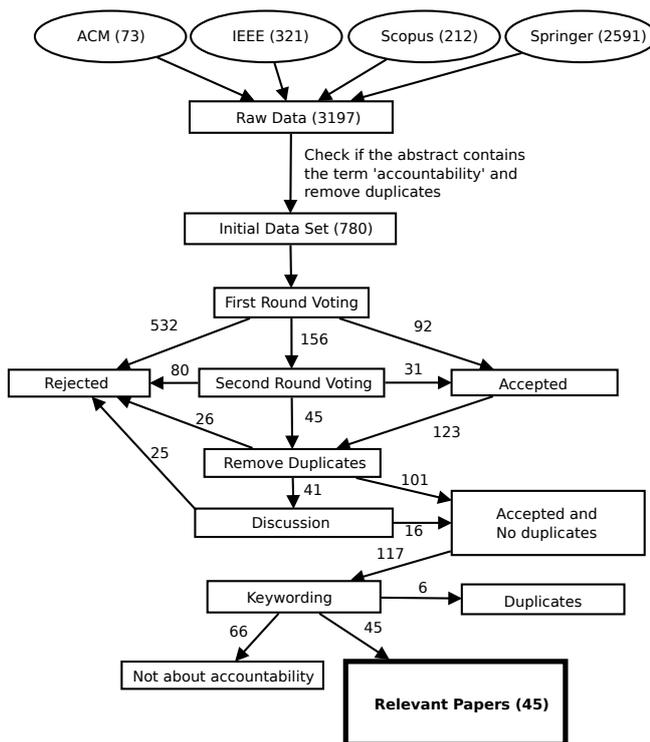


Figure 1. The “sieving” process

systems. In contrast to their work, we focus on implementations and do not restrict our search string to the narrow domain of computer networks. Furthermore Papanikolaou and Pearson [6] give a cross-discipline overview of the term “accountability”. They, however, focus on theoretical definitions and do not consider applications. Our focus is on implementations that we can understand, extend and improve for the TUM LLCM project.

Our **contribution** is a systematic mapping study that identifies which contributions were made over time, the various application domains, layers of abstraction, technologies and protocols in implementing accountability in STS. The results show that even though there exist very few tools for accountability, it is a growing area of research in different domains.

Limitations. To narrow the scope for initial results, we deliberately limited our search to the term “accountability” without synonyms. Moreover, we exclude from our study any work that does not have an implementation, even if it provides guidelines for an implementation. More details about the limitations of this study can be found in Section *Limitations*.

Methodology

We followed the five-step methodology laid out by Petersen et. al. [7]: (1) definition of research questions, (2) conduct search, (3) screening of papers, (4) keywording using abstracts, and (5) data extraction and mapping process. This section describes our instantiation of this methodology.

Definition of Research Questions

We aimed to answer the following research questions:

- RQ 1** Is research into accountability tools a growing area?
- RQ 2** Which application domains have seen most of the accountability implementations?
- RQ 3** Which underlying techniques/protocols are implemented by these tools? At which layers of abstraction do the tools exist? Is there a trend?
- RQ 4** Of what type is the research?
- RQ 5** Are prominent contributors recognizable? How are they related to each other?
- RQ 6** What have the underlying definitions of accountability in common?
- RQ 7** Can we identify common terms in the field?

Conduct Search

In accordance with our research questions, we constructed the search string

```
{accountability AND
(privacy OR safety OR security) AND
(tool OR implementation OR application)}
```

and adapted it to the idiosyncrasies of each digital library. Hence, we obtain a basic set of publications from ACM¹ (73 results), IEEE² (321), scopus³ (212) and Springer⁴ (2591), as shown in Table 1, column ‘Raw’. As a first step, we stored the search results as CSV files.

For this, IEEE and Scopus provided CSV export functionalities, comprising authors, titles, and abstracts. Springer’s export functionality did not include abstracts, hence we used a simple script to access the abstracts from the publication’s URL. To extract the information from ACM, we used the Zotero tool⁵.

During an initial screening of the results, we noticed that many abstracts of Springer publications did not feature the term “accountability”. We randomly selected 40 of those publications and confirmed that indeed none of those were relevant. Hence, we removed all such Springer publications.

After an initial screening for duplicates, we obtained the dataset shown in Table 1, column ‘Cleanup’.

Screening

We used a custom web tool to further screen the papers based on the following inclusion and exclusion criteria:

Inclusion criteria:

- ¹<http://dl.acm.org>
- ²<http://ieeexplore.ieee.org>
- ³<http://www.scopus.com>
- ⁴<http://link.springer.com>
- ⁵<https://www.zotero.org>

Table 1. An overview of our dataset

Source	Raw	Cleanup	Relevant
ACM	73	45	5
IEEE	321	201	25
Scopus	212	212	5
Springer	2591	322	10
Total	3197	780	45

- Publication reports a tool, implementation or application.

Exclusion criteria:

- Publication is not related to privacy, safety, or security.
- Publication reports only an idea, formalism or abstract framework.

In the first round, each paper was considered by two authors of this study and accepted or rejected if their decision was unanimous. In the second round, all papers with disagreements were presented to two other authors. Upon a clear majority of 3-1, the paper was accepted or rejected. After this phase, we manually identified and removed 26 more duplicates.

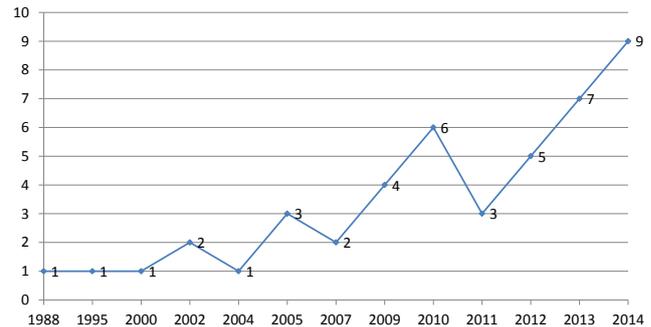
In the following, papers that had received a 2-2 draw were discussed in the presence of all authors and a final verdict was reached. In this phase 25 papers were rejected. Figure 1 summarizes the screening process.

Keywording

In order to classify and further screen the papers, we started with an intuitive set of keywords agreed upon by discussion among the authors (e.g., Security, Monitoring, or Cloud). We also added some keywords under the category of “sanity check” to further exclude irrelevant papers. These keyword-categories were: “No Implementation”, “Not about Accountability”, “PDF not available” and “I am not sure, I need help”. The last category was used if an author was not sure and wanted to discuss the paper with another author. Apart from these initial keywords/categories, every author could create new ones.

Mapping

All 117 “accepted” papers were then randomly divided among the authors. Each author screened the PDF, categorized the paper, and gave a short rationale for the categorization. If the paper did not fit into an existing category, the researcher could create a new category. The categories were shared by all researchers in a “tag-cloud”. In a nutshell, we tried to identify how accountability was understood by different researchers in the community, if they used any novel ways to implement it, and for which domain the implementation was designed. During the process we had several meetings to share the categories that emerged and discussed any unclear publications.

**Figure 2.** Number of papers over the years

Despite the previous screening steps, 66 papers had to be removed because they (1) did not describe an implementation, (2) were not about accountability, or (3) the full text was not available.

After this process, 45 relevant research papers were subject to our study: [8–52], cp. Table 1, column ‘Relevant’.

Findings

Contributions over time (RQ1)

To identify how the number of contributions developed over time, we analyzed the papers according to their year of publication. Figure 2 shows the graph of the distribution from 1988 to 2014, revealing that accountability implementations started gaining interest in 1988 beginning with the work of [17]. For the first few years until the year 2000, this area did not attract much attention with only three papers in 12 years. There are several crests and troughs starting in the year 2000, but the overall interest of the research community has been increasing. In fact, as shown in Figure 2, every trough is at a higher level than the previous one. Since 2011, there has been a consistent growth in the number of implementations. It is also notable that after the publication of the influential paper by Weitzner et al. [4] in 2008, we see relevant publications in every consecutive year.

Application Domains (RQ2)

To answer the second research question, we classified the papers according to the targeted application domains. As shown in Figure 3, accountability concepts have been mostly implemented for the Cloud domain with 8 implementations ([10, 16, 22, 30–32, 36, 38]). Other important domains are Distributed Data Sharing (7 implementations – [20, 24, 32, 38, 41, 45, 52]), Web Applications (6 – [14, 25, 26, 28, 29, 37]), and Healthcare (4 – [9, 11, 12, 52]). For other domains we found at most two implementations.

Since the implementation of accountability mechanisms is a relatively new area of research, there are many domains for which only single implementations exist. These have been grouped as *Others* in Figure 3 and includes E-Voting, Disaster, Outsourcing, Ecoupon, Wireless Networks, Smart Grid, Business Organization, E-commerce, Publishing, Lottery, Insurance, and Location-Based Services.

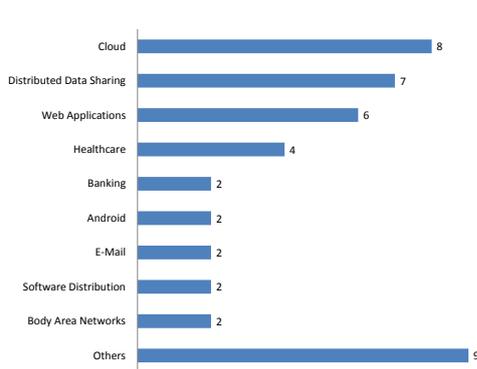


Figure 3. Papers per domain (note that the domains are not mutually exclusive)

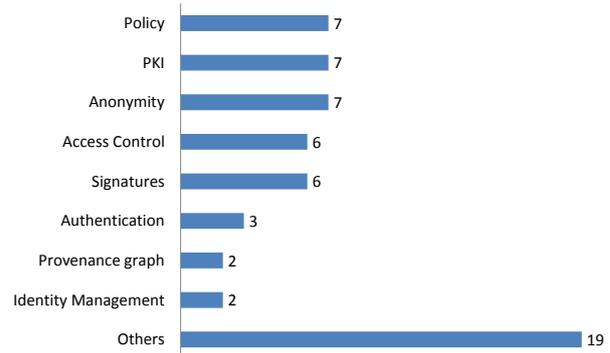


Figure 5. Mechanisms used within accountability implementations (note that some implementations use multiple mechanisms)

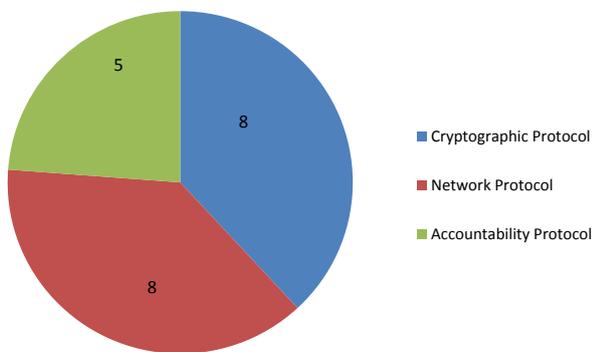


Figure 4. Protocols used in accountability implementations (note that not all papers define a protocol)

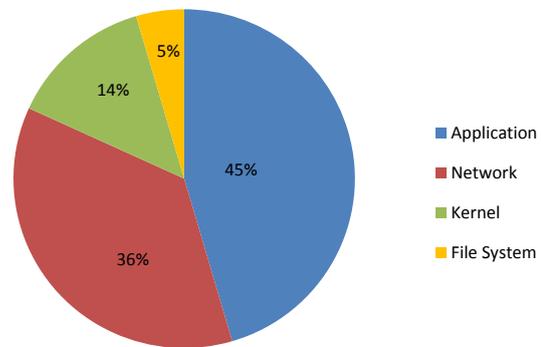


Figure 6. Layers of abstraction where accountability is implemented

Underlying Techniques & Protocols (RQ3)

As depicted in Figure 4, we found three different kinds of protocols that are leveraged by implementations to achieve accountability.

Eight papers use network ([13,21,22,25,28,37,50,51]) or cryptographic ([8,11,13,15,20,25,37,45]) protocols, while five papers make use of accountability protocols ([12,20,25,39,48]). Contrary to our expectation, data provenance protocols are not commonly used for accountability implementations.

We further investigated which mechanisms and techniques are used to implement accountability. As detailed in Figure 5, we found that most solutions are concerned with enforcement of policies (7 solutions – [10,14,23,30,35,40,49]), public key encryption schemes (7 – [8,15,19,20,23,26,44]), anonymity (7 – [8,20,25,37,43,47,48]), access control (6 – [10,23,24,39,40,49]), and signatures (6 – [11,15,19,22,41,45]). Some tools also use authentication (3 – [23,24,26]), provenance graphs (2 – [16,41]), and identity management (2 – [18,49]) to hold entities accountable in systems.

19 further mechanisms appeared in only one implementation each. These are represented as “Others” in Figure 5 and

include Certificates, Trace, Pseudonyms, Pseudonymity, Log Tamper Resistance, Time Synchronisation, Reputation System, Unlinkability, Accountable Anonymity, Online Analytical Processing (OLAP), Questionnaire and report generation, Key Management, Resource Description Framework (RDF), Job-flow Tracking, Fault Detection, Monitoring, Onion Routing, Decentralization, and Shamir’s threshold scheme.

We found that accountability mechanisms are mainly implemented at the application layer (10 instances – [15,18,23,28,30,38,40,43,48,52]) and the network layer (8 – [11,13,20,25,26,28,32,37]), cf. Figure 6. Few solutions are implemented at the kernel layer (3 – [16,17,32]) and the file system layer (1 – [47]).

Research Types (RQ 4)

Our classification of the contributions is based on the classification scheme by Wieringa et al. [53] which was applied to systematic mapping studies methodology by Peterson et al. [7]. We classify the selected papers strictly according to their criteria, which are:

Validation Research: The investigated techniques are novel with a potentially high impact for practice and they have

not yet been adopted in practice. Such techniques are, for example, experiments, i.e. work performed in the lab with a significant amount of real world data.

Evaluation Research: The techniques have been implemented and a thorough evaluation of the technique is conducted. That means, it is shown how the technique is implemented (solution implementation) and what the consequences for the practice are. This also includes the identification of problems in the industry. An example for evaluation research is the reporting of an industry case study.

Solution Proposal: A solution for a problem is proposed. The solution can be either novel or a significant extension of an existing technique. The potential benefits and the applicability of the solution is shown by a small example or a good line of argumentation.

Philosophical Papers: These papers provide new ideas and structures, such as conceptual frameworks of a research field.

Opinion Papers: These papers express the personal opinion about a technique and do not provide implementations.

Experience Papers: Experience papers present and discuss whether and how certain techniques work in practice. Such papers are solely based on personal experiences.

Table 2 maps the selected papers according to these criteria. We realize that all papers focus on solutions and their evaluations. Note that our mapping study focuses on papers that report on techniques that are actually implemented; we excluded meta studies. Hence, we find no papers in the categories experience paper, opinion paper, or philosophical paper.

Figure 7 again categorizes the papers into the above facets, but focuses on the papers' distribution over the years. We realize that initial works provided only solutions, while within the last five years the number of evaluation papers has increased significantly. Validation research is still missing in the field. We could only identify one publication that fits this criteria.

Contributors and Relationships (RQ5)

Collaboration Networks

We analyzed the author networks of the selected papers. First, we find that most authors feature only one publication on accountability implementations, as indicated by the size of the nodes in Figure 8. 13 authors feature two publications, while only one author features three. For the authors with at least two publications, we found that the corresponding papers are closely related follow-up papers.

As also indicated by Figure 8, the analyzed author network is very scattered. The authors of accountability tools do not collaborate across research groups. Again, the only papers published by the same authors are [11, 12], [23, 24], and [16, 31, 32] all of which are a series of papers.

These results lead us to the conclusion and hypothesis that the field of accountability implementations would greatly benefit from more systematic collaborations and research among the identified researchers.

Most Cited Researchers

We further analyzed the references of the 45 selected papers. Our goal was to find out whether they share common literature that is essential for the understanding and implementation of accountability mechanisms. Because some authors made heavy use of self citations, we decided to exclude any self references.

Indeed, we realized that there exist some researchers that are cited across many of the study papers. Table 3 shows those researchers that were cited at least seven times.

Definitions of Accountability (RQ6)

We scanned all 45 papers for the definition of accountability they use. To find the definition we searched the documents for all occurrences of the word "accountability". We then read the text before and after the highlighted term and looked for a definition.

We found that 20 of the 45 papers provide no explicit definition of the term "accountability". 17 papers provide their own definition, not taking other sources into account. These definitions define accountability in terms of responsibility/ assigning blame (6), non-repudiation/ integrity (3), a-posteriori enforcement (3), collect evidence (2), transparency (2), traceability (1).

Only 8 papers rely on a previously-published and peer-reviewed definition of accountability. The definitions were referencing the following papers:

Anderson et al. [54] the "(...) ability to associate an action with the responsible entity"

Bhargav-Spantzel et al. [55] "(...)the ability of holding entities responsible for their actions"

Brzuska et al. [56] "A sanitizable signature scheme satisfies non-interactive public accountability, if and only if for a valid message/signature pair (m, σ) , a third party can correctly decide whether (m, σ) originates from the signer or from the sanitizer without interacting with the signer or sanitizer."

Ko et al. [57] who rely on [58] and use the definition from the "The Best Practices Act of 2010" (we, however, could not find the formulation in the original source): "the obligation and/ or willingness to demonstrate and take responsibility for performance in light of agreed-upon expectations."

Pearson [58] relies on Weitzner et al. [4] and extends the definition of the "Galway project": "Accountability is the obligation to act as a responsible steward of the personal information of others, to take responsibility for

Table 2. Paper categorisation into research type facets; grouped by publisher

Category	ACM	IEEE	Springer	Others
Validation Research	[19]			
Evaluation Research	[8, 12, 43, 51]	[9, 11, 14, 16, 23, 31, 32, 36, 49, 50]	[13, 15, 47, 48]	[46]
Solution Proposal	[27, 30]	[10, 17, 21, 22, 24–26, 28, 29, 37, 38, 40, 42, 44, 45]	[18, 20, 33–35, 39, 41]	[52]

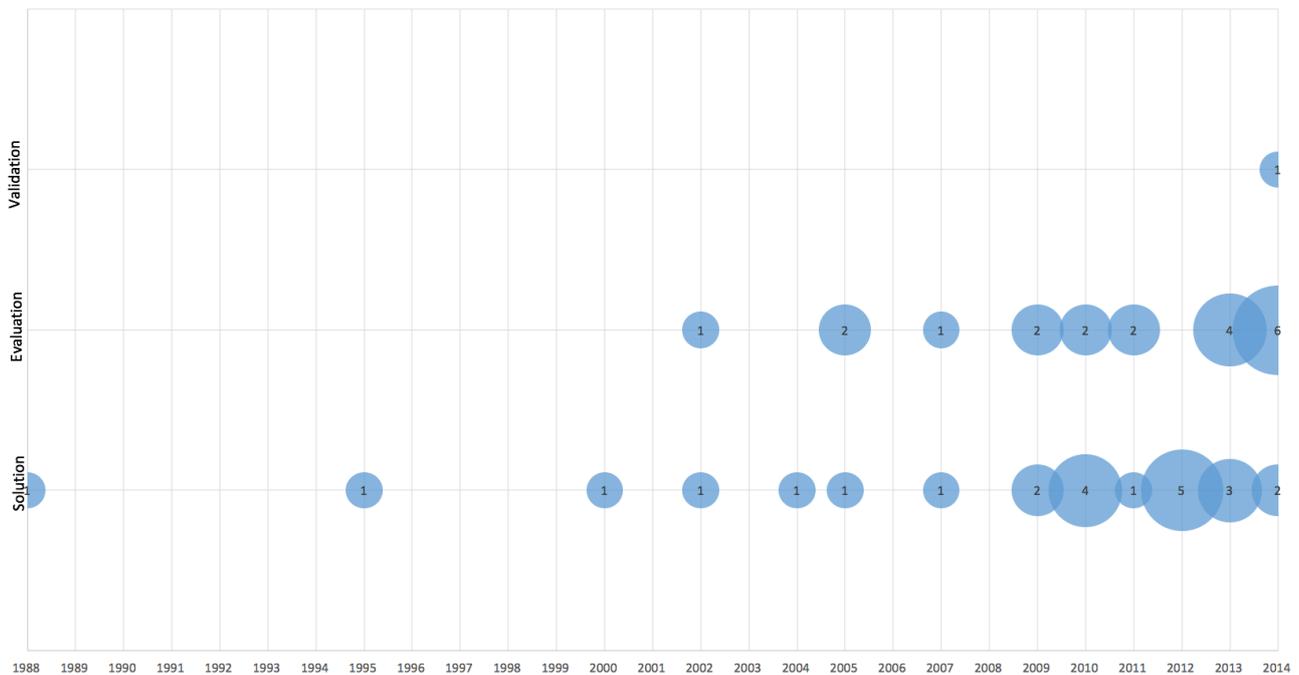


Figure 7. Number of contributions over time and structured according to research type facets

the protection and appropriate use of that information beyond mere legal requirements, and to be accountable for any misuse of that information.”

Xiao [59] “Accountability implies that any entity should be held responsible for its own specific action or behavior so that the entity is part of larger chains of accountability. One of the goals of accountability is that once an event has transpired, the events that took place are traceable so that the causes can be determined afterward.”

These definitions, like the 17 definitions provided by the other papers, are still lacking rigor and rely on a common understanding of the (dictionary-)meaning⁶ of accountability. Again, “responsibility” features strongly in these definitions and we were thus surprised that responsibility does not appear prominently in the papers’ abstracts (see Figure 9) or the most common bigrams (see the following section and Table 4).

⁶The Oxford dictionary defines accountability as “The fact or condition of being accountable; responsibility”. For a more detailed discussion see [6].

Common Terms (RQ 7)

In order to find out which concepts are of particular interest when implementing accountability mechanisms, we analyzed the 45 selected papers for their most common bigrams. For this, we converted the original PDF files to text files and preprocessed them manually, removing bibliographies and running heads. We converted the text to lower case and used Python library `nltk.stem.snowball.SnowballStemmer` to remove common English words as well as terms such as “figure” and “approach”, which are commonly used in academic publications. Finally, the text was tokenized and stemmed using the above library. We identified bigrams using `nltk.collocations.BigramAssocMeasures` and `BigramCollocationFinder`.

Table 4 lists all bigrams that we found to occur at least 50 times in at least four papers. We observe that 27 papers rely on some kind of access control and that 20 papers leverage the concept of public key cryptography (as indicated by the terms public key, private key, and secret key). The concept of a

Table 4. Most common bigrams

1st word	2nd word	Total Count	Papers
access	control	131	27
public	key	145	20
third	party	73	20
private	key	65	17
data	collect	66	13
cloud	computing	82	10
web	service	79	10
personal	data	102	9
secret	key	59	7
data	provenance	83	6
provenance	information	51	6
system	call	59	4
information	process	52	4

accountability implementations?

Cloud computing is en vogue. At the same time, it is one of the application domains where most privacy and data protection concerns have been raised. Distributed data sharing is another such domain. Encryption and access control have been shown to be insufficient for addressing these issues in remote computing and data sharing in general [4]. Hence, it is only obvious that researchers are trying to address privacy and security issues by detective enforcement viz. implementing accountability in these domains (Figure 3). An interesting finding is that Web Applications and Healthcare domains have not attracted equal focus, especially Healthcare where HIPPA (Health Insurance Portability and Accountability Act of 1996) explicitly mandates accountability enforcements.

Another interpretation of Figure 3 is that the need of accountability implementation has been recognized in many application domains, evident from the single papers that discuss implementations in several domains, grouped under “Others”. This leads us to the conclusion that while the field of accountability implementations is focused around cloud computing and distributed data sharing, at the same time it is quite scattered across multiple other application domains.

RQ 3 Which underlying techniques/protocols are implemented by these tools? At which layers of abstraction do the tools exist? Is there a trend?

The underlying techniques in accountability implementations are dominated by cryptography and network protocols. We found only one implementation relying on data provenance and very few accountability-centric protocols which combine, e.g., anonymity with accountability.

We observed three overall trends in mechanisms offered within accountability implementations. First, cryptog-

raphy is dominating the field with, e.g., Public Key Infrastructures, signature-based solutions, and certificates. Second, access control mechanisms are widespread. Either under the term access control or in supporting topics such as policy-based approaches, authentication mechanisms, or identity management. Third, privacy is a recurring theme in particular with respect to anonymity. Further privacy goals such as pseudonymity and unlinkability are supported as well, but to a lesser extend. We sparsely encountered further supporting mechanisms such as provenance and traceability.

We can report that over 80% of the implementations are in the network and application domains. Among these, there is a slight peak towards application. Very little attention has been given to the kernel or file system abstraction layers.

To summarize, we could identify trends towards considering accountability as part of cryptography and network protocols. Less than 20% of the implementations provided protocols in which the primary concern is accountability. A trend is to offer accountability together with mechanisms for cryptography, access control, or privacy. Note that privacy can also be enforced using cryptography or access control. Hence, we suspect that the main trend goes towards privacy and accountability. The trend in terms of abstraction layers is almost equally towards application and network layers.

RQ 4 Of what type is the research?

The research types in the field of implemented accountability approaches are validation, evaluation and solution approaches (c.f. Sect. *Research Types*). It is no surprise that the field started with solution approaches and moved over time to evaluation approaches. The majority of publications in the years 2013 and 2014 are of that type. We have seen only one validation approach. We assume over time the focus of research will go towards evaluation approaches and ultimately validation approaches. Hence, the field evolves towards evaluation research, while we see a clear gap in validation research.

RQ 5 Are prominent contributors recognizable? How are they related to each other?

In contrast to the theoretical discussions of accountability, where we have often cited papers like the one by Weitzner et al. [4] or Feigenbaum et al. [60], there are no especially noticeable contributors. We assume that there are more prominent works on topics related to (but not called) accountability, like fault localization or root cause analysis. This suggests that a clear and thorough overview of the whole field of computer science is needed. This should then yield to a clearer definition and taxonomy of the term accountability and its related concepts.

RQ 6 What have the underlying definitions of accountability in common?

It was surprising that no clear and accepted definition of accountability emerged. We assume that one main reason for this is that it is a common English word and everyone has some intuitive understanding of the term. The lack of a clear definition and differentiation from other terms like “responsibility” or “detection” hinders the scientific discourse and the comparability of the approaches. We hope that in the future works will rely on a peer reviewed definition of accountability and that thus trends and relations among approaches will become more pronounced.

Despite this, all definitions see accountability as some form of a-posteriori mechanism to provide evidence and ultimately assign blame or responsibility. It relies either on logs or some other form of monitoring.

RQ 7 Can we identify common terms in the field?

We could identify common terms from the fields of security and privacy in our analysis. For example, the terms attack, identity, policy are obviously linked to these fields. We are missing terms related to safety such as fault or hazard.

Limitations

There are two main limitations of this mapping study: the first is the selection of papers and the other is our potential bias when reviewing and categorizing the papers. First, by limiting ourself to the term “accountability” we probably missed papers that implement similar concepts but call them differently (e.g., “black box” or “root cause analysis”). We made our choice based on experiences of existing research. Petticrew and Roberts [61] highlight that the two main issues in conducting an literature survey are the sensitivity and specificity of the search. The sensitivity refers to the number of relevant publications of a search. Specificity describes the number of irrelevant studies of a search. The aim is to have a high sensitivity and a low specificity of a search. Synonyms may increase the sensitivity, but it also increases the specificity. Previous experiences of literature studies advocate simple search strings and limited synonyms to achieve an optimal trade-off between specificity and sensitivity e.g. Salleh et al. [62].

Second, it is very possible that we collectively mis-classified some papers. We countered this with a multi-staged voting process and took special care that every paper was reviewed by at least two different researchers.

Furthermore an inherent limitation of mapping studies is the superficial review of the source literature. Especially in the early stages we only looked at the abstract of a paper and not at its content. In the later stages we skimmed through each paper, but no paper was read in its entirety.

Conclusion & Future Work

Through this systematic mapping study, we establish the state of the art in accountability implementations and tools.

We have considered only those papers that describe an implementation. We did not consider contributions that described, even if in detail, how the ideas could be implemented. In this context, an interesting finding is that none of the papers have evaluated their tools for performance. This is important because one key factor that could limit the usefulness of accountability mechanisms is performance efficiency. The reason being that the origin of unwanted events is tracked typically using logging and analysis of “interesting” system events. Depending upon the complexity of the analysis algorithm and the size of the logs, accountability implementations could be very expensive in terms of computation. It would help to get an insight into how the existing implementations fare and if (at least) the concepts could be reused in domains where almost real-time processing is needed, viz., the automotive domain.

Another identified gap is the missing link between the high-level unwanted events that take place in an environment (e.g., personal and medical data is leaked in a Healthcare domain application) and the low-level unwanted events that are logged in the running technical systems (e.g., system calls reading from confidential files and writing to a socket in a network connection). It is important to establish this link because unwanted events are extracted from high-level requirements of privacy, security and safety properties and there is no universally agreed upon semantics of the relevant high-level events (e.g., data leak) in terms of low-level technical events (e.g., system calls writing to sockets). Though this gap has been filled in the context of preventive enforcement of usage control, it is not clear how this could be done for accountability.

One of our goals of this study was to identify which properties are often considered in combination with accountability and {safety, security and privacy}. We found that the most important properties are integrity, provenance, trust, legal compliance, confidentiality, transparency, traceability, auditability and non-repudiation. Most papers have more than one of these properties considered along with accountability. An interesting finding here is that none of the papers implemented a safety property. This discovery points out a gap in the work on accountability for safety-critical systems.

We were also surprised that relevant concepts like reasoning, log analysis and causality did not feature prominently in the result set. Current accountability technologies focus mainly on preventive concepts (Policies and Access Control) or Authenticity/Non-Repudiation (PKI, Anonymity and Signatures). At the high-level view of this mapping study we could not reliably identify an a-posteriori approach. We believe that this needs to change in the future. While it is feasible to manually analyse the logs (flight recorders) the (fortunately) few times a year an aircraft crashes, it becomes in-feasible when a few dozened drones crash every day. Hence, our future

contribution for the TUM LLCM project will focus on these features.

Our conclusion is that though accountability concepts have been around since quite some time, this area has not seen enough implementations, especially of a-posteriori approaches. At the technical level, there exists no generally accepted architecture and we did not come across contributions that give insights into acceptability issues like usability, scalability, etc. At the methodological level, there are no processes for deriving accountability-specific requirements. Thus, there is plenty of room for developing an innovative accountability infrastructure for TUM LLCM.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

The authors gratefully acknowledge the contribution of Severin Kacianka and Florian Kelbert in preparing this report.

References

- [1] Florian Kelbert and Alexander Pretschner. A Fully Decentralized Data Usage Control Enforcement Infrastructure. In *Applied Cryptography and Network Security*, volume 9092 of *Lecture Notes in Computer Science*, pages 409–430. Springer International Publishing, 2015.
- [2] Dean Povey. Optimistic security: A new access control paradigm. In *Proceedings of the 1999 Workshop on New Security Paradigms*, NSPW '99, pages 40–45, New York, NY, USA, 2000. ACM.
- [3] James J Rooney and Lee N Vanden Heuvel. Root cause analysis for beginners. *Quality progress*, 37(7):45–56, 2004.
- [4] Daniel J. Weitzner, Harold Abelson, Tim Berners-Lee, Joan Feigenbaum, James Hendler, and Gerald Jay Sussman. Information accountability. *Commun. ACM*, 51(6):82–87, June 2008.
- [5] Zhifeng Xiao, Nandhakumar Kathiresshan, and Yang Xiao. A survey of accountability in computer networks and distributed systems. *Security and Communication Networks*, 2012.
- [6] Nick Papanikolaou and Siani Pearson. A cross-disciplinary review of the concept of accountability. In *Proceedings of the DIMACS/BIC/A4Cloud/CSA International Workshop on Trustworthiness, Accountability and Forensics in the Cloud (TAFIC)(May 2013)*, 2011.
- [7] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *12th Intl. Conf. on Evaluation and Assessment in Software Engineering*, volume 17. sn, 2008.
- [8] Nikolaos Alexiou, Marcello Laganà, Stylianos Gisdakis, Mohammad Khodaei, and Panagiotis Papadimitratos. VeSPA: Vehicular Security and Privacy-preserving Architecture. In *Proc. 2nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy*, pages 19–24. ACM, 2013.
- [9] M. Ahmed and M. Ahamad. Combating Abuse of Health Data in the Age of eHealth Exchange. In *IEEE International Conf. on Healthcare Informatics*, pages 109–118, September 2014.
- [10] M. Ali and L. Moreau. A Provenance-Aware Policy Language (cProv1) and a Data Traceability Model (cProv) for the Cloud. In *Third International Conf. on Cloud and Green Computing*, pages 479–486, September 2013.
- [11] S.T. Ali, V. Sivaraman, D. Ostry, G. Tsudik, and S. Jha. Securing First-Hop Data Provenance for Bodyworn Devices Using Wireless Link Fingerprints. *IEEE Transactions on Information Forensics and Security*, 9(12):2193–2204, December 2014.
- [12] Syed Taha Ali, Vijay Sivaraman, Diethelm Ostry, and Sanjay Jha. Securing Data Provenance in Body Area Networks Using Lightweight Wireless Link Fingerprints. In *Proc. 3rd International Workshop on Trustworthy Embedded Devices*, pages 65–72. ACM, 2013.
- [13] Changho Choi, Yingfei Dong, and Zhi-Li Zhang. LIPS: Lightweight Internet Permit System for Stopping Unwanted Packets. In *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, volume 3462 of *Lecture Notes in Computer Science*, pages 178–190. Springer Berlin Heidelberg, 2005.
- [14] R.-A. Cherrueau and M. Sudholt. Enforcing Expressive Accountability Policies. In *IEEE 23rd International WET-ICE Conference*, pages 333–338, June 2014.
- [15] Christina Brzuska, Henrich C. Pöhls, and Kai Samelin. Efficient and Perfectly Unlinkable Sanitizable Signatures without Group Signatures. In *Public Key Infrastructures, Services and Applications*, volume 8341 of *Lecture Notes in Computer Science*, pages 12–30. Springer Berlin Heidelberg, 2014.
- [16] Chun Hui Suen, R.K.L. Ko, Yu Shyang Tan, P. Jagadpramana, and Bu Sung Lee. S2Logger: End-to-End Data Tracking Mechanism for Cloud Data Provenance. In *12th IEEE International Conf. on Trust, Security and Privacy in Computing and Communications*, pages 594–602, July 2013.
- [17] D.B. Clifton and E.B. Fernandez. A Microprocessor Design for Multilevel Security. In *Fourth Aerospace Computer Security Applications Conference*, pages 194–198, December 1988.

- [18] Jose M. Such, Agustin Espinosa, and Ana Garcia-Fornes. An Agent Infrastructure for Privacy-Enhancing Agent-Based E-commerce Applications. In *Advanced Agent Technology*, volume 7068 of *Lecture Notes in Computer Science*, pages 411–425. Springer Berlin Heidelberg, 2012.
- [19] Sascha Fahl, Sergej Dechand, Henning Perl, Felix Fischer, Jaromir Smrcek, and Matthew Smith. Hey, NSA: Stay Away from My Market! Future Proofing App Markets Against Powerful Attackers. In *Proc. 2014 ACM Conference on Computer and Communications Security*, pages 1143–1155. ACM, 2014.
- [20] Craig Pearce, Peter Bertok, and Ron Van Schyndel. Protecting Consumer Data in Composite Web Services. In *Security and Privacy in the Age of Ubiquitous Computing*, volume 181 of *IFIP Advances in Information and Communication Technology*, pages 19–34. Springer US, 2005.
- [21] A. Dailianas, Y. Yemini, D. Florissi, and H. Huang. MarketNet: market-based protection of network systems and services—an application to SNMP protection. In *Proc. 19th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, March 2000.
- [22] A.S. De Oliveira, J. Sendor, A. Garaga, and K. Jenatton. Monitoring Personal Data Transfers in the Cloud. In *IEEE 5th Intl. Conf. on Cloud Computing Technology and Science*, volume 1, pages 347–354, December 2013.
- [23] S. Fugkeaw, P. Manpanpanich, and S. Juntapremjitt. A-COLD: Access Control of Web OLAP over Multi-data Warehouse. In *International Conf. on Availability, Reliability and Security*, pages 469–474, March 2009.
- [24] S. Fugkeaw, P. Manpanpanich, and S. Juntapremjitt. AmTRUE: Authentication Management and Trusted Role-based Authorization in Multi-Application and Multi-User Environment. In *The International Conf. on Emerging Security Information, Systems, and Technologies*, pages 216–221, October 2007.
- [25] Gang Xu, L. Aguilera, and Yong Guan. Accountable Anonymity: A Proxy Re-Encryption Based Anonymous Communication System. In *IEEE 18th Intl. Conf. on Parallel and Distributed Systems*, pages 109–116, December 2012.
- [26] A.N. Haidar, S.J. Zasada, P.V. Coveney, A.E. Abdallah, and B. Beckles. Audited Credential Delegation - A User-centric Identity Management Solution for Computational Grid Environments. In *Sixth International Conf. on Information Assurance and Security*, pages 222–227, August 2010.
- [27] Lukasz Jedrzejczyk, Blaine A. Price, Arosha K. Bandara, and Bashar Nuseibeh. On the Impact of Real-time Feedback on Users' Behaviour in Mobile Location-sharing Applications. In *Proc. Sixth Symposium on Usable Privacy and Security*, pages 14:1–14:12. ACM, 2010.
- [28] Kang Wang, A.J. Malozemoff, Ning Jia, Chunhui Han, and M. Maheswaran. A Social Accountability Framework for Computer Networks. In *IEEE Global Telecommunications Conference*, pages 1–6, December 2010.
- [29] Y.J. Kang, A.M. Schiffman, and J. Shrager. RAPPD: A Language and Prototype for Recipient-Accountable Private Personal Data. In *IEEE Security and Privacy Workshops*, pages 49–56, May 2014.
- [30] Gaurangkumar Khalasi and Minubhai Chaudhari. TrustGK Monitor: 'Customer Trust As a Service' for the Cloud. In *Proc. CUBE International Information Technology Conference*, pages 537–543. ACM, 2012.
- [31] R.K.L. Ko, P. Jagadpramana, and Bu Sung Lee. Flogger: A File-Centric Logger for Monitoring File Access and Transfers within Cloud Computing Environments. In *IEEE 10th International Conf. on Trust, Security and Privacy in Computing and Communications*, pages 765–771, November 2011.
- [32] R.K.L. Ko and M.A. Will. Progger: An Efficient, Tamper-Evident Kernel-Space Logger for Cloud Data Provenance Tracking. In *IEEE 7th International Conf. on Cloud Computing*, pages 881–889, June 2014.
- [33] Pramote Kuacharoen. Design and Implementation of a Secure Online Lottery System. In *Advances in Information Technology*, volume 344 of *Communications in Computer and Information Science*, pages 94–105. Springer Berlin Heidelberg, 2012.
- [34] Kwei-Jay Lin and SooHo Chang. A Service Accountability Framework for QoS Service Management and Engineering. *Information Systems and e-Business Management*, 7(4):429–446, 2009.
- [35] Marc Langheinrich. A Privacy Awareness System for Ubiquitous Computing Environments. In *UbiComp 2002: Ubiquitous Computing*, volume 2498 of *Lecture Notes in Computer Science*, pages 237–245. Springer Berlin Heidelberg, 2002.
- [36] F. Masmoudi, M. Loulou, and A.H. Kacem. Multi-tenant Services Monitoring for Accountability in Cloud Computing. In *IEEE 6th Intl. Conf. on Cloud Computing Technology and Science*, pages 620–625, December 2014.
- [37] A. Michalas and N. Komninos. The Lord of the Sense: A Privacy Preserving Reputation System for Participatory Sensing Applications. In *IEEE Symp. on Computers and Communication*, pages 1–6, June 2014.
- [38] D. Mortimer and N. Cook. Supporting Accountable Business to Business Document Exchange in the Cloud. In *IEEE International Conf. on Service-Oriented Computing and Applications*, pages 1–8, December 2010.
- [39] N. Asokan, Alexandra Dmitrienko, Marcin Nagy, Elena Reshetova, Ahmad-Reza Sadeghi, Thomas Schneider, and Stanislaus Stelle. CrowdShare: Secure Mobile Resource Sharing. In *Applied Cryptography and Network*

- Security*, volume 7954 of *Lecture Notes in Computer Science*, pages 432–440. Springer Berlin Heidelberg, 2013.
- [40] J. Pato, S. Paradesi, I. Jacobi, Fuming Shih, and S. Wang. Aintno: Demonstration of Information Accountability on the Web. In *IEEE 3rd Intl. Conf. on Privacy, Security, Risk and Trust and 2011 IEEE 3rd Intl. Conf. on Social Computing*, pages 1072–1080, October 2011.
- [41] Paul Ruth, Dongyan Xu, Bharat Bhargava, and Fred Ragner. E-notebook Middleware for Accountability and Reputation Based Trust in Distributed Data Sharing Communities. In *Trust Management*, volume 2995 of *Lecture Notes in Computer Science*, pages 161–175. Springer Berlin Heidelberg, 2004.
- [42] S. Pearson, P. Rao, T. Sander, A. Parry, A. Paull, S. Patrini, V. Dandamudi-Ratnakar, and P. Sharma. Scalable, accountable privacy management for large organizations. In *13th Enterprise Distributed Object Computing Conference Workshops*, pages 168–175, September 2009.
- [43] Raluca Ada Popa, Andrew J. Blumberg, Hari Balakrishnan, and Frank H. Li. Privacy and Accountability for Location-based Aggregate Statistics. In *Proc. 18th ACM Conf. on Computer and Communications Security*, pages 653–666. ACM, 2011.
- [44] A.D. Rubin. Trusted Distribution of software over the Internet. In *Proc. Symp. on Network and Distributed System Security*, pages 47–53, February 1995.
- [45] V. Sriram, G. Narayan, and K. Gopinath. SAFIUS - A Secure and Accountable Filesystem over Untrusted Storage. In *Fourth International IEEE Security in Storage Workshop*, pages 34–45, September 2007.
- [46] Jose M. Such, Ana García-Fornes, Agustín Espinosa, and Joan Bellver. Magentix2: A privacy-enhancing Agent Platform. *Engineering Applications of Artificial Intelligence*, 26(1):96–109, 2013.
- [47] Ulrich Flegel. Pseudonymizing Unix Log Files. In *Infrastructure Security*, volume 2437 of *Lecture Notes in Computer Science*, pages 162–179. Springer Berlin Heidelberg, 2002.
- [48] Vincent Naessens, Bart De Decker, and Liesje Demuyneck. Accountable Anonymous E-Mail. In *Security and Privacy in the Age of Ubiquitous Computing*, volume 181 of *IFIP Advances in Information and Communication Technology*, pages 3–18. Springer US, 2005.
- [49] Wonjun Lee, A.C. Squicciarini, and E. Bertino. The Design and Evaluation of Accountable Grid Computing System. In *29th IEEE International Conference on Distributed Computing Systems*, pages 145–154, June 2009.
- [50] Yang Xiao, Ke Meng, and D. Takahashi. Implementation and Evaluation of Accountability Using Flow-net in Wireless Networks. In *Military Communications Conference*, pages 7–12, October 2010.
- [51] Wenchao Zhou, Micah Sherr, Tao Tao, Xiaozhou Li, Boon Thau Loo, and Yun Mao. Efficient Querying and Maintenance of Network Provenance at Internet-scale. In *Proc. 2010 ACM SIGMOD International Conf. on Management of Data*, pages 615–626. ACM, 2010.
- [52] Kato Mivule, Stephen Otunba, and Tattwamasi Tripathy. Implementation of Data Privacy and Security in an Online Student Health Records System. Technical report, Department of Computer Science, Bowie State University, 2014.
- [53] Roel Wieringa, Neil Maiden, Nancy Mead, and Colette Rolland. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requir. Eng.*, 11(1):102–107, 2005.
- [54] David G Andersen, Hari Balakrishnan, Nick Feamster, Teemu Koponen, Daekyeong Moon, and Scott Shenker. Accountable internet protocol (aip). In *ACM Computer Communication Review*, volume 38, pages 339–350. ACM, 2008.
- [55] Abhilasha Bhargav-Spantzel, Jan Camenisch, Thomas Gross, and Dieter Sommer. User centricity: a taxonomy and open issues. *Journal of Computer Security*, 15(5):493–527, 2007.
- [56] Christina Brzuska, Henrich C Pöhls, and Kai Samelin. Non-interactive public accountability for sanitizable signatures. In *Public Key Infrastructures, Services and Applications*, pages 178–193. Springer, 2012.
- [57] Ryan KL Ko, Peter Jagadpramana, Miranda Mowbray, Siani Pearson, Markus Kirchberg, Qianhui Liang, and Bu Sung Lee. Trustcloud: A framework for accountability and trust in cloud computing. In *Services (SERVICES), 2011 IEEE World Congress on*, pages 584–588. IEEE, 2011.
- [58] Siani Pearson. Toward accountability in the cloud. *IEEE Internet Computing*, 15(4):64, 2011.
- [59] Yang Xiao. Flow-net methodology for accountability in wireless networks. *Network, IEEE*, 23(5):30–37, 2009.
- [60] Joan Feigenbaum, Aaron D Jaggard, and Rebecca N Wright. Towards a formal model of accountability. In *Proc. 2011 workshop on New security paradigms workshop*, pages 45–56. ACM, 2011.
- [61] M. Petticrew and H. Roberts. *Systematic Review in the Social Sciences: A Practical Guide*. Blackwell Publishing, 2006.
- [62] N. Salleh, E. Mendes, and J. Grundy. Empirical studies of pair programming for cs/se teaching in higher education: A systematic literature review. *IEEE Transactions on Software Engineering*, 37(4):509–525, 2011.

Multi-Layer Monitoring and Visualization

Martin Kleehaus, Jörg Landthaler, Dominik Huth and Florian Matthes

Department of Informatics, Technical University of Munich, Munich
{martin.kleehaus, joerg.landthaler, dominik.huth, matthes}@tum.de

Abstract

This document provides the State of the Art report for the TUM Living Lab Connected Mobility (LLCM) sub-projects Integrated Monitoring (3.2) and Visual Service-Management Control Panel (3.3). The overall goal of the TUM LLCM project is to build a platform that provides various mobility services to end and business clients, while the two sub-projects covered here attempt to provide a monitoring solution for the platform with a focus on the meaningful combination of data collected from different levels of abstraction. Therefore, we identify and examine different research areas present in the academic literature on the topic of monitoring from a high-level perspective. We also present the results of a systematic literature review carried out on the topic of multi-layer monitoring from an Enterprise Architecture Management (EAM) point of view. Additionally, we provide a survey of monitoring tools used in industry as well as an overview of different visualization types utilized in monitoring applications.

Keywords

monitoring; multi-layer monitoring; monitoring taxonomy; enterprise architecture management; literature review; tool survey; visualization types

Introduction

The measurement and control of IT and business services delivered by an Enterprise Architecture (EA) is based on a continuous process of monitoring, reporting, learning and subsequent actions. These steps are fundamental as they provide required enactment proposals to support and improve delivered services. Therefore, it is important to note that, although this monitoring process takes place during service operation, it provides a basis for setting strategy, planning, building and testing services and achieving meaningful improvement in the Enterprise Architecture. In addition, all phases of the Service Lifecycle should ensure that measures and controls are clearly defined, executed and acted upon. This cycle of Service Management is illustrated in figure 1.

The definition of what needs to be monitored is based on understanding the desired outcome of a process, device or system. IT should focus on the service and its impact on the business, rather than just the individual components of technology. For that reason various frameworks emerged in the last decades (like TOGAF [1], ITIL [2], and others) delivering a holistic view on the EA encompassing several layers which 1. categorize the IT in a service oriented way, 2. describe how they impact and align with the business and 3. how the interface between the layers that exchange data and services have to be shaped.

As a consequence, monitoring solutions have been developed to account for these layered architectures and provide an improvement in the measurement and control of the IT and business services. The so called *multi-layer monitoring solutions* are able to obtain and manage monitoring informa-

tion from several EA layers in the same time and track the impact of a workload or a failure from layer to layer. Although, user-centric information like the user behavior are not regarded as a separate EA layer in the aforementioned frameworks, multi-layer monitoring solutions do collect data from this "layer" and analyze for instance how the IT impact the user experience. For that reason, we extend the standard EAM layers with a user layer as it is illustrated in figure 1.

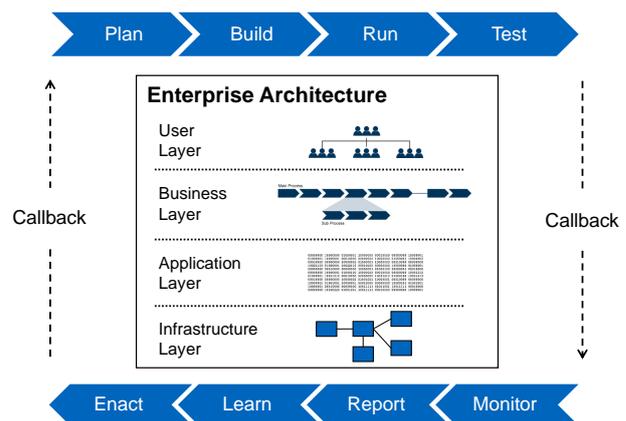


Figure 1. Cycle of continuous service improvement in an Enterprise Architecture

For the LLCM research project, a monitoring solution is envisaged that can monitor a layered target architecture, e.g. a Cloud solution. Core and adjacent aspects such as the meaningful combination of monitoring data acquired from different layers, predictive analytics capabilities, accountabil-

ity support and adequate visualizations are relevant as well. As a first step to approach this research project, we assess the state of the art in multi-layer monitoring in academia and industry. In particular we attempt to answer the following research questions:

- RQ1: Which are the potentially relevant research areas in the domain of monitoring? This research question is limited to research areas that focus on the monitoring of IT systems rather than other monitoring domains, such as civil infrastructure monitoring.
- RQ2: How can the approaches and solutions present in the research areas covered by RQ1 be categorized and how can relations among them be identified?
- RQ3: Which approaches and implementations exist that cover at least two layers from an EAM + User layers perspective?
- RQ4: Which open source or commercial monitoring solutions can be identified that can be relevant for the part-projects?
- RQ5: Which visualization types are used in the software tools covered by RQ4?

The remainder of this document is structured as follows: In Section 2 we identify and briefly explain relevant and neighboring research areas. We build and explain a taxonomy for monitoring in Section 3. Next, we present the results of a systematic literature review on multi-layer monitoring in Section 4. Moreover, we conduct a small tool survey in Section 5 as well as an overview of utilized visualization types in monitoring applications in Section 6. Section 7 concludes this work with a brief summary and a short discussion of the main results. Please find the glossary in Section 8 for an introduction to relevant terms.

Research Areas

In this Section we identify different monitoring research areas in the domain of computer science and business information systems research. From a methodological point of view, we use a combination of a data-, literature- and expert-driven approach and roughly quantify the amount of publications in these research areas. As a first step, we perform a quantitative analysis of keywords associated with *monitoring* in the digital academic library Scopus¹. The identified terms serve as a starting point for the subsequent refinements and also for the identification of suitable keywords for our systematic literature review in Section 4.

Table 1 illustrates that a strictly data-driven procedure returns a significant amount of noise, e.g. *using* is just a common term when describing a tool or a specific method. Yet, gerunds like *using* cannot be excluded from the search,

¹<https://www.scopus.com/>

Keyword	Occurrences
monitoring	26282
system	6118
using	3601
sensor	2249
health	1826
network	1771
wireless	1759
data	1664

Table 1. Unconsolidated co-occurrence frequency of nouns and gerunds with term *monitoring* in article titles on Scopus when restricting to results in computer science in the years 2005-2016

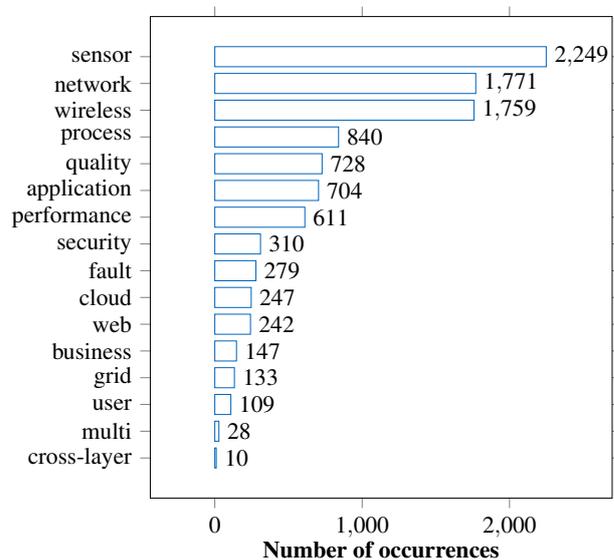


Figure 2. Consolidated number of co-occurrences in article titles with the term *monitoring*

because *monitoring* can be both a noun and a gerund and the underlying assumption is that we find as many occurrences of *monitoring* as we identified publications. Furthermore, also nouns do not necessarily provide insight into the research area, e.g. *data*. The reduction of the list has to be done in an informed manner. A consolidated list of co-occurring words, together with their occurrence frequency, is depicted in Figure 2.

Subsequently to this process, we consulted literature and experts in the field, which yielded - as our final result - the following research areas:

- **Cloud monitoring** describes monitoring within a cloud computing environment. As defined by Aceto et al. [3], it is concerned with the platforms, techniques and tools for monitoring cloud infrastructures (IaaS), services and applications (SaaS), as well as for platforms (PaaS).

- **Business process monitoring**, according to Aalst et al. [4], is the knowledge of process design in order to steer operational business processes. Another term associated with business process monitoring is **business activity monitoring**.
- **Fault detection**, a research area with a long history, aims at an early discovery of unintended behavior of a system. In Isermann [5] it is interpreted as an entry point towards analysis and decisions on actions in order to ensure operability of a system.
- **User experience monitoring** addresses the interaction of a user with any kind of system. The crucial part is measuring the user's personal perception of the interaction, as described by Albert [6]. **User behavior monitoring** is applied to gain knowledge about activities that a user performs, for instance which actions are commonly executed in an application.
- **Network monitoring** covers the monitoring of multiple servers or devices and their communication.
- **Wireless sensor networks** are networks of a large amount of physical sensors whose purpose is monitoring a large system. They are able to communicate with each other and form a network. Applications include the military, health, environment, or business domain, e.g. for surveillance or the management of store inventory.
- **Quality of Service** monitoring is used to ensure the fulfillment of service level agreements (SLA) that can be the base measure for payment of a service. Simple examples for variables to be monitored are response time or server throughput. **Application performance monitoring** can be seen as an adaptation of QoS Monitoring for Cloud environments.
- A grid, as described in Foster and Kesselman [7], originally was a distributed computing infrastructure for research in advanced engineering, but grid structures have been applied to commercial areas as well. In a flexible system with technical resources leaving and entering the grid, **grid monitoring** is the discovery and tracking of grid resources, compare Zaniolas and Sakellariou [8]. As a recent development, the area of **smart grid monitoring** is evolving due to the shift of energy production capacities from centralized powerplants to microproducers, such as photovoltaic collectors on private homes.
- **Web monitoring** specifically handles the domain of web applications. Possible objectives are performance monitoring or gaining insight into customer behavior and preferences.
- In the context of computer science, **Security monitoring** aims at discovering possible vulnerabilities in software systems.

Monitoring Taxonomy

The continuous measurement and observation of Key Performance Indicators (KPI) of an enterprise architecture is an unconditional necessity in order to keep the IT system behavior and the daily business operations under control. These KPIs are derived from several architecture layers describing the user experience, business operations, running applications and the underlying IT infrastructure. In this section, we introduce a holistic view on the basic concepts of IT monitoring which sets the context for the following sections in this paper. The concepts are classified in a taxonomy which is illustrated in figure 3.

Layers

According to an enterprise architecture view employed by standardized frameworks like TOGAF [1] the main layers which are relevant for monitoring purposes can be modeled as the *IT infrastructure* encompassing all technological aspects, the *application layer* which defines the software running in the IT infrastructure, the *business processes* operating on top of the aforementioned layers and the *user layer* which can be considered as the executive body of the enterprise. Although the *user layer* is not regarded as a separate layer in the architecture domain in most frameworks, it plays an important role regarding monitoring aspects as it uncovers crucial insights about user behavior and how stakeholders interact with the system. All layers are closely interrelated and are shaping the common understanding of an enterprise architecture. The following sections provide a detailed description of each layer.

IT Infrastructure

The IT infrastructure can be monitored at the hardware and network layer. These layers can be seen as places to put probes on the monitoring system. In fact, the layer at which the probes are located has direct consequences on the KPIs that can be monitored and analyzed:

- **Hardware:** at this layer we consider the physical components of the computing and networking equipment as well as the physical infrastructure like disk, RAM, CPU etc. Hardware monitoring basically reads out hardware sensors providing the current condition of the IT components, like CPU temperature, disk speed, RAM utilization and all devices communicating with each other in an IT infrastructure.

Monitoring metrics in the hardware level are computation-based. The assessment primarily focuses on performance purposes. Relevant metrics are, for instance, server throughput (which defines the number of requests per second), CPU Speed, CPU time per execution (defined as the CPU time of a single execution), CPU utilization, memory page exchanges per second (assesses the number of memory pages per second exchanged through the I/O), memory page exchanges per execution (defined as the number of memory pages used during an

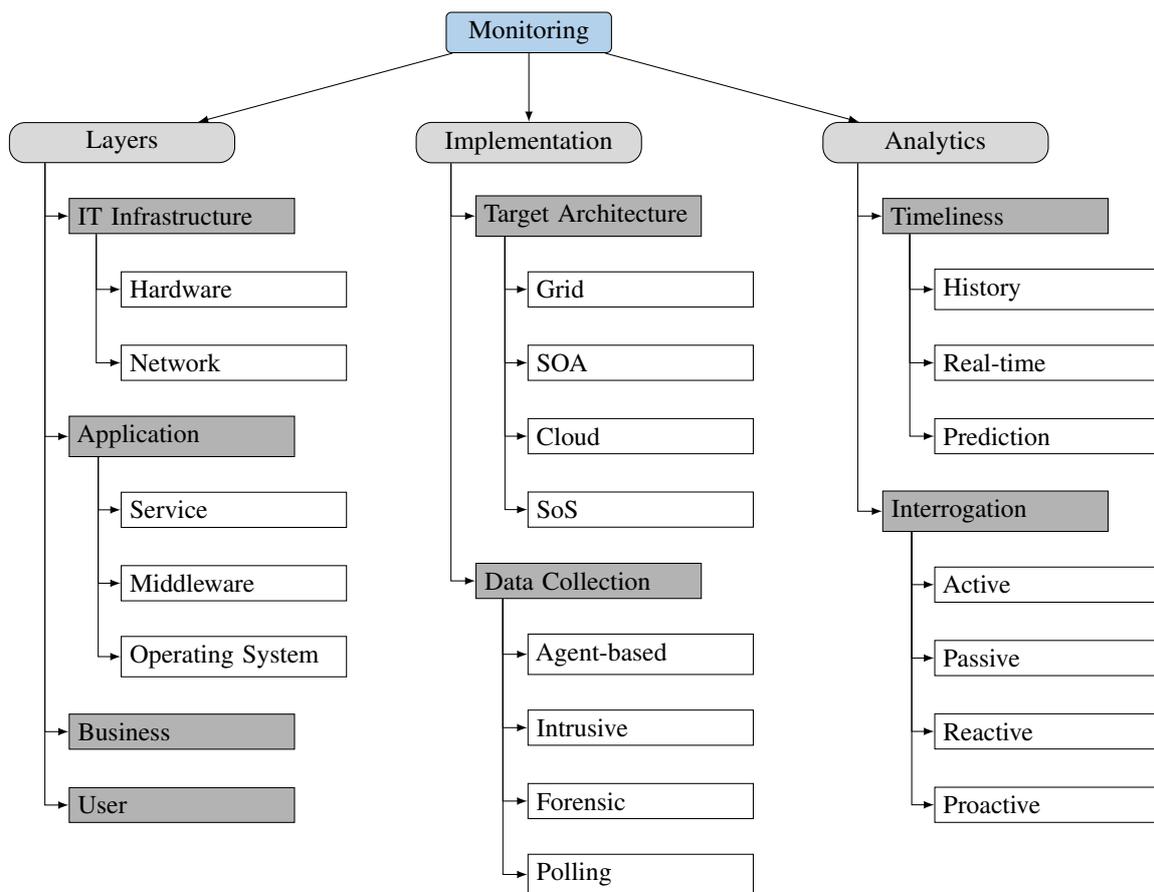


Figure 3. Monitoring Taxonomy

execution), disk/memory throughput, throughput/delay of message passing between processes, response time and others. All of them can be evaluated in terms of classical statistical indicators (mean, median, etc.) as well as in terms of temporal characterization and therefore visualized as time series data [3].

- **Network:** at this layer we consider the network links and paths between the hardware components. Network monitoring verifies the performance of the network and provides the ability to proactively respond to network outages. Metrics for assessing the performance level of network infrastructure can be divided into four main groups [9] [10] [11]: availability, loss and error, delay and bandwidth. Availability metrics assess how robust the network is, i.e. the percentage of time the network is running without any problem impacting the availability of services. Loss and error metrics measure the fraction of packets lost in a network due to buffer overflows or other reasons, or the fraction of erroneous bits or packets. Delay metrics provide information about One Way Delay (OWD), Round Trip Time (RTT) and Delay Variation (IPDV, or "jitter") of the packets transferred

by a network. Finally, bandwidth metrics assess the amount of data that a user can transfer through the network in a time unit, like the traffic volume.

Application

The application layer defines the software components in an IT infrastructure and can be categorized in the following [12], [13]:

- **Operating System:** at this layer we consider the software components forming the physical and virtual operating system. Operating system monitoring examines resource usage and aims to find out how efficiently resources are being used, at what proportions and by whom.
- **Service:** this layer presents the software running on the hardware components or the service which is provided by the system with direct or no user interface. Application metrics provide mainly information about the operation state, availability and the performance of the application.
- **Middleware:** this layer can be considered as the service provided to software applications beyond those

available from the operating system. Middleware in distributed computing includes software web servers and application server frameworks. For monitoring purposes, this allows collecting per-request information, keeping track of the amount of open sessions and states of transactions.

Business

Monitoring the business layer of an enterprise refers to the aggregation, analysis, controlling and presentation of real-time information about business activities, like the current status and the results of various operations, processes, and transactions. The main purpose of business activity monitoring (BAM) is to improve the speed and effectiveness of business operations [14]. This is feasible by collecting information from multiple application systems and other internal and external sources. Unlike traditional monitoring approaches, BAM draws its attention to several applications which combined support the whole business process. However, monitoring the business processes is still challenging compared to the previous EAM layers since services are intangible as they do not have material existence, often inseparable because the execution and consumption of services occurs frequently in parallel, immersive since services are often executed in collaboration with consumers, and bipolar, because services are often executed by a blend of human and technological resources [15].

Relevant key performance indicators for monitoring business activities can be classified into four key attributes [14]:

- *Volumes* counts for values of different aspects of the business process and its associated transactions. Examples are number of transactions, process events, tickets closed, compliance events for audit or process revenue, etc.
- Another attribute is *velocity* which indicates the time-related aspect of business operations, like idle-times between business activities or events, time remaining for process completion, process throughput, or life-time of tickets.
- The collection of *errors* during the business process is a further indicator which can be monitored. Required data are obtained at transaction level. Examples are occurrences of transactions which are executed out of sequence, duplicate transactions, or timeouts of steps or entire processes.
- The last category relates to *special conditions* which are defined by the user and represent the key to developing comprehensive key performance indicator-based measurements, allowing the user to combine the volume, velocity, and error measurements with business-specific knowledge and understanding. For example, a company might want to be alerted to any orders beyond a certain size or the presence of non-standard shipping instructions.

User

Monitoring the user aspects of an enterprise architecture mainly focuses on the user behavior [16]. In particular *real user monitoring* (RUM) is an approach which enables the analysis and construction of user behavior profile based on the transactions made on websites or applications [17]. This technology is a form of passive monitoring, relying on services that continuously observe the system in action, tracking availability, functionality, and responsiveness. While some "bottom-up" forms of RUM rely on capturing server-side information in order to reconstruct end-user experience, "top-down" client-side RUM can investigate, directly, how real human beings interact with the monitored application. Top-down RUM focuses on the direct relationship between application speed and user satisfaction which provides insights in order to optimize the overall application performance.

User based monitoring metrics as it is described in [18] consider every statistic which relates to the end-user experience, behavior, or customer related economic point of view. Classic user-based metrics in websites are, for instance, the web page load time, number of page views, the average time spent on the site, the percentage of returning visitors, the bounce rate (which is defined as the percentage of visitors to a particular website who navigate away from the site after viewing only one page), session length, conversion rate (what indicates the percentage of website visitors who can be converted to buying customers), website response time, and others.

Implementation

Monitoring solutions differ in particular regarding the architecture they are targeting to analyze and which approach they apply to collect and store the required data for further analysis. For instance, graph databases are well suited for storing monitoring data from network infrastructures. The following chapters provide an overview about implementation-oriented aspects of monitoring.

Target Architecture

Besides the traditional IT architecture consisting of the aforementioned layers new paradigms have evolved in the last years shifting the focus of providing information and services to a more network oriented, distributed and unstructured way. This introduces more complexity which poses a challenge for monitoring systems as they either make data accessibility difficult, since the IT infrastructure does not reside in one place anymore or they produce too much information in unstructured format. The following architectures are currently presenting a huge focus in the monitoring scope and still challenging:

- **Grid Monitoring:** Due to the distributed and heterogeneous resource compilation of grid infrastructures the establishment of a monitoring system is very challenging. Every component within the grid may fall into a different administrative control which makes the centralization of the monitoring data very difficult. Furthermore, grid systems are highly dynamic as resources

may join and leave during the time. Hence, monitoring systems must be extensible and scalable in order to cope efficiently with a growing number of resources, events and users [8].

- **Service-oriented Architecture (SOA) Monitoring:** A SOA application consists of a set of software components which provide services via communication protocols in order to fulfill business requirements. The communication protocols which are supported by SOA frameworks are manifold like HTTP, SOAP, FTP, REST, etc. In addition the SOA applications provide the communicated data in various formats like XML, JSON, etc. Both aspects makes it difficult to monitor these services and the data exchange as each protocol and data format requires different processing methods. The complexity grows as new wrappers or adapters have to be defined in order to handle these various formats. This also requires analysis and decisions on how to represent monitored data and if format conversion is required or not [19].
- **Cloud Monitoring:** Due to the very high complexity of cloud systems, certain phenomena are observed in the first place and disappear afterwards. For example, considering a probe in an application that runs in the cloud and collects information on the rate at which it exchanges information with other applications running in the same cloud, this rate might also comprise the transfer rate of the network. This depends on whether the two applications run on the same physical host or not, and this information is not always exposed by the service provider. Similar issues arise for evaluating the performance of computation: the time required for a task completion can depend on the actual hardware that is executing the instructions (usually exposed only as a CPU model or equivalent) and on the workload due to other virtualized environments running on the same physical server which are not exposed to the consumers at all [3].
- **System-of-Systems (SoS) Monitoring:** Many software systems have system-of-systems (SoS) architectures comprising interrelated and heterogeneous systems. Aforementioned systems evolve certain behavior that only emerges at runtime due to complex interactions between the involved systems and their environment. Monitoring the behavior of SoS is thus very challenging since existing approaches are often limited to particular architectural styles or technologies and are thus hard to apply in SoS architectures [20].

Data Collection

Most IT infrastructure monitoring solutions focus on an agent-based or agent-less solutions. The differences in both approaches are described in the following:

- **Agent-based Monitoring:** In an agent-based monitoring solution, a software component (agent) is installed or deployed on a monitored node with the primary purpose of collecting information and pushing it over the network to a central location. Agents have more capabilities than agentless monitoring solutions and enables access to deeper levels of root-cause analysis and trouble shooting. However, agents need additional resources and may stress the monitored node.
- **Agent-less Monitoring:** In the agent-less monitoring approach, data is obtained from applications or network devices without installing any additional software. Instead, the monitoring solution use various protocols to gather the monitoring data such as SNMP, WMI, HTTP, POP, FTP, etc. or leverages the application programming interface (API) provided by the applications which are already installed on the monitored node. In particular network traffic monitoring and analysis of log files can be performed without the installation of agents.

Although agent-based monitoring provides a more accurate way to analyze the IT system, there are situations where agents are not the best approach, especially in scenarios where too many agents are already running in the system, and a further installation would introduce risk and performance issues into the system. In this scenario, agent-less auditing can provide more value.

The following points address further perspectives which emphasize how data can be collected from the several probes and how they impact the monitored system:

- **Intrusiveness:** Monitoring solutions which are highly intrusive require significant modification to the application. They can be regarded as agent-based solutions, however their agents needs to be integrated directly into the programming code, hence a simple installation on the application layer is not sufficient. For instance, most monitoring tools which observe the user behavior on website require the introduction of JavaScript snippets directly in the header section. This approach is followed, inter alia, by Google Analytics. Most monitoring approaches prevent intrusive agents as maintaining low intrusiveness allows to minimize the instrumentation costs.
- **Forensic:** This mode of gathering data refers to analyze frequently various log or trace files produced by the observed infrastructure. The logs will hereby scanned for known text patterns and rules that indicate important events and failures. Forensic monitoring represents a push based approach of observing data as it depends heavily on the outputs the infrastructure delivers to the probe. Hence, the quality of the event analysis depends on the diversity of the log file content. However, one

advantage of forensic monitoring is the low intrusiveness as it can be managed without agents which leads to very low instrumentation costs.

- **Polling:** The polling principle describes a pull approach for gathering data and refers to actively sampling the status of the observed infrastructure or application. During the polling loop, the IT infrastructure is polled via network protocols like SNMP calls, accessed via SSH to execute scripts or dump files or via execution of other application specific commands. The advantage of this mode is that there is little impact on the infrastructure/application being polled since the host resources like CPU utilization are needed only during the polling. The rest of the time the monitoring application is not stressing the infrastructure where the probe is located. However, status changes or raised events can only be recognized as soon as the polling is performed by the monitoring process. In addition, if polling takes too long important events or defects may not be recognized in time.

Analytics

Data gathering and data analysis are two fundamental elements of monitoring systems. First, monitoring collects and tracks desired hardware and software metrics. Afterwards, analysis evaluates these metrics to identify events, system or application states and frequently occurred patterns for troubleshooting, resource provisioning, or other management actions. Different analytical aspects have to be taken into account encompassing the timeliness and interrogation of monitoring solutions and which metrics they are using. In this section, we define and motivate such analytical aspects and describe the advantages and issues arising from them.

Timeliness

The timeliness of monitoring systems describes the time-dependent perspective of observed data. This perspective can be separated into two parts: First, a monitoring system is timely as soon as it is able to provide information at the time that users need to access it [21]. This refers to supply the user with the current status or changes in real-time, mostly prepared in an user-comprehensible visualization. Without this attribute, queries like "which machines have CPU utilization above 90%?" would never be possible, which would render the monitoring solution useless. Therefore most of the monitoring tools which are focusing on analytical aspects are timely in this perspective.

This will lead to the second perspective that is addressed in this taxonomy which draws attention to the analytical capability of monitoring approaches like data mining, machine learning or techniques that assist in automatic problem diagnosis and root-cause analysis. Hence, while monitoring has been shown feasible at scale and in real-time, analysis is typically performed after a volume of monitoring data has been collected. The following attributes can be identified:

- *Historical analysis* of monitoring data provides insight about the past behavior of the observed systems. The data history will be materialized in the database in order to track back the status changes in anytime. Historical analysis is not always possible in every monitoring scenario. For instance, storage of observed streaming data would exhaust memory consumption of the system sources very fast.
- *Real-time analytics* address the challenge to capture, aggregate and incrementally analyze data on demand and in real-time. Certain events which indicates crucial changes in the system status or behavior will be instantly processed and forwarded to the administrator. It has to be taken into account that real-time monitoring solutions may not necessarily safe the data history. They are only interested in the current status of observed infrastructure.
- *Predictive monitoring* is capable to predict future behaviors by analyzing historical data and leveraging machine learning algorithms. The historical data is in particular required for finding patterns, anomalies or correlations between monitoring data. Afterwards the obtained findings can be used to predict future behavior of the monitored systems as soon as certain events occur that will have an impact on the system's behavior. It is not always required to store the data history in order to find patterns in the dataset or to predict future behaviors. For instance, a bulk of unsupervised learning algorithms [22] [23] [24] have been developed which can be used to cluster streaming data efficiently without the need to keep the data history in memory. Hence, this will not lead to the problem of exhausting memory consumption as described in the first point.

Interrogation

The way how the IT infrastructure is monitored can be abstracted into four categories [2]:

- *Active monitoring* refers to the ongoing "interrogation" of a system in order to analyze and determine its current status and to predict future behaviors. This approach is resource-intensive and is usually reserved to proactively monitoring the availability of critical systems or attempting to resolve an incident or diagnosing a problem.
- *Passive monitoring* is more common and is addressing issues in the system by analyzing historical log data. The main difference to active monitoring is that the passive approach shows how the system handles existing conditions, but provides less insights into how the system will deal with future events.
- *Reactive monitoring* reacts to certain type of events or failures and triggers particular actions. For instance,

server performance degradation may trigger a reboot, or a system failure will generate an incident. Quality of service monitoring for example is always reactive as it determines the normal status of a system and triggers an event as soon as these conditions are not met anymore.

- *Proactive monitoring* is designed to detect patterns of events which indicate that a system or service is about to fail. Proactive monitoring is always analyzing historical or streaming data in order to create patterns which determine on the one hand the normal condition of a system and on the other hand anomalies which have been detected previously.

It has to be mentioned that reactive and proactive monitoring could be active or passive. For instance, in a proactive - passive scenario event records are correlated over time to build trends for Proactive Problem Management. Reactive - active scenarios are used to diagnose which system is causing the failure and under what conditions, e.g. "ping" a device, or run and track a sample transaction through a series of devices.

Multi-Layer Monitoring

In this Section, we present the scope, methodology and results of our systematic literature review on the topic of multi-layer monitoring. One goal of the sub-project monitoring is to look at the monitoring domain from an EAM perspective, in particular to combine monitoring data from different layers in a meaningful way. Therefore, we now choose a standard EAM model with three layers: *Business*, *Application* and *Infrastructure* layer plus one additional *User* layer. The intention of this work is to identify prototypical implementations and approaches for monitoring that collect or even combine data from at least two or more of the EAM + User layers. We follow the methodology for systematic literature reviews proposed by Kitchenham [25] and limit this literature review to computer science and business information systems research. In the following, we first describe the main steps of the systematic literature review and subsequently the results.

First, we searched for different keyword combinations in academic literature portals ACM Digital Library and Scopus. We did not identify unique keywords for research on monitoring from the described EAM perspective. From the keyword analysis in Section 2 we conclude that e.g. the term *process* is not restrictive enough, while combinations shown in Table 2 yield feasible result sets. In addition to that, the combinations of *layer*, *level*, *cross* and *multi* yield very relevant results. In combination with a subsequent analysis of the references we hope to identify a large portion of the relevant literature on the topic. We performed a match-all search on the keywords in document titles only. Additionally, we restrict the literature research to a time period starting from the year 2000 to present.

The 161 publications returned from the keywords were manually inspected. We dismissed those that contained certain

keywords in the title from which it can be safely assumed that they fall into other research areas such as water monitoring or healthcare monitoring. If the title alone was not distinctive enough, we tried to make a decision based on the abstract. In cases that the abstract was not expressive enough, too, we selected the publication for further detailed inspection. Some publications were not available in full-text and some publications emerged from several keyword combinations and have been counted only once. This pre-selection process resulted in 57 documents that were investigated further.

In a first round of manual inspection the first three authors tried to answer the following questions about each publication:

1. Which EAM + User layers are covered by the approach described in the publication?
2. If applicable: What is the meaning of the term multi/cross-layer/level as used in the publication?
3. Is there a prototypical implementation or tool and is it available for researchers?
4. If identifiable: What is the architecture of the system that is monitored by the approach?
5. Which references are relevant for further investigation?

Subsequently, we concentrated our effort on ten publications that cover at least two EAM + User layers and also have a prototypical implementation, regardless if it is publicly available or not. We also included one of our own publications in the comparison. In Tables 3 and 4 we summarize our findings about the identified prototypes with respect to implementation and (predictive) analytics capabilities. In the following we also describe the main idea of each of the identified approaches as well as technical and otherwise relevant details.

- **CLAMS, Alhamazani et al. e2014, [26]:** The Cross-Layer Multi-Cloud Application Monitoring-as-a-Service Framework (CLAMS) presents a novel approach for the collection of monitoring data for applications running across multiple cloud providers, e.g. Amazon AWS and Microsoft Azure. It monitors Quality of Service (QoS) parameters stemming potentially from all common cloud layers (Software-as-a-Service, SaaS; Platform-as-a-Service, PaaS and Infrastructure-as-a-Service, IaaS), in order to monitor the quality of the applications running in the cloud. The proof of concept prototype collects hardware metrics, e.g. CPU workload via SIGAR², as well as network metrics via SNMP. All data is stored in local monitoring managers in their own databases for each cloud provider. A super-manager application (running also in a cloud environment) discovers monitoring managers via (selective) broad-casting or decentralized discovery mechanisms.

²<http://hyperic.com/products/sigar>

Keyword	Found	Relevant	Duplicates	Inaccessible	1. round	2. round
monitoring multi level	33	11	0	1	10	2
monitoring multi layer	26	8	0	1	7	3
monitoring cross level	3	0	0	0	0	0
monitoring cross layer	30	17	2	1	14	1
monitoring systems-of-systems	19	11	1	0	10	3
monitoring enterprise architecture	6	4	0	2	2	0
monitoring application infrastructure	28	7	0	1	6	0
monitoring infrastructure business	1	1	0	0	1	0
monitoring application business	15	11	2	7	6	0
crossreferences						1
total	161	70	5	13	56	10

Table 2. Publications found on the scientific publication platforms ACM Digital Library and Scopus using our selection of keywords for the document titles. *For Scopus the search was restricted to the Computer Science Subject Area.

It collects the complete data from the monitoring managers via possibly multiple communication methods: publish/subscribe, client/server, web services or SNMP; also push or pull strategies are possible. The super managers stores the data in a MySQL³ database. Moreover, monitoring managers and super-managers can build a hierarchy with multiple levels. The user interacts via a console with the application. However, the approach does not implement any data aggregation mechanisms and does not cover business process metrics or user experience metrics.

- **ECMAF, Zeginis et al. 2013, [27], [28], [29]:** Event-Based Cross-Layer Service Monitoring and Adaptation Framework (ECMAF) is an event-based approach for the monitoring and adaption of cross-layer services. It also is the name of a prototypical implementation reifying the approach. One key idea of the approach is to automatically derive a dependency model documenting static and dynamic dependencies among the entities of the different SOA/Cloud layers infrastructure and services as well as a business process management layer. The dependencies are captured using an Web Ontology Language (OWL) dialect called OWL-Q, [30]. The dependency model in combination with the ability to detect event patterns and the possibility to define solution strategies empowers the framework to automatically resolve violations of defined KPIs. Similar to the CLAMS framework of Alhamazani et al. the ECMAF framework has been extended to support multi-cloud setups. In this case so-called monitoring engines reside in the clouds of the different cloud providers and the so-called adaption engine collects data from the individual monitoring engines via the Siena⁴ event bus. In contrast to CLAMS, ECMAF transmits only events data to the adaption engine. In particular, the frame-

work collects data from the infrastructure layer using Nagios⁵ and the Astro [31] framework. It also collects data using the monitoring functionalities offered by the cloud providers. The monitoring data is stored in OpenTSDB⁶, a relational database optimized for time series data. The authors also provide a taxonomy of events divided into functional and non-functional for each layer as well as an elaborate event model covering events stemming from all layers covered by ECMAF as well as EAM. As far as we know, the ECMAF framework comes without any visualization for the user.

- **ECoWare, Baresi et al. 2013, (Guinea et al. 2011), [32], [33]:** The approach focuses on the event based monitoring of KPIs from multiple layers in SOA and Cloud architectures. A special feature of the corresponding prototypical ECoWare framework is that it supports a so-called "Multi-layer Collection and Constraint Language" (mlCCL) that enables the framework user to specify the collection, aggregation and analysis of the monitored events. Moreover, it is possible to correlate violations from one layer with behaviors at other layers. Therefore, all events are stored in a format called Service Data Objects (SDOs). This standardized object format contains properties about events as well as one or more captured values and hence enables the aggregation of data from multiple events. The SDOs can reference other SDOs, which allows for complex compositions of monitoring events. It is possible to specify the aggregation points within multiple SDOs and to define actions conditioned on SDO values. The implemented framework Event Correlation Middleware (ECoWare) has agents that extract probes from within services (via Aspect Oriented Programming, AOP) and also collects data from the infrastructure layer using

³<https://www.mysql.com>

⁴<http://www.inf.usi.ch/carzaniga/siena>

⁵<https://www.nagios.org>

⁶<http://opentsdb.net>

the collectd⁷ tool. The collected data enveloped by SDOs are communicated over the network to a central application via the Siena P/S bus. The prototype collects application and hardware layer metrics. It uses Apache's Commons Net implementation of the NTP protocol to synchronize clocks on distributed resources and the data is visualized in the EcoWare Dashboard. This user frontend is a Java application that supports on- and off-line charting of time series data. However, data older than 24 hours are deleted. Moreover, the EcoWare framework has been combined with the Cross-Layer Adaptation Manager (CLAM) [34] approach to enable the adaption of services based upon the detection of problems whose phenomena occur at different layers.

- **MLAC, Landthaler et al. 2016, [35]:** The MLAC approach, implemented in a minimal viable prototype, attempts to combine events from arbitrary sources from all EAM layers in order to support root cause analysis. Therefore, the information contained in EA databases is used to find already known dependencies among the entities where the events stem from. Subsequently, anomalies detected in monitoring data or events encoding operational activities, e.g. version deployments or marketing activities are correlated, if they appear at the same time and a dependency among the entities is documented in the EA. Therefore, the EA needs to be represented as a graph. The minimal viable prototype is programmed in Java and stores the event data in a HSQLDB⁸. The approach has been evaluated using an artificial webshop example. A particular focus of the evaluation is the applicability of the BIRCH algorithm for the detection of level shift changes in time series data. The static EA is visualized using a graph visualization library for Java and detected possible correlations are dynamically indicated within this graph.
- **Monalytics, Kutare et al. 2010, [36]:** The main focus of the approach presented by Kutare et al., also reflected by its name, is the combination of monitoring and analytics, in particular for the use cases of large-scale data center systems and cloud infrastructures with use cases being the detection of runtime component misbehavior and performance aware load balancing. Therefore, a hierarchically organized structure (implemented through computational communication graphs) of so-called monitoring brokers report pre-analyzed and aggregated monitoring data to the Monalytics platform. Monitoring brokers can aggregate monitoring data from all of their children (agents and monitoring brokers) by recursively aggregating relevant data from their children. The aggregation covers the grouping of different data types as well as the aggregation of ranges of values of the same type, e.g. mean values. Hence, a special fea-

ture of this approach is the so-called data-local analysis of monitoring data, i.e. data analysis close to the data source. The agents potentially monitor applications, operating system, hypervisor and hardware metrics as well as physical sensors. Though not ultimately clear to us, we guess that application metrics are not collected so far, i.e. merely infrastructure metrics are taken into account. The system also deals with the special challenges of different monitoring rates and dynamic changes of the applications and virtual machines. The monitoring brokers can raise notifications and pass raw data to higher level monitoring brokers and the monalytics platform, too. Moreover, the approach incorporates discovery mechanisms such that the monitoring brokers establish a hierarchy automatically. The events are communicated using the EVPath⁹ eventing system. Monalytics is implemented in C/C++. There seems to be no visualization component developed yet.

- **ReMinds, Vierhauser et al. 2014, 2015, [37], [20], [38], [39], [40]:** The ReMinds framework is a fully elaborated tool suite for systems-of-systems monitoring. It is currently applied to an industrial use case in the area of metallurgical plants. Heterogeneous systems composed of different programming languages and architectural styles are instrumented, events are collected in a centralized component and subsequently processed using a complex event processing engine. Therefore, industrial processes control, optimization and production planning systems are instrumented using agents that extract probes from applications (Java, C++, scripting languages supported), analyze log files or intercept application-to-application communication. User interactions are captured, too. Event data is collected via an event broker mechanism using a JSON exchange format and stored in a relational database (Oracle, MySQL) or in distributed file systems. The ReMinds framework uses an event model that allows for the abstraction of different types of events, event aggregation and event relationships definitions. Moreover, a domain specific constraint checking language has been developed for the purpose of analyzing the collected event data against violations. More specifically, the constraint checking language allows the specification of temporal, structural and data constraints. Finally, five user frontends have been developed to provide the industry experts easy ways of interaction with the ReMinds toolsuite: console, Eclipse RCP, .NET Viewer, web client and constraint editor. The core of the ReMinds framework has been developed in Java and uses a bunch of different libraries (e.g. JMS, RMI, Hibernate and Apache Active MQ). An extension to Cloud systems is planned.

⁷<https://collectd.org>

⁸<http://hsqldb.org/>

⁹<http://www.cc.gatech.edu/systems/projects/EVPath>

- **SOA4All, Mos et al. 2009, [41], [42], [43], [44]:** The EU research project SOA4All developed a SOA service delivery platform with a special focus on the scalability of the platform. Within this context a multi-layer monitoring solution (Analysis Platform) has been developed. The Analysis Platform consists of different components, including already existing solutions, e.g. for semantic business process analysis by the SENTINEL framework. The Analysis Platform communicates with the rest of the SOA4All components and systems via a Distributed Service Bus (DSB). A Monitoring Mediator component in the Analysis Platform receives events from the DSB. This is the main entry point of information to the Analysis Platform. However, parts of the Analysis Platform also draw data from other sources bypassing the Monitoring Mediator. The Monitoring Mediator also stores the data in an Analysis Warehouse. A Basic Event Processor pre-processes events and aggregates basic events for further processing in the so-called K-analytics component of the Analysis Platform. The SENTINEL component draws data from the Monitoring Mediator, but also draws data bypassing the Monitoring Mediator. The Monitoring UI retrieves data from the K-analytics component as well as the SENTINEL framework. Similar to other approaches, this approach includes several complex event models, including different event types for infrastructure, application/service, business and user activity events that cover various metrics, e.g. average response time of services, number of users per service, available bandwidth, DSB statistics, current availability of a service, to name a few of different levels and scopes. The framework includes many visualizations including those of the integrated existing tools, e.g. PEtALS (distributed service bus monitoring), IC2D (infrastructure of distributed and grid systems monitoring) and the SENTINEL framework comes also with dedicated user interfaces. They are configurable in the sense that dashboards can be composed of different widgets separately for each user. The widgets cover process lists, process event lists, charts presenting the results of SPARQL <https://www.w3.org/TR/rdf-sparql-query/> queries, a graph-based overview of semantically related services, raw messages lists, service lists, performance charts and alert lists besides others. However, the information collected at different layers seems to be visualized in separate widgets, i.e. there are no combined visualizations.
- **Song et al. 2010, 2013, [45], [46], [47]:** Song et al. argue that monitoring is usually seen as a cross-cutting concern and most approaches are based on external, centralized solutions. The authors further argue that this violates a key idea of layered systems: the separation of concerns. They propose the only decentralized monitoring approach we have identified so far. It uses a model-driven engineering (MDE) approach: First, constraints and adaption strategies for constraint violations need to be defined a priori. Next, during runtime, for each layer meta-models are build repeatedly. For the illustrating use case presented by Song et al., the layers encompass layers for the infrastructure, services and business processes. On constraint violations within one layer, the adaption strategies are applied and the resulting model changes are propagated to all other layers using a synchronization engine that is based on a bidirectional model transformation method. Hence, other layers can use adaption strategies to respond to adaptations in an originating layer. E.g. a server breakdown can be resolved by a pre-defined strategy that starts another server on another node. The models cover concepts, concepts' properties, and relationships among the concepts. For the services and business process layers, runtime EMF-models are build from XML configuration files. For the infrastructure layer, relevant KPIs are polled through server APIs and configuration files. For this approach the monitoring data is not persisted, but conflict events enter a message queue, which is processed during runtime. The approach has been implemented in Java for a crisis management application and an extension of the approach to Cloud environments is planned, too.
- **SoSMaRT, Hershey et al. 2007, 2009, 2010, [48], [49], [50], [51]:** One major contribution by Hershey et al. is the so-called SOA Monitoring Reference Architecture (SOARMA). Hershey et al. present a four-layered reference architecture with layers for: governance, business, application and infrastructure monitoring domains. Orthogonal to that, four planes have been identified that cover additional perspectives: user, system signaling, security and management/operations. Also, a large bunch of possible target metrics for service-based systems is presented. Additionally, a multi-layer monitoring approach has been proposed and according to the authors has also been implemented. The approach provides different monitoring views for each monitoring layer of the SOARMA. However, to us it remains unclear, to which degree the application is interactive for the user. The approach has been explained for the specific use case of denial-of-service attacks on military communication networks. Here, network metrics are captured via SNMP and remote agents as well as user experience metrics, in particular a VoIP quality metric. Unusual metric values (events) can then be seen in different layers and planes of the approach, which facilitates root cause analysis. The implementation is described as a system-of-systems architecture, however, we were unable to extract more details of the implementation.
- **CEP4CMA, Mdhaffar et al. 2014, [52], [53]:** Complex Event Processing for Cloud Monitoring and Anal-

ysis (CEP4CMA) is an approach to efficiently and rapidly identify the root causes of problems in a Cloud setup. The approach collects performance metrics from all three common Cloud layers: IaaS, PaaS and SaaS. However, based on theoretical and experimental investigations, relationships among metrics across and within layers and also across and within metric categories, e.g. CPU or memory, have been identified a priori. In particular, the correlation among metrics has been calculated on test data and metrics with typically high correlation have been reduced to one of these metrics. Moreover, the relationships among the metrics have been explored manually for causal relationships. From these considerations rules have been derived and implemented in a complex event processing engine. Hence, less metrics need to be captured and due to the application of rules, common root causes for problems can be identified automatically. From an implementation point of view, Monitoring Agents collect metrics from all Cloud layers and send the data to a central Analysis Agent that performs the complex event processing. The Monitoring Agents use different existing monitoring tools, e.g. Xenmon [54], Ganglia [55], IoStat¹⁰ and MpStat¹¹ as well as tools developed by the authors, e.g. AOP4CSM [56] and a so-called JVMSensor. With the complete monitoring data communication from the Monitoring Agents to the central Analysis Agent being a potential performance bottleneck, the approach has also been extended to support the orchestration of multiple CEP4CMA instances, called D-CEP4CMA [53].

- **SSC, Shone et al. 2013, [57]:** Shone et al. present a framework to detect security threats in systems-of-systems architectures using monitoring data including hardware and application metrics. Therefore, individual components of system-of-systems are assessed individually. Thus, parts of the System-of-Systems Composition (SSC) Monitoring Framework, in particular the so-called Monitoring Daemons implemented in C, need to be installed on all component systems. Within a 10-day training period, the system identifies the normal behavior of roughly 100 different metrics, including e.g. CPU load, file system metrics, configuration files and service requests. Also, the learned thresholds are adapted when new data enters the monitoring system. The approach implements complex algorithms to identify behaviors that are considered as abnormal. It identifies ranges of values considered as normal for each metric and e.g. also recognizes the frequencies of quitting these ranges. If, within a component, misbehavior is detected, then events are communicated to a central Decision Module. The metrics can be also be correlated across components, because the Decision

Module can access historic data that is collected from the Monitoring Daemons. The correlation among metric behaviors is taken into account, too. Therefore, the Decision Module uses an elaborate scoring mechanism (Most Appropriate Collaborative Component Selection, MACCS). MACCS uses several similarity measures among the metrics. Finally, risk levels (normal, low, high) are calculated for each component. The approach has been applied to an artificial web services setup with a DoS attack simulating abnormal behavior. The Analysis Agent is developed in Java and there is no frontend or visualization of the results.

In summary, there is quite some research present in the area of multi-layer monitoring. In the last decade, SOA was enhanced with Cloud solutions and these layered approaches call for multi-layer monitoring solutions. However, there are multiple proposals for layering IT systems and this was also reflected in the publications we read: We encountered a multitude of different definitions of layers/levels in multi-layer/level monitoring ranging from simply functional/non-functional levels to level definitions covering all SOA or EAM layers. From a high level-perspective, SOA, Cloud and SoS solutions map best to the EAM + User layers. Only few approaches combine runtime monitoring with user interaction and user experience monitoring. Also, the vertical range of prototypes varies a lot. While certain approaches focus on particular parts of monitoring, e.g. the data collection or data aggregation, others provide full stack implementations covering all steps from collection, communication, storage, aggregation, analysis of data to their visualization. Moreover, most approaches use a centralized, external component that serves as a storage, processing, analysis and visualization unit. Only Song et al. apply a decentralized monitoring approach.

Monitoring Tools

In this Section we provide a basic overview of potential tools that could be used as part of a monitoring solution for an open mobility platform. We focus here on tools that are already popular in industry. We also look at monitoring tools from different domains and different layers, in particular infrastructure, network, business process/activity, web and user experience/behavior monitoring. Moreover, some of these tools, e.g. the ELK stack recently gained a lot of attention in the research community. Likewise, dynatrace ruxit, Netuitive and SOASTA are trending in enterprise contexts. The main intention is to give a short overview on these tools, rather than to provide comprehensive comparisons, which have already been conducted, e.g. for Nagios, Cacti, collectd, IBM Tivoli, Ganglia, Amazon Cloud Watch, Microsoft Azure Watch, Monitis, RevealCloud, OpenNebula, CloudHarmony, LogicMonitor, and related tools by Alhamazani et al. [58] and Fatema et al. [59].

¹⁰<http://linux.die.net/man/1/iostat>

¹¹<http://linux.die.net/man/1/mpstat>

Name	Layers				Target Arch.	Data Collection				Data Storage	Impl. Lang.	User Frontend	P
	I	A	B	U		A	I	F	P				
CLAMS	●	●	○	○	Cloud	?	●	○	○	MySQL/RDB	Java	Console	○
ECMAF	●	●	●	○	Cloud	●	○	○	○	OpenTSDB/RDB	Java**	○	○
ECoWare	●	●	○	○	Cloud	●	●	○	●	DB	Java**	Desktop	●
MLAC	○	●	●	○	○	○	○	○	○	HSQLDB	Java	Desktop	○
Monalytics	●	●	○	○	(Cloud)	●	○	○	○	?	C/C++	○	○
ReMinds	●	●	○	●	SoS / (Cloud)	●	●	●		DB/HDFS	Java**	Multiple*	-
SOA4All	●	●	○	○	SOA	?	?	?	?	HSQL/MySQL	Java	?	●
Song et al.	●	●	●	○	SOA / (Cloud)	○	○	●	○	○	Java	?	○
SoSMaRT	●	○	○	●	SOA	●	?	?	●	?	?	Desktop	○
CEP4CMA	●	●	○	○	Cloud	○	○	○	○	?	Java**	○	○
SSC	●	●	○	○	SoS	○	○	○	○	SQLite	C**	○	○

Table 3. Implementation oriented overview of identified prototypical implementations from the systematic literature review: ○ not used, ○ Planned, ● used, ? marks that we were unable to extract the information. The layers correspond to **I (Infrastructure)**, **A (Application)**, **B (Business)**, **U (User)**. For the data collection dimension we separate **A (agents used)** for all layers, and the characteristics **I (Intrusive)**, i.e. sending data from within an application, **F (Forensic: configuration and log files)** and **P (Polling)** for the application layer. In the **P** column, we record, if a prototypical implementation is publicly available. *For the ReMinds Framework see textual description of the framework. **Multiple tools are used, potentially using multiple implementation languages.

Name	Timeliness			Interrogation				Metrics			
	Historical	Real-Time	Predictive	AT	PS	RA	PA	CT	NW	BN	UC
CLAMS	○	●	○	○	●	●	○	●	●	○	○
ECMAF	●	●	●	●	○	●	●	●	●	●	○
ECoWare	●	●	●	●	○	●	○	●	○	○	○
MLAC	○	●	○	○	●	●	●	●	○	●	○
Monalytics	●	●	●	●	○	●	●	●	○	○	○
ReMinds	○	●	○	○	●	●	○	●	○	●	●
SOA4All	●	●	●	○	●	●	●	●	●	●	●
Song et al.	●	●	●	●	○	○	●	●	○	●	○
SoSMaRT	○	●	○	○	●	●	●	●	●	●	●
CEP4CMA	○	●	○	●	○	●	○	●	●	○	○
SSC	○	●	○	●	○	●	●	●	○	○	○

Table 4. Analytics oriented overview of identified prototypical implementations from the systematic literature review: ○ not used, ○ Planned, ● used. All dimensions directly correspond to our taxonomy presented in Section 3. **AT:** Active, **PS:** Passive, **RA:** Reactive, **PA:** Proactive, **CT:** Computational, **NW:** Network, **BN:** Business, **UC:** User-centric

- Nagios¹²: Nagios is a well known open-source monitoring tool. It collects various monitoring KPIs and provides visualization using charts through a web-based frontend. The tool itself is rather slim, however it can be extended using plugins and a large amount of different plugins has been developed so far. The plugins enhance Nagios to support different alerting mechanisms, failure detection methods and of course a plethora of agents, mostly for infrastructure and software KPIs. It also supports alerting on server breakdowns or threshold exceeding. The monitoring data is stored in a relational database, by default MySQL. Nagios has been developed for *nix systems, however there are workarounds such that it can also be used for collecting monitoring data in a MS Windows environment.
- Cacti¹³: Cacti is another popular open-source tool, mainly written in C. The main focus is the visual presentation and exploration of time series data via a web-based frontend. It also provides an extension infrastructure. In contrast to Nagios, it is a frontend to RRDtool. RRDtool stores data in a round-robin fashion, i.e. a database allocates a pre-defined amount of space (corresponding to a fixed number of monitoring events) and once the database is full, newly entering data overwrites existing data. Additionally, Cacti uses a MySQL database for the storage of its configuration.
- collectd¹⁴: The collectd tool is an open-source monitoring data collection tool in particular for server performance and network monitoring. It runs as a daemon under *nix environments. For collectd, there exists a large number of plugins. Therefore, it supports the monitoring data collection of a multitude of KPIs and applications similar to Nagios. In contrast to Nagios, it does not write the monitoring data to RRD files and does not support the visualization or exploration of the collected data. However, it can be combined e.g. with Cacti as a frontend.
- The so-called ELK stack, consisting of Elasticsearch¹⁵, Logstash¹⁶ and Kibana¹⁷ is an open source analysis and discovery solution developed by Elastic. It allows near real-time analysis in big data environments. Elasticsearch is a search-based discovery tool that typically serves as the underlying technology for applications with complex search functionalities, i.e. online stores or wikis. The data is collected via Logstash, a solution originally intended for log collection but extended with enrichment and transformation features. Kibana is used for visualization of the data, with the option to use area charts, data tables, line chart, KPI, pie charts, tile maps and bar charts. Since it is search-based, the created dashboard needs to query Elasticsearch in defined time intervals. While the ELK stack can be employed for a wide range of purposes, it does not provide ready-to-use functionalities for monitoring. Data collectors for the infrastructure, application or business layer have to be implemented within Elasticsearch.
- The IBM APM solution¹⁸, formerly known as IBM Tivoli, claims to be the best selling solution in Application Performance Management in 2013. It is intended for monitoring at the infrastructure and application layers, both for on-premise and cloud environments or even a hybrid landscape. An agentless data extraction approach is offered for simple and fast implementation, as well as the option to develop custom agents for more detailed data acquisition. The solution features automatic detection and isolation of performance issues and supports the user in identifying trends for a proactive management of the application landscape. There is also support for application detection within the landscape.
- Oracle's Business Application Monitoring (BAM)¹⁹ is available as part of the Oracle SOA or BPM suite and therefore depends on an implementation of Oracle standard software. It comes with predefined data collectors and dashboards that update in real time and is geared towards business analysts with the need to get insights into business processes. The focus in an EAM perspective is strictly on the business layer, while monitoring the infrastructure and application layers underneath is out of the scope of Oracle BAM. One interesting feature is the full integration in the Oracle suite, which allows detailed analysis and advanced features like incremental update of KPIs instead of full database queries. On the other hand, the solution is limited to Oracle applications and is therefore not suitable for monitoring the wide range of applications and processes in an enterprise.
- ExtraHop²⁰ is a real-time streaming and analytics platform that collects its data from network traffic. They follow an agentless approach to gather network data and process it automatically to gain insights about the complete infrastructure and application layers of an enterprise. The applications and interconnections are discovered automatically from transactional data. ExtraHop calls this approach *wire data*. While the monitoring of business processes is not supported directly, the platform is extensible in itself and is able to stream the application data to other BI tools for analysis.

¹²<https://www.nagios.org/>

¹³<http://www.cacti.net/>

¹⁴<https://collectd.org/>

¹⁵<https://www.elastic.co/>

¹⁶<https://www.elastic.co/products/logstash>

¹⁷<https://www.elastic.co/products/kibana>

¹⁸<http://www.ibm.com/middleware/us-en/knowledge/it-service-management/application-performance-management.html>

¹⁹<http://www.oracle.com/technetwork/middleware/bam/overview/index.html>

²⁰<https://www.extrahop.com/>

- Netuitive²¹: Netuitive provides an adaptive monitoring and analytics platform for cloud infrastructures and web applications. The solution investigates metrics from several data sources encompassing operating system, applications, middleware and web browser. Overall Netuitive supports over 65 integrations. The metrics will be used to create patterns in order to describe the infrastructure behavior by implementing machine learning algorithms. As soon as the algorithm has learned the normal behavior of the infrastructure, Netuitive discovers correlations across the metrics and detects relevant anomalies. An interesting aspect is that time series are detected as outliers rather than single values of a time series.
- SOASTA mPulse²²: With mPulse, SOASTA provides a real-time user monitoring solution that tracks user-based performance indicators for technical (page load time, resource timing, navigation timing, and others) and business (page views, bounce rate, conversion rate, revenue made, overall visit, and others) purposes directly from a user browser or mobile application. mPulse requires javascript-based agents which have to be included as tags in the web-application code. The tool also gathers mobile user metrics like user location, device type, carrier speed and application usage. The solution provides a REST API interface that allows users to customize metrics, interact with repository objects and read or write seed data content.
- Dynatrace ruxit²³: Ruxit is a spin-off by Dynatrace and was released 2014. It is a run-time cross-layer monitoring solution which provides insights from the infrastructure layer up to the user interactions with the applications. In addition, this solution also provides mobile application monitoring for iOS and Android systems. The technology requires an agent installation on every system which has to be observed. Ruxit applies an extensive set of machine learning algorithms in order to find patterns, abnormal behaviors, correlations between the observed layers, and predict future behaviors. As soon as a failure was identified ruxit performs root-cause analysis and proposes possible solutions in real-time. Due to this strong focus on data analytics approaches ruxit delivers a monitoring solution which does not only observe the current status of the infrastructure and notifies the administrator about occurred problems, but also provides valuable insights about the acquired data.
- Nimsoft²⁴: The monitoring solution was founded 1998 and acquired by CA Inc. in 2010. It supports multi-

layers monitoring of both virtual and physical cloud resources. Nimsoft provides a holistic view on monitoring resources which are distributed on different cloud infrastructure e.g. a consumer can view resources on Google Apps, Rackspace, Amazon, Salesforce.com and others through a unified monitoring dashboard. In addition, the monitoring solution enables its consumers to monitor both private and public clouds.

- Google (Universal) Analytics²⁵: In contrast to the previous listed tools, Google Analytics is a well known and widely used tool for the analysis of website usage. This encompasses, among other capabilities, website usage, user analysis, marketing KPIs, e.g. conversion rates or customer journey analysis or a technology stack analysis of the users. Google Analytics comes with a free version for smaller websites (less than 10 million hits per month) and a premium version for paying customers. The premium version also has extended capabilities. The main users of Google Analytics are typically stakeholders of marketing activities. In order to optimize websites, e.g. webshops, Google Analytics provides a plethora of analysis capabilities, e.g. a bounce rates analysis for pages in a conversion funnel. From a technical point of view, agents consist of a small Javascript snippet that needs to be included in the website source code (on all pages) and consequently sends all user interactions and data to Google servers. The data collected in Google Analytics can be exported in various data formats, e.g. in CSV or MS Excel formats. Google Analytics has been extended to Google Universal. The major difference of Google Universal to Google Analytics is the cross-device tracking of users.
- Piwik²⁶: Piwik is an open-source alternative to Google Analytics. In contrast to Google Analytics, it can be installed on servers controlled by the operator of a website or as a hosted solution with other operators. The monitoring data is also captured via a Javascript snippet or a one-pixel-sized image that results in a call to the Piwik server. The server software is written in PHP and the frontend is similar to Google Analytics web-based. The full log data is stored in files, which are divided by days and months. All data can be queried from external tools via a REST API.

Visualization of Monitoring Data

The monitoring solutions described in Section 5 employ a number of visualizations to display the data that was obtained and possibly processed. Depending on the type of data, some visualizations might be more suitable to support human understanding of the results and, ultimately, decision making as a response to the observed system behavior. In this section we

²¹<http://www.netuitive.com/>

²²<http://www.soasta.com/performance/>

²³<http://www.dynatrace.com/en/ruxit/>

²⁴<http://www.ca.com/us/products/>

[ca-unified-infrastructure-management.html](http://www.ca.com/us/products/ca-unified-infrastructure-management.html)

²⁵https://www.google.com/intl/de_de/analytics/

²⁶<https://piwik.org/>

want to give an overview of the visualization types most commonly used by industry solutions. All of the examples in this report combine multiple visualization options to sophisticated dashboards that capture data from different data sources.

Most of the visualization techniques are well-known from end-user spreadsheet applications. We describe them in short and organize them by increasing complexity.

- The most straightforward visualization that is utilized is displaying KPIs as a number. This option requires the user to be able to categorize and judge the information. A time dependence is not possible, i.e. the number can only reflect the state of the monitored resource at one specific point in time. Examples can be seen in Figure 4.
- Traffic lights are very common tools for management and are heavily employed in monitoring solutions, e.g. Nagios. If an application is unavailable or a certain threshold value is exceeded, this is signaled through a yellow or red light, indicating a need for action. Oracle BAM (see Figure 5) also uses gauges, where the indicator points in a green, yellow or red area and gives additional information through the position within that area.
- A pie chart consists of a filled circle that is divided in subsections and colored accordingly, as in the lower right of Figure 5. Each subsection represents the number of observations for that particular type relative to the number of observations for all types, i.e. the percentage. A variation of pie charts are sunburst charts (compare Figure 4), torus-shaped figures, where only the sections of the torus are colored. Some visualization libraries also provide the possibility to zoom into sunburst charts and investigate the composition of subsections in yet another sunburst chart. This type does not allow time-dependent visualization. A more scalable alternative to a pie chart is a treemap (compare Figure 6), representing the respective percentages with the size of rectangles instead of fixed portions of a circle.
- Some of the tools, for example Kibana (Figure 4) and Oracle BAM (Figure 5) make use of differently sized bubbles to visualize an amount depending on two-dimensional influence factors. It can be used to show results on a map, for example the number of events or sales revenues in different areas of operation.
- Bar charts are made up of columns whose length or height corresponds to the cumulative number of observations of an event type (in case the observed matter are distinguishable events, e.g. access to different functionalities) or in a fixed time period when showing a history of accesses to one specific functionality. An extension of bar charts are box plots, where ranges of values can be displayed and some statistical properties can be included in the graphic. As an example, it could show the

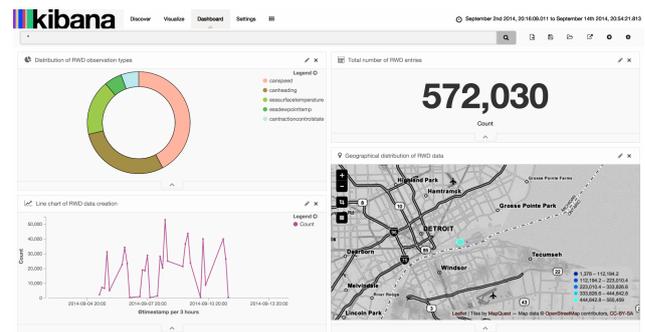


Figure 4. Dashboard with sunburst, KPI, time series and map visualization in Kibana, [60]

observed quantiles for response latencies. Examples for both variations can be found in Figure 7.

- A time series plot draws a continuous line as time progresses and is therefore particularly suitable to visualize high-frequency data, which can be seen in Figures 8 and 9.
- One very advanced feature from *ruxit* is the discovery and visualization of application landscapes, Figure 10. It recognizes physical and virtual hosts, application servers and applications, and the interaction among these components, and displays them in a graph. An interesting question that is open for investigation is how well this approach scales in a large-organization context. The application also features a visualization of consecutive events over different systems or applications contributing to an error message with the goal to identify the root cause of a problem. *Nagios XI* (Figure 11) also features a visualization of a network graph, but it was not possible to obtain further information on whether the tool is able to detect entities automatically.

Conclusion

Within the framework set by the TUM LLCM project, this work presents our results for the assessment of the current state of the art on the topic multi-level monitoring and visualization. We first pointed out relevant and neighboring research areas for monitoring in computer science in general. These results were used to compile a taxonomy for monitoring, where we defined a framework for classification of monitoring solutions. Due to the scope of this work, we categorize solutions by what is monitored and which of the EAM layers are covered. We identified implementation types for the target architecture and data collection methods and listed analytics approaches with respect to time, interrogation and metrics.

Within this framework, we conducted a systematic literature review focusing on implementations for multi-level monitoring approaches, with layers corresponding to the common

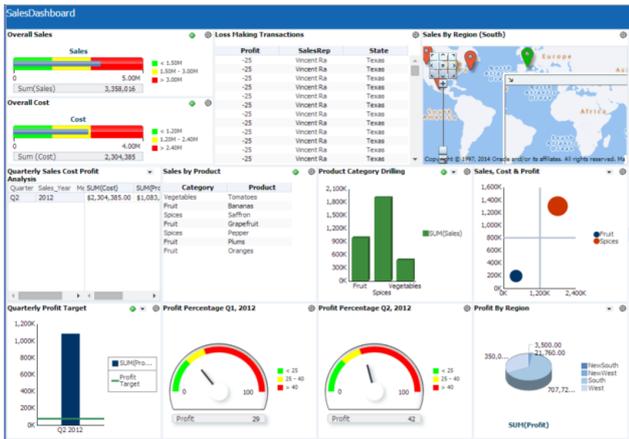


Figure 5. Oracle BAM dashboard with bar charts, map visualization, bubble chart, pie charts and gauges, <http://www.oracle.com/technetwork/middleware/bam/learnmore/dashboard-visualizations-2295973.pdf>



Figure 6. Oracle BAM visualization of a treemap, <http://www.oracle.com/technetwork/middleware/bam/learnmore/dashboard-visualizations-2295973.pdf>

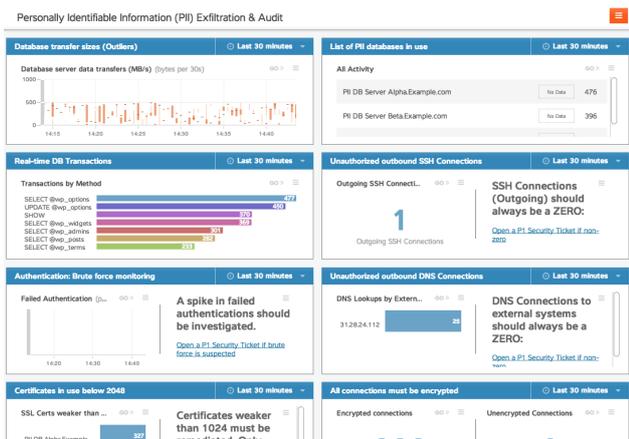


Figure 7. Dashboard with box plots and bar charts in Extrahop, <https://assets.extrahop.com/images/productui/PII%20Exfiltration%20and%20Audit.png>

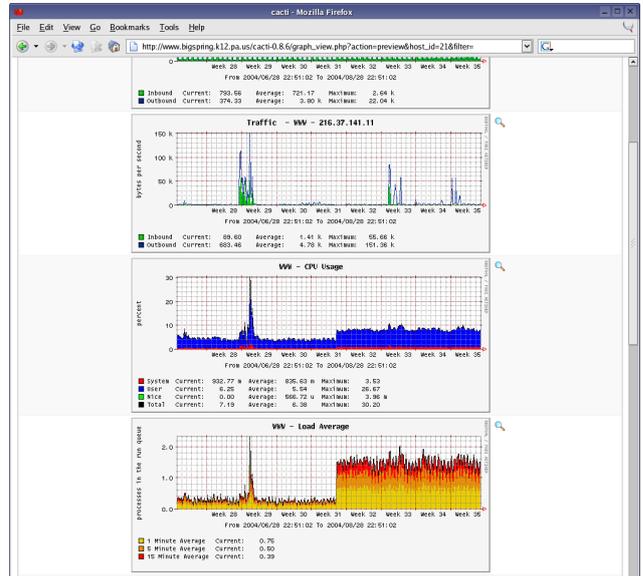


Figure 8. Time series plot from the visualization tool Cacti, http://www.cacti.net/get_image.php?image_id=40&x=1039&y=1107&quality=90



Figure 9. Multi-layer dashboard in ruxit, <https://blog.ruxit.com/wp-content/uploads/2014/09/image02.png>

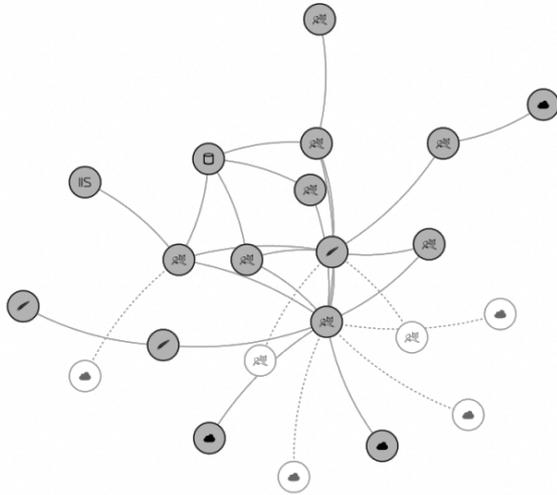


Figure 10. Application landscape graph in *ruxit*, <https://blog.ruxit.com/wp-content/uploads/2014/09/smartscape.png>

EAM layers plus an additional User layer. Eleven prototypical approaches, conceptual or with a working prototype, were identified. They were analyzed in detail in Section 4. Further, we briefly surveyed existing industry solutions commonly applied in monitoring and gave an overview and examples of different methods used for visualization of monitoring data.

Our major findings are that a tremendous amount of research is conducted within each of the layers we considered. Also, there is an active research area on the topic of multi-layer monitoring, in particular focusing on the monitoring of SOA and Cloud architectures. Monitoring is usually considered as a cross-cutting concern and the pre dominant approach is to build a central component or system that encompasses all monitoring concerns. This centralized component is external to the system to be monitored.

From a high-level perspective, we can conclude that most multi-level monitoring research is approached from SOA and Cloud point of views, but not from EAM perspectives. For EAM, it would be beneficial to have a correct overview (amount, type, etc.) of all entities in an IT landscape, also throughout all layers. Likewise, monitoring could benefit from EA information, e.g. to improve the identification of intra- and inter-layer links among entities of the EAM layers or for the visualization of an IT landscape, even IT landscape visualizations enhanced with live monitoring capabilities.

Glossary

- **Instrumentation:** Instrumentation in this work describes the process of monitoring a target system, i.e. receiving metrics about the system.
- **Probe:** While there are different definitions for probes, we use the definition of Mansouri-Samani and Sloman [61]: A probe is a component that is used to extract or intercept information from a target system.
- **SNMP:** SNMP is a network protocol that allows the monitoring of devices that are connected to a computer network. Hardware or software agents are installed or integrated in a device that are able to determine the status of the device.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] Van Haren. *TOGAF Version 9.1*. Van Haren Publishing, 10th edition, 2011.
- [2] Cabinet Office. *ITIL Service Operation 2011 Edition*. The Stationery Office, Norwich, 2011.

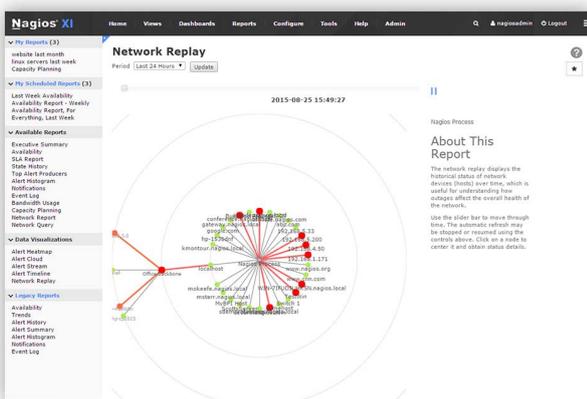


Figure 11. Network graph in *Nagios XI*, https://www.nagios.com/wp-content/uploads/2016/02/Advanced_Infrastructure_Management_Capabilities_Drop.jpg

- [3] Giuseppe Aceto, Alessio Botta, Walter De Donato, and Antonio Pescapè. Cloud monitoring: A survey. *Computer Networks*, 57(9):2093–2115, 2013.
- [4] W M P Van Der Aalst, A H M Hofstede, and M Weske. Business Process Management: A Survey. *Business Process Management*, pages 1–12, 2003.
- [5] Rolf Isermann. Process fault detection based on modeling and estimation methods-A survey. *Automatica*, 20(4):387–404, 1984.
- [6] William Albert and Thomas Tullis. *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes, 2013.
- [7] Ian Foster and Carl Kesselman. *The Grid 2: Blueprint for a new computing infrastructure*. Elsevier, 2003.
- [8] Serafeim Zanikolas and Rizos Sakellariou. A taxonomy of grid monitoring systems. *Future Generation Computer Systems*, 21(1):163–188, 2005.
- [9] Andreas Hanemann, Athanassios Liakopoulos, Maurizio Molina, and D. Martin Swamy. A study on network performance metrics and their composition. *Campus-Wide Information Systems*, 23(4):268–282, 2006.
- [10] Alessio Botta, Antonio Pescapè, and Giorgio Ventre. Quality of service statistics over heterogeneous networks: Analysis and applications. *European Journal of Operational Research*, 191(3):1075–1088, 2008.
- [11] Massimo Bernaschi, Antonio Pescapè, Napoli Federico, Filippo Cacace, Bio-medico Roma, and Stefano Za. Analysis and Experimentation over Heterogeneous Wireless Networks. *First IEEE International Conference on Testbeds and Research Infrastructures for the Development of NeTworks and CoMMunities (TRIDENT-COM'05)*, pages 182–191, 2005.
- [12] Slawek Ligus. *Effective Monitoring and Alerting: For Web Operations*. O'Reilly Media, 2012.
- [13] Mandar Sahasrabudhe, Meenakshi Panwar, and Sagar Chaudhari. Application performance monitoring and prediction. *2013 IEEE International Conference on Signal Processing, Computing and Control (ISPCC)*, pages 1–6, 2013.
- [14] David W. McCoy. Business Activity Monitoring: Calm Before the Storm. Technical report, Gartner, April 2002.
- [15] Matthias Winkler, Jorge Cardoso, and Gregor Scheithauer. Challenges of business service monitoring in the internet of services. In *Proceedings of iiWAS*, pages 613–616. ACM, 2008.
- [16] Ghulam Ali, Noor A. Shaikh, and Zubair A. Shaikh. Agent-based user-profiling model for behavior monitoring. *2009 International Conference on Future Networks, ICFN 2009*, pages 3–7, 2009.
- [17] Katsunori Oyama, Atsushi Takeuchi, Hua Ming, and Carl K. Chang. A concept lattice for recognition of user problems in real user monitoring. *18th Asia-Pacific Software Engineering Conference*, pages 163–170, 2011.
- [18] Yiduo Mei, Ling Liu, Xing Pu, and Sankaran Sivathanu. Performance measurements and analysis of network I/O applications in virtualized cloud. *2010 IEEE 3rd International Conference on Cloud Computing*, pages 59–66, 2010.
- [19] F. Z. Safy, M. El-Ramly, and A. Salah. Runtime monitoring of soa applications: Importance, implementations and challenges. In *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*, pages 315–319, March 2013.
- [20] Michael Vierhauser, Rick Rabiser, Paul Grunbacher, Christian Danner, Stefan Wallner, and Helmut Zeisel. A flexible framework for runtime monitoring of system-of-systems architectures. *Working IEEE/IFIP Conference on Software Architecture 2014, WICSA 2014*, pages 57–66, 2014.
- [21] Chengwei Wang, Karsten Schwan, Vanish Talwar, Greg Eisenhauer, Liting Hu, and Matthew Wolf. A flexible architecture integrating monitoring and analytics for managing large-scale data centers. In *Proceedings of the 8th ACM International Conference on Autonomic Computing, ICAC '11*, pages 141–150, New York, NY, USA, 2011. ACM.
- [22] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, SIGMOD '96*, pages 103–114, New York, NY, USA, 1996. ACM.
- [23] L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani. Streaming-data algorithms for high-quality clustering. In *Proceedings of the 18th International Conference on Data Engineering, ICDE '02*, pages 685–694, Washington, DC, USA, 2002. IEEE Computer Society.
- [24] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams: Theory and practice. *IEEE Trans. on Knowl. and Data Eng.*, 15(3):515–528, March 2003.
- [25] B. Kitchenham. Procedures for performing systematic reviews. Technical report, Keele University and NICTA, 2004.
- [26] Khalid Alhamazani, Rajiv Ranjan, Karan Mitra, Prem Prakash Jayaraman, Zhiqiang Huang, Lizhe Wang, and Fethi Rabhi. Clams: Cross-layer multi-cloud application monitoring-as-a-service framework. In *Proceedings of the 2014 IEEE International Conference on Services Computing, SCC '14*, pages 283–290, Washington, DC, USA, 2014. IEEE Computer Society.

- [27] Chrysostomos Zeginis, Kyriakos Kritikos, Panagiotis Garefalakis, Konstantina Konsolaki, Kostas Magoutis, and Dimitris Plexousakis. *Service-Oriented and Cloud Computing: Second European Conference, ESOC 2013, Málaga, Spain, September 11-13, 2013. Proceedings*, chapter Towards Cross-Layer Monitoring of Multi-Cloud Service-Based Applications, pages 188–195. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [28] Chrysostomos Zeginis, Konstantina Konsolaki, Kyriakos Kritikos, and Dimitris Plexousakis. *Service-Oriented Computing - ICSOC 2011 Workshops: ICSOC 2011, International Workshops WESOA, NFPSLAM-SOC, and Satellite Events, Paphos, Cyprus, December 5-8, 2011. Revised Selected Papers*, chapter ECMAF: An Event-Based Cross-Layer Service Monitoring and Adaptation Framework, pages 147–161. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [29] Chrysostomos Zeginis, Konstantina Konsolaki, Kyriakos Kritikos, and Dimitris Plexousakis. *Web Information Systems Engineering - WISE 2012: 13th International Conference, Paphos, Cyprus, November 28-30, 2012. Proceedings*, chapter Towards Proactive Cross-Layer Service Adaptation, pages 704–711. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [30] K. Kritikos and D. Plexousakis. Semantic qos metric matching. In *2006 European Conference on Web Services (ECOWS'06)*, pages 265–274, Dec 2006.
- [31] F. Barbon, P. Traverso, M. Pistore, and M. Trainotti. Runtime monitoring of instances and classes of web service compositions. In *2006 IEEE International Conference on Web Services (ICWS'06)*, pages 63–71, Sept 2006.
- [32] Luciano Baresi and Sam Guinea. Event-based multi-level service monitoring. In *ICWS*, pages 83–90. IEEE Computer Society, 2013.
- [33] Sam Guinea, Gabor Kecskemeti, Annapaola Marconi, and Branimir Wetzstein. *Service-Oriented Computing: 9th International Conference, ICSOC 2011, Paphos, Cyprus, December 5-8, 2011 Proceedings*, chapter Multi-layered Monitoring and Adaptation, pages 359–373. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [34] Asli Zengin, Annapaola Marconi, and Marco Pistore. Clam: Cross-layer adaptation manager for service-based applications. In *Proceedings of the International Workshop on Quality Assurance for Service-Based Applications, QASBA '11*, pages 21–27, New York, NY, USA, 2011. ACM.
- [35] Jörg Landthaler, Martin Kleehaus, and Florian Matthes. Multi-level event and anomaly correlation based on enterprise architecture information. In *Proceedings of the 12th International Workshop on Enterprise & Organizational Modeling and Simulation, EOMAS '16*, Ljubljana, Slovenia, 2016. Springer LNBP.
- [36] Mahendra Kutare, Greg Eisenhauer, Chengwei Wang, Karsten Schwan, Vanish Talwar, and Matthew Wolf. Monalytics: Online monitoring and analytics for managing large scale data centers. In *Proceedings of the 7th International Conference on Autonomic Computing, ICAC '10*, pages 141–150, New York, NY, USA, 2010. ACM.
- [37] Michael Vierhauser, Rick Rabiser, Paul Grünbacher, Klaus Seyerlehner, Stefan Wallner, and Helmut Zeisel. Reminds: A flexible runtime monitoring framework for systems of systems. *Journal of Systems and Software*, 2015.
- [38] M. Vierhauser, R. Rabiser, P. Grünbacher, and A. Egyed. Developing a dsl-based approach for event-based monitoring of systems of systems: Experiences and lessons learned (e). In *Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conference on*, pages 715–725, Nov 2015.
- [39] Michael Vierhauser. A requirements monitoring infrastructure for systems of systems. In *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering, ASE '14*, pages 887–890, New York, NY, USA, 2014. ACM.
- [40] M. Vierhauser, R. Rabiser, P. Grünbacher, and B. Aumayr. A requirements monitoring model for systems of systems. In *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, pages 96–105, Aug 2015.
- [41] Adrian Mos, Carlos Pedrinaci, Guillermo Alvaro Rey, José Manuel Gómez, Dong Liu, Guillaume Vaudaux-Ruth, and Samuel Quaireau. Multi-level monitoring and analysis of web-scale service based applications. In *Service-Oriented Computing. IC-SOC/ServiceWave 2009 Workshops - International Workshops, ICSOC/ServiceWave 2009, Stockholm, Sweden, November 23-27, 2009, Revised Selected Papers*, pages 269–282, 2009.
- [42] Adrian Mos, Carlos Pedrinaci, Guillermo Álvaro Rey, Iván Martínez, Christophe Hamerling, Guillaume Vaudaux-Ruth, Dong Liu, and Samuel Quaireau. Soa4all analysis platform d2.3.2 service monitoring and management tool suite first prototype: Prototype documentation. Technical Report 215219, INRIA, France, September 2009.
- [43] Adrian Mos, Carlos Pedrinaci, Guillermo Álvaro Rey, Iván Martínez, Christophe Hamerling, Dong Liu, Samuel Quaireau, and Fy Ravoajanahary. Soa4all analysis platform d2.3.3 service monitoring and management tool suite second prototype: Prototype documentation. Technical Report 215219, INRIA, France, September 2009.
- [44] Adrian Mos, Carlos Pedrinaci, Guillermo Álvaro Rey, Iván Martínez, Christophe Hamerling, Dong Liu, Samuel Quaireau, and Fy Ravoajanahary. Soa4all analysis platform d2.3.3 service monitoring and management tool

suite second prototype: Prototype documentation. Technical Report 215219, INRIA, France, September 2009.

- [45] Hui Song, Amit Raj, Saeed Hajebi, Siobhán Clarke, and Aidan Clarke. Model driven engineering of cross-layer monitoring and adaptation. In *MODELSWARD 2013 - Proceedings of the 1st International Conference on Model-Driven Engineering and Software Development, Barcelona, Spain, 19 - 21 February, 2013*, pages 331–340, 2013.
- [46] Hui Song, Gang Huang, Yingfei Xiong, Franck Chauvel, Yanchun Sun, and Hong Mei. *Model Driven Engineering Languages and Systems: 13th International Conference, MODELS 2010, Oslo, Norway, October 3-8, 2010, Proceedings, Part II*, chapter Inferring Meta-models for Runtime System Data from the Clients of Management APIs, pages 168–182. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [47] Hui Song, Yingfei Xiong, Franck Chauvel, Gang Huang, Zhenjiang Hu, and Hong Mei. *Models in Software Engineering: Workshops and Symposia at MODELS 2009, Denver, CO, USA, October 4-9, 2009, Reports and Revised Selected Papers*, chapter Generating Synchronization Engines between Running Systems and Their Model-Based Views, pages 140–154. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [48] P. Hershey and C. B. Silio. Systems of systems approach for monitoring and response across net-centric enterprise systems. In *Systems Conference, 2010 4th Annual IEEE*, pages 1–6, April 2010.
- [49] P. Hershey and D. Runyon. Soa monitoring for enterprise computing systems. In *Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International*, pages 443–443, Oct 2007.
- [50] P. Hershey and C. B. Silio. Systems engineering approach for event monitoring and analysis in high speed enterprise communications systems. In *Systems Conference, 2009 3rd Annual IEEE*, pages 344–349, March 2009.
- [51] P. C. Hershey, J. M. Pitts, and R. Ogilvie. Monitoring real-time applications events in net-centric enterprise systems to ensure high quality of experience. In *MILCOM 2009 - 2009 IEEE Military Communications Conference*, pages 1–7, Oct 2009.
- [52] Afef Mdhaffar, Riadh Ben Halima, Mohamed Jmaiel, and Bernd Freisleben. *Networked Systems: Second International Conference, NETYS 2014, Marrakech, Morocco, May 15-17, 2014. Revised Selected Papers*, chapter CEP4CMA: Multi-layer Cloud Performance Monitoring and Analysis via Complex Event Processing, pages 138–152. Springer International Publishing, Cham, 2014.
- [53] Afef Mdhaffar, Riadh Ben Halima, Mohamed Jmaiel, and Bernd Freisleben. D-cep4cma: a dynamic architecture for cloud performance monitoring and analysis via complex event processing. *IJBDI*, 1(1/2):89–102, 2014.
- [54] Diwaker Gupta, Rob Gardner, and Ludmila Cherkasova. Xenmon: Qos monitoring and performance profiling tool. Technical report, Hewlett-Packard Laboratories, 2005.
- [55] Matthew L Massie, Brent N Chun, and David E Culler. The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing*, 30(7):817 – 840, 2004.
- [56] A. Mdhaffar, R. B. Halima, E. Juhnke, M. Jmaiel, and B. Freisleben. Aop4csm: An aspect-oriented programming approach for cloud service monitoring. In *Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on*, pages 363–370, Aug 2011.
- [57] N. Shone, Qi Shi, M. Merabti, and K. Kifayat. Misbehaviour monitoring on system-of-systems components. In *2013 International Conference on Risks and Security of Internet and Systems (CRiSIS)*, pages 1–6, Oct 2013.
- [58] Khalid Alhamazani, Rajiv Ranjan, Karan Mitra, Fethi Rabhi, Prem Prakash Jayaraman, Samee Ullah Khan, Adnene Guabtani, and Vasudha Bhatnagar. An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art. *Computing*, 97(4):357–377, 2015.
- [59] Kaniz Fatema, Vincent C. Emeakaroha, Philip D. Healy, John P. Morrison, and Theo Lynn. A survey of cloud monitoring tools: Taxonomy, capabilities and objectives. *Journal of Parallel and Distributed Computing*, 74(10):2918 – 2933, 2014.
- [60] Ömer Uludag. Descriptive study and experimental analysis of the elk stack applicability for big data use cases. Master’s thesis, Technische Universität München, Germany, April 2016.
- [61] M. Mansouri-Samani and M. Sloman. Monitoring distributed systems. *IEEE Network*, 7(6):20–30, Nov 1993.

Sensing On Demand

Vittorio Cozzolino and Jörg Ott

Department of Informatics, Technical University of Munich, Munich
{cozzolin, ott}@in.tum.de

Abstract

Cloud computing and Internet-of-Things (IoT) are two diametrically opposed technologies both extensively used for different purposes. The adoption of IoT technologies has lately become more and more pervasive, moreover the convergence between Cloud Computing and IoT has become a hot topic over the last few years. The presence of manifold IoT deployments emphasize the necessity to build a platform able to aggregate data and services offered by different providers. In this paper, we explore existing solutions aimed to exploit the interplay of Cloud Computing technologies and IoT deployments. We present state-of-the art implementations and highlight key concepts and architectural principles. The aim of this paper is to show the concept of sensing-on-demand as a service and provide a better understanding of design challenges of the integration of IoT and Cloud Computing.

Keywords

Internet of Things; Cloud Computing; Fog Computing; Pervasive Computing; Sensors Networks

1. Introduction

Sensing-on-demand is an archetype of service where automated systems as well end-users have the possibility to retrieve information from manifold IoT deployments through a unified platform. The integration of IoT and Cloud computing is the first step to take to aggregate in a transparent way different services offered by multiple providers. IoT deployments are currently demanded to execute specific tasks and have application-specific network and hardware design. Building a platform to share the underlying network architecture and hardware capabilities belonging to different providers and hosting multiple applications would lead to various potential benefits. For instance, it could increase the utilization of sensing and communication resources, whenever the underlying network infrastructure covers the same geographic area and the sensor nodes monitor the same physical variables of common interest for different applications [3]. We consider virtualization as a fundamental element to leverage the intrinsic differences across various IoT deployments and, generally, sensors networks.

Virtualization allows to hide the inner complexity of each different IoT network and expose only valuable information to the outside. Therefore, our objective is to identify the best possible solution to correctly virtualize, model and exploit assorted sensing resources. Along the integration process, there is the desire to offer a service following and on-demand paradigm. The resource offered by the envisioned platform shall be accessible and usable by multiple users concurrently. Therefore, we need to ensure different requirements in a multi-tenant environment: security, isolation, correct resource allocation and task scheduling.

The goals of this paper is to underline the potential gain of interconnecting IoT technologies or, generally speaking,

any kind of device at the edge of the Internet with Cloud Computing infrastructures. Section II presents an overview of the involved technologies. Section III discusses some key applicative domains. Section IV discusses the state-of-the-art in sensing platforms and the already proposed solutions. Section V focuses on open research issues. Finally, section VI and VII present observation and future perspectives and conclude the paper.

2. Technological Background

The Internet-of-Things represents one of the most disruptive technologies, i.e., technology that will change the way we use IT, extending and empowering existing ubiquitous and pervasive computing scenarios. The IoT is a multidisciplinary domain that covers a large number of topics from purely technical issues (e.g. routing protocol, semantic queries), to a mix of technical and societal issues (e.g., security, privacy, usability) [2]. As smart devices are becoming more pervasive in our daily life, the IoT concept is steadily becoming the next technological revolution by enabling an exchange of data never available before and interconnecting a plethora of different objects.

Before the interest's upsurge in the IoT, wireless sensors networks (WSNs) dominated the sensing ecosystem. WSN research was focused on delivering optimized solutions for resource-constrained devices able to solve specific issues [2] and supported by localized, fine-tuned infrastructures. In contrast, the major goal IoT research is to inter-connect and integrate manifold WSN deployment and build a unified infrastructure. Instead of having a plethora of isolated WSN islands, it aims to lay down bridges to create a WSN mesh network able to answer different demands.

IoT is generally characterized by tiny smart devices,

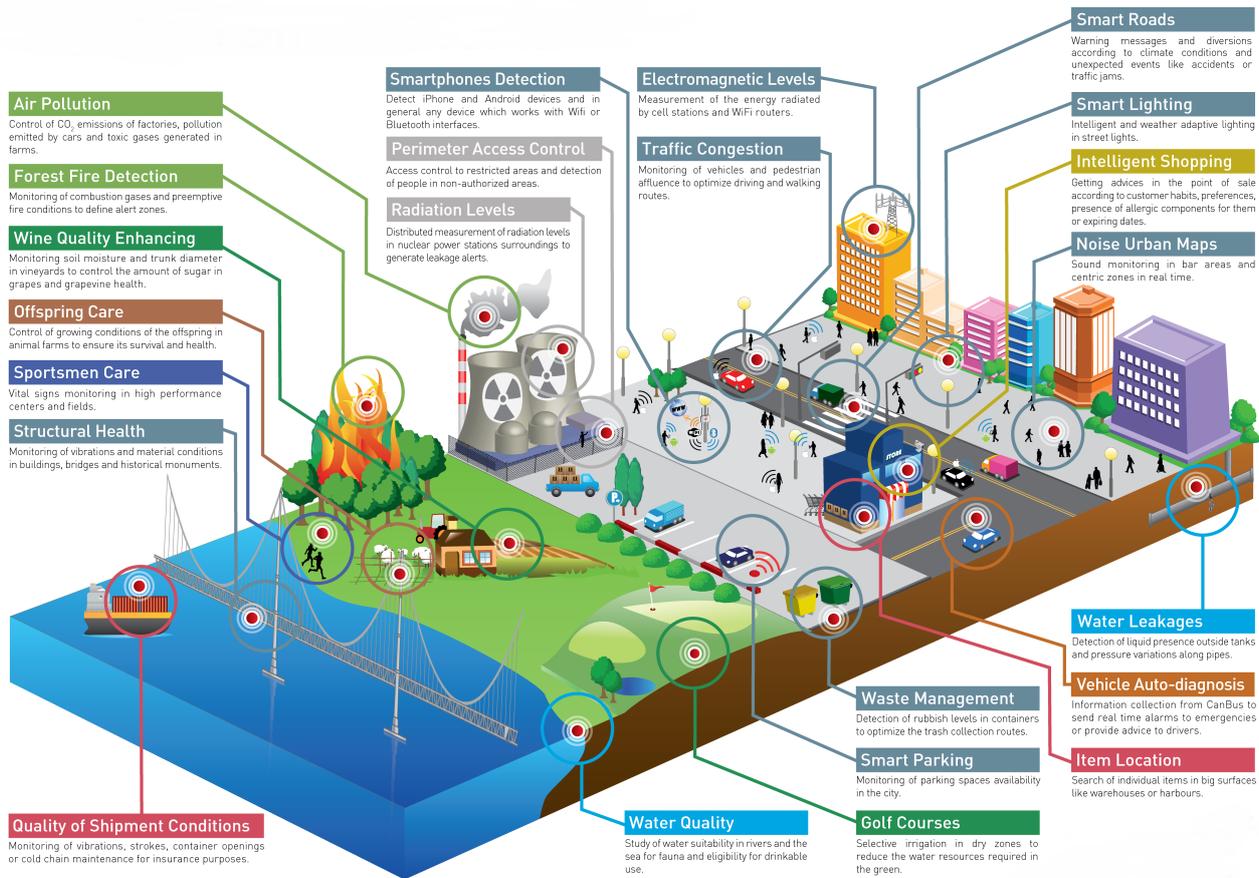


Figure 1. Smart city [1]

widely distributed, with limited storage and processing capacity, which involve concerns regarding reliability, performance, security, and privacy [4]. Furthermore each one of these objects is readable, recognizable, locatable, addressable, and/or controllable via the Internet using different technologies like RFID [5], wireless LAN, wide-area network, or other means. Currently deployed smart systems are semantically separated and generally dedicated to a specific task [3]. Figure 1 shows Smart Cities, highly automated environments aimed to enhance quality, performance and interactivity of urban services while reducing costs and resource consumption. Considering the high density of IoT networks and systems present in a Smart City, the critical mass of diverse services offered is remarkable. Therefore, combining and taking advantage of such a broad range of services is a primary goal to guarantee a better experience for end-users.

Subsequently, we advocate the importance of extending the IoT ecosystem to consumer-centred tools like smartphones, smart cars, smart houses and smart devices in general. This would trigger a shift of paradigm where the IoT becomes the Internet of Everything (IoE). We consider that Cloud Computing technologies play a fundamental role in the process of integration and virtualization of services owned by different

providers.

Cloud computing has virtually unlimited capabilities in terms of storage and processing power, it's a much more mature technology, and has most of the IoT issues at least partially solved. The National Institute of Standard and Technologies (NIST) describes cloud computing as [7]:

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

The architecture of a cloud computing environment can be divided into four layers: the hardware/data-centre layer, the infrastructure layer, the platform layer and the application layer. Cloud computing is well known to provide multiple services [8]. In our specific scenario, we are more concerned about infrastructure as a service (IaaS), which focuses on providing the required abstraction layer to support the provisioning of infrastructure grid. Moreover, cloud computing is an essential part in the process of virtualizing and aggregating services

offered by different providers.

Cloud computing and IoT are therefore two critical technologies for realizing the ubiquitous communications vision. The cloud can provide large-scale and long-lived storage and processing resources for personalized ubiquitous applications delivered through the IoT and it serves as important back-end resources. However, cloud-based platforms are physically far from the real nodes connected to them.

On the other hand, device-centric technologies and applications, such as IoT, constitute part of a local and distributed infrastructure, providing a continuous stream of data generated by sensors and actuators. Large amounts of heterogeneous and personalized data coming from distributed sources (e.g., nodes) have to be handled in a transparent and secure manner [5]. Consequently, a fast and successful deployment will not be possible without proper planning of the necessary resources (e.g., computing power, storage, network band).

3. Applicative Domains

The integration of IoT and Cloud enables a new set of services and applications that encompass both Machine-to-Machine (M2M) and Machine-to-Human (M2H) communications. In this section we shortly describe only a few promising applicative domains enabled by this new paradigm.

3.1 Automotive and Smart Mobility

The advances in cloud computing and IoT have provided a promising opportunity to resolve the challenges caused by the increasing transportation issues. Modern vehicles are increasingly equipped with a large amount of sensors, actuators, and communication devices such as: mobile devices, GPS devices, and embedded computers[9]. Moreover, they can communicate with other vehicles or exchange information with the external environments over various protocols, including HTTP, TCP/IP, SMTP, WAP, and Next Generation Telematics Protocol (NGTP) [4].

The functionalities offered by cloud computing and IoT provide a promising opportunity to further address the increasing transportation issues, such as heavy traffic, congestion, and vehicle safety. In the past few years, different solutions have been proposed to create intelligent transportation systems with the support of cloud computing.

As an example, ITS-Cloud proposes a novel approach to improve vehicle-to-vehicle communication and road safety [10]. Another approach can be found in [11], where a cloud-based urban traffic control system is based on a service-oriented architecture (SOA). Nevertheless, advances in both cloud computing and IoT field, research on integrating IoT with vehicular data clouds is still in its infancy and literature on this topic is highly insufficient [12]. IoT-based vehicular data clouds still have to overcome noticeable challenges before being efficiently used and deployed at large scale: scalability, integration of different technologies, performance, reliability, security, privacy and lack of global standards are

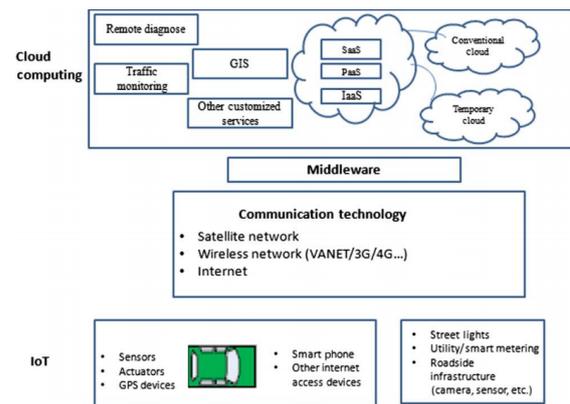


Figure 2. Automotive and Smart Mobility sample architecture [4]

just some of these challenges that open interesting discussions and research opportunities.

In the near future, IoT vehicular data clouds are expected to be the backbone of future ITSs with the ultimate goal of making driving safer, more enjoyable and efficient [12]. Though with many challenges, IoT and cloud computing provide tremendous opportunities for technology innovation in the automotive industry [13], and will serve as enabling infrastructure for developing vehicular data clouds [14].

3.2 Smart Cities, Heterogeneous Sensing Infrastructures

At a holistic level, cities are "systems of systems", and this could stand as the simplest definition for the term. The main features of a smart city include a high degree of integration of information technology and a comprehensive application of information resources. Recently, ICT solutions have become the neuralgic point around which modern economies and the advent of smart cities revolves. Consumer-centric devices at the edge of the Internet such as smart-phones, music players, and in-vehicle sensors are key enablers of mobile sensing and crowd-sensing. These devices will fuel the evolution of the IoT as they feed sensor data to the Internet at a societal scale.

The main challenge of the smart city scenario is to harness the power of different ICT networks (networks of people, of knowledge, of sensors) to create a collective knowledge that empowers citizens, through participation and interaction.[4] By implementing a strong interaction between cloud resources and IoT networks it would be possible to create a common layer acting as a sink for the information flows coming from different heterogeneous sensing infrastructure and able to expose data in a uniform way. A number of frameworks has been proposed, typically consisting of a sensing platform and a cloud back-end storage. Kantarci, *et al.* proposes a framework based on reputation-based crowd-sourcing aimed mostly at public safety [15]. The work of [16][17] present an ecosystem for mobile crowd sensing applications which relies on the Cloud-based publish/subscribe middleware to acquire sensor data from mobile devices in a context aware

and energy efficient manner.

Other proposed solutions suggest to use Cloud architectures as a tool to discover, connect and integrate multiple sensors networks creating a platform that supports ubiquitous connectivity and real-time applications for smart cities [18][19]. Kumar *et al.* proposes the development of a smart city using an intelligent and energy efficient illumination system that would also offer ubiquitous communication [20].

Developing such an infrastructure is not a trivial task, the desire of exploiting a heterogeneous environment comes with solid challenges: security, flexibility, scalability, optimized and efficient sensor resource utilization (processing power, storage, I/O) and sensing task scheduling. Moreover, the lack of a common infrastructure to interconnect and share information between smart cities it's a major scalability drawback that generates operation and regional fragmentation that prevent innovative synergies. [21][22]

3.3 Environmental Monitoring

Environmental monitoring can greatly benefit from the interplay between IoT and cloud computing. The combined use of Cloud and IoT can contribute the deployment of a high speed information system between the entity in charge of monitoring wide-area environments and the sensors/actuators properly deployed in the area. Environmental monitoring is strongly enabled by WSNs that are able to bring to IoT applications noticeable capabilities for both sensing and actuation. Some examples are long-term monitoring of water level (lakes, streams and rivers), gas concentration in air, humidity and temperature, radiation levels, structures integrity (dams, bridges).

However, as mentioned in [23], despite the enormous progress achieved in WSNs, it still remains a major drawback that they are domain-specific and task-oriented, tailored for particular applications with little or no possibility of reusing them for newer applications. Rather than trying to change existing WSN deployments through cloud computing, it's necessary to interconnect different sensing domains and create a shared pool of information accessible by end-users as well as other platforms.

The terms Sensing as a Service (SnaaS) and Sensor Event as a Service (SEaaS) are coined to describe the process of making the sensor data and event of interest available to clients and applications, on the fly and over the cloud infrastructure. The cloud-based data access is able to bridge latency-energy requirements of low power communication segments and the ubiquitous and fast access to data for end users [24]. Moreover, it provides an interface to interact, manage and process complex events generated by sensors.

Sensor cloud is an infrastructure that allows truly pervasive computation using sensors as interface between physical and cyber-worlds, the data-compute cluster as the cyber-backbone and the internet as the communication medium. [26]

The main challenges in this field pertain to infrastruc-

ture scaling, security (information leak, potential breaches, and data corruption), computational resources not sufficient to deal with changing environmental conditions and proper communication protocols.

4. State-Of-The-Art

In this section we present the state-of-the-art in sensing platforms and analysed critically. Our study is focused on analysing existing solutions that proof the feasibility of the interaction between Cloud services and IoT deployments.

The existing work can be categorized in open IoT testbeds/platforms and open IoT software libraries. The first category focuses on real deployments providing both the hardware and software support for the offered services while the latter documents software libraries made available as Open Source that provides reusable components that can be used for developing IoT applications or other enablers.

It is worth mentioning that there are also multiple commercial solutions in both categories but, considering the academic scope of this paper, we won't consider them.

4.1 Open IoT Testbeds and Monitoring Platforms

The following subsection discusses testbeds and monitoring platforms which can be categorized into generic, IoT-oriented and federated. The categorization is concentrated mostly on one dimension: the platform's purpose. Federated testbeds can be seen as infrastructures designed with the purpose of integration of multiple systems to achieve a greater goal. Figure 3 depicts the geographic distribution of platforms we are going to present in the next subsections.

4.1.1 Generic Platforms

Under the umbrella of the generic platform it is possible to find all of those solutions that, in our specific case, are not directly addressing issues related to the IoT world but still worth mentioning. The reasoning behind it is that we are strongly interested into the architectural design more than the scope of the platform. In particular, hereby are going to be listed some of the most interesting and well-known networks measurements platforms.

RIPE Atlas [27] is one of the best examples of platform that ranges from cloud services to physical devices. Ripe Atlas is a global network of probes that measure Internet connectivity and reachability. There are thousands of active probes in the RIPE Atlas network and the network is constantly growing. The RIPE NCC collects the data from this network and provides useful maps and graphs based on the aggregated results. Moreover users can run custom and predefined measurements on the network even though they are just limited to simple tasks (e.g., ping, traceroute). Along the same direction, SamKnows [29] is worth mentioning, which deployed thousands of probes in US and Europe, but as RIPE it only supports limited performance measurements.

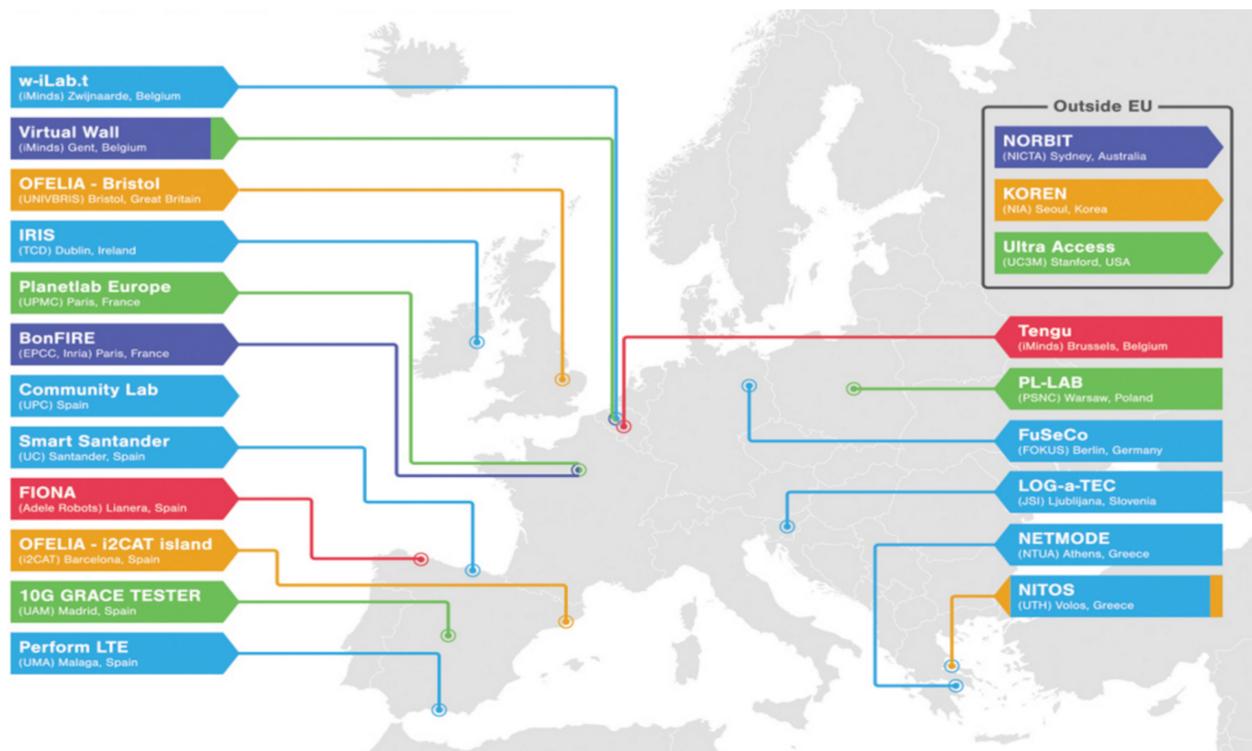


Figure 3. Geographical distribution of Future Internet’s research test-beds [44]

Similarly to RIPE Atlas, the Archipelago (Ark) Measurement Infrastructure [33] is a distributed measurement platform supported by hardware measurement nodes (2nd gen. Raspberry Pi) distributed with as much geographical and topological diversity as possible for a wider view of the global Internet. It also aims to reduce the effort needed to develop and deploy sophisticated large-scale measurements and to provide a step toward a community-oriented measurement infrastructure.

Another example is PlanetLab [30]. PlanetLab is a global overlay network for developing and accessing broad-coverage network services, it allows multiple services to run concurrently and continuously, each in its own slice and currently consists of 1353 nodes at 717 sites. A slice is a horizontal cut of global PlanetLab resources and each service (a set of distributed and cooperating programs delivering some higher-level functionality) runs in a slice of PlanetLab. A slice encompasses some amount of resources across a set of individual PlanetLab nodes distributed over the network and can be seen as a network of virtual machines, with a set of local resources bound to each virtual machine.

A similar platform to PlanetLab is BISmark [31]. BISmark is a deployment of home routers running custom software, and back-end infrastructure to manage experiments and collect measurements. The project began in 2010 as an attempt to better understand the characteristics of broadband access networks and has enabled studies of access link performance, network connectivity, Web page load times, user behaviours and activity. The project faced far more exacer-

bated challenges because it relied on home routers which are a resource-limited, prone to downtimes, devices.

CitySense [25] is a work-in-progress research project for an urban-scale wireless networking testbed. It will consist of 100 Linux-based embedded PCs outfitted with dual 802.11a/b/g radios and various sensors, mounted on buildings and street-lights across the city of Cambridge. The goal of the project is building an urban mesh network directly programmable by end-users providing an experimental apparatus for urban-scale distributed systems and networking research efforts. One interesting aspect of the CitySense is that, compared to other projects, the testbed is an outdoor, permanent installation. Other projects are mostly indoor, laboratory testbeds.

GENI, the Global Environment for Networking Innovation, is one of the biggest distributed virtual laboratories for transformative, at-scale experiments in network science, services, and security. GENI allows users to: obtain and connect compute resources, install custom software control traffic at network switches level and install custom network protocols.

FutureGrid [32] is a project aimed to understand the behaviour and utility of Cloud computing approaches. It provides computing capabilities that allow researcher to tackle complex issues related to authentication, authorization, scheduling, virtualization, middleware design, interface design and cyber-security. The test-bed includes a geographically distributed set of heterogeneous computing systems, supporting virtual machine-based experiments as well as op-

erating systems on native hardware for experiments.

4.1.2 IoT-Oriented Platforms

IoT-oriented platforms are self-explanatory: their main purpose is to study and evaluate the potential gains derived from the integration of clouds infrastructure and IoT ecosystems. Clearly, none of these solutions is comprehensive from the point of view of targeted devices heterogeneity but still are extremely valuable examples to be taken into account when developing a new platform of the same kind.

FIT IoT-Lab [35][34] (previously SENSLAB [36]) is an open access platform part of the "Future Internet of Things" (FIT2) experimental platform, with over 2700 wireless sensor nodes deployed across six sites in France, today IoT-LAB is the largest open low-power wireless remote testbed in the world. A variety of fully programmable wireless sensors are available, with different processor architectures; meaning that a user has a full "bare-metal" access to the nodes. Moreover, some mobile nodes with predefined trajectories are provided to the user in several sites. Each mobile node is embedded on a robot (Turtlebot2 or Wifibot) with advanced functionalities like navigation, obstacle avoidance and automatic docking. IoT-LAB provides full control of network nodes and direct access to the gateways to which nodes are connected, allowing researchers to monitor energy consumption and network-related metrics of nodes. The IoT-Lab is composed of three types of hardware nodes (WSN430, M3, A8) and the architecture is based on 3 layers: open node, gateway, control node.

NITOS [37] [38] is an integrated testbed with heterogeneous features, it focuses on supporting experimentation-based research in the area of wireless networks. The main components of NITOS are: a wireless experimentation testbed consisting of powerful nodes (some of them mobile), a software defined radio (SDR) testbed, a Software Defined Networking (SDN) testbed and a distributed Wireless Sensor Network (WSN) testbed able to sense and gather environmental measurements from agricultural installations. The infrastructure uses of a multiple software libraries and tools to provide resource abstraction and management (NITOS Scheduler), network virtualization, and a framework dedicated to experiment measurements handling (OML).

FuSeCo Playground [39] is a pioneering reference facility, integrating various state of the art wireless broadband networks and helping to coin the vision of a Future Internet in areas like Smart Cities, Automotive, eHealth, eGovernment, Smart Metering and more. It provides core network technologies such as multi-access network environments, M2M and IoT, sensors networks and mobile broadband, and SDN and openflow environments. The testbed addresses large and small scale equipment vendors, network operators, application developers and research groups to testwise deploy and extend their components and applications. It is worth mentioning that there is an ongoing project called FLEXCARE [40], carried on by TU Berlin, aimed to extend the FuSeCo Playground testbed with a new LTE testbed facility.

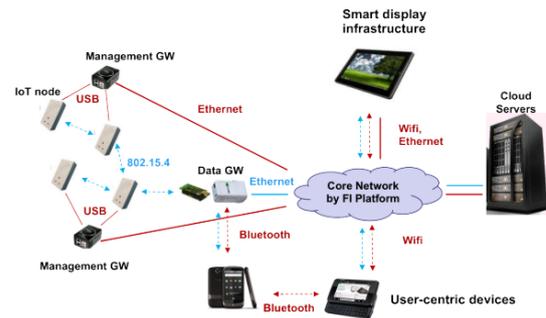


Figure 4. SmartCampus network architecture [64]

The w-iLab.t [41] is an experimental, generic, heterogeneous wireless testbed deployed by iMinds [42]. w-iLab.t provides a permanent testbed for development and testing of wireless applications via an intuitive web-based interface, where registered users can create their own executables, upload these executables, associate those executables with a selection of sensor nodes, and schedule the job to be run on w-iLab. It hosts different types of completely configurable wireless nodes: sensor nodes, WiFi based nodes, sensing platforms, and cognitive radio platforms (that are limited to operating in the ISM bands due to license restrictions.)

SmartCampus [64] is a user-centric testbed for experimental IoT research, which is part of the European SmartSantander experimental facility [63]. It is meant to monitor the behaviour of users inside a real world office. Figure 4 shows the architecture which is composed of three layers : i) a Server tier that hosts all the back-end functionalities, ii) an embedded Gateway (GW) tier which forms the testbed infrastructure and allows the iii) IoT tier to be connected and reachable to a backbone network through WiFi or Ethernet [64]. The testbed dispose of 200 IoT nodes, 100 GWs and 30 Android 4 based Smartphones. The Smartphones can communicate directly with the GWs via Bluetooth or WiFi. The IoT devices provide manifold sensing capabilities: relative amount of light, relative noise level, temperature and motion through a PIR sensor and a vibration sensor to determine tampering with the device.

One minor example is Indriya [43]: a large-scale, low-cost wireless sensor network testbed deployed at the National University of Singapore. The goal of the platform is to understand performance differences and correlations that may exist among different WiFi channels, it is also equipped with different types of sensor boards, thus allowing evaluation of WSN applications.

4.1.3 Federated Platforms

Building a federated testbed entails a supplemental network and software infrastructure over the stand-alone testbed facility [3]. Federated platforms are the first real step towards the creation of an unified platform that harbours a heterogeneous environment. Through the federation of testbeds and platforms addressing different matters and focusing only

on specific community within the Future Internet ecosystem, innovative experiments become possible that break the boundaries of these domains.

Fed4FIRE [44] is an integrating project under the European Union's Seventh Framework Programme (FP7) aimed to the creation of a traversal, open, accessible and reliable platform for the Future Internet Research able to easily harbour different Internet research communities. Fed4FIRE aims to include in its platform roughly 18 different testbeds with heterogeneous scopes and architectures. For example, optical networking, wireless networking, software defined networking, cloud computing, grid computing, smart cities, IoT, etc..

OneLab [45] is a consortium consisting of five different higher education and research institutions and provides access to PlanetLab Europe, NITOS, CorteXlab and IoT-LAB testbeds. It was designed to offer both wireless and fixed-line emulated environments and reproducibility of experimentation in different fields: IoT, wireless systems, cloud service and network virtualization. The access to the federation of testbeds is obtained through MySlice [46], a free OS resource management tool for testbeds. After gaining access to the platform, the Manifold Application Programming Interface (API) handles the distribution of the task across the multiple testbeds available accordingly to the requested resources and tasks.

BonFire [47] is a cloud facility based on an Infrastructure as a Service delivery model adopting a multi-platform federated approach. It comprises 7 geographically distributed testbeds across Europe (EPCC, HP, iMinds, Inria, USTUTT, PSNC and Wellness Telecom (WT)), which offer heterogeneous Cloud resources, including compute, storage and networking. To simplify the interaction between the embedded testbeds, their basic resources offered are exposed using a version of the Open Cloud Computing Interface (OCCI).

Community-Lab [49][48] is a project complementing the Fed4FIRE federated testbed. It's a new facility for experimentally-driven research built on the federation of existing community IP networks, like Guifi.net (Catalonia, Spain), Funk-Feuer (Vienna, Austria) and AWMN (Athens, Greece), constituted by more than 20,000 nodes and 20,000 km of links. The project is characterized by three main factors: large scale testbed with thousand of nodes, integration of existing production networks and the support for long term studies of new network protocols and parallel experimental services by means of node and network virtualization.

FIESTA-IoT (Federated Interoperable Semantic IoT Testbeds and Applications) is a project backed by the European Horizons 2020 Programme. It provides a meta-testbed IoT/cloud infrastructure to enable the submission of experiments over the interconnected/interoperable underlying testbeds. By utilizing a single set of credentials, researchers and engineers will be able to design and execute experiments across a virtualised infrastructure, i.e. access the data and resources from multiple testbeds and IoT platforms using a common approach. FIESTA offers tools i) to design and execute experimental

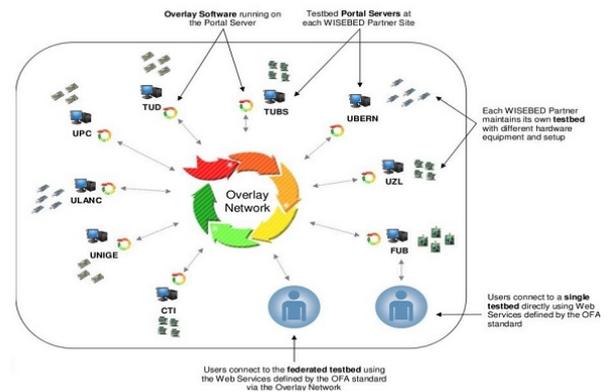


Figure 5. WISEBED Overall Architecture [50]

workflows, ii) dynamically discover IoT resources, and iii) access data in a testbed agnostic manner. The FIESTA-IoT project is supposed to comprise four international testbeds: Smart Santander (Spain), Smart Campus (United Kingdom), KETI (Korea), 4G/LTE/IMS M2M Testbed (France).

WISEBED [50] Experimental Facility (WEF) is currently a federation of independent sensor networks located at 9 locations throughout Europe. Its goal is to establish a large scale pan-European network of WSNs. WEF is composed of 750 motes (iSense, TelosB, MicaZ, Tmote Sky, SunSPOT, Mica2) offering a wide variety of sensing tools, connected with both wired and wireless backbone and covering also outdoor areas. Moreover, the testbed employs mobile sensor node as a combination of Roomba robots plus iSense mote devices. The architecture of the WISEBED system is based on a hierarchy of layers where each layer is comprised of one or more peers: services which are activated by the system as a response to various events. The bottom layer contains the wireless sensor nodes that are running iSense, Contiki, TinyOS devices. A single instance of a virtual testbed server exposes the capabilities and resources offered by the federated testbeds. Resource specification of each single node are presented in human-readable format.

4.2 Open IoT Software Libraries

The following subsection discusses available IoT software libraries offering a framework to enable an interaction between Cloud back-end services and remote devices. Being software-only solutions, they are not device specific and present an adequate abstraction level to be considered as guidelines in the creation of a middleware layer.

Software libraries require the hardware support to materialize the idea of a shared sensing platform, but their role is not marginal. Important concerns are the modelling and designing of the way information are exchanged in the network and of the chain of control from the cloud level all the way down to the IoT devices.

FIWARE [51] is a platform that provides an enhanced OpenStack-based cloud environment plus a rich set of open standard APIs that make it easier to connect to the IoT. Their goal is to ease the development of Smart Applications in multiple vertical sectors. The core element of FIWARE are Generic Enablers (GEs): public and royalty-free packages containing different software stacks based on their purpose. Specifically, IoT GEs have been spread in two different domains: Gateway and Backend. While IoT Gateway GEs provide inter-networking and protocol conversion functionalities between devices and the IoT Backend GEs, the IoT Backend GEs provide management functionalities for the devices and IoT domain-specific support for the applications [52]. The FIWARE ecosystem handles well-known IoT protocol standard (Ultralight2.0/HTTP, MQTT/TCP, LWM2M/CoAP, SIGFox Cloud) and exposes the same data REST API to developers. Moreover, from the hardware perspective, FIWARE is compatible with a plethora of commercial and open hardware platforms (Arduino, Cloudino, Intel Edison, RaspberryPi, etc.). Figure 6 shows the FIWARE structure.

Building the Environment for Things as a Service (BE-TaaS) [53] is a novel, horizontal platform for the deployment and execution of content centric Machine-to-Machine (M2M) applications relying on a local cloud of gateways. Each gateway requires the presence of a running instance of the BETaaS run-time environment, creating a form of a logical overlay. Therefore, the logical federation of networks forms a local cloud in which each gateway shares the functionalities offered by the things of its M2M systems with the rest of the network. At the bottom level, BETaaS integrates heterogeneous M2M systems at the Physical layer into an unified M2M system due to the Adaptation layer [54].

BUTLER [55] is a European Union FP7 project focused on the IoT research. It aims to enable the development of secure and smart life assistant applications due to a context and location aware, pervasive information system. It is composed of four architectural layers: Communication (end-to-end communication infrastructure), Data/Context Management (data models, APIs), System/Device Management and Service Layer (discovery, binding, deployment and provisioning of context-aware services). Moreover, the architecture considers a split into three different entities: BUTLER SmartObject, BUTLER SmartServer and BUTLER SmartMobile (modeling the devices and gateways, the server and the clients, respectively). The most interesting component is the SmartGateway: bridge devices able to offer an homogeneous access to heterogeneous networks of IoT nodes. Each IoT device is mapped as a SmartObject and accesses indirectly by the end-user through a BUTLER SmartService. Therefore, BUTLER Service and Resource model allows exposing the resources (i.e., sensor data, properties, actions on the physical environment, etc.) provided by an individual service [56].

SiteWhere is an open source server application and framework for the Internet of Things. It provides a system that facilitates the ingestion, storage, processing, and integration

of device data. It offers three main services: IoT Server Platform, Device Management, Integration. SiteWhere allows IoT devices to register with the back-end server platform and push data into the database system. It is worth underlying that there is no abstraction layer defined, the bottom layer is supposed to make HTTP REST calls to forward sensors data to the back-end as JSON messages. SiteWhere is designed to be a store-and-process system for the IoT that makes no assumption on the underlying hardware being, at the same time, flexible but also unable to really control the hardware layer. From the components perspective, it takes advantage of different technologies: Apache Tomcat, Spring Framework, MongoDB, Apache HBase, InfluxDB and Apache Spark as analytic engine [57].

Kaa [58] is a highly flexible, hardware-agnostic open-source platform for building, managing, and integrating applications in the Internet of Things. Kaa infrastructure consists of the Kaa server and endpoint SDKs. The Kaa server implements the back-end part of the platform, exposes integration interfaces, and offers administrative capabilities. An endpoint SDK is a library which provides communication, data marshalling, persistence, and other functions available in Kaa for specific type of an endpoint (e.g. Java-based, C++-based, C-based, Objective-C-based). A group of server nodes represent a cluster while endpoints are specific Kaa clients registered in a Kaa deployments. Each client runs an application developed specifically for its platform (Arduino, SMT32 etc.) but implements the same Kaa library to communicate with the server. Kaa offers a set of tools to be integrated with platform-specific software to communicate with a central back-end server. From the hardware perspective, it can be seen as an extension module. Kaa supports different operative systems (Android, iOS, Linux, Windows etc.) as well as hardware platforms (Intel Edison, beaglebone, RaspberryPi etc.) [59].

servIoTicy [60] is an open-source, state-of-the-art platform for hosting Internet of Things (IoT) workloads in the Cloud. It provides multi-tenant data stream processing capabilities, a REST API, data analytic, advanced queries and multi-protocol support in a combination of advanced data-centric services. The architecture of servIoTicy is composed of different elements. The Front-End of platform is a REST API that allows external services to communicate with servIoTicy. The Stream Processing Topology is responsible for the execution of the code associated to Data Processing pipes as well as the forwarding to external entities. Finally, the data backend includes the Data Store and the Indexing Engine providing search capabilities based on different criteria. servIoTicy doesn't provide any specific software to be installed on the hardware devices, it offers a library to be integrated in the IoT application to send the data to the back-end infrastructure [61].

Karibu [66] is an open-source, data collection architecture and reference implementation designed to meet the needs of the urban IoT community. It enables sensors data collection from different sources and reliably stores the data in a scalable

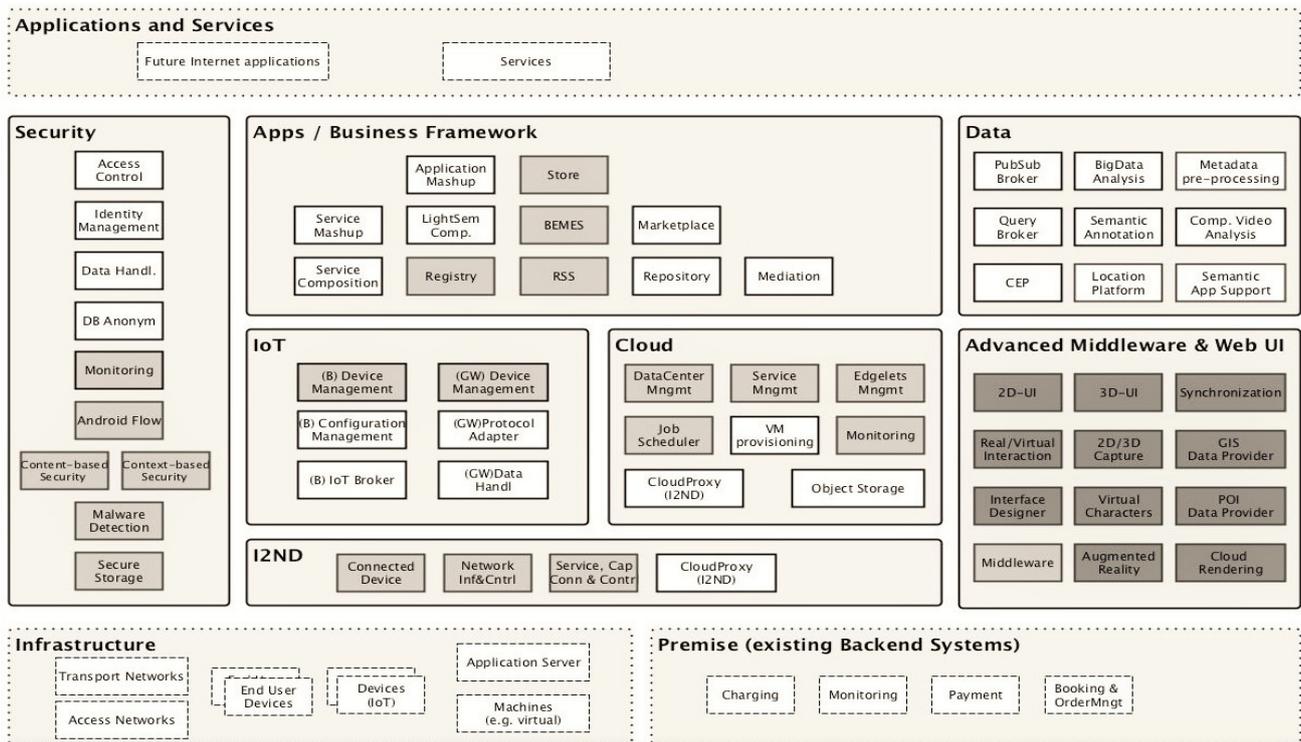


Figure 6. FIWARE City Map [52]

backend system. Karibu aims to ensure four main architectural attributes: data integrity, modifiability, availability, and performance. These quality attributes guided the developer team towards specific design choices. The architecture uses a producer/consumer interaction paradigm. A client library deployed on the data producers (smartphones) collects data and send them to the backend infrastructure using an SSL-encrypted connection. The Karibu daemon continuously fetch messages from the messaging systems, performs minimal processing into a suitable storage format, and finally stores data in a replicated primary database system. Karibu addresses the heterogeneity of the underlying hardware devices by taking advantage of a library to be integrated into device-specific software. Unfortunately, the authors did not specify in which language the library has been developed, leaving unclear the range of compatible devices.

5. Research Issues

Based on the explanatory study of the previous section, we hereby list the main issues related to the process of creating a sensing platform based on IoT and Cloud technologies interplay.

5.1 Scalability

While small-scale testbed ranging from 10s to hundreds of nodes were sufficient to provide meaningful results in WSN experiments [2], IoT experiments focused on multiple networks integration require a greater number of involved de-

vices. As the number of devices grows, the data produced by these devices grow unboundedly. Thus, handling the growth of number of devices and information they produce is a massive challenge in IoT [70]. Moreover, increasing the testbed size leads to new challenges and requires automated fault-recovery, events managements, flexible subscription schema, plug-and-play capabilities and scalable data collection procedures. Guaranteeing such properties in respect of users and things is still an open issue.

5.2 Standardization and Heterogeneity

The success of IoT depends on standardization, which provides interoperability, compatibility, reliability, and effective operations on a global scale [71]. Consequently, the current lack of standards is actually a consistent issue towards the integration of IoT and Cloud technologies. Resorting only to Cloud-side solutions to simplify the standardization effort it's not sufficient and leads to increased network load and data processing delay. IoT Devices are deployed by different persons/authorities/entities. These devices have different operating conditions, functionalities, resolutions, etc [70]. To facilitate the interconnection of smart heterogeneous objects, it's crucial to concentrate efforts toward the creation of an homogeneous set of tools and service APIs able to command and control experiments across heterogeneous testbeds. Another important requirement is to provide a common development framework to simplify the programmability of heterogeneous devices.

5.3 Concurrency and QoS

An on-demand platform has to support multiple concurrent users executing tasks involving an arbitrary set of devices. Realizing an infrastructure supporting the multiplexing of concurrent tasks guaranteeing minimization of tasks interference, nodes' load balancing, task scheduling and optimized allocation is complex and challenging tasks. Task allocation has been well studied for WSNs and right now the existing algorithms are less capable of handling situations where we have multiple sink-nodes (gateways) [3] distributed inside interconnected testbeds. Quality of Service (QoS) requirements in multi-tenant environments is critical since the system is required to handle different applications simultaneously. In addition, we have to consider the possibility of having users with different privileges and demands. This implies the mandatory presence of algorithms able to correctly schedule tasks based on the different priority while trying to keep all the others with an acceptable degree of fulfilment.

5.4 Mobile Devices

Mobile devices play a key role in the IoT ecosystem providing real-world information about entities actually moving inside a smart systems. These devices will fuel the evolution of the IoT as they provide sensor data readings to the Internet at a societal scale [72].

Integrating mobile devices with stationary sensing smart objects is a key requirement for future IoT applications. The great majority of end-users will interact with their surrounding with a mobile device as both consumers and producers of information. Therefore, we would increase noticeable the amount of information gathered by the sensing platform by transforming users into active contributors. This shift of paradigm require the development of suitable software layers to support full-duplex communications backed by data cleansing and validation algorithms.

5.5 Network Layer

The design of architectures and protocols for distributed systems is a key issue for general networked systems and for IoT in particular, given it's fragmented and distributed nature. The network layer is strongly and highly affected by the nodes' cardinality in IoT networks. The massive amount of data streaming from the environment to the Internet is a side effect of the IoT type of scenarios: this means a potentially very large amount of information injected into the network [74].

Another challenge is the creation of multiple flexible virtual networks coexisting at the same time. Each user's task will require different resources covering a specific set of nodes, consequently an overlay network has to be created and maintained until the task is completed. Moreover, the actual resources offered by the nodes in network need to be discovered and published, otherwise tasks cannot be deployed efficiently.

Problems like leader-election, node counting and averages computation are a core topic in the distributed systems literature [75] [76]. In addition, they affect also IoT networks.

Routing poses another interesting challenge considering the unreliability of wireless link and the presence of mobile nodes. Sharing the network between multiple application implies also the necessity to build routing protocols making decisions based on additional, application-level parameters: QoS requirements, nodes' load and location, nodes' capabilities, among others.

5.6 Data Modelling and Data Mining

IoT networks and WSNs encompass manifold different devices and sensors providing information formatted without following a shared, unified data model. In terms of testbeds interconnection and integration, creating a comprehensive data model able to be extended following specific rules and compatible with different systems is fundamental to build a consistent and solid data pool. The presence of an unified data models would unlock the development of generic APIs and interfaces, enormously simplifying the software layer. This way, any application would be able to tap into the aggregated database generated by interconnected testbeds without the need for data conversion algorithms.

Data mining is another consistent issue. Trillion of devices will be connected in the near future and the amount of information generated will be humongous. In a simple supermarket there are 700,000 RFID tags. If the supermarket has readers that scan the items every second, about 12.6 GB RFID data will be produced per second, and the data will reach 544TB per day [73]. Therefore, it is necessary to develop optimized and efficient algorithms to extrapolate valuable information filtering out "garbage data".

5.7 Security and Privacy

Security and privacy are both research challenges receiving increasing attention especially in multi-tenant, shared environments. Security issues have been reported as the biggest concern preventing enterprises and organizations from adopting cloud services, according to recent researches [67]. The presence of IoT technologies adds an additional layer of complexity, introducing also devices as actuators, able to directly affect entities in the real-world. Security represent also the capacity to defend a system from external malicious attacks. The creation of an interconnected network of IoT deployments increases the attack surface exponentially along with the chances of being attacked (malware injection into physical nodes, wireless medium disruption, gateway tampering, data corruption). Nowadays, privacy is an important concern raised by the growing popularity of IoT [68], [69]. Smart devices often collect sensitive user data and an attacker might take direct control on the device or wiretap the communication, stealing such personal data. Providing isolation mechanism and carefully tailoring access grants and policies to sensitive data is still a challenges, especially when data integrity has to be ensured.

6. Observations and Future Perspective

In the process of designing an on-demand sensing services, we consider crucial a new approach where IoT resources are shared. Considering the aforementioned solutions, the IoT federated platforms followed the path of unification and cooperation to reduce overall deployment costs and expand the testbeds capabilities. Cooperation is a key factor in the development of a comprehensive platform but it poses many challenges. Different hardware, testbed scope, customizability degree and software layer are just a few complicated matters to be addressed in the process of fusing heterogeneous testbeds. Nevertheless, the potential gains is not trivial.

After considering the applicative scenarios and the existing solutions in the field, we strongly support virtualization as a key element in the effort of integrating manifold sensing systems and exploit their inter-play. Virtualization helps attenuate IoT deployment differences by providing a solid, secure and flexible abstraction layer. Each sensing installation is wrapped by a black-box that hides its intrinsic complexity. What it's seen from the outside is an interface able to respond to specific commands. By shaping this abstraction layer, we can modify the way systems and users interact and create a flexible, pliable service changing with our needs.

On the other hand, virtualization ensures security and isolation. Tasks performed by different users and/or systems (Machine-to-Machine) will be scheduled, executed and isolated from each-other ensuring a high degree of privacy. Moreover, a virtualized environment is easier to monitor and control. In this scenario, Cloud computing has the central role of coordinator/controller and of interface with the external world. End-users would contact the Cloud infrastructure to book on-demand sensing tasks and interact with a highly abstracted system.

None of the solutions studied in this paper utilize the advantages of virtualization, as we envision them. In most cases the resources abstraction is carried by a software layer (middleware) specifically tailored for a single platform. To enable the sharing of IoT and sensing resources it's important to abandon vertical approaches and move to a horizontal solution. A requirement is also a standardization effort of the IoT devices and data models. IoT devices need to head toward homogenized hardware and software specifications to strongly reduce technological incompatibilities. On the other hand, a unified data model is paramount to exploit the inter-correlation of information generated by multiple IoT deployments and generate added value. When information can be compared and processed following the same steps, the real power of a unified platform can be harnessed. Abolishing conversion procedures and multiple, tedious steps of data cleansing would increase the speed at which raw data is converted into valuable insights.

7. Conclusion

The upcoming capillary diffusion of IoT devices opens exciting possibilities to different applications in various fields. By designing a shared infrastructure, deployment costs would be strongly reduced and also administrative tasks simplified. With the support of Cloud computing, IoT deployments at the edge of the network can converge and create a cooperative sensing layer. Following such an approach would not only maximize the utilization of the available sensing resources and capabilities but also offer to end-users a comprehensive, flexible on-demand service.

The issues to be solved to build such and homogeneous architecture are not trivial and a noticeable amount of work is required before a clear solution can be stated. Our paper present a list of existing solutions that can be seen as reference points in the process of building a sensing infrastructure combining IoT technologies and Cloud computing.

Our final considerations about the discussed scenario are focused on the architecture virtualization. From our perspective, shifting from single-purpose IoT deployments toward general-purpose, flexible ones is the first step toward the mitigation of the aforementioned challenges.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation. Bavaria, an initiative of the Bavarian State Government.

References

- [1] Libelium, *Smart World* <http://www.libelium.com/>
- [2] Gluhak, Alexander, et al., *A survey on facilities for experimental internet of things research*. IEEE Communications Magazine 49.11 (2011): 58-67.
- [3] Farias, Claudio M. De, et al., *A Systematic Review of Shared Sensor Networks*. ACM Computing Surveys (CSUR) 48.4 (2016): 51.
- [4] A. Botta, W. de Donato, V. Persico, A. Pescapé, *Integration of cloud computing and Internet of Things: A survey* Future Generation Computer Systems (2015), <http://dx.doi.org/10.1016/j.future.2015.09.021>
- [5] Suciu, G., Vulpe, A. , Halunga, S. , Fratu, O. , Todoran, G. , Suciu, V. , *Smart Cities Built on Resilient Cloud Computing and Secure Internet of Things* Control Systems and Computer Science (CSCS), 2013 19th International Conference on.
- [6] Jung, Gueyoung, Nathan Gnanasambandam, and Tridib Mukherjee, *Synchronous parallel processing of big-data analytics services to optimize performance in federated clouds* Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on. IEEE, 2012.

- [7] Mell, Peter, and Tim Grance, *The NIST definition of cloud computing*. 2011
- [8] Qi Zhang, Lu Cheng, Raouf Boutaba, *Cloud computing: state-of-the-art and research challenges*. Journal of internet services and applications 1.1 (2010): 7-18.
- [9] He, Wu, Gongjun Yan, and Li Da Xu., *Developing vehicular data cloud services in the IoT environment*. Industrial Informatics, IEEE Transactions on 10.2 (2014): 1587-1595.
- [10] Trivedi, Prashant, Kavita Deshmukh, and Manish Shrivastava., *Cloud Computing for Intelligent Transportation System*. International Journal of Soft Computing and Engineering, IJSCE (2012).
- [11] Jaworski, Paweł, et al., *Cloud computing concept for intelligent transportation systems*. Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on. IEEE, 2011.
- [12] Iwai, Akihito, and Mikio Aoyama., *Automotive cloud service systems based on service-oriented architecture and its evaluation*. Cloud Computing (CLOUD), 2011 IEEE International Conference on. IEEE, 2011.
- [13] Goggin, Gerard., *Driving the internet: mobile internet, cars, and the social*. Future Internet 4.1 (2012): 306-321.
- [14] Gubbi, Jayavardhana, et al., *Internet of Things (IoT): A vision, architectural elements, and future directions*. Future Generation Computer Systems 29.7 (2013): 1645-1660.
- [15] Kantarci, Burak, and Hussein T. Mouftah., *Mobility-aware trustworthy crowdsourcing in cloud-centric Internet of Things*. Computers and Communication (ISCC), 2014 IEEE Symposium on. IEEE, 2014.
- [16] Antonic, Aleksandar, et al., *A mobile crowdsensing ecosystem enabled by a cloud-based publish/subscribe middleware*. Future Internet of Things and Cloud (FiCloud), 2014 International Conference on. IEEE, 2014.
- [17] Podnar Zarko, Ivana, Aleksandar Antonic, and Krešimir Pripužic., *Publish/subscribe middleware for energy-efficient mobile crowdsensing*. Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication. ACM, 2013.
- [18] Petrolo, Riccardo, Valeria Loscrì, and Nathalie Mitton., *Towards a smart city based on cloud of things*. Proceedings of the 2014 ACM international workshop on Wireless and mobile technologies for smart cities. ACM, 2014.
- [19] Mitton, Nathalie, et al., *Combining Cloud and sensors in a smart city environment*. EURASIP journal on Wireless Communications and Networking 2012.1 (2012): 1-10.
- [20] Kumar, Narendra., *Smart and intelligent energy efficient public illumination system with ubiquitous communication for smart city*. Smart Structures and Systems (ICSSS), 2013 IEEE International Conference on. IEEE, 2013.
- [21] Suciuc, George, et al., *Smart cities built on resilient cloud computing and secure internet of things*. Control Systems and Computer Science (CSCS), 2013 19th International Conference on. IEEE, 2013.
- [22] Ballon, Pieter, et al., *Is there a Need for a Cloud Platform for European Smart Cities?* eChallenges e-2011 Conference Proceedings, IIMC International Information Management Corporation. 2011.
- [23] Khan, Imran, et al., *Wireless Sensor Network Virtualization: A Survey*. 2015
- [24] Lazarescu, Mihai T., *Design of a WSN platform for long-term environmental monitoring for IoT applications*. Emerging and Selected Topics in Circuits and Systems, IEEE Journal on 3.1 (2013): 45-54.
- [25] Murty, Rohan Narayana, et al., *Citysense: An urban-scale wireless sensor network and testbed*. Technologies for Homeland Security, 2008 IEEE Conference on. IEEE, 2008.
- [26] Rao, B. B. P., et al., *Cloud computing for Internet of Things and sensing based applications*. Sensing Technology (ICST), 2012 Sixth International Conference on. IEEE, 2012.
- [27] *RIPE Atlas*. Available from: <https://atlas.ripe.net>.
- [28] *Testbed Map* <http://www.fed4fire.eu/wp-content/uploads/2015/10/testbeds-europe-map-3-rgb-1000px.jpg>
- [29] *SamKnows*. Available from: <http://samknows.com/>.
- [30] Chun, Brent, et al., *Planetlab: an overlay testbed for broad-coverage services*. ACM SIGCOMM Computer Communication Review 33.3 (2003): 3-12.
- [31] Sundaresan, Srikanth, et al., *BISmark: a testbed for deploying measurements and applications in broadband access networks*. 014 USENIX Conference on USENIX Annual Technical Conference (USENIX ATC 14). 2014.
- [32] Von Laszewski, Gregor, et al., *Design of the futuregrid experiment management framework*. Gateway Computing Environments Workshop (GCE), 2010. IEEE, 2010.
- [33] *Archipelago (Ark) Measurement Infrastructure*. Available from: <http://www.caida.org/projects/ark/>
- [34] Adjih, Cédric, et al., *FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed*. Proceedings of the 2nd IEEE World Forum on Internet of Things (WF-IoT). 2015.
- [35] *FIT IoT-Lab*, Available from: <https://www.iot-lab.info/what-is-iot-lab/>.
- [36] *sensLAB, Very Large Scale Open Wireless Sensor Network Testbed*. 2010, Available from: <http://www.senslab.info/>.
- [37] Dimitris Giatsios, Apostolos Apostolaras, Thanasis Korakis, and Leandros Tassioulas, *Methodology and Tools for Measurements on Wireless Testbeds: The NITOS Approach*. 2014, Available from: <http://nitlab.inf.uth.gr>.

- [38] NITOS FI, Available from: <http://www.ict-openlab.eu/technologies/testbeds/nitos.html>.
- [39] FuSeCo Playground, Available from: <https://www.fokus.fraunhofer.de/go/en/>.
- [40] FLEXARE, Available from: <http://www.av.tu-berlin.de/research/development/projects/flexcare/>.
- [41] w-iLab.t, Available from: <http://doc.ilabt.iminds.be/ilabt-documentation/wilabfacility.html>.
- [42] iMinds, Available from: <https://www.iminds.be/en>.
- [43] Doddavenkatappa, Manjunath, Mun Choon Chan, and Akkihebbal L. Ananda., *Indriya: A low-cost, 3D wireless sensor network testbed*. Testbeds and Research Infrastructure. Development of Networks and Communities. Springer Berlin Heidelberg, 2012. 302-316.
- [44] FED4FIRE, Available from: <http://www.fed4fire.eu>.
- [45] ONELAB, Available from: <https://onelab.eu/services>.
- [46] MySlice, Available from: <http://myslice.info/info>.
- [47] BonFIRE, Available from: <http://doc.bonfire-project.eu/R4.1/>.
- [48] Community-Lab, Available from: <http://wiki.confine-project.eu/intro:community-lab>.
- [49] Neumann, Axel, et al., *Community-lab: Architecture of a community networking testbed for the future internet*. Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on. IEEE, 2012.
- [50] GChatzigiannakis, Ioannis, et al., *WISEBED: an open large-scale wireless sensor network testbed*. Sensor Applications, Experimentation, and Logistics. Springer Berlin Heidelberg, 2009. 68-87.
- [51] FIWARE, Available from: <https://www.fiware.org/>.
- [52] FIWARE Wiki, Available from: <https://forge.fiware.org/>.
- [53] BEETAS, Available from: <http://www.betaas.eu/>.
- [54] Kyriazakos, S, Anggorojati, B, Prasad, N, *BETaaS platform—a things as a service environment for future M2M marketplaces*. Internet of Things. User-Centric IoT. Springer, 2015. 305-313.
- [55] BUTLER, Available from: <http://www.iot-butler.eu/>.
- [56] INNO, Ericsson, Telecom Italia, Gemalto, CEA, FB-Consulting, ISMB, IhomeLabs, Swisscom, STMicroelectronics, Université de Luxembourg, Katholieke Universiteit Leuven, Cascard, TST, Jacobs University, Utrema, Zigpos, *D3.1 Architectures of BUTLER Platforms and Initial Proofs of Concept*. <http://www.iot-butler.eu/download/deliverables>, D3.1. October 2012.
- [57] SiteWhere, Available from: <http://documentation.sitewhere.org/>.
- [58] Kaa, Available from: <http://www.kaaproject.org/>.
- [59] Kaa Project Docs, Available from: <http://docs.kaaproject.org/display/KAA/Design+reference>.
- [60] servIoTicy, Available from: <http://www.servioticy.com/>.
- [61] Villalba, Álvaro, et al., *servIoTicy and iServe: A Scalable Platform for Mining the IoT*. Procedia Computer Science 52 (2015): 1022-1027.
- [62] FIESTA-IoT, Available from: <http://fiesta-iot.eu/iot-experiments-as-a-service/>.
- [63] Smart Santander, Available from: <http://www.smartsantander.eu/>.
- [64] Nati, Michele, et al., *Smartcampus: A user-centric testbed for internet of things experimentation*. Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on. IEEE, 2013.
- [65] Kyle Banker, *MongoDB in Action*. Manning Publications Co., Greenwich, CT, 2011.
- [66] Henrik Bærbak Christensen, Henrik Blunck, Niels Olof Bouvin, Robert S. Brewer, and Markus Wüstenberg, *Karibu: a flexible, highly-available, and scalable architecture for urban data collection*. In Proceedings of the First International Conference on IoT in Urban Space (URB-IOT '14). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 96-98.
- [67] Chantry, D., *Mapping applications to the cloud*. Technical Report, January 2009.
- [68] M. Amadeo et al., *Information-centric networking for the internet of things: challenges and opportunities*. IEEE Network, vol. 30, no. 2, March-April 2016, pp. 92-100.
- [69] F. Fund et al., *Under a cloud of uncertainty: legal questions affecting internet storage and transmission of copyright-protected video content*. IEEE Network, vol. 30, no. 2, March-April 2016, pp. 32–38.
- [70] Sarkar, Chayan, et al., *A scalable distributed architecture towards unifying iot applications*. Internet of Things (WF-IoT), 2014 IEEE World Forum on. IEEE, 2014.
- [71] Da Xu, Li, Wu He, and Shancang Li., *Internet of things in industries: a survey*. Industrial Informatics, IEEE Transactions on 10.4 (2014): 2233-2243.
- [72] Ganti, Raghu K., Fan Ye, and Hui Lei., *Mobile crowd-sensing: current state and future challenges*. IEEE Communications Magazine 49.11 (2011): 32-39.
- [73] Bin, Shen, Liu Yuan, and Wang Xiaoyi., *Research on data mining models for the internet of things*. Image Analysis and Signal Processing (IASP), 2010 International Conference on. IEEE, 2010.
- [74] Miorandi, Daniele, et al., *Internet of things: Vision, applications and research challenges*. Ad Hoc Networks 10.7 (2012): 1497-1516.

- [75] M. Jelasity, A. Montresor, O. Babaoglu, *Gossip-based aggregation in large dynamic networks*. ACM Trans. Comput. Syst. 23 (1) (2005) 219–252.
- [76] J. Garay, Y. Moses, *Fully polynomial byzantine agreement for $n > 3t$ processors in $t + 1$ rounds*. SIAM J. Comput. 27 (1) (1998) 247–290.

Privacy-Preserving Proximity Services

Michael Haus, Karim Emara and Jörg Ott

Department of Informatics, Technical University of Munich, Munich
{haus, emara, ott}@in.tum.de

Abstract

In the last years, the paradigm of personal computing changed drastically, moving away from stationary PCs and heavyweight laptops to mobile devices. This change is based on the ubiquity of mobile interconnected devices leading to great opportunities for services that utilize location, such as navigation or communication with nearby friends. Location-based Services (LBS) are widely used based on a centralized architecture and absolute GPS positions. We focus on Proximity-based Services (PBS) based on peer-to-peer architecture to detect what is around us. In addition, we provide further insights about which data are potentially useful to create meaningful proximity information. Many LBS and PBS achieve their functionality without advanced privacy protection mechanisms. However, mobile data especially location data is sensitive, because adversaries can infer whereabouts of mobile users. Moreover, the uniqueness of human mobility traces is high yielding to a high identification rate of individual users. Therefore, we review the most recent literature in the domain of private proximity testing including attack models.

Keywords

Proximity-based Services; Location-based Services; Private Proximity Testing

1. Introduction

We have a paradigm shift in personal computing, moving away from stationary PCs and heavyweight laptops to mobile devices. In 2012, mobile phones and tablets outsold PCs and notebooks by a ratio of 5.5x and the gap will further increase up to 10.2x in 2018 [1, 2]. Besides that, the study [3] reports that 19% of the world's mobile users already using LBS and the most popular application is the navigation via maps and GPS. Furthermore, one in five (22%) of LBS users enrich their social lives by finding friends in the nearby environment. Therefore, we have a broad basis of potential users that can use Proximity-based Services (PBS) and Location-based Services (LBS) on their mobile devices.

LBS mainly rely on the absolute position of an user to answer the question "where we are?". In contrast, PBS are based upon context information to find co-location with other points of interest to answer the question "who are we with?". PBS are a subclass of the well-known LBS and their goal is to improve the users' daily lives by providing a personalized service to enable sharing of location information and location-aware information retrieval. Therefore, the LBS focus on a centralized architecture, in which the location server acts as a Trusted Party (TP) which receives coordinates from the users to provide location-specific information, e.g. nearby friends. The assumption of a fully trusted server is unrealistic and the use of global positioning systems limits the functionality of LBS to outdoor environments. In comparison, PBS use relative positioning between entities in a smaller local reference frame to solve the issues of LBS. LBS use global positioning systems, while PBS use also other positioning techniques

such as Bluetooth and Wi-Fi which are working indoor in an energy-efficient manner.

The location data used by LBS and PBS is sensitive and must be protected against privacy attacks. De Montjoye *et al.* [4] state that mobile data is among the most sensitive data currently collected. The uniqueness of human mobility traces is high, which allows to uniquely identify 95% of mobile users by four spatio-temporal points. This enables location tracking, which includes threats, such as stalking, mugging or empty home for burglary (absence). The study [5] reports that 51% of the participants using LBS, but only 18% share their location with others. The cause is that 52% of the users have strong privacy concerns for location sharing with other people and even higher for sharing with businesses. In general, 83% worry about the overall loss of privacy.

Our paper addresses the following research questions with respect to private proximity services:

- RQ1 How to detect spatial proximity of two or more users efficiently, promptly and correctly? We focus on PBS to recognize nearby users without relying on a TP. The available mechanisms take advantage of Bluetooth, Wi-Fi or sound data to calculate relative distances between mobile users.
- RQ2 How to secure the proximity solution, which does not require the disclosure of the user locations? We review techniques for Private Proximity Testing (PPT), which uses cryptographic algorithms to enable a pair of mobile users to privately test whether they are nearby within a specific distance threshold.

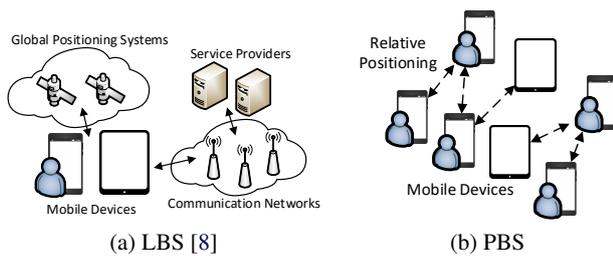


Figure 1. Common architectures of location aware systems.

The remaining sections of this survey are organized as follows: Section 1 answers RQ1, it presents different PBS, which use position, sound or multimodal data to detect mobile users in vicinity. Section 2 answers RQ2, it shows multiple PPT approaches to perform proximity tests in a private manner. Section 3 highlights potential research directions for private proximity services. Finally, Section 4 briefly summarizes the characteristics of LBS and PBS.

2. Proximity-based Services (PBS)

This section shows the evolution from LBS to PBS and their different characteristics. The term proximity is defined as "the state of being near to somebody" [6]. A proximity location-sensing technique determines when an object is near to a known location [7].

2.1 Evolution of Location Aware Systems

The widespread use of LBS is based on the mainstream popularity of mobile devices, such as smartphones and tablets. Initially, the development of LBS was started by the following systems: Active Badge system, Microsoft's RADAR system, MIT's Cricket project and Intel's Place Lab project. Figure 1(a) shows the common LBS architecture consisting of four major entities: mobile devices, global positioning systems, communication networks and service providers [8]. Users send queries to LBS servers including their location obtained via GPS of the mobile device. The LBS servers are service providers answering the queries with tailored information based on the location in the queries. All queries and responses are transmitted via communication networks, such as third-generation (3G) networks. We give two examples of nowadays LBS. First, Tiramisu [9] provides real-time information for the local public transportation, e.g. to find the nearest bus stop. Second, Walkly [10] as safety application to define the destination and estimated arrival time of your journey, if you fail to reach the destination, your trusted network will be notified and prompted to take action. The categorization of existing LBS show their application diversity: marketing, emergency, geotagging, tracking, navigation, gaming, social media, sports, billing and points of interest (POIs) [11].

Now, we present characteristics and disadvantages of LBS to motivate the need of PBS. Current LBS rely on a centralized architecture with a location server acting as TP. However, it is questionable whether the assumption of a TP is realistic [12].

In addition, most of the LBS use global positioning systems, which limits their functionality to outdoor environments, although people spend the majority of their time indoors [13]. Moreover, the global positioning techniques are usually energy demanding, which is an issue especially for resource constrained mobile devices.

PBS use relative positioning between entities in a smaller local reference frame in comparison to LBS with a global positioning. The PBS are trying to solve the issues of LBS by focusing on an infrastructure-less environment without a TP as highlighted in Figure 1(b). The goal of PBS is to calculate the relative distance (proximity) between users and POIs to identify the closest POIs inside an area of interest [14]. PBS focus on an advanced definition of proximity, not strictly geographic, which is defined as semantic proximity [15]:

Information about a location, its environmental attributes (e.g. noise level, light intensity, temperature, and motion) and the people, devices, objects and software agents that it contains.

There are multiple wireless technologies and positioning techniques that can estimate the proximity among POIs as illustrated in Table 1.

Application examples of PBS include public safety, localized social networking, home automation and networking, local data transfer (offloading) and mobile advertisements [16]. After the detection of potential nearby communication partners, you are able to ask questions like "Is there anybody on the train who can lend me a power adapter?" or "What's the guest Wi-Fi password at the airport?" As a result, the user can easily access locally-related information. Applications of this kind can be referred to as the term Mobile Social Networks in Proximity (MSNP) [17]. MobiClique [18] is another example, which alerts users when other users are in physical proximity and share a relationship based on profile and friend list. Besides that, the recent AllJoyn [19] initiative enables ad hoc, secure, proximity-based, device-to-device communication without a cloud or intermediary server.

The architecture of PBS has the following characteristics:

- Peer-to-peer operation: Mobile devices use their hardware capabilities to achieve localization and communicate directly to other users.
- Local validity: The information shared with PBS is locally relevant with little interest to the rest of the world.
- Temporal validity: Information provided by PBS is usually valid for a limited amount of time and it is not useful to store that information at a central storage.

Based on these characteristics, the peer-to-peer architecture of PBS has the following advantages: no dependence on central (third-party) entities, limited surveillance or censorship and inherently stronger anonymity due to missing central authority. Another benefit of PBS is data offloading, because in most

Table 1. Comparison of short-range wireless transmission techniques [20, 21].

Wireless technology	Bluetooth 4.0	D2D	WiFi Direct	LTE Direct
Max. transmission distance	10 - 100 m	10 - 1000 m	200 m	500 m
Max data rate	24 Mb/s	1 Gb/s	250 Mb/s	-

cases it is unnecessary to upload large pieces of content to a central authority due to geographic and time-dependent constraints, meaning that the information is only relevant for a small number of users. This property is particularly useful when we consider the prediction of Cisco [22], that the global mobile data traffic grew 74 % in 2015 and will further increase nearly eightfold between 2015 and 2020. On the other hand, the infrastructure-less services are tricky in terms of limited range, reliability, scale and trust.

2.2 Approaches for PBS

This section describes the state-of-the-art of PBS. The existing solutions for proximity estimation can be based on position, sound or multimodal data. Moreover, we introduce the current industry efforts to provide PBS.

2.2.1 Industry Efforts

First, we present current industry efforts in the field of PBS. In 2015, the 3GPP group standardized Device-to-Device Proximity Services (ProSe) for LTE [16]. The popularity of PBS is largely driven by social networking applications, in which the direct communication between nearby mobile devices is particularly interesting. The basic functionality defines proximity in a broader sense than just by physical distance. It is also based on channel conditions, signal to interference plus noise ratio (SINR), throughput, delay, density and load. Deutsche Telekom states an evolving demand for proximity services and introduces LTE proximity services called LTE Radar based on the standardization of ProSe [23]. In addition, Qualcomm developed a new technique known as LTE Direct for device-to-device (D2D) proximal discovery [21]. The challenges can be grouped into four major categories: (1) energy efficiency in case of continuous device discovery, (2) long enough ranges and high enough capacity to enable broad set of use cases, (3) interoperable discovery between different mobile apps, operating systems and devices and (4) privacy barriers to approaches that track the user's location. The overall goal is to make the best use of all technologies for proximity services, such as LBS for user-initiated search, LTE Direct for always-on device-to-device proximal discovery and Proximity beacons [24] for micro-location awareness and geo-fencing. Another initiative known as Wi-Fi Aware [25] provides always-on, real-time discovery of what is available nearby. In general, there are two design approaches to enable proximity aware systems:

- WAN top-down approach: expand heterogeneous cellular network to include D2D capability, e.g. 3GPP LTE D2D

- WLAN bottom-up approach: expand and integrate existing standalone D2D solutions, e.g. WiFi Direct, Bluetooth

Furthermore, the service types of proximity systems can be classified into: (1) standalone and self-organizing (WiFi Direct, Bluetooth), (2) network assisted or network controlled (LTE + WiFi Direct) and (3) network integrated and heterogeneous network (3GPP LTE D2D).

2.2.2 Position-based PBS

Most of the PBS use short-range wireless such as Bluetooth and WiFi to locate nearby devices based on the position. One of the first systems was IFind [26] for real-time location monitoring. Banerjee *et al.* [13] introduced Virtual Compass to create a 2D localisation map of nearby devices based on users relative distances. The system measures the received signal strength (RSSI) of directly exchanged messages using Wi-Fi and Bluetooth. Both measurements are combined for distance estimation of neighboring peers with higher accuracy. In addition, the system considers several mechanisms to save energy. First, it increases the Bluetooth scan interval when the neighbor graph does not change. Second, a cloud service collects Wi-Fi positions of the users and determines whether the nearby devices are currently active or not. This service informs users when other active devices are in vicinity. Otherwise, Wi-Fi and Bluetooth can be disabled if the device is completely alone. PeerSense [27] scans the neighborhood to detect nearby devices in combination with a social network for authentication. Thus, the system shows only those people, e.g. friends who have allowed you to recognize them. Friends Radar [28] provides decentralized location updates in peer-to-peer fashion using XMPP and GPS. Only known contacts or friends are visible, as extension for indoor environments the application uses signal strength techniques instead of GPS. Comm2Sense estimates the distance between subjects applying data mining techniques to analyze Wi-Fi RSSI [29, 30]. Many approaches use Bluetooth and WiFi for proximity detection due to ease of implementation and wide availability in many mobile devices [31], [32]. Similar peer-based localization methods, include NearMe [33] and BlueEye [34]. Bostanipour and Garbinato [35] studies the effect of parameters to improve the detection probability. The evaluation results show that an increased transmission power and increased time range between two consecutive broadcasts enhance the recognition. Another important aspect is the energy efficiency, because continuous proximity sensing rapidly drains battery of mobile devices. eDiscovery [36, 37] identified that Bluetooth high-power state consumes less energy

than lower-power state of WiFi. The approach dynamically changes duration and interval of Bluetooth discovery based on number of discovered peers.

2.2.3 Sound-based PBS

Most research in location sensing answered the question "where we are?" (physical location). This section focus on sound-based proximity systems and their main purpose is to answer the question "who are we with?" (co-location). To obtain accurate proximity information, many approaches use ambient noise as spatiotemporal identifier, because ambient sound contains abundant information which depends on time and space [38]. For example, SoundSense [39] applies a variety of sound features to discover sound events that are specific to individual users. Tan *et al.* [40] has the goal to detect groups of people attending the same meeting and automatically enable content sharing among them. They capture silence signatures of the sound, which are robust against differences in loudness and placement relative to the speaker. The similarity measure is based on the cosine metric to determine if two people sharing the same context. The audio co-location system achieves a worst case accuracy of 96%. By using sound as proximity information, the system is mostly independent from infrastructure and sense as many contexts as possible, indoor and outdoor. Other approaches [41, 42] use complex acoustic signatures to represent rich context information. However, sound information is highly sensitive and therefore the system should protect the users' privacy, such as the silence signatures of the previous system. Wyatt *et al.* [43] use only features from which speech cannot be reconstructed to improve the privacy. Thiel *et al.* [44] perform ambient sound analysis to detect mobile phones in vicinity. First, the approach uses Bluetooth pairing of two devices to increase the system accuracy, because persons are already in the same space. Afterwards, one device emits a repeating sound pattern in inaudible narrow spectrum and the other device tries to detect them by signal auto correlation. The work [38] calculates cross correlation in the frequency domain for the similarity measure of ambient sound. The system discriminates 5 rooms with the accuracy of true positive 94.9% and false positive 0.1%. However, the sound is not only appropriate for phone-to-phone proximity, audio tones also enable accurate distance measurement between mobile phones. The approaches [45, 46] estimate the distance between a pair of mobile phones based on propagation delays of audio beacons that are transmitted by each phone. Qiu *et al.* [47] propose a 3-dimensional relative localization between smartphones equipped with two microphones for accurate detection of social interactions among mobile phone users. Lane *et al.* [48] provide an overview about the state-of-the-art in mobile audio sensing.

2.2.4 Multimodal-based PBS

As previously mentioned our special interest is on semantic proximity to collect more data about the surrounding environment for meaningful proximity modeling. For example,

Table 2. Physical parameters to classify a proximity hierarchy [50].

	Environment	Object
Static state	temperature, humidity, pressure, ambient acoustic	orientation, tilt, altitude, light
Dynamics	motion (person moving), light changes, acoustic (speakers, door slamming)	acceleration, wind, light changes

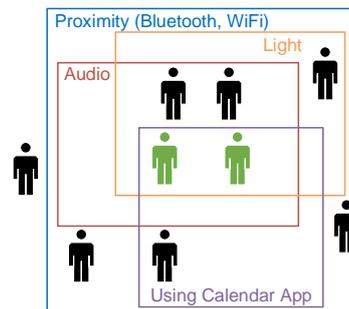


Figure 2. Examine the context of all nearby devices and look after similarities to group users that match in multiple modalities (green) [51].

the car detects the driver in near distance and automatically opens the door. However, the driver sits close by a café and is not moving. Therefore, we have to use complementary sensor data, such as position and acceleration to identify more complex situations. Varshavsky *et al.* [49] proposed "Amigo" to detect whether the devices are within the same vicinity based on similar radio signal environment. The Smart-Its Project [50] focused on phone-to-phone proximity and defined a proximity hierarchy to combine different modalities, see Table 2. The idea is to use a complementary set of sensors to obtain accurate proximity estimates.

Freitas and Anind [51, 52] presents the DIDJA Toolkit, a similar approach as the proximity hierarchy, considering multiple different modalities to form groups. No single modality works well in all conditions. Figure 2 shows the use case to opportunistically detect nearby people and build a group to automatically share the free calendar periods. A combination of comparison methods depending on the context type decides whether the users are within a group or not, sharing a similar context. The input values are Bluetooth readings, audio amplitude to preserve user privacy and measurements from thermometer, light sensor and accelerometer. The survey about spontaneous device association [53] contains further approaches and use cases.

2.3 Related Topics

2.3.1 Community Detection

We need an algorithm for community detection and grouping depending on the number of nearby users to ensure a useful service. In case of crowded environments, there are many potential communication partners and we have to limit the members of the communication group and structure the mobile ad hoc network [54] to maintain a fast communication, e.g. by reducing latency. For instance, discover unknown clusters or groups of mobile users sharing the same social behavior or interests [55] or provide assistance to known travel groups staying together. Usually, the social communities are formed by two major approaches. First, the self-reported social network is based on the user's declared interests or friendships. Second, in a detected social network, infer the community based on certain patterns from data traces. Plantié and Crampes [56] present the state-of-the-art in community detection including traditional methods such as k-means, statistical inference-based methods, hierarchical clustering and lastly methods to find overlapping communities [57]. The algorithms for community detection can be divided into heuristic measures and influence maximization. For example, the betweenness metric identifies bridge nodes between two groups. We can partition the network into smaller groups by removing these betweenness edges. In case of influence maximization, the algorithm forwards messages to nodes with large influence (i.e. many connections) on other nodes in the network to enhance message dissemination. Another scenario in which community detection is helpful concerns major events, where we group users and send them to different entrances of public transport.

2.3.2 Service Discovery

Until now, most research is done for connectivity in mobile ad hoc networks [58, 59]. However, service discovery is another key issue, because users want to intuitively share information and services after discovering nearby nodes. The service discovery architectures can be classified into directory-based, directory-less and a hybrid combination of them [58]. The directory-based architecture consists of three node roles: server (service provider), client (service requestor) and a service directory as agent to enable communication between server and clients. In contrast, the directory-less architecture does not use a central service directory for negotiation between service provider and client. The service provider broadcasts service advertisements and the service requestor broadcasts service requests. We prefer a distributed service discovery, because the architecture is simpler due to a missing central control directory. In addition, the distributed architecture is more appropriate for mobile ad hoc networks which share multiple attributes with proximity-based applications. However, directory-less service discovery has some drawbacks. First, higher communication costs to maintain consistency and replicate service information between multiple nodes [59]. Second, a major problem is the frequency of messages, which may result in network congestion. There

are multiple solutions such as probabilistic and intelligent forwarding to solve the issue of network utilization. Further information about service discovery protocols can be found in [60]. Sundramoorthy *et al.* [61] present an overview of design aspects and solutions tailored for service discovery involving system size, resource constraints (e.g. bandwidth, energy), security and system heterogeneity.

3. Private Proximity Testing (PPT)

As discussed in the previous section, proximity services support many applications such as social, location-based and advertising applications. However, one disadvantage of these services is that the user's exact location is usually revealed whether to the server or other users, regardless of their proximity to the requesting user. This shared information threatens the user's privacy, although sharing the exact location is not necessary to fulfill the application requirements.

There are several approaches that preserve location privacy and each approach is suitable for specific types of applications. Privacy-preserving approaches include anonymization, obfuscation, transformation, encryption and broadcasting dummy information [12]. Private Proximity Testing (PPT) is based on cryptographic primitives to enable a pair of mobile users to test if they are nearby within a specific distance threshold. PPT protects the location data against a wide range of attacks, because it reveals no sensitive information to anyone, including the service provider [62].

3.1 System Model

There are several system models assumed in related work but they usually composed of the following entities: a trusted party (TP), a service provider (SP) and mobile users. Figure 3 highlights the PPT system model. The TP is responsible for managing and distributing the cryptographic keys among users and the SP. This trusted party can be a dedicated entity for the PPT system that bootstraps the whole system and generate keys as in [63]. Otherwise a third party, e.g. social networks can be employed to offload the trust establishment among users to its infrastructure as proposed in [62, 64]. The SP is a central entity that routes user messages to each other, computes proximity tests privately, or even stores encrypted user's locations. The SP is usually considered untrusted and should not learn the user's locations or proximity test results. Depending on the application scenario, mobile users may share strong, weak or no relationship. The user's goal is to test (or to be notified) if another user is located in the vicinity without revealing the exact location or the test result to anyone.

A PPT service has a set of features that characterize a given approach. These features can be categorized into functionality, security and efficiency as discussed in the next sections. To simplify the explanation of the rest of this section, we assume the use case that Alice and Bob are two users and Alice wants to learn if Bob is in her proximity within a threshold δ .

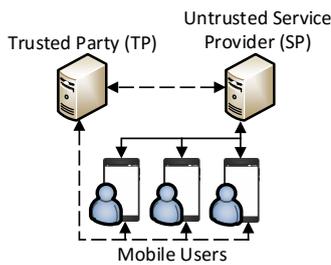


Figure 3. The PPT system model. First, the TP manages the cryptographic keys among users and the SP (dashed line). Second, the SP computes the proximity tests privately and routes user messages (solid line).

3.2 Functionality Features

3.2.1 Architecture

Proximity testing can be handled in both centralized or decentralized ways. In the centralized approach, SP employs a server that may collect and store encrypted user locations periodically and do some computations. The advantages of this approach are that it may reduce the computations required from the user's devices and allow asynchronous communication among users when one user is offline [62]. In addition, it allows computing proximity of multiple users in one round without the need to send a query message for each users pair which minimizes the communication and computation overhead at the client side [63, 65].

In the decentralized approach, users communicate together and do all computations required to find the proximity result without relying on a third party. Note: decentralized is the procedure of proximity testing rather than the protocol operations. For example, an underlying infrastructure such as an SP server [62] or a social network [64] can still be used in a decentralized approach. This infrastructure supports the protocol in exchanging messages among users and/or distribute cryptographic keys, but the testing procedures are only performed on the end devices.

3.2.2 Location Obfuscation and Quantization

Basically, proximity can be tested by calculating the Euclidean distance and test if it lies within the given range or threshold. However, the user's location should be cloaked at first to satisfy ϵ -geo-indistinguishability by adding an appropriate noise (e.g. Laplacian noise) to the location [63]. Geo-indistinguishability, proposed by Andres *et al.* [66], is a variation of differential privacy and defined as follows:

A mechanism satisfies ϵ -geo-indistinguishability iff for any radius $r > 0$, the user enjoys ϵr -privacy within r .

Doing Euclidean distance calculations on encrypted location data could require computationally intensive homomorphic encryption schemes [67]. Instead, many research works propose to partition the service area into grid cells [62–64] where the user's location is expressed in terms of the identity of the containing cell. The bigger the cell size, the higher the privacy

level. This location quantization facilitates testing proximity using encryption techniques, e.g. private equality testing.

3.2.3 Proximity threshold

The proximity threshold can be configured globally for all users or individually by each user. When it is global, Alice is considered proximate to Bob when Bob is proximate to Alice, and vice versa. Moreover, this threshold can be defined as an absolute Euclidean distance [68] or in terms of grid cells [62, 64]. Both definitions are valid when the service area is assumed to be a free non-constrained area. However, they will not work when semantic barriers exist. For example, if two users are located on different sides of train railways or one user is in a shopping mall while the other is just passing by, then these users are not practically proximate. Thus, for this kind of applications it is better to define the threshold as shortest path distance, introduced in [69].

3.2.4 Asymmetry

Proximity testing is asymmetric when the requesting user will learn if the queried user is nearby, but not vice versa [62]. This feature has an advantage that it emulates the asymmetric nature of a social relationship – Bob may be willing to let Alice learn proximity to him, but not inverse. Thus, asymmetry preserves the privacy of Alice. Its disadvantage is that a symmetric proximity test doubles the communication costs, while the service is basically asymmetric.

3.3 Security Features

3.3.1 Encryption

PPT is often considered as an instance of a secure multi-party computation (SMC) problem, where multiple parties compute the output of a function without learning each other's inputs [70]. SMC is also referred as secure function evaluation (SFE) and is either solved by a problem-specific or generic approach. The problem-specific approach exploits the function properties to design a certain protocol that is more efficient than those that would result from a generic solution. The generic approach seeks to design a general solution for the SMC problem by transforming arbitrary functions into secure functions [71]. Homomorphic encryption [72] and garbled circuits [73] are two main techniques for the generic approach.

Homomorphic encryption is a kind of encryption technique that allows anyone (not just the key holder) to perform computations on ciphertext, such that the obtained result, when decrypted, matches with the result of the same operations performed on the plaintext. *Partial* homomorphic systems allow only specific operations to be carried out on the ciphertext, such as Goldwasser-Micali [74], ElGamal [75] and Paillier [76]. Later, Craig Gentry [72] described the first conceivable construction for a *fully* homomorphic encryption, that supports arbitrary computation on ciphertext. In the context of PPT, partial homomorphic systems are often sufficient since the proximity function can be usually constructed using only a small set of primitive operations. PPT problem is often

reduced to *private matching* problem (i.e., private equality testing (PET) or private set intersection (PSI)). In this problem, each party hold a set of inputs and needs to jointly calculate the intersection of the input sets without revealing any further information [77].

Garbled circuits, introduced by Andrew Yao [73] for secure two-party computation, allow modeling of an arbitrary function f as a boolean circuit. The basic idea is that one party (circuit generator) prepares an encrypted version of a circuit computing the desired function; the second party (circuit evaluator) then computes the output of the circuit without learning any intermediate values [71]. Starting with a boolean circuit for f (on which both parties agree in advance), the circuit generator associates two random cryptographic keys w_0, w_1 with each wire of the circuit (w_0 encodes a 0-bit and w_1 encodes a 1-bit). Then, for each binary gate g of the circuit, the generator computes ciphertexts. The resulting four ciphertexts, in random order, constitute a garbled gate. The collection of all garbled gates forms the garbled circuit that is sent to the evaluator. In addition, the generator reveals the mappings from output-wire keys to bits. Assuming that the decryption process is able to detect incorrect decryption, the evaluator attempts to decrypt all four encryptions of each gate, but exactly one will decrypt correctly. Thus, it is able to obtain the correct garbled value without learning anything about the computation or the values. One issue in this technique is that it is secure as long as both parties do not deviate from the protocol procedures. Furthermore, Hallgren *et al.* [68] showed that PPT protocols based on homomorphic encryption are more efficient than those based on garbled circuits.

Searchable encryption [78] is another related cryptographic approach, but it cannot be used in PPT, because it produces deterministic ciphertexts from plaintext. Since location data has a low-min entropy that cannot exceed 2^{40} [79], an adversary can generate all possible encrypted locations and exhaustively search for the user location.

3.3.2 Verification

It is desirable for the PPT service to provide users some kind of verification regarding the proximity result obtained from a server. For the verification of proximity results, Bob should return authenticated data to the server which in turn generates proof information that Alice can use to verify the computation results, as proposed in [65]. In addition, location information itself should be verifiable to trust the received proximity result and prevent location spoofing. One solution for this issue is to use location tags, a set of features collected from signals in the physical environment [62]. To guarantee the effectiveness of these tags, they must fulfill two properties: (1) the tag is similarly reproducible by any device when located at the certain position and time and (2) hard to produce when the adversary is not physically at the required place and time.

3.4 Efficiency Features

3.4.1 Computation

The computation of a PPT service is required to be efficient so that users can perform many tests with their friends on their mobile devices without draining the device battery. Computational factors include the underlying cryptographic primitive, the key length and the number of modular exponentiation required per user. In addition, the optimization of the cryptographic implementation plays an important role. For example, implementations using integers or elliptic curves significantly decrease the running time for methods based on discrete log problems, as shown in [79].

3.4.2 Communication

The communication overhead among system entities should be minimized to save bandwidth and reduce power consumption. The service architecture influences the communication overhead, because it determines the data amount and message frequency exchanged between entities. For a fully decentralized service, proximity testing is performed in a pairwise way, which requires every user to open a separate connection for each friend. Instead, using an intermediate entity, such as an SP server, may reduce this overhead. The user forwards all messages to the server, which in turn de-multiplexes them to the concerned users [62].

3.5 Adversary Models and Attacks

3.5.1 Adversary Model

The adversary model assumed in PPT protocols is related to the secure computation method. There are two main types of adversaries:

Honest-but-curious adversary. The attacker follows the protocol specifications, but keeps a record of all its intermediate computations [80]. They may try to obtain more (private) information beyond that provided by the normal protocol execution. Although this model is weak, it is considered realistic by several research works, since violating this model damage the reputation of the performed entity, e.g. a service provider or a friend [67]. Honest-but-curious adversary is also called semi-honest and passive.

Malicious adversary. The attacker can arbitrarily deviate from the protocol specifications according to the attacker's goal. Malicious behaviors may include aborting the protocol execution at any point, e.g. after obtaining the desired result and reporting bogus location information to the other system entity [80, 81].

3.5.2 Attacks

This section shows several attacks against location aware systems which are protected by PPT.

Localization attack. Since location information has low-min entropy, an adversary can apply an exhaustive search on the encrypted user locations to determine the exact location. For example, in a centralized PPT when the server calculates proximity computation for users, the server can exploit this function in an offline message recovery. First, the server

selects location candidates where the user can be present and encrypts them by the user's public key. Then it uses the proximity function on the encrypted locations previously received from the user and the selected location candidates to determine the user's locations over time.

Bogus information. A malicious user may generate false location or false intermediate information and send it to the other party to produce incorrect results. For example, a user may generate garbage information to deny proximity. Moreover, the user sends customized encrypted information that leads to false positive proximity result to gain the exact location of the other user.

Replay attack. In this attack, a user or server may record previously generated location information to produce a fake proximity result. For example, a user keeps the (verified) location information that produces a negative proximity result to be re-used later when the user wants to hide the proximity to the requesting user. This attack is more relevant when location tags are employed.

Multi-run PPT attack. If the PPT service is ideal and deterministic, it can be prone to this type of attack. An ideal proximity test produces an accurate result whenever the two parties are located within the specified distance threshold. In this case, the attacker can run the proximity test several times to determine the user's exact location by moving around and applying triangulation (assuming the user is stationary) [62, 68].

Collusion attack. When several users or a server and other users collude together to gain more information than permitted. One example of this attack considers when a user A specifies different proximity thresholds for different friends B and C matching trust levels in them. Users B and C may collude together to let the less-trusted friend know more than permitted to know. Another example similar to multi-run attack, several users run the proximity test for the same user in the same time to determine the exact location using triangulation [68].

3.6 Approaches

In this section, we focus on location privacy approaches for LBS. First, general privacy-preserving approaches are reviewed followed by the relevant work on private proximity detection services.

3.6.1 Location Privacy Approaches

Location privacy is mainly related to LBS, which uses a Trusted Third Party (TTP), that receives location data from the mobile users to provide location-specific information. This centralized approach is vulnerable by multiple adversaries. Location privacy is mainly covered by anonymity and obfuscation approaches.

Anonymity techniques aim to hide the person's identity. Most approaches are based on k -anonymity [82], in which the target user is indistinguishable from the other $k - 1$ users. The location server (LS) calculates the obfuscation area containing k users and the LBS only receives the obfuscation area and is not able to identify a specific user. Another idea of Dürr *et al.* [83] splits positions into shares and distributes them among

non-trusted LSs. Thus, the attacker must compromise several LS to get sufficient location information for identifying users. Position dummies [84] is another concept where the user sends multiple false positions ("dummies") to the LS together with true user position.

Obfuscation degrades the quality of location information to protect the user identity. Gutscher *et al.* [85] performs geometric operations such as shift or rotate over the positions before sending them to the LS. Further approaches include path cloaking [86] or virtual trip lines [87]. Ardagna *et al.* [88] expands the obfuscation area based on a probability distribution function, which calculates the probability that a user is located in a specific area. This approach prevents map matching attacks.

3.6.2 PPT Approaches

Mascetti *et al.* [89, 90] present a set of protocols including Hide&Crypt to share a secret key and encrypt the locations before transmission using SMC. Narayanan *et al.* [62] proposed three protocols. The first two protocols employ private equality testing to check if three-layered hexagons are overlapping. The third protocol represents locations as location tags and thus employs private set intersection to compute if both parties have overlaps in their tags. Saldamli *et al.* [67] builds upon the second protocol in [62] to propose a Vectorial Private Equality Testing (VPET) protocol. VPET decreases the use of cryptographic primitives by blinding the values through a simple geometric representation of the values.

Novak and Li [64] proposed the Near-pri scheme which is mainly based on Paillier encryption technique. Authors divide the earth into small sections of 10 m^2 and each user maintains a Policy P of a factor of 10 m . The user's location will be shared with other users if their locations are within P . If Alice wants to learn Bob's location, Bob generates several first degree polynomials. Each polynomial is rooted at one value from his set $[L_b - P_b, L_b + P_b]$ where L_b is the latitude of Bob. Bob sends the encryption of the negated coefficients $(E(-C_i) \forall i \in [1, n])$ from these polynomials to Alice. Alice computes $E(L_a) * E(-C_i) \% n^2$, and the resulting value can only be decrypted by Bob. If the decrypted result is 0, Bob knows that Alice's latitude is $L_a = C_i$. This procedure is repeated for the longitude and both parties are considered nearby if Alice's latitude and longitude are within the policy of Bob. To avoid a huge number of polynomials that Bob should generate, authors employed binary tree to store the sections that Bob's policy area spans. Bob generates polynomials for parent nodes that spans only the Bob's policy area (i.e., not the tree root for example because it covers external sections as well). Authors employed Facebook Chat in message transmission, but they found out that this is the largest bottleneck in their system with a speed of roughly 8 Kbps . They recommend to offload all message transmission to a third party server to attain the 3G or 4G speeds.

Zhuo *et al.* [65] proposed a scheme that enables users to verify the obtained proximity result. They divided the service area into a grid of square cells where a user can define her

discoverable range (DR) as a set of cell numbers. Specifically, Alice encrypts her cells set using ElGamal algorithm and sends it to the server along with her public key. Then, the server broadcasts this request to all users. If Bob is interested, he encrypts his DR using Alice's public key, generates an authenticated data $auth_{Bob}$ based on DR and sends both to the server. The server performs the proximity computation, generates proof information and sends all this information back to Alice. Authors showed in the evaluation section that the waiting time till test results are obtained increases with the increase of the size of DR and the number of responses. In general, the waiting time is reasonably long (e.g., Alice waits about 30 s for 15 responses, if her proximity area threshold is only 320^2 m).

Kotzanikolaou *et al.* [79] proposed a lightweight PPT protocol which requires only one public-key exponentiation per user. In this protocol, a user computes private and public keys based on her actual location and uses them in a simple modular exponentiation to perform the equality testing. Authors compared their protocol with other two protocols and showed efficient performance (e.g., it takes less than 0.12 ms using MIRACL library). The disadvantage of this protocol is that it does not support setting proximity range for users.

Huang *et al.* [63] proposed EPPD scheme in which users frequently upload their encrypted locations to SP. When a proximity test is initiated by a user, SP finds all friends who are located in an area intersects with the region specified by the requesting user. Then SP forwards a secured randomized query to those users who replies SP with the encrypted proximity results. Finally SP relays all responses to the requesting user.

4. Research Directions

We highlight several opportunities for further research in the domain of privacy-preserving proximity-based applications.

Hybrid architecture. The PBS have a limited range based on the used wireless communication technologies, such as Bluetooth and Wi-Fi. We focus on a hybrid solution between centralized LBS and peer-to-peer PBS to extend the reachability and strengthen the reliability, such as a cloudlet based proximal discovery service [91]. For example, Liberouter [92] as low-cost router platform provides a WLAN access point without relying on Internet infrastructure. The users are able to access content of the neighborhood stored locally on the platform. In addition, the principle of floating content [93] is a fully distributed variant of an ephemeral content sharing service. The approach uses store and forward for message dissemination and is solely dependent on the mobile devices in the vicinity. The lifetime and distribution of locally created content depends on interested nodes being available. Besides that, the performance and network structure of PBS as peer-to-peer application can be improved by selecting a powerful node called superpeer [94] acting as a server for a set of clients. The node selection is based on criteria, such as highest bandwidth, storage or low latency between clients and server.

Semantic proximity. Existing work focus mainly on strict geographic definition of proximity based on absolute distances, i.e. user A is 200 m away from user B. This definition is not practical in real applications due to the spatial barriers, as discussed in Section 3.2.3. Our research efforts are directed along the semantic proximity to create a meaningful place identity through different surrounding modalities. For instance, measure the proximity in terms of reachability by different transport modalities, e.g. walking or public transport. In this case, the proximity threshold will be expressed in time and the modal of transport. Besides that, another interesting question concerns, what sensor data [95,96] is characteristic and available [97] for a certain place? The in-depth analysis of mobile social signal processing [98] present frameworks to collect a wide range of sensor data to detect activity or location.

Area of interest. The user defines an area of interest to be notified, when other users are within the zone. Most approaches use a simplistic circle around the current user location and shortest path distance for proximity detection [69]. However, this proximity definition is restricted to non-constrained Euclidean spaces, e.g. users on different sides of the river but within distance threshold. In addition, the rigid distance threshold does not allow to choose different areas of interest. There are some grid-based extensions such as vicinity region to support convex or concave shaped interest zones [99,100]. All existing approaches focus on distance definitions with respect to an area of interest. Our idea is to introduce a new metric, the walking time, e.g. 5 minutes to the next point of interest by considering obstacles for a more realistic solution. Thereby, we could extend vicinity regions to closed regions of arbitrary shapes.

Many-to-many proximity tests. Proximity testing is usually performed in one-to-one or one-to-many paradigms, i.e. a pair of users check their proximity or a user scans for nearby friends. What if there is a group of people who navigate in a city and they need to re-join into sub-groups according to their proximity. This use case may need to perform many-to-many proximity tests which is definitely inefficient if they are performed one by one. Thus, one research direction is to perform many-to-many proximity testing in private and efficient way.

Proximity test considers user movements. Another research direction concerns the user mobility. Many research works assume users do not change their locations until the proximity test is performed. Testing proximity of moving users may support several interesting applications. In this case, the speed, direction and transport modality can accompanied with the encrypted location to find where and when users can actually meet.

5. Conclusion

The high relevance and widely usage of LBS is based on providing personalized information automatically adjusted by the current user location. The system relies on a centralized

architecture using the GPS signal of the user. The benefits of this architecture include unlimited range and a large installation base. On the other hand, the system has a high battery drain from constant network pings and a privacy barrier due to possible location tracking. The PBS focus on a smaller scale using wireless network technologies such as Bluetooth and WiFi with a limited range of approximately 50 meters. The advantages are lower power consumption, privacy sensitive and indoor support. The PPT aims to secure the centralized approach of LBS using mainly homomorphic encryption and garbled circuits. One major concern is the energy consumption, because these cryptographic methods are demanding in terms of energy consumption. In the next step, we will specify a prototype system based on the findings of this survey.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation. Bavaria, an initiative of the Bavarian State Government.

References

- [1] Gartner. Gartner says worldwide pc, tablet and mobile phone combined shipments to reach 2.4 billion units in 2013. [Online]. Available: <http://www.gartner.com/newsroom/id/2408515>
- [2] ——. Worldwide device shipments to grow 1.9 percent in 2016, while end-user spending to decline for the first time. [Online]. Available: <http://www.gartner.com/newsroom/id/3187134>
- [3] TNS. (2012) Mobile life study. [Online]. Available: <http://www.tnsglobal.com/press-release/two-thirds-world/T1textquoterights-mobile-users-signal-they-want-be-found>
- [4] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, “Unique in the crowd: The privacy bounds of human mobility,” *Scientific reports*, vol. 3, 2013.
- [5] Microsoft. (2011) Location based services usage and perceptions survey presentation. [Online]. Available: <https://www.microsoft.com/en-us/download/details.aspx?id=3250>
- [6] A. S. Hornby, S. Wehmeier, and M. Ashby, Eds., *Oxford Advanced Learner’s Dictionary of Current English*, 7th ed. Oxford University Press, 2005.
- [7] J. Hightower and G. Borriello, “A survey and taxonomy of location systems for ubiquitous computing,” 2011.
- [8] K. G. Shin, Xiaoen Ju, Zhigang Chen, and Xin Hu, “Privacy protection for users of location-based services,” *IEEE Wireless Communications*, vol. 19, no. 1, pp. 30–39, February 2012.
- [9] Tiramisu transit. [Online]. Available: www.tiramisutransit.com
- [10] Walkly. [Online]. Available: <http://www.walklyapp.com/>
- [11] H. S. Maghdid, I. A. Lami, K. Z. Ghafoor, and J. Lloret, “Seamless outdoors-indoors localization solutions on smartphones,” *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–34, 2016.
- [12] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, “A classification of location privacy attacks and approaches,” *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, 2014.
- [13] N. Banerjee, S. Agarwal, P. Bahl, R. Chandra, A. Wolman, and M. Corner, “Virtual compass: Relative positioning to sense mobile social interactions,” in *Pervasive Computing*, ser. Lecture Notes in Computer Science. Springer, 2010, vol. 6030, pp. 1–21.
- [14] G. Karabulut-Kurt, “On the performance of proximity-based services,” *Wireless Communications and Mobile Computing*, vol. 13, no. 15, pp. 1397–1405, 2011.
- [15] M. Knappmeyer, S. L. Kiani, E. S. Reetz, N. Baker, and R. Tonjes, “Survey of context provisioning middleware,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1492–1519, 2013.
- [16] X. Lin, J. Andrews, A. Ghosh, and R. Ratasuk, “An overview of 3gpp device-to-device proximity services,” *IEEE Communications Magazine*, vol. 52, no. 4, pp. 40–48, 2014.
- [17] Y. Wang, A. V. Vasilakos, Q. Jin, and J. Ma, “Survey on mobile social networking in proximity (msnp): Approaches, challenges and architecture,” *Wireless Networks*, vol. 20, no. 6, pp. 1295–1311, 2014.
- [18] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, “Mobiclique: Middleware for mobile social networking,” in *Proceedings of the 2nd ACM Workshop on Online Social Networks (WOSN)*, 2009, pp. 49–54.
- [19] Alljoyn. [Online]. Available: <https://allseenalliance.org/>
- [20] D. Feng, L. Lu, Y. Yuan-Wu, G. Li, S. Li, and G. Feng, “Device-to-device communications in cellular networks,” *IEEE Communications Magazine*, vol. 52, no. 4, pp. 49–55, 2014.
- [21] Qualcomm Technologies Inc., “Lte direct always-on device-to-device proximal discovery,” August 2014.
- [22] Cisco, “Visual networking index: Global mobile data traffic forecast update, 2015–2020,” 03.02.2016.
- [23] T. Henze, “Lte proximity services,” in *LTE World Summit*, 2014.

- [24] Xin-Yu Lin, Te-Wei Ho, Cheng-Chung Fang, Zui-Shen Yen, Bey-Jing Yang, and Feipei Lai, “A mobile indoor positioning system based on ibeacon technology,” in *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 4970–4973.
- [25] Wi-Fi Alliance. (2015) Wi-fi aware. [Online]. Available: <http://www.wi-fi.org/discover-wi-fi/wi-fi-aware>
- [26] S. Huang, F. Proulx, and C. Ratti, “Ifind: A peer-to-peer application for real-time location monitoring on the mit campus,” in *Proceedings of the 10th International Conference on Computers in Urban Planning and Urban Management (CUPUM)*, 2007.
- [27] A. Gupta, M. Miettinen, M. Nagy, N. Asokan, and A. Wetzel, “Peersense: Who is near you?” in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom)*, 2012, pp. 516–518.
- [28] R. Mayrhofer, C. Holzmann, and R. Koprivec, “Friends radar: Towards a private p2p location sharing platform,” in *Proceedings of the 13th International Conference on Computer Aided Systems Theory (EUROCAST)*, 2011, pp. 527–535.
- [29] I. Carreras, A. Matic, P. Saar, and V. Osmani, “Comm2sense: Detecting proximity through smartphones,” in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2012, pp. 253–258.
- [30] V. Osmani, I. Carreras, A. Matic, and P. Saar, “An analysis of distance estimation to detect proximity in social interactions,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 5, no. 3, pp. 297–306, 2014.
- [31] T. M. T. Do and D. Gatica-Perez, “Groupus: Smartphone proximity data and human interaction type mining,” in *15th Annual International Symposium on Wearable Computers (ISWC)*, 2011, pp. 21–28.
- [32] T. Higuchi, H. Yamaguchi, and T. Higashino, “Clearing a crowd: Context-supported neighbor positioning for people-centric navigation,” in *Proceedings of the 10th International Conference on Pervasive Computing*, 2012, pp. 325–342.
- [33] J. Krumm and K. Hinckley, “The nearest wireless proximity server,” in *Proceedings of the 6th International Conference on Ubiquitous Computing (UbiComp)*, 2004.
- [34] A. Ghose, C. Bhaumik, and T. Chakravarty, “Blueeye: A system for proximity detection using bluetooth on mobile phones,” in *Proceedings of the ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp Adjunct)*, 2013, pp. 1135–1142.
- [35] B. Bostanipour and B. Garbinato, “Improving neighbor detection for proximity-based mobile applications,” in *IEEE 12th International Symposium on Network Computing and Applications (NCA)*, 2013, pp. 177–182.
- [36] B. Han, J. Li, and A. Srinivasan, “On the energy efficiency of device discovery in mobile opportunistic networks: A systematic approach,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 786–799, 2015.
- [37] B. Han and A. Srinivasan, “ediscovery: Energy efficient device discovery for mobile opportunistic communications,” in *Proceedings of 20th IEEE International Conference on Network Protocols*, 2012, pp. 1–10.
- [38] H. Satoh, M. Suzuki, Y. Tahiro, and H. Morikawa, “Ambient sound-based proximity detection with smartphones,” in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2013, pp. 1–2.
- [39] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, “Soundsense: Scalable sound sensing for people-centric applications on mobile phones,” in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2009, pp. 165–178.
- [40] W.-T. Tan, M. Baker, B. Lee, and R. Samadani, “The sound of silence,” in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2013, pp. 1–14.
- [41] T. Nakakura, Y. Sumi, and T. Nishida, “Neary: Conversation field detection based on similarity of auditory situation,” in *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2009, pp. 1–6.
- [42] B. Zhang and M. D. Trott, “Reference-free audio matching for rendezvous,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 3570–3573.
- [43] D. Wyatt, T. Choudhury, J. Bilmes, and H. Kautz, “A privacy-sensitive approach to modeling multi-person conversations,” in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.
- [44] B. Thiel, K. Kloch, and P. Lukowicz, “Sound-based proximity detection with mobile phones,” in *Proceedings of the Third International Workshop on Sensing Applications on Mobile Phones (PhoneSense)*, 2012, pp. 1–4.
- [45] C. Peng, G. Shen, and Y. Zhang, “Beepbeep: A high-accuracy acoustic-based system for ranging and localization using cots devices,” *ACM Transactions on Embedded Computing Systems*, vol. 11, no. 1, pp. 1–29, 2012.
- [46] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, “Push the limit of wifi based localization for smartphones,” in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2012, pp. 305–316.

- [47] J. Qiu, D. Chu, X. Meng, and T. Moscibroda, “On the feasibility of real-time phone-to-phone 3d localization,” in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2011, pp. 190–203.
- [48] N. D. Lane, P. Georgiev, and L. Qendro, “Deeppear: Robust smartphone audio sensing in unconstrained acoustic environments using deep learning,” in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2015, pp. 283–294.
- [49] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, “Amigo: Proximity-based authentication of mobile devices,” in *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp)*, 2007, pp. 253–270.
- [50] S. Antifakos and B. Schiele, “Beyond position awareness,” *Personal and Ubiquitous Computing*, vol. 6, no. 5, pp. 313–317, 2002.
- [51] A. A. de Freitas and A. K. Dey, “Using multiple contexts to detect and form opportunistic groups,” in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*, 2015, pp. 1612–1621.
- [52] ———, “The group context framework: An extensible toolkit for opportunistic grouping and collaboration,” in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*, 2015, pp. 1602–1611.
- [53] M. K. Chong, R. Mayrhofer, and H. Gellersen, “A survey of user interaction for spontaneous device association,” *ACM Computing Surveys*, vol. 47, no. 1, pp. 1–40, 2014.
- [54] J. Y. Yu and P. Chong, “A survey of clustering schemes for mobile ad hoc networks,” *IEEE Communications Surveys & Tutorials*, vol. 7, no. 1, pp. 32–48, 2005.
- [55] N. Kayastha, D. Niyato, P. Wang, and E. Hossain, “Applications, architectures, and protocol design issues for mobile social networks: A survey,” *Proceedings of the IEEE*, vol. 99, no. 12, pp. 2130–2158, 2011.
- [56] M. Plantié and M. Crampes, “Survey on social community detection,” in *Social Media Retrieval*, ser. Computer Communications and Networks. Springer, 2013, pp. 65–85.
- [57] J. Xie, S. Kelley, and B. K. Szymanski, “Overlapping community detection in networks,” *ACM Computing Surveys*, vol. 45, no. 4, pp. 1–35, 2013.
- [58] C. Ververidis and G. Polyzos, “Service discovery for mobile ad hoc networks: A survey of issues and techniques,” *IEEE Communications Surveys & Tutorials*, vol. 10, no. 3, pp. 30–45, 2008.
- [59] M. Girolami, S. Chessa, and A. Caruso, “On service discovery in mobile social networks: Survey and perspectives,” *Computer Networks*, vol. 88, pp. 51–71, 2015.
- [60] A. N. Mian, R. Baldoni, and R. Beraldi, “A survey of service discovery protocols in multihop mobile ad hoc networks,” *IEEE Pervasive Computing*, vol. 8, no. 1, pp. 66–74, 2009.
- [61] V. Sundramoorthy, P. Hartel, and H. Scholten, “A taxonomy of service discovery systems,” in *Context-Aware Computing and Self-Managing Systems*, ser. CRC Studies in Informatics Series. CRC Press, 2009, pp. 43–77.
- [62] A. Narayanan, N. Thiagarajan, and M. Lakhani, “Location privacy via private proximity testing,” in *Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS)*, 2011.
- [63] C. Huang, R. Lu, H. Zhu, J. Shao, A. Alamer, and X. Lin, “Eppd: Efficient and privacy-preserving proximity testing with differential privacy techniques,” in *Proceedings of the IEEE International Conference on Communication (ICC)*, 2016.
- [64] E. Novak and Q. Li, “Near-pri: Private, proximity based location sharing,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2014, pp. 37–45.
- [65] G. Zhuo, Q. Jia, L. Guo, M. Li, and Y. Fang, “Privacy-preserving verifiable proximity test for location-based services,” in *Proceedings of the IEEE Global Communications Conference*, 2015, pp. 1–6.
- [66] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, “Geo-indistinguishability: Differential privacy for location-based systems,” in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2013, pp. 901–914.
- [67] G. Saldamli, R. Chow, H. Jin, and B. Knijnenburg, “Private proximity testing with an untrusted server,” in *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2013, pp. 113–118.
- [68] P. Hallgren, M. Ochoa, and A. Sabelfeld, “Innercircle: A parallelizable decentralized privacy-preserving location proximity protocol,” in *Proceedings of the 13th Annual Conference on Privacy, Security and Trust (PST)*, 2015, pp. 1–6.
- [69] L. Šikšnys, J. R. Thomsen, S. Šaltenis, and M. L. Yiu, “Private and flexible proximity detection in mobile social networks,” in *Proceedings of the 11th International Conference on Mobile Data Management (MDM)*, 2010, pp. 75–84.
- [70] G. Zhong, I. Goldberg, and U. Hengartner, “Louis, lester and pierre: Three protocols for location privacy,” in *Proceedings of the 7th International Symposium on Privacy Enhancing Technologies (PET)*, 2007, pp. 62–76.
- [71] Y. Huang, D. Evans, J. Katz, and L. Malka, “Faster secure two-party computation using garbled circuits,” in *USENIX Security Symposium*, vol. 201, no. 1, 2011.

- [72] C. Gentry, “A fully homomorphic encryption scheme,” Ph.D. dissertation, Stanford University, 2009, crypto.stanford.edu/craig.
- [73] A. C. Yao, “Protocols for secure computations,” in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS)*, 1982, pp. 160–164.
- [74] S. Goldwasser and S. Micali, “Probabilistic encryption & how to play mental poker keeping secret all partial information,” in *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing (STOC)*, 1982, pp. 365–377.
- [75] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” in *Advances in cryptology*. Springer, 1984, pp. 10–18.
- [76] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT)*, 1999, pp. 223–238.
- [77] M. J. Freedman, K. Nissim, and B. Pinkas, “Efficient private matching and set intersection,” in *Advances in Cryptology (EUROCRYPT)*. Springer, 2004, pp. 1–19.
- [78] M. Bellare, A. Boldyreva, and A. O’Neill, “Deterministic and efficiently searchable encryption,” in *Advances in Cryptology (CRYPTO)*. Springer, 2007, pp. 535–552.
- [79] P. Kotzanikolaou, C. Patsakis, E. Magkos, and M. Korakakis, “Lightweight private proximity testing for geospatial social networks,” *Computer Communication*, vol. 73, no. PB, pp. 263–270, Jan. 2016.
- [80] O. Goldreich, *Foundations of Cryptography*, 2nd ed. Cambridge University Press, 2009, vol. 2.
- [81] J. D. Nielsen, J. I. Pagter, and M. B. Stausholm, “Location privacy via actively secure private proximity testing,” in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom)*, 2012, pp. 381–386.
- [82] M. Gruteser and D. Grunwald, “Anonymous usage of location-based services through spatial and temporal cloaking,” in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2003, pp. 31–42.
- [83] F. Durr, P. Skvortsov, and K. Rothermel, “Position sharing for location privacy in non-trusted systems,” in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2011, pp. 189–196.
- [84] P. Shankar, V. Ganapathy, and L. Iftode, “Privately querying location-based services with sybilquery,” in *Proceedings of the 11th International Conference on Ubiquitous Computing (UbiComp)*, 2009, pp. 31–40.
- [85] A. Gutscher, “Coordinate transformation - a solution for the privacy problem of location based services?” in *Proceedings of the 20th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2006, p. 354.
- [86] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, “Preserving privacy in gps traces via uncertainty-aware path cloaking,” in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*, 2007, pp. 161–171.
- [87] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson, “Virtual trip lines for distributed privacy-preserving traffic monitoring,” in *Proceedings of the 6th International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2008, pp. 15–28.
- [88] C. A. Ardagna, M. Cremonini, and G. Gianini, “Landscape-aware location-privacy protection in location-based services,” *Journal of Systems Architecture*, vol. 55, no. 4, pp. 243–254, 2009.
- [89] S. Mascetti, C. Bettini, D. Freni, X. S. Wang, and S. Jajodia, “Privacy-aware proximity based services,” in *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM)*, 2009, pp. 31–40.
- [90] D. Freni, “Privacy-preserving techniques for proximity based lbs,” in *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM)*, 2009, pp. 387–388.
- [91] J. Michel and C. Julien, “A cloudlet-based proximal discovery service for machine-to-machine applications,” in *Mobile Computing, Applications, and Services*. Springer, 2014, pp. 215–232.
- [92] T. Kärkkäinen and J. Ott, “Liberouter: Towards autonomous neighborhood networking,” in *Proceedings of the 11th IEEE/IFIP Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2014, pp. 162–169.
- [93] J. Ott, E. Hyytiä, P. Lassila, J. Kangasharju, and S. Santra, “Floating content for probabilistic information sharing,” *Pervasive and Mobile Computing*, vol. 7, no. 6, pp. 671–689, 2011.
- [94] G. Jesi, A. Montresor, and O. Babaoglu, “Proximity-aware superpeer overlay topologies,” *IEEE Transactions on Network and Service Management*, vol. 4, no. 2, pp. 74–83, 2007.
- [95] M. Beigl, A. Krohn, T. Zimmer, and C. Decker, “Typical sensors needed in ubiquitous and pervasive computing,” in *First International Workshop on Networked Sensing Systems (INSS)*, 2004, pp. 153–158.
- [96] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, “A survey of mobile phone sensing,” *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

- [97] Sensors overview. [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_overview.html
- [98] N. Palaghias, S. A. Hoseinitabatabaei, M. Nati, A. Gluhak, and K. Moessner, "A survey on mobile social signal processing," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–52, 2016.
- [99] X. Lin, H. Hu, H. P. Li, J. Xu, and B. Choi, "Private proximity detection and monitoring with vicinity regions," in *Proceedings of the 12th International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE)*, 2013, pp. 5–12.
- [100] B. Mu and S. Bakiras, "Private proximity detection for convex polygons," in *Proceedings of the 12th International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE)*, 2013, pp. 36–43.

Data-Driven Continuous Architecture Engineering

Ilias Gerostathopoulos and Christian Prehofer

Department of Informatics, Technical University of Munich, Munich
{ilias.gerostathopoulos, christian.prehofer}@tum.de

Abstract

In this work package, we focus on software architecture engineering for increasing the qualities of Connected Mobility Systems (CoMoSs). We identify the potential for methods that continuously validate new developments and changes based on the value they deliver. To reach the ambitious goal of so-called “data-driven continuous architecture engineering” we need to cater to the need for (i) rapid development cycles that incorporate the feedback of end-users and (ii) analyzing large amounts of data coming from sensors and actuators deployed within a CoMoS that track various aspects of both the deployed product and of the development process that led to it. As a result, in this report we provide an overview of Continuous Integration practices, focusing on how to optimize and learn from existing best practices in the field, and of Big Data analytics concepts and tools, that are currently the enabling technology for various methods of data analysis and data-driven decision making.

Keywords

Software Architecture; Continuous Integration; Big Data Analytics

1. Introduction

Connected Mobility Systems (CoMoSs) refer to the orchestration of devices and services to offer value-added functionalities in the mobility market [1]. An example of a CoMoS is a smart parking system where external Web services provide a homogeneous view over the availability of parking slots in a city, as recorded by roadside sensors, to cars driving in the city.

There are many challenges in the systematic development and maintenance of CoMoSs that relate to both their size and complexity and to the business needs and customers' expectations. On the technical side, one needs to deal with the different device APIs, the dynamicity and unpredictability of the physical world where the devices reside, and the sheer size of the application code that needs to be developed and maintained. As an example, infotainment systems deployed in modern cars are known to comprise several million lines of code. Since cars in CoMoSs are just one of the target platforms (which include servers, end-user devices, road and city infrastructure devices) we expect CoMoSs to be even bigger and more complex than current infotainment systems.

On the business side, there is a tremendous need for delivering new features and enhanced products as quickly as possible so that companies obtain or keep a competitive advantage. Products need also be continuously improved based on customer feedback. To address these needs, more and more companies, even in the traditional embedded domain, adopt agile processes such as Scrum [2] and Kanban [3], and modern software engineering technologies and practices such as Continuous Integration [4] and Continuous Delivery [5].

In this work package, we aim to perform software engineering research with focus on software architecture (SA)

to tackle some of the aforementioned challenges. We use SA in a broader sense than just referring to the structure of the software part of a CoMoS; SA for us encompasses the *set of significant decisions* about the organization of a software-intensive system that have an effect to the system's qualities [6]. Such qualities refer to attributes of the product or service end-users interact with, such as usability, performance, availability, interactivity, privacy, etc. They may also refer to the process of building the product—in such cases, we speak of code maintainability, feature coverage, testability, product extensibility and modularity, developers' productivity, etc.

1.1 Scope and Goals of the report

In order to address the above requirements, this report covers the latest techniques and approaches towards architecture engineering. We focus on topics which we consider specifically important for the rapidly evolving CoMoSs systems. These topics are as follows:

- Latest approaches for improving the architecture engineering process by relying on analytics over development, user, and system data (Section 2). We also provide an overview of the related area of software analytics (Section 2.1).
- A detailed account on the continuous integration practices that can be integrated into our architecture engineering approach (Section 3). We focus here on the differences between the continuous integration systems in industry and the ways to model such systems to better understand and compare them.
- As a key enabler of our envisioned approach, we cover Big Data technologies that can enable scalable dis-

tributed analytics (Section 4). We focus here on the Hadoop ecosystem.

- Reference architectures for CoMoSs (Section 1.2). Since we focus more on the architecture engineering process, and in particular in informing the architecturally significant decisions based on data, in this report we do not provide a detailed account on the reference architectures for CoMoSs.

1.2 Reference Architectures for CoMoSs

A software reference architecture (RA) is a software architecture that provides a template for creating concrete architectures for a particular domain [7]. It encodes proven architectural solutions and common assets that can be reused and provides a vehicle for stakeholder communication. A RA is usually derived via generalization over a set of concrete architectures. In the domain of CoMoSs, there are two problems in coming up with a RA. First, since it is a fast evolving domain, concrete architectures tend to change over time to satisfy different stakeholder requirements. Second, since it is a highly competitive market, there is not much information available about the technical architectures of the companies leading it (in this case, Google and Apple).

We can however, look into the large amount of software architecture and platform research in domains related to CoMoSs, such as Internet of Things (IoT) [8] and Cyber-Physical Systems (CPS) [9, 10]. In the IoT domain, there have been several European projects focusing on a software reference architectures for IoT such as IoTA [11], COMPOSE [12], and ALMANAC [13]. Similarly, in the CPS domain, projects such as AMADEOS [14], TAPPS [15], and Demanes [16] have been looking into the design of efficient, scalable and trusted CPS. Whereas these approaches have been looking primarily at the technical challenges of CoMoSs engineering, we believe that enhancing the state of the art in SE of CoMoSs will require a broad view that combines fast development cycles with continuous feedback from customers and the system under development. We describe this vision next, and explain how it fits with the research planned within this work package.

2. Data-Driven Continuous Architecture Engineering

Since CoMoSs are typically composed of a large number of devices (servers, vehicles, mobile phones, sensors, actuators, etc.), they have the ability to record a large amount of data. Since cost of data storage has dropped dramatically, it is also feasible to store all the recorded data. Recorded data allows us to track aspects that pertain both to the developed products (usage patterns, user satisfaction, error reports, etc.) and the development process that led to them (development time, commits, tests, deployments, etc.). What if we could “close the loop” in software architecture engineering by analyzing the vast amount of available data and to continuously improve the quality of a CoMoS?

This idea is the basis of the approach that we call *data-driven continuous architecture engineering*. According to this, data related to both the runtime phase of a CoMoS and the development life cycle should be recorded and analyzed in order to identify correlations between development methods and end-products, perform experiments measuring end-user behavior and generally assessing the value a new development delivers. The analytics results should then lead to improvement of development methods, approval or discarding of features, prioritization of test activities, etc. A graphical overview of the approach is given in Figure 1.

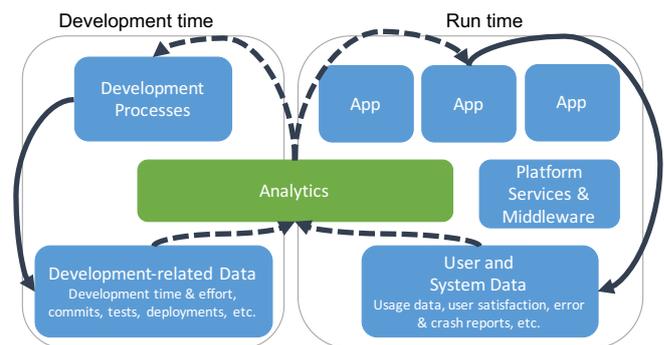


Figure 1. Overview of data-driven continuous architecture engineering.

The proposed approach relates to a number of other approaches in literature. *Evidence-based software engineering* [17] is a recently proposed vision of being able to validate any new development or change to a system from the perspective of the value it delivers. In this, new developments and changes are evaluated based on performing end-user experiments (e.g. A/B testing [18]). *Data-driven software engineering* is a different practice that focuses on continuous collection of data to quantify metrics related to produce quality and make estimates of post-release failures early in the development cycle [19]. Finally, *value-based software engineering* is a related practice that focuses on increasing a company’s business value by improving the economic efficiency of the software they develop [20].

We see two main requirements in implementing our approach: (i) software needs to be integrated and delivered in short iterations to end-users so that feedback can be obtained as early as possible, (ii) data needs to be recorded, combined and analyzed in a systematic way to derive actionable insights in optimizing a new architecture or process.

To address requirement (i), we focus on Continuous Integration (CI) practices. CI refers to the software development practice where members of a team integrate their work frequently—at least daily—leading to multiple integrations per day [4]. Each integration is performed at a shared mainline and is supported by automated software testing and building activities in order to detect integration errors as early as possible. CI is an agile practice rooted in extreme programming

methods and is reported to increase developer productivity and communication and improve release frequency and predictability. To achieve the latter, CI is usually extended by the practice of Continuous Delivery, which refers to the delivery of the (integrated) code to an environment where business logic tests and reviews can be performed [5].

To address requirement (ii), we focus on analytics on so-called “Big Data”. This refers to data analytics approaches that are able to scale to large amounts of data (e.g. petabytes or exabytes) that are typically unstructured or semi-structured, i.e. they do not follow a particular schema, as in the case of data stored in traditional relational database management systems (RDBMS). The sources of these data have been scientific experiments (e.g. in nuclear physics in CERN¹), the world wide web, and, more recently, mobile and IoT devices. From our perspective, sources of data related to data-driven continuous architecture engineering are both running systems (system logs, crash reports, user and environment data) and development tools (continuous integration servers, software repositories, issue tracking systems).

We aim to use data-driven continuous architecture engineering to address the following research questions:

- How can we ensure the quality and performance of a CoMoS platform based on Big Data analytics?
- How can we design and manage different stakeholder views in a CoMoS, including application developers and platform management?
- How can we provide an integrated CI environment for a CoMoS, considering the different devices and platform parts?
- How can we ensure the adaptability and evolution of a CoMoS based on data-driven architecture decisions?

In the rest of the section we elaborate on the different possibilities of using data to enhance the development practices and runtime aspects of CoMoSs by providing an overview of the area of software analytics. Then, in order to investigate the feasibility of this approach, in Section 3 we zoom in in Continuous Integration, while in Section 4, we provide an overview of the most prominent concepts and technologies in analyzing Big Data—a rapidly growing area in the last years.

2.1 Software Analytics for Data-Driven Continuous Architecture Engineering

Software analytics refers to obtaining insight from software artifacts to facilitate decision making [21]. It focuses on the trinity of software development, systems, and users, with the goal of improving development productivity, software quality, and user experience [22]. In general, software analytics relies on (i) large-scale computing to handle large datasets, (ii) machine learning-based analysis algorithms, and (iii) information visualization for presenting insights (Figure 2).

¹European Organization for Nuclear Research, <http://home.cern>

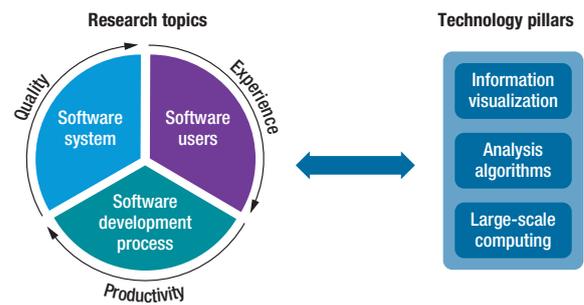


Figure 2. Research topics and technology pillars of Software Analytics (extracted from [22]): “Large-scale computing” refers to Big Data technologies, overviewed in Section 4.

Taking an architecture-centric approach, in this work package we intend to use software analytics to facilitate architecturally significant decisions in CoMoSs. In the rest of this section, we provide an overview of the software analytics domain, highlighting the latest trends.

In a recent literature review on software analytics [23], the following artifacts were identified as common data sources:

- Source code repositories for source code (including code comments)
- Version tracking systems for commits’ metadata (including commit messages) and source code versions
- Issue tracking systems for bug/defect/issue reports and issue requests
- Code reviewing systems
- Emails, mailing lists, and wikis
- User reviews
- Developers surveys
- Operating system logs, performance counters, service transaction logs
- Execution traces
- Process data, product data, organizational data, project schedules, milestones, etc.

Not only the sources of data are diverse, but also the analysis goals. For illustration, approaches presented at the latest international conference on Mining Software Repositories (May 2016) [24] included mining:

- issue reports to predict an issue’s lifetime [25];
- execution traces to prioritize code changes related to performance regressions [26];
- historical test runs to detect UI performance degradations [27];

- energy measurements of a large number of applications to predict the energy consumption of any foreign application [28].

Software analytics has also been used in a direct relation to software architecture. One such use is in architecture recovery techniques, which mine source code to extract a system's actual architecture [29]. Such techniques have been used, e.g. in the study of architecture decay, in which the extracted architectures of a system are compared across different versions [30]. They have also been used in bottom-up approaches to recommending architectural tactics based on topics and domain concepts found in the source code of a large corpus of projects [31].

The above examples highlight the possibilities in software analytics for obtaining insight from data. Independent of the concrete techniques, data sources, and analysis goals, we observe a number of trends in the area of software analytics:

1. **Use of and linkage of different artifacts.** The use of more than one artifact (e.g. use of source code and commit messages) is important; according to some authors, it is what distinguishes software analytics from direct software analysis [23]. When several artifacts are used, they should be linked together to get more complex insights.
2. **Use of distributed processing systems for analysis.** Since the size of software and related artifacts is growing and will continue to do so, software analytics approaches are increasingly considering Big Data technologies to scale the analysis efficiently [31, 32]. For example, Boa², a popular language and infrastructure for mining software repositories is based on translating the analytics tasks to MapReduce jobs that are then executed on a Hadoop cluster [33].
3. **Emphasis on actionable results.** The output of analytics has to be *insightful* information, in the sense that it conveys knowledge that is meaningful and useful for practitioners performing a specific task, but also *actionable*, i.e., information with which practitioners can devise concrete ways to complete that task [22]. This is why analytics are typically coupled by recommendations for courses of action (e.g. prioritize these features, resolve these issues first, etc.)
4. **Immediacy.** Traditional analytics consumes static historical data, performs post-hoc analysis on them, and builds models used for prediction or explanation. Since we move towards shorter release cycles, a key requirement for actionable analytics is that they must be available in *real time*—faster than the rate of change of effects within the system. Such kind of streaming analytics [32] will allow for continuous and localized

learning of models from data streams. Big Data streaming technologies such as Spark Streaming (Section 4) can be employed here.

5. **Targeting multiple software practitioners.** The recent literature review pointed out almost half of the software analytics approaches examined targeted exclusively developers [23], with some approaches targeting also project managers and testers. However, decision making is mainly performed by project managers; supporting them with actionable insight is (or should be) the main goal of software analytics [34]. At the same time, there are approaches aiming to add real time analytics capabilities to existing software products (e.g. to improve their performance) that try to separate the concerns of the data scientist from the concerns of the application developer, pointing out that the former should be responsible for devising experiments for analysis and learning [35].

3. Continuous Integration for Data-Driven Continuous Architecture Engineering

Recent research on understanding and enhancing industrial CI practices has revealed a lot of discrepancy in what is understood and implemented under the CI umbrella [36]. Actual CI systems and practices diverge at a number of variation points, as they provide different answers to questions such as how and when should teams integrate with one another, what is the mainline for each team/project/department, how are the results of automated builds fed back to the interested parties, etc. At the same time, there is a discrepancy among software professionals in the perception of the positive effects of CI [37]. Despite the overall consensus that CI brings several advantages, the presence and extent of such advantages depend on the particular variant of CI practice—in particular on how well it fits the goals and settings of the organizational unit using it.

As there clearly exists no universal CI practice with associated benefits, it is important to investigate the relationship between CI variants and their outcome. A first step towards this direction is to use an approach to accurately and unambiguously describe CI systems—embodying different variants of the CI practice—so that they can be documented, compared, and evaluated. In the following, we detail on two such approaches, after providing a comprehensive overview of the variation points within CI practice.

3.1 Variation Points in Continuous Integration

We describe here fifteen variation points in the implementation of a Continuous Integration (CI) system. Along this line, each CI system is thus considered a distinct CI *variant* and is assembled by selecting one alternative at each variation point. This variation-point analysis is based on the results on a systematic literature review investigating whether there

²<http://boa.cs.iastate.edu/>

is disparity or contention evident in the descriptions of various aspects of CI found in literature [36]. The same review has also been used as input for the creation of the modeling technique presented in Section 3.2.

- *Build duration.* Refers to the time needed between a developer checks-in a change until he/she gets notified of verdict (success/failure). It highly depends on what is included in the build (build *scope*, a separate variation point). Indicative values: several minutes, over an hour.
- *Build frequency.* Refers to how often builds are performed. This is independent from integration frequency (a separate variation point), which refers to how often changes are brought in to the product development mainline. Indicative values: several times per day, once per day, weekly.
- *Build triggering.* Builds are typically, but not exclusively, triggered by source code changes. Other triggering sources are fixed schedule (time-based triggering) and version updates of component dependencies.
- *Definition of failure and success.* A build is typically considered failed if any test fails during the build. More relaxed alternatives allow acceptance tests to break over the course of an iteration. More strict alternatives define additional success requirements such as certain level of code coverage and absence of severe code analysis warnings.
- *Fault duration.* Refers to how long it takes before a failed build is fixed. It depends on the definition of failure and success (a separate variation point). Indicative values: less than thirty minutes, within one hour, until the end of the iteration.
- *Fault handling.* Refers to how faults, once detected, are handled, i.e. by whom and in which priority. This can be performed (i) by the developer checking in the fault (assuming that the offending commit can always be identified), (ii) by developers having checked in source code since the last successful integration, (iii) by the last developer to checked in source code, (iv) by a dedicated team. Regarding priority, broken builds are typically treated as top priority tasks; however, there are more relaxed approaches where fault handling depends on the type and severeness of the fault.
- *Integration frequency.* Refers to how often developers check in changes in source code. It is in general independent from build frequency (a separate variation point). While the term "continuous" integration hints towards the continuous checking in of changes—leading to very high integration frequencies—it is generally expected that developers integrate their changes every few hours or at least once per day.
- *Integration on broken builds.* Refers to whether checking-in on top of revisions that failed are allowed. Alternatives range from strict ones involving automated blocking of check-ins on broken builds, to discouraging (without preventing) developers to check in on broken builds, to allowing check-ins at any time.
- *Integration serialization and batching.* While checked-in changes typically trigger new builds, there are alternative ways to handle cases where multiple changes are made during the timespan of a single (long-running) build. The two extremes are (i) serializing the check-in process so that each check-in builds the mainline on an integration machine, and (ii) batching all accumulated check-ins into a single build.
- *Integration target.* This aspect refers to where developers check in their changes, i.e. to which branch in the version control system. Alternatives include merging directly into the mainline, using a development branch for merging and then pushing revisions to the mainline, and using team-specific integration branches.
- *Modularization.* CI is typically not modularized, which means that the entire software is built and tested upon changes. Noteworthy alternatives include products built by large number of components, the source code of which is controlled independently. In such cases, each component follows its own CI cycle. Such modularized approaches are claimed to reduce feedback times, as only the components that are affected by the changes are rebuilt and re-tested [36].
- *Pre-integration procedure.* Refers to the actions performed prior to checking in source code. Alternatives range from relaxed ones, where no local testing is assumed, to more thorough ones, where developers are expected to manually compile, develop, and run the unit tests suite, to strict ones involving code reviews and implementing and locally executing the necessary unit and integration tests before checking in changes.
- *Scope.* Refers to the amount and type of activities included in the CI practice. CI includes at a minimum source code compilation & unit testing. Extensions include more advanced testing activities such as integrations tests, functional and/or non-functional (e.g. performance) system tests, and/or acceptance tests, static and/or dynamic code analysis activities, packaging activities, and deployment activities (typically regarded as part of Continuous Delivery).
- *Status communication.* This aspect is concerned with who to communicate the CI status, e.g. notifications of build failures and how. Alternative notification targets are (i) the last person to check in, (ii) the whole development team, (iii) team leaders only. Alternative

communication methods include (i) emails, (ii) RSS feeds, (iii) web pages, and (iv) dashboards.

- *Test separation.* Refers to segmenting test suites into multiple parallel or sequential activities. The most common approach is to have monolithic test suites. In case there is test segmentation, this is based on either functional areas or components. Test separation is also typically performed at the time axis by separating short (e.. unit tests) from long-running (e.g. acceptance tests, performance tests) test activities. The idea is that slower tests have different triggering conditions and frequencies.

Optimizing CI to the specific needs and settings of a company/project involves coming up with a CI variant by picking one option at each the variation point that maximizes the benefits of using CI in the company/project. Such tailoring of CI involves design trade-offs, as choosing one alternative over another might increase certain benefits and decrease others at the same time. As an example, choosing the fault handling alternative of always having the developer who last checked in changes fix the broken build may decrease the developer's productivity (measured e.g. by number of commits per day), while it eases communication (measured e.g. by number of emails exchanged).

In the following, we present a modeling technique for documenting CI variants that can be used as a basis for design exploration and trade-off analysis.

3.2 Automated Software Integration Flow

The Automated Software Integration Flow (ASIF) is a modeling technique that offers a graphical view of a CI system where nodes represent inputs, activities, and triggers and edges represent consuming relationships and triggering relationships [38, 36].

A CI system in ASIF is essentially represented by a Directed Acyclic Graph (DAG) of interconnected automated activities, based on the fact that a “build” typically consists of a number of interconnected tasks which conditionally trigger each other and which may be executed sequentially, in parallel or on different schedules altogether. The DAG captures the “integration flow anatomy” of the CI system under study and documents choices related to the modularization and build triggering variation points (Section 3.1). The rest of the variation-point choices are documented as attributes to the input and activity nodes of the DAG.

From a different perspective, ASIF can also be considered as a *domain specific modeling language* which focuses on the domain of CI. As such, the DAG representation is just a concrete graphical syntax of the language. Its meta-model, together with the mapping of language concepts, such as automated activities and inputs, to diagrammatic elements is depicted in Figure 3.

An example of an ASIF model is depicted in Figure 4. The Meta-model specifies that automated activities, inputs

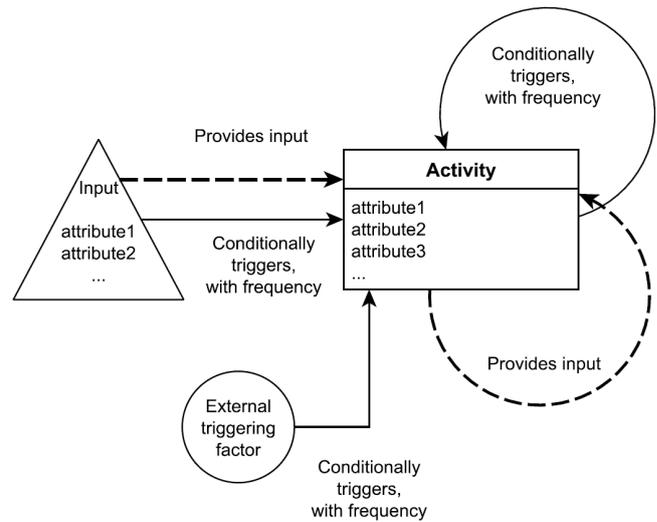


Figure 3. Meta-model of ASIF, extracted from [38].

and triggers are represented by rectangles, triangles, and circles, respectively. Consuming and triggering relationships are represented by dashed and solid lines, respectively. It refers to a product development project at Ericsson AB. As depicted in the diagram, changes in the software configuration management (SCM) provide input to the daily product build and the daily unit test and analysis activities. The daily build is triggered both manually and in an automatic schedule. It receives input from external dependencies and from the SCM mainline and provides input to an acceptance test activity and a product smoke test screening activity. In parallel to the main acceptance test activity, which typically takes up to 7 hours to complete, a basic subset of acceptance tests are periodically scheduled (hourly basic acceptance test). This activity uses the latest product build that passed the screening activity, with delta packages added on top as parts of the product are changed.

ASIF does not prescribe which attributes to use in the model, as this depends in general on the particular goal and context of using the technique. For example, if one is focusing on end-to-end timing analysis of a CI system, only the time-related attributes (duration, execution frequency, etc.) should be included. However, based on experience in using the technique in industry [39] the following *input* attributes are important:

- type of branch/repository—e.g. private, team, development, release.
- steps required before integrating new code—e.g. reviews, local tests, manual approval.

Similarly, the following *activity* attributes are deemed important:

- average duration of activity, measured in minutes.

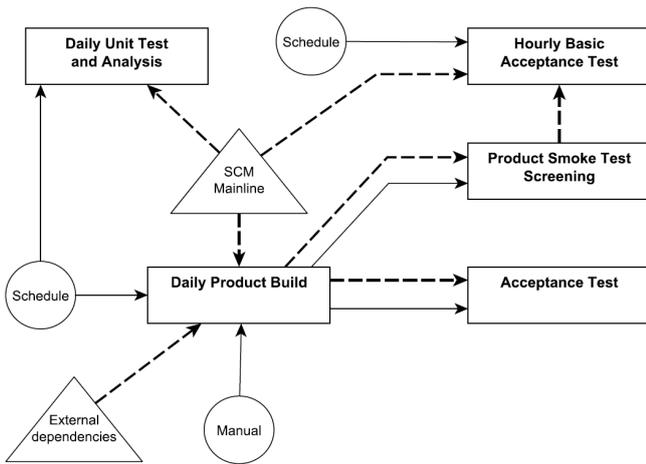


Figure 4. An example of an ASIF model, extracted from [40]. Figure 3 can be used as a legend to this diagram. Node attributes (e.g. average duration of the `Acceptance Test` activity) have been omitted for readability.

- whether any form of static or dynamic code analysis is performed—e.g. memory consistency, code coverage, style checks, complexity analysis.
- whether the activity involves any kind of deployment—e.g. to User Acceptance Testing, to customers.
- definition of success of the activity—e.g. “tests passed”, “metrics are satisfactory”, “artifacts are built”.
- how the activity status is communicated—e.g. emails, web pages, reporting meetings.
- average interval between failure and first subsequent success of activity (fault duration)—e.g. minutes, hours, days.

Even though the ASIF model is fairly simple, using more attributes (for example, with the intention to cover all the variation points of Section 3.1) can quickly make it information-dense. Including more information is generally facilitating different kinds of analysis, but makes the model more difficult to create and maintain.

So far, ASIF has been evaluated in two multiple-case studies with promising results [40, 39]. In particular, it provides a common language for documenting CI systems in a tool-agnostic way. This has shown to improve the understanding and communication of complex CI systems even among the engineers of these systems; it has also shown potential in identifying and planning improvements to those systems.

3.3 Continuous Integration Visualization Technique

The Continuous Integration Visualization (CIViT) technique provides a graphical representation of the end-to-end testing activities of a product or a product platform [41, 42]. End-to-end testing includes all verification and validation activities

that range from unit tests performed when a developer checks in code to product release testing. CIViT has been inspired by the results of a multiple-case study with the aim of improving the understanding of how testing activities are arranged in industry settings [41]. Its goal is to support companies in implementing CI by providing appropriate communication means about test activities and their coverage.

In CIViT, each testing activity is modeled as a rectangle split into four parts representing different types of testing (Figure 5). *New functionality* testing refers to testing the functionality currently under development. *Legacy functionality* testing refers to testing of functionality that has already been built and operates correctly. *Quality requirements* testing refers to testing of performance, safety, security, and other qualities of the system under test. Finally, *edge case* testing refers to testing “unlikely or weird situations” [41], often discovered through considerable investigative effort.

For each testing type, two dimensions are deemed to be of importance and are consequently depicted in CIViT: the degree of coverage and the level of test automation. Color coding is used to intuitively depict ranges in the values of these dimensions, as described in Figure 5. It is important to note, though, that obtaining a conclusive test coverage estimation is difficult, as manually estimated numbers are found to be highly subjective in previous case studies [39].

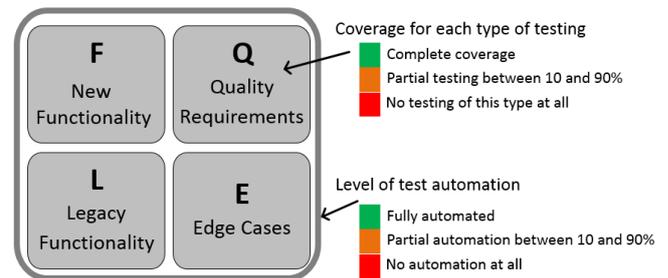


Figure 5. The four testing types in CIViT and their color coding scheme.

Having the modeling construct of a testing activity in place, CIViT provides a two-dimensional graph, where different testing activities are mapped out (Figure 6 provides an example). The dimensions of the graph are *scope* and *periodicity*, which are common among all testing activities. Scope refers to the segment of the overall system that is tested and has five values ranging from `Component` (a small system module that is developed from a single developer or a small team) to `Customer` (testing at the customer’s site). Periodicity refers to the average time between the start of a testing activity and the acquiring of feedback (e.g. success/failure verdict, failing modules/components, etc.) from the activity.

The strength of CIViT lies in providing documentation and overview over the testing activities of a product. This can in turn prove useful in identifying problems such as slow feedback loops, duplicate testing activities, missing coverage

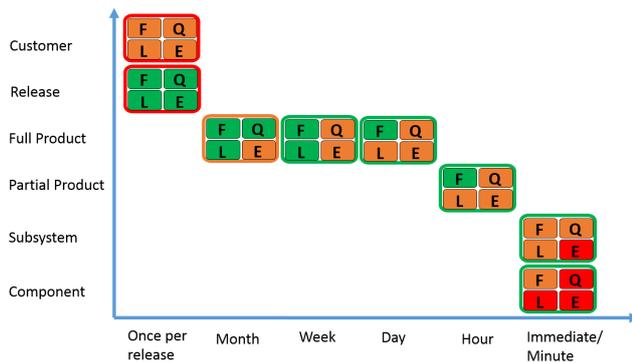


Figure 6. A hypothetical example of a CIViT model. We can observe that automatic testing is performed within minutes, hours, and days focusing on the new functionality (unit tests, integration tests). Additionally, semi-automatic testing is performed on the full product on a weekly and monthly basis; additional manual testing is performed once during releases and at customer site.

of particular type of testing. It can also help prioritize the improvements in the testing infrastructure, by e.g. increasing the periodicity, scope, coverage or automation of a particular testing activity.

3.4 Combining ASIF with CIViT

Since ASIF focuses on modeling software integration flows while CIViT focuses on modeling end-to-end testing activities, they can be viewed as complementary and we can try to combine them in a single architecture framework, where each of the techniques will represent a different view over the CI system. Initial evidence on the potential of such an alignment has been provided by a recent study where ASIF and CIViT were both used to model four CI systems in industry [39].

4. Big Data Technologies for Data-Driven Continuous Architecture Engineering

In Big Data technologies, Hadoop³ has become the de facto standard over the past ten years. Hadoop is an open source ecosystem of tools supported by the Apache Software Foundation⁴. The main advantage of Hadoop is that it provides a way to perform cost-effective analytics using commodity (i.e. no special-purpose) servers in an unprecedented scale. It is being used by a number of top companies such as Yahoo!, Microsoft, LinkedIn, Facebook, Twitter, with the notable exception of Google, which builds and maintains its own suite of Big Data tools.

We will try to explain the success of Hadoop by examining some of its core constituents, i.e. its distributed file system and its popular programming model. We will then zoom in in two important technologies in performing analytics in a Hadoop

environment. Finally, we will describe a relatively new programming framework that can be “attached” to Hadoop and offers orders-of-magnitude increase in performance of analytics computations.

4.1 Hadoop Distributed File System

The Hadoop Distributed File System (HDFS) is the most common choice for the storage layer of a Hadoop installation⁵. HDFS is a distributed file system that scales to thousands of nodes (i.e. servers) and provides built-in fault tolerance by data replication [43, 44, 45]. As a stark difference to other filesystems, HDFS follows a write-once-read-many operational model where a file cannot be changed once created, written, and closed. This prevents many data coherency issues and enables high throughput data access, while it still fits well the requirements of MapReduce-based and of analytics applications in general.

HDFS is implemented as a userspace filesystem in Java, which uses the native filesystem at each node, such as ext3 or NTFS, to store data. Files in HDFS get divided into blocks of typically 128 MB which get stored as separate files in the local filesystem. A replication factor (typically three) determines how many identical copies will be created and saved in the cluster. Having additional copies of a single block (and consequently of a single file) allows for a high degree of fault-tolerance in cases when nodes become unavailable.

HDFS stores file system metadata and application data separately. For metadata storage, one node in the cluster implements the centralized NameNode service which is responsible of maintaining the HDFS directory tree and the mapping between an HDFS file name, its blocks, and the nodes on which these blocks are stored. These nodes implement the DataNode service which is responsible for storing data blocks on behalf of local or remote clients. A client is a library that provides an Application Programming Interface (API) to users, with common operations such as reading, writing, and deleting files and directories.

To read or write a file, a client application contacts the NameNode to obtain a list of blocks and destinations (i.e. DataNodes) from which to read or write. In case of reading, the client establishes connection to the “closest” DataNode and requests specific block IDs. In case of writing, the client pipelines the blocks that constitute the file to be written to the DataNodes chosen by the NameNode. Two important remarks are that (i) HDFS makes use of the locality of the nodes to increase read bandwidth, and (ii) having a dedicated coordination point helps in load-balancing in the cluster (as DataNodes are chosen also based on their load). Finally, it is important to stress that the client is shielded from all the complexity of replication management and query routing; the internal workings between the two services are transparent to the client that views a simple API.

One of the main criticisms of earlier HDFS implementa-

³<http://hadoop.apache.org/>

⁴<http://www.apache.org/>

⁵Other options include Amazon’s S3, FTP, and Windows Azure Storage Blobs.

tions was that the NameNode introduced a single point of failure. In response, in later stages a backup service in the form of a secondary NameNode was added.

A simplified version of the HDFS architecture is depicted in Figure 7.

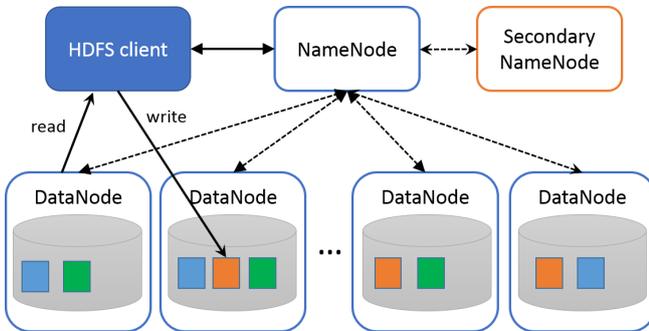


Figure 7. Simplified HDFS architecture.

4.2 MapReduce

MapReduce (MR) is a parallel programming model that was described in a seminal paper by Google in 2004 [46]. It can be used for processing large datasets and is amenable to a number and variety of real-world analytics tasks. Its main advantages is its conceptual simplicity in combination with the fact that it partially shields developers from the complexity of parallel and distributed programming [47].

In the MR model, a user-defined program is divided into a *map* function and a *reduce* function. The *map* function takes as input a key/value pair and produces one or more intermediate key/value pairs. The *reduce* function takes as input a pair resulting from grouping together intermediate values with the same intermediate key (a task typically performed by an extra *combiner* function) and produces a single value as output. The *map* and *reduce* contain all the application logic of a program.

The canonical example of MapReduce is an application that counts the words in a potentially very large document. In this case, the *map* function receives as input a set of pairs of $\langle id, line \rangle$ and returns a set of pairs of $\langle word, 1 \rangle$ for each word it encounters. A *reduce* function receives a pair of $\langle word, integer\ list \rangle$, adds the integers in the list (which correspond to occurrences of the word) and returns the results.

As indicated also by the example, the *map* and *reduce* functions can run independently on each pair, allowing for enormous amounts of parallelism. Indeed, an MR program (also called job) typically spawns several hundreds of identical *map* and *reduce* tasks, each of which receives different input. Figure 8 shows the pattern of a MR program.

MR is implemented in Hadoop by two Java services, JobTracker and TaskTracker [48]. JobTracker is a centralized service responsible for splitting the input data into pieces for processing in the individual *map* and *reduce* tasks and for scheduling each task on a cluster node for execution. Each node runs the TaskTracker service which

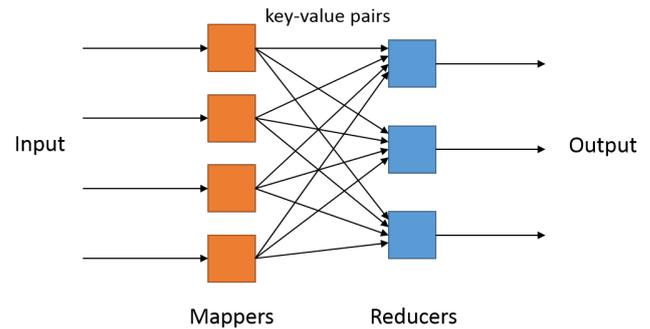


Figure 8. Overview of MapReduce pattern.

reports back to the JobTracker on task execution completion. An important remark here is that MR tasks have the blocking property, which means that no output is used until the task is completed. This allows for recovering from failures (e.g. node downtimes) by restarting tasks on healthy nodes. This is also a responsibility of JobTracker in Hadoop.

Finally, it should be noted that although Hadoop MR is written in Java, users can issue MR jobs written in different languages, e.g. Python or R.

4.3 Hive

Hive is a data warehousing solution built on top of Hadoop [49]. Its main goal is to simplify the querying and analysis tasks in Hadoop by providing a familiar SQL-like syntax for performing these tasks. Hive alleviates the problem of writing custom MR programs that are hard to maintain and reuse and allows non-programmers to interact with Hadoop for reporting and ad-hoc data analysis.

Hive provides an SQL-like declarative language called *HiveQL* for specifying queries. Queries are internally compiled into MapReduce jobs⁶ and executed on a Hadoop cluster. In particular, Hive supports Data Definition (DDL) statements for creating tables, data manipulation (DML) statements such as *load*, and typical SQL statements such as *select*, *join*, *union*, *group by*, *order by*, etc.

Database schemas are kept in a system catalog called *metastore*, which is physically stored in a relational database. When working with Hive, a user can create tables schemas and load data to them from files in the HDFS. (This effectively means that files are moved to the Hive-controlled filesystem namespace of HDFS.) Hive supports reading and writing in a number of serialization formats including CSV and JSON.

Once a query is issued, it gets translated into an execution plan. In case of DDL statements, the plan consists only of metadata operations, while *LOAD* statements are translated to HDFS operations. In case of *INSERT* statements and regular queries, the plan consists of a directed-acyclic graph (DAG) of MapReduce jobs, which get executed in the Hadoop cluster.

⁶Hive can also compile to Apache Tez and Spark jobs.

```

1. -- Create a list of all parking slots and their position
2. A = LOAD 'parkingSlots.csv' as
3.   (pl_id:int, available:int,
4.    longitude:double, latitude:double, timestamp:long);
5. B = FOREACH A GENERATE pl_id, longitude, latitude;
6. C = DISTINCT B;
7.
8. -- Create a list of all cars that are driving
9. D = LOAD 'cars.csv' as
10.  (car_id:int, speed:int, longitude :double,
11.   lat:double, timestamp:long);
12. E = FILTER D BY speed > 5.0;
13.
14. -- Join the data by GPS distance
15. F = JOIN C by timestamp, E by timestamp;
16. G = FOREACH F GENERATE *, Distance(C::longitude,
17.   C::longitude, E:: longitude, E::latitude)
18.   as distance;
19. H = FILTER G BY distance < 5.0;
20.
21. -- Count the amount of cars for each PL
22. I = GROUP H BY pl_id;
23. J = FOREACH I {
24.   distCars = DISTINCT H.car_id;
25.   GENERATE $0, COUNT(distCars);
26. };

```

Figure 9. An example Pig program that calculates the number of moving cars in the vicinity of a parking slot, for each point in time, based on the positions of parking slots and cars driving in a city.

4.4 Pig

Pig [50, 51] is a scripting layer on top of Hadoop MR. It can be used as alternative to Hive for simplifying the querying and analysis tasks. However, whereas Hive targets data analysts with SQL expertise, Pig targets mainly developers with procedural programming expertise.

Pig provides a procedural query language called *Pig Latin* [50, 52]. A Pig Latin program is a sequence of statements, each of which specifies only a single data transformation. Statements are constructed with the use of SQL-style high-level data manipulation constructs, e.g. JOIN, GROUP, ORDER, DISTINCT, FILTER, FOREACH, and others. An illustrative example is depicted in Figure 9. As an important difference to SQL, where only flat tables are allowed, Pig Latin has a nested data model that allows non-atomic data types such as tuple, set, and map to occur as fields of a table. This provides more intuitive and flexible programming abstractions.

Apart from using its built-in constructs, Pig allows users to provide User-Defined Functions (UDFs), typically written in Java, that extend the functionality of Pig. As an example, the `Distance()` UDF (Figure 9, line 16) returns the Euclidean distance of two positions.

A Pig Latin program essentially can be represented by a directed acyclic graph (DAG) where nodes represent data transformations and links represent data flow. This is called *logical plan*. Logical plans get translated to physical plans, which in turn get translated to MR jobs⁷ by the Pig compiler. As an example, the example program of Figure 9 is represented by the DAG of Figure 10 and then split into two MR jobs as indicated in the Figure.

⁷Pig can also compile to Apache Tez and Spark jobs.

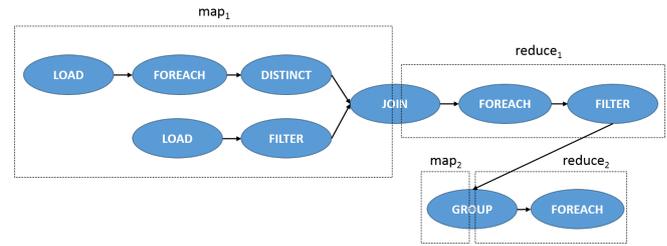


Figure 10. Logical plan for example Pig program and its mapping to MR jobs.

4.5 Spark

Spark is a computing framework for large clusters⁸ [53]. It has been conceived to deal with two main shortcomings of traditional MR-based computations on top of HDFS: (i) they do not support interactive data exploration and analytics due to high latency in the scale of minutes and hours, and (ii) they do not support iterative jobs, where a function is repeatedly applied to a dataset—a common case in many multi-pass machine learning computations. Spark deals with both these issues by keeping data in memory at each cluster node and preventing the reloading of data from disk as much as possible.

The main abstraction in Spark is that of a Resilient Distributed Dataset (RDD) [53, 54, 55]. An RDD is a read-only, partitioned collection of records. RDDs can only be created by deterministic operations on (i) data in non-volatile storage (e.g. HDFS) and (ii) other RDDs via transformations such as *map*, *filter*, *sort*, *join*, and *union*. RDDs do not have to be materialized at all times; instead, an RDD has enough information about how it was derived from other RDDs (and transitively from other stable datasets)—its origin or *lineage*—to reconstruct itself by computing its partitions from stable storage. This provides strong fault tolerance and recoverability.

Each RDD is represented in Spark via a common interface that exposes: (i) the set of partitions, which are atomic pieces of the dataset, (ii) a set of dependencies on parent RDDs, (iii) a function for computing the dataset based on its parents, (iv) metadata about its partitioning scheme and data placement (to support data locality of operations). Dependencies are distinguished between *narrow* ones, where each partition of the parent RDD is used by at most one partition of the child RDD (e.g. results of *map* or *filter* operations), and *wide* ones, where multiple child partitions may depend on it (e.g. results of *group* and *join* operations). While the former can (and should) be computed on a single cluster node in a pipelined fashion, the latter require data from all parent partitions to be shuffled across the nodes.

Another main abstraction in Spark is that of *shared variables* across all cluster nodes. These can be either *broadcast variables* or *accumulators*, referring to read-only data such as lookup tables and variables with add-only semantics, used to

⁸<http://spark.apache.org/>

conveniently implement parallel sums, respectively.

Spark is implemented in Scala⁹, a JVM-based language with functional features such as closures. The RDD abstraction is thus provided as a language-integrated API in Scala. To use Spark, developers write a *driver program* that connects to a cluster of *workers*. The driver defines one or more RDDs and invokes actions on them. Actions are specified by passing Scala closures (function literals) as parameters to generic RDD operations. For example, the following example passes a function that checks whether the “ERROR” substring is included to the *filter* RDD operation.

```
val file = spark.textFile("hdfs://...")
val errors = file.filter(_.contains("ERROR"))
```

Apart from transformation RDDs operators that produce new RDDs, Spark supports RDDs actions that produce either a single output (e.g. *count*, *reduce*), multiple outputs (e.g. *collect*) or outputs to stable storage (e.g. *save*). When such actions are executed, Spark’s scheduler examines the RDD’s lineage graph to produce a DAG of execution stages (Figure 11). The boundaries for these stages are the shuffle operations required for wide dependencies. The scheduler launches tasks to compute the missing RDD partitions from each stage until the target RDD is computed. Tasks are assigned to workers based on data locality using delay scheduling [56].

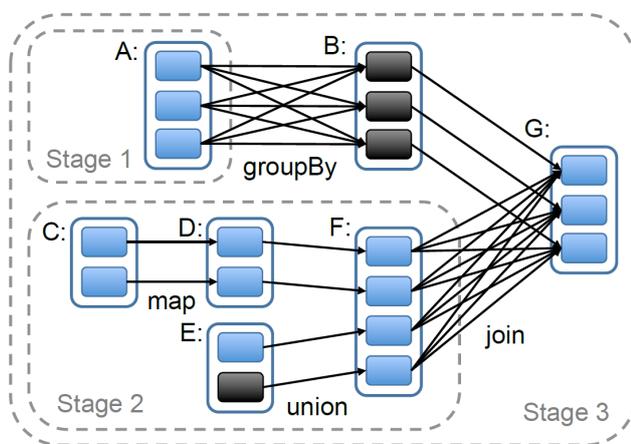


Figure 11. Example of Spark execution stages: boxes with solid outlines are RDDs, black rectangles represent partitions that are already in memory. Extracted from [55].

Spark can run over different cluster managers including Apache Mesos¹⁰, Hadoop YARN [57], Amazon EC2¹¹, and its built-in standalone cluster manager. Apart from Scala, it allows writing driver programs in Java, Python and R. Most importantly, it comes with a number of accompanying libraries to support real-time SQL querying (Spark SQL, successor of Shark project [58]), graph processing (GraphX [59]), machine

learning (MLlib¹²) and stream analytics based on discretized streams (Spark Streaming [60]). Spark enjoys a very active open source community support having more than 1000 contributors [61].

4.6 Discussion

In this section, we described the Big Data tools related to Hadoop and the concepts behind them that have “stood the test of time”. There have been many other projects that are partially overlapping either in goals or in functionality with the ones described here:

- Dryad [62] was a Microsoft framework for parallel computing that stood as an alternative to Hadoop MapReduce (MR), but was discontinued in 2013, when Microsoft switched to Hadoop for its Big Data solutions.
- Apache Impala¹³, Apache Drill¹⁴ (an open source version of Google’s Dremel [63]), and Presto¹⁵ are all different SQL querying engines for Hadoop and thus alternatives to Hive. They do not rely on MR for job execution, but on their own execution engines.
- Apache Tez¹⁶ is a successor of Hadoop MR engine, which offers increased performance by combining multiple MR jobs into a single Tez job represented by a DAG of tasks. Hive and Pig are currently using Tez by default; however, compilation to MR jobs is still an option in latest releases.

At the same time, the Hadoop ecosystem consists of a number of other important components not described in this document, as they do not focus specifically on analytics. For instance, Apache YARN [57] (stands for “Yet Another Resource Negotiator”) is a distributed application management framework. It was introduced in the 2nd major release of Hadoop with the goal to decouple MR’s resource management and scheduling capabilities (part of YARN) from the data processing components (part of so-called MapReduce 2.0). YARN can manage the resources of non-MapReduce workloads (e.g. of graph processing systems such as Apache Giraph¹⁷).

Finally, a current hot topic in Big Data analytics is stream processing. In this arena, the main competitors are Spark Streaming [60] and Apache Flink¹⁸ [64]. While the first one relies on micro-batches, the second one supports scanning of incoming data as they come—tuple-at-a-time semantics. Apache Storm¹⁹ was also a notable tool for stream processing, but its popularity is decreasing since the advent of Flink (it

¹²<http://spark.apache.org/mllib/>

¹³<http://impala.io/>

¹⁴<https://drill.apache.org/>

¹⁵<https://prestodb.io/>

¹⁶<https://tez.apache.org/>

¹⁷<http://giraph.apache.org/>

¹⁸<https://flink.apache.org/>

¹⁹<http://storm.apache.org/>

⁹<http://www.scala-lang.org/>

¹⁰<http://mesos.apache.org/>

¹¹<https://aws.amazon.com/ec2/>

is indicative that even Twitter, the biggest supporter of Storm has recently switched to Flink).

5. Conclusions

In this report, we first briefly described the challenges of a systematic development and maintenance of Connected Mobility Systems (CoMoSs) and of obtaining a reference architecture for such systems. We then described the concept of “data-driven software architecture engineering” that we plan to employ in improving the development and testing methods and processes for CoMoSs. According to this, data related to both the runtime phase of a CoMoS and the development life cycle should be recorded and analyzed in order to identify correlations between development methods and end-products, perform experiments measuring end-user behavior and generally assessing the value a new development delivers. In other words, the approach proposes the use of software analytics in informing architecturally significant decisions.

In the main part of the report, we focused on what we regard as the two main requirements for data-driven software architecture engineering, i.e. Continuous Integration (CI) and Big Data analytics. In particular, we first described state-of-the-art methods for documenting and eventually improving the state of the CI practice. Subsequently, we provided an overview of Big Data analytics concepts and tools focusing on the Hadoop ecosystem.

We are currently experimenting with the different technologies in the Big Data analytics area. In particular, we have been implementing a graphical tool to ease the definition of analytics based on the Pig Latin language [50]. We have also been working on a reference problem/exemplar for Big Data analytics in the domain of CoMoSs, with the goal to identify interesting challenges in integrating Big Data analytics to a real-life example. Finally, we have started investigating the problem of preserving privacy constraints in the integration of results from Big Data analytics jobs run in different clusters. This can enable to transcend the organizational and company silos in analytics with the hope of deriving even more useful insights.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] Roland Berger Strategy Consultants. think:act Study – Connected Mobility 2025, January 2013. https://www.rolandberger.de/media/pdf/Roland_Berger_TaS_Connected_Mobility_E_20130123.pdf.
- [2] Mike Cohn. *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley Professional, Upper Saddle River, NJ, 1 edition edition, November 2009.
- [3] Karl Scotland. Aspects of Kanban. <http://www.methodsandtools.com/archive/archive.php?id=104>.
- [4] Martin Fowler. Continuous Integration. <http://martinfowler.com/articles/continuousIntegration.html>.
- [5] Jez Humble and David Farley. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley, 2010.
- [6] A. Jansen and J. Bosch. Software Architecture as a Set of Architectural Design Decisions. In *5th Working IEEE/I-FIP Conference on Software Architecture (WICSA'05)*, pages 109–120, 2005.
- [7] Robert Cloutier, Gerrit Muller, Dinesh Verma, Roshanak Nilchiani, Eirik Hole, and Mary Bone. The Concept of Reference Architectures. *Syst. Eng.*, 13(1):14–27, February 2010.
- [8] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A Survey. *Comput. Netw.*, 54(15):2787–2805, October 2010.
- [9] NIST. Cyber-Physical Systems: Situation Analysis of Current Trends , Technologies , and Challenges. Technical report, 2012.
- [10] By Kyoung-dae Kim and P R Kumar. Cyber-Physical Systems: A Perspective at the Centennial. *Proceedings of the IEEE*, 100(Special Centennial):1287–1308, 2012.
- [11] Internet of Things - Architecture EU project. <http://www.iot-a.eu>.
- [12] Collaborative Open Market to Place Objects at your Service EU project. <http://www.compose-project.eu/>.
- [13] Reliable Smart Secure Internet of Things for Smart Cities EU project. <http://www.almanac-project.eu>.
- [14] Architecture for Multi-criticality Agile Dependable Evolutionary Open System-of-Systems EU project. <http://amadeos-project.eu/>.
- [15] Trusted Apps for open CPSs EU project. <http://www.tapps-project.eu/>.
- [16] Design, Monitoring and Operation of Adaptive Networked Embedded Systems EU project. <http://www.deman.es.eu/>.
- [17] Jan Bosch and Helena Holmström Olsson. Data-driven continuous evolution of smart systems. pages 28–34. ACM Press, 2016.
- [18] Ron Kohavi, Thomas Crook, and Roger Longbotham. Online Experimentation at Microsoft. In *Third Workshop on Data Mining Case Studies and Practice*, 2009.

- [19] Christian Bird, Brendan Murphy, Nachiappan Nagappan, and Thomas Zimmermann. Empirical Software Engineering at Microsoft Research. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work, CSCW '11*, pages 143–150, New York, NY, USA, 2011. ACM.
- [20] Barry W. Boehm. Value-Based Software Engineering: Overview and Agenda. In Stefan Biffl, Aybüke Aurum, Barry Boehm, Hakan Erdogmus, and Paul Grünbacher, editors, *Value-Based Software Engineering*, pages 3–14. Springer Berlin Heidelberg, 2006. DOI: 10.1007/3-540-29263-2_1.
- [21] Tim Menzies and Thomas Zimmermann. Software analytics: so what? *IEEE Software*, 30(4):31–37, 2013.
- [22] Dongmei Zhang, Shi Han, Yingnong Dang, Jian-Guang Lou, Haidong Zhang, and Tao Xie. Software Analytics in Practice. *IEEE Software*, 30(5):30 – 37, October 2013.
- [23] Tamer Mohamed Abdellatif, Luiz Fernando Capretz, and Danny Ho. Software Analytics to Software Practice: A Systematic Literature Review. In *Proceedings of the First International Workshop on BIG Data Software Engineering, BIGDSE '15*, pages 30–36, Piscataway, NJ, USA, 2015. IEEE Press.
- [24] *MSR '16: Proceedings of the 13th International Conference on Mining Software Repositories*, New York, NY, USA, 2016. ACM.
- [25] Riivo Kikas, Marlon Dumas, and Dietmar Pfahl. Using dynamic and contextual features to predict issue lifetime in GitHub projects. pages 291–302. ACM Press, 2016.
- [26] Qi Luo, Denys Poshyvanyk, and Mark Grechanik. Mining performance regression inducing code changes in evolving software. pages 25–36. ACM Press, 2016.
- [27] María Gómez, Romain Rouvoy, Bram Adams, and Lionel Seinturier. Mining test repositories for automatic detection of UI performance regressions in Android apps. pages 13–24. ACM Press, 2016.
- [28] Shaiful Alam Chowdhury and Abram Hindle. GreenOracle: estimating software energy consumption with energy measurement corpora. pages 49–60. ACM Press, 2016.
- [29] Thibaud Lutellier, Devin Chollak, Joshua Garcia, Lin Tan, Derek Rayside, Nenad Medvidović, and Robert Kroeger. Comparing Software Architecture Recovery Techniques Using Accurate Dependencies. In *Proceedings of the 37th International Conference on Software Engineering - Volume 2, ICSE '15*, pages 69–78, Piscataway, NJ, USA, 2015. IEEE Press.
- [30] Duc Minh Le, Pooyan Behnamghader, Joshua Garcia, Daniel Link, Arman Shahbazian, and Nenad Medvidovic. An Empirical Study of Architectural Change in Open-Source Software Systems. pages 235–245. IEEE, May 2015.
- [31] Mehdi Mirakhorli, Hong-Mei Chen, and Rick Kazman. Mining Big Data for Detecting, Extracting and Recommending Architectural Design Concepts. In *Proceedings of the First International Workshop on BIG Data Software Engineering, BIGDSE '15*, pages 15–18, Piscataway, NJ, USA, 2015. IEEE Press.
- [32] Georgios Gousios, Dominik Safaric, and Joost Visser. Streaming Software Analytics. In *Proceedings of the 2Nd International Workshop on BIG Data Software Engineering, BIGDSE '16*, pages 8–11, New York, NY, USA, 2016. ACM.
- [33] Robert Dyer, Hoan Anh Nguyen, Hridesh Rajan, and Tien N. Nguyen. Boa: A Language and Infrastructure for Analyzing Ultra-large-scale Software Repositories. In *Proceedings of the 2013 International Conference on Software Engineering, ICSE '13*, pages 422–431, Piscataway, NJ, USA, 2013. IEEE Press.
- [34] Raymond PL Buse and Thomas Zimmermann. Analytics for software development. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, pages 77–80. ACM, 2010.
- [35] Patrick Tendick and Audris Mockus. Decisions As a Service for Application Centric Real Time Analytics. In *Proceedings of the 2Nd International Workshop on BIG Data Software Engineering, BIGDSE '16*, pages 1–7, New York, NY, USA, 2016. ACM.
- [36] Daniel Ståhl and Jan Bosch. Modeling continuous integration practice differences in industry software development. *Journal of Systems and Software*, 87:48–59, January 2014.
- [37] Daniel Ståhl and Jan Bosch. Experienced benefits of continuous integration in industry software product development: A case study. In *The 12th IASTED International Conference on Software Engineering, (Innsbruck, Austria, 2013)*, pages 736–743, 2013.
- [38] Daniel Ståhl and Jan Bosch. Continuous Integration Flows. In Jan Bosch, editor, *Continuous Software Engineering*, pages 107–115. Springer International Publishing, 2014. DOI: 10.1007/978-3-319-11283-1_9.
- [39] Daniel Ståhl and Jan Bosch. Industry application of continuous integration modeling: a multiple-case study. pages 270–279. ACM Press, 2016.
- [40] Daniel Ståhl and Jan Bosch. Automated Software Integration Flows in Industry: A Multiple-case Study. In *Companion Proceedings of the 36th International Conference on Software Engineering, ICSE Companion 2014*, pages 54–63, New York, NY, USA, 2014. ACM.
- [41] Agneta Nilsson, Jan Bosch, and Christian Berger. Visualizing Testing Activities to Support Continuous Integration: A Multiple Case Study. In Giovanni Cantone and Michele Marchesi, editors, *Agile Processes in Software Engineering and Extreme Programming*, number 179 in Lecture Notes in Business Information Processing, pages

- 171–186. Springer International Publishing, May 2014. DOI: 10.1007/978-3-319-06862-6_12.
- [42] Agneta Nilsson, Jan Bosch, and Christian Berger. The CIViT Model in a Nutshell: Visualizing Testing Activities to Support Continuous Integration. In *Continuous Software Engineering*, pages 97–106. Springer, 2014.
- [43] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The Hadoop Distributed File System. In *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST) '10*, pages 1–10, Washington, DC, USA, 2010. IEEE Computer Society.
- [44] Dhruva Borthakur. HDFS Architecture Guide, April 2013. https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.
- [45] J. Shafer, S. Rixner, and A. L. Cox. The Hadoop distributed filesystem: Balancing portability and performance. In *2010 IEEE International Symposium on Performance Analysis of Systems Software (ISPASS)*, pages 122–133, March 2010.
- [46] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation - Volume 6, OSDI'04*, pages 10–10, Berkeley, CA, USA, 2004. USENIX Association.
- [47] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM*, 51(1):107–113, January 2008.
- [48] Apache. Hadoop MapReduce Tutorial, April 2013. https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html.
- [49] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. Hive: A Warehousing Solution over a Map-reduce Framework. *Proc. VLDB Endow.*, 2(2):1626–1629, August 2009.
- [50] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig Latin: A Not-so-foreign Language for Data Processing. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08*, pages 1099–1110, New York, NY, USA, 2008. ACM.
- [51] Alan F. Gates, Olga Natkovich, Shubham Chopra, Pradeep Kamath, Shravan M. Narayanamurthy, Christopher Olston, Benjamin Reed, Santhosh Srinivasan, and Utkarsh Srivastava. Building a High-level Dataflow System on Top of Map-Reduce: The Pig Experience. *Proc. VLDB Endow.*, 2(2):1414–1425, August 2009.
- [52] Apache. Pig Latin Basics, April 2014. <http://pig.apache.org/docs/r0.14.0/basic.html>.
- [53] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster Computing with Working Sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing, HotCloud'10*, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.
- [54] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient Distributed Datasets: A Fault-tolerant Abstraction for In-memory Cluster Computing. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, NSDI'12*, pages 2–2, Berkeley, CA, USA, 2012. USENIX Association.
- [55] Matei Alexandru Zaharia. *An Architecture for and Fast and General Data Processing on Large Clusters*. PhD thesis, University of California at Berkeley, 2013.
- [56] Matei Zaharia, Dhruva Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, and Ion Stoica. Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling. In *Proceedings of the 5th European Conference on Computer Systems, EuroSys '10*, pages 265–278, New York, NY, USA, 2010. ACM.
- [57] Apache Hadoop YARN. <http://hortonworks.com/apache/yarn/>.
- [58] Reynold S. Xin, Josh Rosen, Matei Zaharia, Michael J. Franklin, Scott Shenker, and Ion Stoica. Shark: SQL and Rich Analytics at Scale. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, SIGMOD '13*, pages 13–24, New York, NY, USA, 2013. ACM.
- [59] Reynold S. Xin, Joseph E. Gonzalez, Michael J. Franklin, and Ion Stoica. GraphX: A Resilient Distributed Graph System on Spark. In *First International Workshop on Graph Data Management Experiences and Systems, GRADES '13*, pages 2:1–2:6, New York, NY, USA, 2013. ACM.
- [60] Spark Streaming. <http://spark.apache.org/streaming/>.
- [61] Apache Spark Project Statistics. <https://www.openhub.net/p/apache-spark> (accessed 01.06.2016).
- [62] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, and Dennis Fetterly. Dryad: distributed data-parallel programs from sequential building blocks. In *ACM SIGOPS Operating Systems Review*, volume 41, pages 59–72. ACM, 2007.
- [63] Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer, Shiva Shivakumar, Matt Tolton, and Theo Vassilakis. Dremel: Interactive Analysis of Web-scale Datasets. *Proc. VLDB Endow.*, 3(1-2):330–339, September 2010.

- [64] Alexander Alexandrov, Rico Bergmann, Stephan Ewen, Johann-Christoph Freytag, Fabian Hueske, Arvid Heise, Odej Kao, Marcus Leich, Ulf Leser, Volker Markl, Felix Naumann, Mathias Peters, Astrid Rheinländer, Matthias J. Sax, Sebastian Schelter, Mareike Höger, Kostas Tzoumas, and Daniel Warneke. The Stratosphere Platform for Big Data Analytics. *The VLDB Journal*, 23(6):939–964, December 2014.

Models and Tools for Indoor Maps

Georgios Pipelidis and Christian Prehofer

Department of Informatics, Technical University of Munich, Munich
{georgios.pipelidis, christian.prehofer}@tum.de

Abstract

This is a state-of-the-art review of indoor mapping techniques, tools and models. The main challenges of indoor mapping and existing initiatives, such as the Enhanced-911 and the Enhanced-112 are listed. A review of localization techniques is also presented. Criteria and metrics for evaluating these techniques and ways of fusing them are listed. Additionally, techniques for generating indoor maps from data coming from: (1) voluntarily contribution of users, (2) existing indoor plans or even (3) transparently generated from users, are reviewed. Furthermore, Geographic Information Systems and technologies for defining topological relationships are introduced. Spatial data visualization techniques are listed. Finally, models for representing indoor spatial data are reviewed and a comparison of these models is provided. Last but not least, a "Major Market Players" analysis is presented.

Keywords

Indoor Mapping; Geographic Information Model; Indoor Localization

1. Introduction

In the recent years, devices equipped with services capable of estimating the location of an entity (e.g. a human, an object etc.), called Location Services (LS), are pervasive (e.g. smartphones, wearables etc.). LSs are vastly used for adding value to the functionality of additional services (i.e. navigation or recommendation of nearby restaurants). These services are called Location Based Services (LBS) [1]. A LBS implies the existence of localization technology (i.e. LS) and a map. As a map, it is defined a model that describes the geometry, the topology and some semantic information of a place.

Even though people spend approximately 80% of their time indoors [2], [3], LBSs are mostly developed for outdoor environments, where the localization and mapping problems have largely been addressed. Unfortunately, the same does not apply on indoor environments. The Global Positioning System (GPS) cannot work indoors, since its signal cannot penetrate solid objects, such as walls. Additionally, most of the indoor places lack of indoor maps, while most of the existing indoor maps do not confirm to any standardized format. Understanding the indoor environments is therefore of great importance.

The number of users in a LBS, are exponentially increased while their accuracy is increased [4]. Hence, and by taking into consideration the advancement of indoor localization technologies, alternative methods of generating or integrating indoor maps have to be researched (i.e. [5]), in order to cover the increasing demand. As an aside need, modeling semantic information for an indoor place (i.e. information that can be used for localization or navigation) is equally important with modeling the geometric and topological properties of a place [6].

The rest of this paper is organized as follows: In Section II the role of location in LBS, a rough background is described through their use cases, their challenges and public initiatives in US and EU. The section III provides a brief background coverage of different localization techniques and their evaluation criteria. Section IV provides a state of the art review of techniques for indoor mapping. A series of models designed or enhanced for modeling indoor spatial information are reviewed, and a rough comparison of these models are provided in section V. A Major Market Players analysis is provided in the section VI. Finally, the section VII concludes the paper.

2. Background

In this section a brief introduction of Location Based Services is provided, through some exemplar use cases and public initiatives. Additionally, in this section existing and upcoming challenges of LBSs are listed and a description of the term "location" in LBS is provided.

2.1 Use Cases

some examples where indoor LBSs are used or could be used to improve the quality of life are:

- **Indoor telematics:** is the most popular LBS system, such as navigation.
- **Presence feature:** provides to the user which of his friends are in a close distance.
- **Indoor Car-to-Car communication:** enables the exchange of warning messages (i.e. empty parking spots).
- **Fleet management:** control and coordinate entire fleets of robots indoors.

- **Virtual reality:** could be enhanced and merged with the real world, where the user's location would become an essential aspect of the play.
- **Hospitals:** could locate their equipment, provide navigation to their visitors and monitor the exercise and location of their patient.
- **Equipment monitoring:** provides tools for identifying and report broken equipment (i.e.: burned-out luminaries).
- **Internet of Things:** taking decisions based on the location of user equipment. (i.e. switch on or off the lights when the user is present or absent).
- **Fire fighter assistance:** provide navigation to fire fighters through safe zones or even localize peoples in danger indoors
- **Surveillance:** enables parents to localize their children, pet owners to find their animals or police to track convicts or terrorists.
- **Indoor Evacuation Simulation:** enable users to conduct multi-agent indoor evacuation simulations.

Moreover, recommender mechanisms can benefit from LBS by providing to the mobile user with nearby points of interest (i.e. restaurants etc.). Furthermore, marketing can be improved by providing the consumer with information about products and services of local relevance. Finally, analytics can benefit shop owners, since they can find products which are visited the most or least and make a better use of their space or even get benefit by trading this information.

2.2 Challenges

This chapter lists some of the open challenges and drawbacks of indoor LBSs. They can be organized into three categories: (1) challenges of indoor localization techniques, (2) challenges of indoor mapping and (3) challenges of modeling of indoor spatial information.

1) Localization: The most essential drawback of Indoor localization is the lack of a prevailing indoor positioning technology. Every technology has its benefits and drawbacks. Consider the most prevailing technologies:

- **BLE beacons based localization:** Such dedicate hardware is a resource demanding technology, since they have to be densely installed. As a result, they are limited of being installed in large building structures (i.e. airports [7]), while they require complicated installations. They are mostly operating with batteries, as a result they are an energy constrained technology.
- **Magnetic field based localization:** This technology from the other side, requires permanent structures in a building (i.e. walls) reach in structural steel elements,

that will vary on the steel content and structure. Usually, this is not the case. Additionally, the disturbances tend to occur near walls which disables the technique from operating in large indoor areas like big halls etc.

- **WiFi based localization:** Such technologies seem to be ubiquitous but they work only under specific circumstances. Algorithms that use trilateration for positioning presume the synchronization of the access points (AP) and keeping them synchronized is a challenge. Algorithms use angle of arrival require optimized antennas for localization, as a result, they cannot be used with existing smartphones. Algorithms that used received signal strength for localization, can be dramatically influenced by the presence of people, since the microwave frequency used in WLAN can be absorbed by the human body.

Fusing the above mentioned technologies in an infrastructure independent way is an ongoing research.

2) Mapping: Indoor localization, in most of the cases, requires indoor maps. Indoor mapping indicates the existence of models that describe geometry of places and objects, topological relationships between these places (i.e. adjacency and connectivity) and semantical annotation of the space which indicate: (1) the way that the space is used (e.g. stairs, elevator etc.) and (2) unique identifiers of the place (e.g. the received signal strength in a room from multiple APs).

Beyond the technical challenge of making the maps, mapping indoor places is a resource demanding procedure with an enormous amount of cost. Additionally, environment characteristics are never static (i.e. objects displaced etc.). Hence, indoor maps often become outdated, while their maintenance effort increase the overall cost. Legal challenges are often the case, since in most cases the indoors are privately owned places.

3) Modeling: Storing indoor maps require an enormous amount of data, considering the fact that recently, only the building footprints in OSM surpassed the amount of data on streets. Additionally, there is not a well agreed upon model for this procedure. Filtering outliers, extraction of topological information from spatial information and enhancement of extracted features with semantic information are technologies under research. Additionally, since the procedure of mapping is often crowdsourced, it has been emerged the need of mechanisms that manage uncertainty from various user inputs and mechanisms that bind different inputs from the same floorplans.

Furthermore, indoor localization cannot use the maps without semantically enhanced and uniquely identified locations.

Finally, modeling accuracy (provide the correct position), availability (provide results within a constrained time limit), stability (provide consistent results) and ambiguity (provide uncertainty of the results) remains a challenge. Last but not

least, there is not an explicitly defined taxonomy of indoor environments.

2.3 Public Initiatives

Today, there are some governmental initiatives for LBS. US has launched the E-911, which stands for enhanced 911 for the enhancement of emergency services. Its role is to localize people who call 911. Most of the people call 911 from mobile phones, hence localizing them is already a challenge. The E-911 uses cellular networks for localizing people and its goal is to enhance their accuracy by updating the current infrastructure.

EU from the other side launched some activities for Enhanced 112 (E-112) in 2000 and founded the Coordination Group on Access to Location Information for Emergency Services (CGALIES).

2.4 The term Location in Location Based Service

The term “location” is associated with a certain place in the real world. Location denotes a place of an object in the real world, and hence this kind of location belongs to the class of physical locations. The cyberspace Internet has brought another concept of location where virtual meetings take place (e.g. a distributed computer game). This is called a virtual location. LBSs predominantly refer to physical locations, with an exception of augmented reality.

Physical locations can be further broken down into three subcategories: (1) Descriptive locations: natural geographic objects (2) Spatial locations: a single point in the Euclidean space (position) expressed by coordinates. (3) Network locations: the topology of a communications network. The target persons of a LBS can be pinpointed by all these location description models. Spatial location or position information represents an appropriate means for exactly pinpointing an object on Earth.

A LBS needs to map between different location categories. For example, distance calculations can only be done by descriptive locations, while routing can be expressed better by descriptive locations. For expressing spatial locations, it is necessary to use: (1) a coordinate system, (2) a datum and (3) a projection (i.e. on a map). Coordinate systems used for describing locations are the Cartesian and the ellipsoidal. The Cartesian describes a location by specifying its distances to predefined axes. The ellipsoidal describes a location by its angles to an equatorial and polar plane. A datum defines the size and shape of the Earth as well as the origin and orientation of the coordinate system that is used to reference a certain position.

3. INDOOR LOCALIZATION

In 1978 the first GPS satellite was launched and in 1995 GPS worked with its full capability for the first time. Unfortunately, the satellite signals are not strong enough to work indoors [8]. This has as a result for alternative techniques for indoor

localization to be emerged. On this chapter the most popular techniques are presented.

3.1 Identification of Entrances

Many approaches have been suggested for entrance localization or outdoor to indoor transition and vice versa. A simple technique has been suggested by [9] and [10], where the drop of confidence or inability of GPS is obtained as an indication of this transition. Digital cameras in smartphones have been also suggested [11] together with image processing techniques. A promising technique has been suggested by [12] and [13], where light sensors, cell tower signal and magnetic field sensors, together with assistive technologies, such as the acceleration and proximity sensor and time, are fused for identifying the IO transition.

Using the light sensor, is due to the observation that the light intensity indoors is lower than outdoor or semi-outdoor environment (i.e. existence of walls), while indoor fluorescent light exhibits a periodical pattern, due to alternating power (AC). In this scenario, the proximity sensor can be used as a confidence indicator, since it can successfully identify whether the light sensor is blocked by an object, hence the measurement is not accurate. Additionally, the time can indicate whether it is night or day. This approach is rotation, weather, and time invariant.

Cellular tower signal detection, detects the attenuation of signal due to the existence of walls. Received Signal Strength (RSS) variation within a short period of time (i.e. 10 sec) can indicate the IO transition, since indoors a mobile device is exhibits higher degree of the cellular signal strength attenuation than outdoors due to the reflection of the signal from walls. The opposite effect occurs for the WiFi RSS. Additionally, the number of cellular antennas, in the case of cellular RSS, and the number of AP, in case of WiFi RSS, which exhibit this effect by the number of existing antennas and APs, expresses the confidence of the IO-transition. The magnetic sensor can detect disturbances due to steel elements inside walls of a building. Hence, the intensity of the magnetic field can be used as indicator for identifying the IO-transition.

3.2 Methods for Indoor Localization

Wireless Local Area Network

In 1997 IEEE Standard 802.11 was set and the first version of Wireless LAN was born. WiFi can be used as an enabler for indoor LBS. WiFi uses electromagnetic waves to transmit data and it operates in broadband (2.4 GHz and 5 GHz). Energy transformation due to reflection (i.e. because of existence of walls or windows) causes the signal amplitude to consequently become smaller. This is the main idea behind indoor localization based on WiFi.

There are several approaches for indoor localization based on WiFi, among the most popular are: (1) Based on proximity sensing [14], this demands a database of station IDs and their geolocation, then the position is determined by measuring the RSS. (2) Trilateration, the distance is calculated from the station to a device. With more than one stations the device

position can be approximately estimated. Several methods for trilateration exist, some of them are (a) based on Time of Arrival (ToA) [15], it estimates the distance based on the Round Trip Time (RTT) of a message. (b) Time Difference of Arrival (TDoA) [16], it uses the difference between the arrival times of the signals to determine the position. (c) RSS [17], uses propagation-loss of the WiFi signals to compute the distance. (3) Another method for localization is based on triangulation or Angle of Arrival (AoA) [18], where the distance is trigonometrically estimated but special antennas are mandatory for this approach. (4) Another popular approach for indoor localization based on WiFi is by wave propagation estimation based on Friis formula, where the received and transmit power needs to be known, as well as the signal wavelength and the distance is derived by the Friis formula similar to [19]. (5) Finally, localization can also be done by pattern recognition and fingerprinting methods.

Geomagnetism

In this technique, the location is estimated based on disturbances of the earth's magnetic field caused by structural steel elements in a building [20]. Its unique characteristics are that spatially it varies but it is a permanent characteristic of space. For accurate mapping and localization, a 3D axis electronic compass equipped with an internal tilt compensated algorithm to measure the heading of the sensor, can be used.

Active badge

There are numerous localization techniques available using badges. A characteristic example is [21]. Here, they have designed a tag that emits a unique infrared (IR) signal for approximately a tenth of a second every 15 seconds. The signal is received by a network of sensors installed in a particular building. It uses IR because it is cheap, it can operate with a 6m range and is mostly reflected by walls which can help for a better discretization between different rooms in a building. Finally, a master station processes the data detecting for badge transmissions while providing with location data to the clients.

Visual Light Positioning (VLP)

Two techniques for indoor localization based on VLP are available:

1) Code Division Multiple Access: For some indoor environments RF signals are not desirable (i.e. hospitals) due to Electromagnetic Interference. Moreover, with the introduction of white LEDs as illumination source, a new communication technology arises called Visible Light Communication (VLC) [22]. In this technology, the information is transmitted by modulating the light intensity. It is not limited to indoor localization. Its main advantages are its energy efficiency and long lifetime. A VLC system can be employed for indoor and outdoor applications.

2) Polarization-based modulation: As already mentioned, anchor locations can be broadcasted through VLC. Additionally, it is argued [23] that this procedure can be done in a

simpler way, by modulating the light, in a way to be uniquely identified through its polarization. In this way the effect of flickering from the conventional VLC, which rely on intensity-based modulation of LED lights can be avoided.

The suggested way can also be supported by constrained wearable devices, while it can support sources beyond LED light. It can even use sun light, eliminating the dependency on LED. It works as follows, a dispensor is added to the VLC transmitter. The dispensor has a special property which is called optical rotatory dispersion (ORD). It implies that this material can rotate the light's polarization differently with different frequencies (colors) of light. Finally, through the dispensor, a novel modulation scheme is employed, which is called Binary Color Shift Keying (BCSK).

Dead Reckoning

The idea of dead reckoning, also known as deduced reckoning, is that the current location can be estimated based on the previous location, the distance traveled and the direction of motion. Today dead reckoning can be applied on pedestrian data collected by smartphones [24], although there are still several challenges.

The first challenge lies in the estimation of the distance traveled. It can be theoretically calculated by integrating the acceleration twice with respect to time. However, this procedure leads to errors and displacement that will grow cubically with time. To handle *saengwongwanich2014indoorrobo* this errors, usually this procedure is done based on activity recognition. The framework is as follows: (1) Identify if the person is walking. (2) Estimate the number of steps. (3) Estimate step length. (4) Compute the distance traveled.

The second challenge is the attitude estimation. Identifying the walking direction of a person, using a smartphone device, can be a challenging task, since smartphone pose (i.e. phone rotated by 90° from the walking direction etc.) and location (i.e. in the pocket or hand or bag pack etc.) can vary together with the activity of the user (i.e. walking, standing etc). Usually, attitude is estimated as a classification problem, hypothesizing that during a gait cycle the maximum acceleration occurs towards the walking direction, while at the same time there is a minimum in the lateral acceleration [25], [26], [27]. Attitude can be expressed with three mathematical representations. Euler angles, rotation matrix or quaternions.

A third challenge is that the PDR error leads to displacement that will grow cubically with time. This problem can be eliminated by introducing landmarks, where the error can always be restarted. Landmarks, can be uniquely identified areas in a place. A landmark can be identified either by the existed sensors of a smartphone (i.e. a landmark can be a unique WiFi or GSM RSS, geomagnetic fingerprint of the place or even a sound fingerprint), or by performing activities (i.e. climbing stairs, doors, elevators).

Cellular networks

In cellular network positioning, radio signals are used for localization. Their advantage is that they have low energy con-

sumption and higher availability. It works as follows: a user's phone is localized based on the cell tower location where it is connected. Unfortunately, this is only an approximation of the actual trajectory but more precise techniques have also been suggested [28], by taking advantage of signal strength and other advanced methods.

Radio Frequency Identification (RFID)

Hypothesizing that in the near future every object in an environment will be equipped with small, cheap RFID tags of a detection range approximately 6m. Using these tags an entity equipped with RFID antennas can localize itself in relation to these objects [29].

Acoustic fingerprint

Based on the observation that in indoor environment distinct functional areas have different configurations of furniture locations, a room precision localization system obviates the needs for infrastructures, is argued to be possible as shown in [30] that rooms' acoustic properties can be characterized by Room Impulse Response (RIR).

Bluetooth Low Energy (BLE)

BLE stands for Bluetooth Low Energy and is part of Bluetooth 4.0 [31]. BLE works with beacons designed for localization based on BLE technology. Low energy protocol allows Beacons to work on a single battery for a long time. The spectrum of the signal ranges between (2.4GHz – 2.4835GHz) and it can be detected in up to 100m. The difference to Bluetooth is that it has lower transfer rates. The localization is done based on the Received Signal Strength Indication. The observation, similar to localization with WiFi RSS, is that the signal decreases predictably as the devices goes further.

Ultra Wideband (UWB)

Various methods of localization based on Ultra Wideband technologies exist [32]. Similar to WiFi localization techniques, localization can be achieved: (1) by the angle of arrival (AOA), which measures the angles between a given node and a number of reference nodes to estimate the location. (2) The signal strength (SS), where the distance between nodes is estimated by computing the energy intensity of the signal. (3) Time delay information, estimates the distance between nodes by estimating the travel time of the received signal.

3.3 Evaluation of Localization Techniques

Evaluation of localization methods is an open challenge, since many environmental characteristics can influence their accuracy. Different infrastructures can favor some methods while disfavoring others. Some evaluation criteria as described by [33] are:

1) Scalability: It concerns whether the algorithm is accurate enough for hundreds, or even thousands of nodes as it is for less than ten. It also examines in case the localization system is centralized, if there are some potential bottlenecks or in case the localization system is distributed, whether an algo-

rithm can be easily developed and deployed for a distributed system easily.

2) Accuracy: It concerns whether the estimated positions match the ground truth positions. Since this is an application-dependent task, accuracy is expressed based on the inter-node spacing. (i.e. if the average node spacing is 100m, up to 1m error may be acceptable, it cannot be the same when the average node spacing is 0.5 m). Metrics applied on this technique are:

- When the actual node position (ground truth) and physical network topology are given, the error can be expressed as follows:
 - Mean absolute error: by the residual error between the estimated and actual node positions for every node in the network, after summarizing them and averaging the result.
 - FROB: (Frobenius): by computing the residual error between all nodes in the network. Assuming that the estimated and actual inter-node distances are determined, it determines the root mean square of the total residual error, which represents the global quality of the localization algorithm.
 - GER (Global Energy Ratio): by the normalized distance error between all nodes.
 - GDE (Global Distance Error): by taking the RMS error over the network of n nodes and normalizes it by the average radio range.
 - ARD (Average Relative Deviation): by normalizing the average of the estimated distances between all nodes in the network and the estimated location.
 - BAR: the sum-of-squares normalized error taken from matching the estimated location with the actual location.
- Without ground truth:
 - Average Distance Error: by subtracting from the observed range between two nodes their estimated distance.
 - SPFROB (shortest-path FROB): based on the shortest path between two nodes, rather than Euclidean distance.

3) Resilience to Error and Noise: It concerns whether the localization algorithm can deal with errors and noise in the input data, as well as, whether noise, bias or uncorrelated error in the input data affect the algorithm's performance.

4) Coverage: It concerns the area covered by the network and the algorithm can apply localization, given a specific network topology/deployment. Usually, it depends on the deployed network density. It also concerns the effort needed to add another node to the network after the initial localization algorithm has completed. Metrics:

- Density: The average number of neighbors a node has.
- Anchor Placement: Computed using the Geometric Dilution Of Precision (GDOP) metric. It describes the geometric “strength” of the nodes current positions with respect to the target.

5) Cost: How expensive is the algorithm (i.e. per-node hardware or software cost) in terms of: Power consumption, Time taken to localize a node, Communication and Pre-deployment set-up (i.e. need for, and number of anchors). Metrics:

- Anchor to Node Ratio: the number of anchors in the network divided by the number of nodes
- Communication Overhead: the average number of packets sent per node
- Power Consumption: combination of the power used to perform local operations and the power used to send and receive messages associated with localization.
- Algorithmic Complexity: computational complexity in time and space of localization algorithms
- Convergence Time: the time taken for the initial measurement gathering and the localization algorithm convergence.
- Hybrid Metrics:
 - Performance Cost Metric (PCM): The performance cost and the localization error are weighted by a parameter that determines the relative importance.

3.4 Fusing Localization Techniques

The best localization technique is probably the fusion of two or more of the above localization algorithms. Several techniques are available for fusing, and not limited, spatial data:

- *Voting [34]*: localization algorithms are used and the location is estimated based on the result of the majority of the localization algorithms.
- *Boosting [35]*: (Bootstrap aggregating) where the position is decided by weighting the results of the different algorithms.
- *Cascading [36] and Stacking [37]*: the decision is given by another classifier which is trained with the results of the previous classifiers. This classifier is also called meta-classifier while the procedure is called meta learning.
- *Bagging [38]*: different features are evaluated by the same localization algorithm and then a second algorithm decides over the different data.

4. INDOOR MAPPING

The existence of worldwide indoor maps database can lead to a significant growth of LBS [39]. For example, prior knowledge about the environment can greatly support search and rescue missions in emergency situations. Furthermore, most existing location-sensing techniques rely on fixed references to determine the location of tagged devices. Which implies that location information is only available in environments where references with known positions are deployed. However, in many scenarios where location information is very useful, a reference system is not likely to have been previously deployed.

4.1 Methods for Mapping Indoor Information

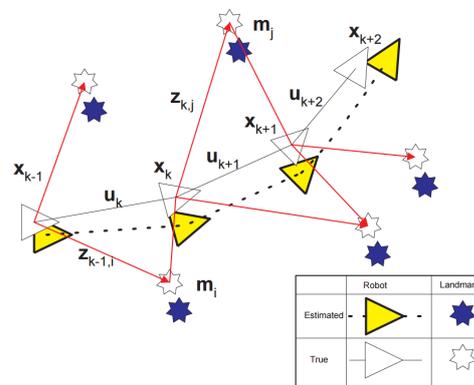


Figure 1: A simultaneous estimate of both robot and landmark locations is required. The true locations are never known or measured directly. Observations are made between the sensing device and landmark location [40].

In this section different mapping techniques are reviewed. Techniques that carry knowledge regarding fixed location in the building and others that do not. Since the indoor mapping procedure is largely crowdsourced, different approaches that enable the crowdsourcing map creation are mostly presented.

Simultaneous Localization And Mapping (SLAM)

SLAM addresses the problems of localization and mapping as one [40]. Its main contribution is that it uses the correlations between observed landmarks for reducing the localization error (Figure 1). As a result, an “entity” can construct a map of the environment (from landmarks) and use this map to deduce its location [41].

In a specific timestamp, the location, the orientation of an “entity” and the observed landmark can be retrieved by computing the posterior probability of observed features such as: the set of its prior actions, the location and orientation of the surrounding landmarks and its previous location and orientation. Hence, the current position of an object can be computed using probabilistic approaches (i.e. Bayes Theorem), if a state transition model and an observation model is defined. The state transition model is usually assumed to be a Markov process, since the next state depends only on the previous state (Markov property) and the directions are independent of the

observations and the map (Time invariance). The observation model, describes the probability to observe a landmark when the location, orientation and other landmark locations are known.

Since this problem can be formulated in a probabilistic way, there is a need for representing of the observation and the motion models in such a way that it will enable an efficient and consistent computation of its prior and the posterior distributions probabilities. There are two popular computational solutions for this problem:

- The extended Kalman filter (EKF-SLAM) [42].
- The use of Rao-Blackwellised particle filters (Fast-SLAM) [43].

Light Detection And Ranging (LiDAR)

Light Detection And Ranging use lasers to measure the distance between objects inside a building (i.e. walls, floors, ceilings etc.) [44]. It works similar to sonar or radar sensors but instead of sound or radio waves it uses light. It works as follows, a LiDAR unit, often mounted on a robot or vehicle, scans the environment using green or infrared light. The position of the unit is estimated either by dead reckoning or by other localization technique. It works as follows: A burst of light energy, called pulse, is emitted by the LiDAR unit, then the reflected light energy, called return, is recorded by the LiDAR sensor. Finally, the travel time is estimated by recording the time taken for the light to the object and back. The product of the travel time and the speed of light, divided by two will then return the distance of the object from the LiDAR system, while the localization technique will return the location of the LiDAR unit in the building. IMU sensors are essential for this procedure, since they are used to estimate the tilt (i.e. Yaw, Pitch and Roll) of the LiDAR unit. Finally, a point cloud is generated and by identifying contours (i.e. points of similar distance) a map can be extracted. Semantic annotations are usually made manually by expert surveyors.

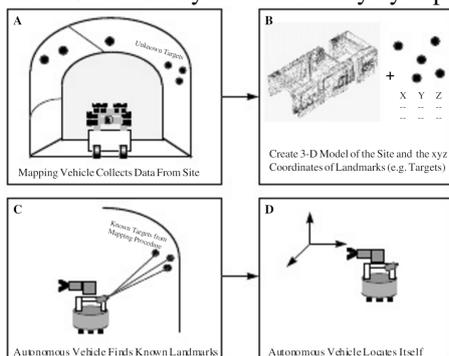


Figure 2: The mapping procedure using LiDAR [44].

Automatically Integration from Industry Foundation Classes (IFC)

IFC can provide an architectural model that enables spatial division and varieties of information of the building (i.e. furniture, building size, material type, wall colors etc.), these characteristics can be used for enhancing the indoor navigation

procedure [45], since more descriptive spatial characteristics can be offered. Unfortunately, today IFC models are only used to model a single building, limiting them from exchanging information in a GIS environment. The reason is that it represents objects in a local coordinate system while GIS use a global. Hence, in order to use IFC objects in GIS, they have to be translated to the global system coordinates.

Additionally, IFC and GIS have different ways of describing objects [46]. IFC uses a Swept Solid representation, where objects are described as 3D solid models, or Constructive Solid Geometry (CSG) where complex objects are described by a set of primitive solids (i.e. boxes) combined with Boolean operations. GIS objects, from the other side, are surface models established based on their boundaries. As a result, translating IFC data to GIS and vice versa remains a challenge.

Fortunately, even though IFC needs approximately 900 classes for describing a complete project, the classes used for presenting geometric and attribute information, and hence need to be translated to GIS, are limited to 17, for which there exist corresponding classes in GML. As a result, the entire model cannot be described with GIS but important information can easily be translated.

IFC is not designed to be used in navigation. As a result, deriving topological relations from IFC between indoor places (i.e. adjacency and connectivity between rooms), can be done with alternative ways [47]. A naïve way, where each door of an IFC model can be seen as a point, while each wall can be seen as a vertice that connects different doors. Unfortunately, this does not hold. A more generic indoor topology (Figure 3), can be constructed if each door is perceived as a line segment and partitions that intersect with the perpendicular line of the door segment are selected (to represent connectivity). If there are more than one partition (a door can connect only two places), the perpendicular lines from the surrounding walls, of the selected partition, which are connected with the perpendicular line of the door, are taken into consideration and the partition attached to the wall with the shortest distance is selected.

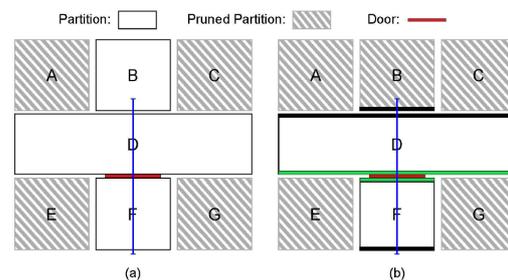


Figure 3: An example of the generic method [47].

Designed by the crowd

Volunteered Geographic Information (VGI) has become more attractive today with new and easy in use smartphone applications. An example is the 3D Modeler [48]. Its design revolves around a client server architecture. It enables smartphone users to interact with a 3D modeling applications in order

to construct components of a building (i.e. rooms, hallways, furniture etc.). After a user submits his model, it is uploaded on the server, in order to be disseminated to other users who can enhance it or vote for its accuracy (the dimensions and location of modeled objects) and its completeness (whether it contains all existing components or not). The quality of the model is decided based on the user votes and predefined thresholds, while color scale is used to communicate the status of the object to other users.

Its main weakness is that it lacks semantic and topological representation that can enable the localization procedure (e.g. RSS, Magnetic field fingerprint etc.).

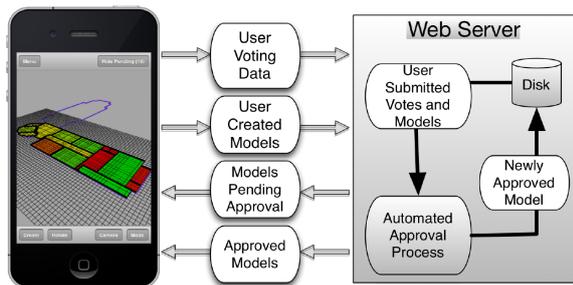


Figure 4: Mobile 3D Modeler System Architecture [48].

Computer Vision techniques

1) structure from Motion: A map can be extracted via Computer Vision (CV) techniques, since CV can provide geometric characteristics (i.e. position, size and the orientation) of individual landmarks (i.e. stairs, doors, walls etc.) [49]. It works as follows: After images are captured, from a camera for which the intrinsic and extrinsic parameters [50] are known and unique features have been extracted (i.e. BRISK features [51]), using a Structure from Motion (SfM) algorithm [52], a 3D point cloud of the building can be extracted. Finally, by applying edge detection algorithm (i.e. Cunny algorithm [53]), shape recognition algorithm (i.e. Hough Transform [54]) and segmenting the results, by grouping parallel lines into groups and rejecting lines of no geometric importance, the landmark countours can be identified.

Finally, by projecting the point cloud into a 2D structure and by labeling the highest density places as wall segments the whole structure can be denoted by $L = (P, Q)$, where P are the main geometric vertices and Q are connecting points.

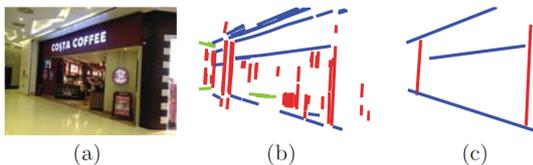


Figure 5: Geometric vertices detection work flow: (a) original image. (b) detect line segments parallel to the three orthogonal axes. (c) merged long line segments corresponding to the landmark's major contour lines. Different colors represent different dimensions. source: [49].

2) Depth Sensors: A 3D point cloud can be the result of a depth sensor. This point cloud can be used for extracting a map from this point cloud similar to [55]. It works as follows: An infrared projector, projects a unique pattern (i.e. a speckle pattern [56]). An infrared sensor, whose relative distance to the projector and rotation is known, recognizes these markers. A depth map is constructed by analyzing the unique pattern of infrared light markers by triangulating the distance between the sensor the projector and the object. The technique of analyzing a known pattern is called structured light (project a known pattern onto the scene and infer depth from the deformation of that pattern). Finally, combining structured light with CV techniques, for example depth from focus (uses the principle that stuff that is blurrier is further away) and depth from stereo (Stuff gets shifted more when are close and the scene is in angle, than stuff that is far away), the depth of different areas can be estimated.



Figure 6: (Left) 3D maps generated by Depth Sensor. [55].

Indoor Mapping Using Impulse Radio Network

For mapping a 2D floorplan, consider the following scenario, where two radios are operating close to a corner comprising two walls [57] (Figure 7). Hence, there will be four channels (two broadcasting and two receiving) with their corresponding impulse responses. Additionally, their pulses consist of the line-of-sight (LOS), single reflections and some double reflections, assuming that by suitable thresholding, all higher-order reflections can be ignored. If the receivers can estimate the TOAs, while they are insensitive to amplitude and phase changes, the distances to possible walls (reflections) can be estimated. In the specific scenario, it is expected that two are single reflections, one from each of the two walls, and the last one is the double reflection from both walls. In the first step the system takes into consideration, besides the LOS, only single reflections, hence the system will classify an additional wall.

If the distance between the two radios is known, it may be a case where the same wall is identified from both radios. Hence, one wall can be positioned in relation to the two radios as the common tangent of the two circles with radius the estimated distances by excluding inner common tangents, since they are potentially blocking the pulses between the radios. Finally, by identifying whether two walls predict double reflections with delays that fit the actual measured and intervals that fit the actual measured, there will be filtered all the rest instead of the actual scenario and its mirror image. This scenario can be scaled for mapping multiple walls with higher degree of reflections.

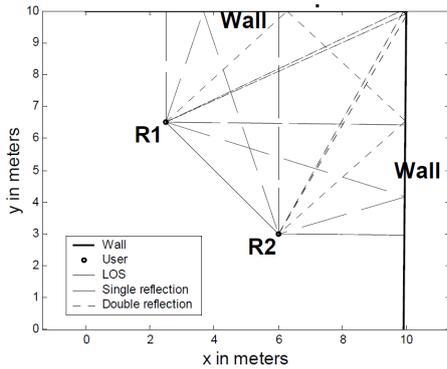


Figure 7: Two radios communicate in a 2-wall environment [57].

Dynamic Mapping

Maps can be transparently and autonomously generated based on activity recognition from IMU of smartphones [58]. It works as follows: Measurements from embedded mobile device sensors are collected while users moving naturally inside buildings [59]. The collected data can be then used for estimating the traces of users. Movements can be seen as motion constraints while different kind of POIs (e.g. doors, stairs etc.) as landmarks in a SLAM algorithm. The traces consist of various steps which if annotated by their exact locations can produce a point cloud of the require surface.

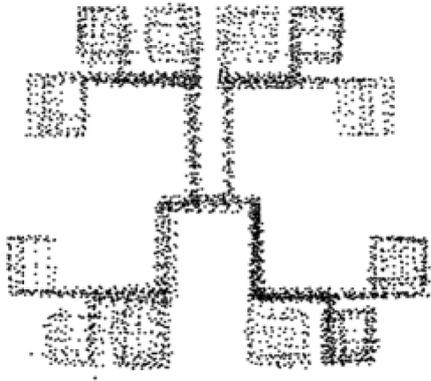


Figure 8: Point cloud constructed by mapping traces of people freely walking in the structure. Source: [39]

4.2 Keep Indoor Maps Up-To-Date

Localizing an “entity” indoors requires constantly up-to-date databases, where uniquely spatially identifiable locations are stored. This procedure is highly resource and time demanding. A solution to this problem is the “organic” contribution to this procedure by users who have been successfully localized. This concept is called “organic maps” and it is achieved by combining different methods of localization or context recognition (i.e. WiFi RSS and Calendar Data).

The first organic maps were designed to ask from the user to mark his/her location [60], [61], while recently [62] it is suggested to acquire this information dynamically. In this way an “organic” system can eliminate the need of explicit actions by users and in a sense to be aware of its status. The

idea is as follows: whenever a user’s device observes an unrecognized signature (i.e. RSS), but the device itself has been successfully localized, the unrecognized signature is tagged with the particular location and stored in the database.

“Organic” mapping is mostly used with fingerprint-based methods, since they rely on databases composed by pairs of <fingerprint, ground-truth location>[60]. An organic map can be self-aware of its accuracy. The accuracy of a positioning system is usually expressed as the maximum (or) average positioning error expressed in meters. This information can be combined with other sources of information such as calendar information and the accuracy can be retrieved (i.e. if the user is localized by the RSS in a room and this is disproved by his calendar, the accuracy of the system can be dynamically quantified).

Managing uncertainty is a great challenge for “organic” mapping. Considering the fact that for determining when a user’s input is actually required, is a challenging procedure, while determining if it the given information is accurate, demands a highly sophisticate and dynamic system. Additionally, the binding process is more error prone than conventional mapping techniques, since user’s inputs can be more inaccurate than trained surveyors. Furthermore, the quality of an “organic” system depends on the willingness of the users to contribute during the entire life cycle of the system.

Finally, it must be highlighted that an organic map presumes the existence of the map, since it is only designed to keep it update. Hence is differs from dynamic map generation, where the map entirely is generated from user data.

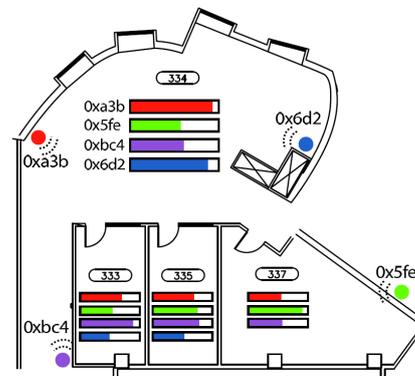


Figure 10: RF fingerprints. The bars in each space illustrate the RSSI from each in-range AP. [61]

5. INDOOR DATA MODELS

A data model is a tool for defining structures that organize sets of data. These structures can be a set of tasks such as storing or exchanging data. A data model can enable the storage of data in a consistent way and structure the stored data in such a way that the importing and exporting type of data will be explicitly defined. The indoor space is different than the outdoor, due to its architecture constrains such as stairs, doors, corridors, floors and walls.

In this chapter the spatial data and GIS systems will be explained. Ways to visualize spatial objects will be listed and topological relationship models will be explained. Finally, different models for representing spatial data will be described.

5.1 Spatial Data and GIS

For transforming descriptive locations to spatial and vice versa, spatial databases are key technologies. They are important for indicating the positions of targets with respect to geographical content. They are used for mapping spatial location onto meaningful descriptive location information and vice versa. This process is called geocoding or reverse geocoding.

Spatial databases use Data Base Management Systems (DBMS). A spatial DBMS forms an integral part of a Geographic Information System (GIS). GIS is “a computer system for capturing, managing, integrating, manipulating, analyzing, and displaying data which are spatially referenced to Earth” [63]. GISs, distinguish between two levels of abstraction: (1) The geographic data model, a conceptual view of geographic content (2) The spatial data model, deals with all aspects of the physical data management.

5.2 Visualization of Spatial Data

Spatial data can be represented by either “raster” or “vector” mode. Raster mode is similar to a bitmap image, where the analysis is done by “tessellation” (tiling of a plane using different shapes), while the number of bits necessary for representing a raster attribute is referred to as “depth”. In raster mode, a spatial object is represented by a collection of pixels and its position is given by the integer coordinates of these pixels within a grid. In vector mode, the spatial objects are represented by means of coordinates of a reference system. The simplest spatial object is a single point. Two endpoints model a straight line, an ordered list of points model a polyline and a polygon is when the start node and end node of a polyline match. The models for describing spatial objects that will be described on this chapter are based on the vector mode.

5.3 Topological relationships and Navigation

Essential features for navigation are the explicitly defined topological relationships. Topological relationships can be derived during runtime, by performing operations on the spatial objects stored in the database, or they can be explicitly modeled when setting up the database. Models for the explicit modeling of topological relationships are: (1) the spaghetti model, where each spatial object is stored entirely independent of any other. (2) The network model, which covers the relationships between points and polylines and (3) topological models, which are also capable of reflecting adjacent relationships between polygons.

5.4 Models for Indoor Spatial Information

A plethora of models for describing indoor information exist. Some of them have been emerged from the gaming and

architectural industry, while others have been evolved from existing models which are used outdoors. In this section we review the most prominent of them.

COLLABorative Design Activity (COLLADA)

COLLADA [64], is an XML file format for cross-platform interchange of 3D assets among various graphics software applications. It supports geometry with full skinning, advanced material and visual effects animation, physical properties and collisions. It is maintained by the nonprofit technology consortium, the Khronos Group, and has been adopted by ISO as a publicly available specification, ISO/PAS 17506. The exporting and importing files are identified with a “.dae” (digital asset exchange) filename extension. It is designed by Sony Computer Entertainment for supporting game engines. It is used by software tools such as Computer Aided Design (CAD) as well as systems and three-dimensional modeling software of city models like CityEngine. It is also supported by Google Earth. It allows to identify surface materials (i.e. friction or gravity) which enable it from being used in physics simulation tools.

Recently, in order to bridge the gap between WebGL and COLLADA, Motorola initiated the COLLADA2JSON project for JSON format design from the ground up conforming with the requirements of WebGL and Web3D by enabling Rest3D APIs.

IndoorOSM

Indoor Open Street Maps [65], is an indoor mapping tagging schema, which now is defunct due to technical problems (i.e.: tag collisions, massive use of relations etc.). It is designed for mapping of indoor spaces by taking into consideration special properties like floors, doors, windows, nodes, relations, keys as well as 3D properties (i.e. height vertical connections etc.). It enables the multi-level representation of the indoor data and allows overlapping elements to be filtered.

More precisely, a building is represented as a relation of type “building”. General characteristics of the building (i.e. address, name, height etc.) are key values to the main relation. Different levels of the building are defined as children of the main relation and type “level”, while entrances/exits are mapped as relation members. Rooms and corridors are mapped as nodes and ways respectively. Rooms, stairways and corridors are mapped as closed-ways, while doors or windows are mapped as single nodes with information about their size, accessibility, name or type defined as key-value pairs. Vertical connections (i.e. stair, escalators) are mapped as a closed way. Suggestions exist to map the perpendicular coordinate with the use of the attribute “ele”, which is used to indicate altitude or the attribute “layer”, which is used outdoors to express bridges or highway intersections.

IndoorGML

IndoorGML [66] is an Open Geospatial Consortium (OGC) standard for indoor information. It was emerged to fill the need for standard on accurate definition of indoor spaces for

enabling indoor navigation. It is an XML schema operating on the application layer. The indoor location acquisition procedure differs from outdoors due to its higher complexity because of the existence of structural constraints (i.e. corridors, doors, rooms, elevator, stairs etc.) and the unavailability of GPS as well as the alternative indoor localization techniques available. It provides a data model which contains:

- Cellular space: The space is defined via the use of cells. The cells are defined from the decomposition of indoor space to its smallest organizational or structural units.
- Semantic representation: For achieving cellular modeling structural constraints are important. Semantics can be a driver of the identifying this constraints (i.e. each cell is restricted to the coverage of Wi-Fi).
- Geometric representation: Representation of geographical space and its elements is done via external references such as CityGML LoD4 which follows ISO19107 definitions.
- Topological representation: Connectivity and adjacency is modeled via Node-Relation Graph (NRG). NRG follows Poincare duality. Rooms are mapped to nodes (3D to 0D object). Surface shared by two objects (i.e. corridors) mapped to edges (2D to 1D object).
- Multi-Layered representation: Multi-Layered Space-Event Model (MLSEM) allows for multiple layers of data representation, enabling multiple thematic layers (i.e.: Wi-Fi can be used in one of this layers as a way to decompose the indoor space).
- Sub-spacing: Allows the sub-spacing of indoor spaces via decomposition of cellular space and its node representation in a dual space NRG to multiple sub-nodes, in order to better reflect hierarchical structures (i.e. dividing a corridor by segments).
- Anchor nodes: Enable interoperability of indoorGML with outdoor datasets by externally referencing anchor spaces and anchor boundaries.

OpenDRIVE

OpenDRIVE [67] emerged as a standardized model for data exchange between different driving simulators. It describes entire road networks by relating data that belong to the road environment. It does not take into consideration objects that can interact with the environment. It is managed by VIRESS Simulatiotechnologie GmbH and an open community.

OpenDRIVE has been recently enhanced by 3D Mapping Solutions GmbH in such a way to support the digitalization of real world data in a submillimeter accuracy, as they argue, which contributed to the emerge of autonomous driving cars as well as high level simulation tools.

For mapping a place using this approach, first a point cloud is extracted, using LiDAR. This point cloud encodes

information of the road, which the os used to identify single-point objects (e.g. road signs etc.). These data will be later classified following the OpenDRIVE defused standards.

OpenDRIVE is used to model different road types (i.e. car, bicycle, rail or pedestrian), various types of crossroads (i.e. start/end of crossing, extra lines on crossing etc.), different number of lanes as well as different road curvature (by following predefined standards).

OpenDRIVE is used to model road topography with lane level precision. It can be argued that crossroads can be precisely modeled dynamically by spline fitting of existing crowd-sourced data, while the number of lanes on a road can also be dynamically estimated, using the road width and the applied standards in the area.

OpenDRIVE can also be used for simulating:

- Vehicle dynamics: since the curvature of roads can be precisely modeled.
- Energy saving: by modeling distances and road inclination.
- Driver comfort: since potholes or other road diversities cab be exact modeled and the wheel suspension can be estimated.
- Traffic system planning: the capabilities of traffic system planning can be enhanced since its evolution can be simulated.

OpenDRIVE has some open challenges. For example there are not standards established on how to map indoor places such as indoor parking places and road tunnels.

Building information modeling (BIM)

BIM [68] is in the middle of Architecture Engineering and Construction (AEC) phases. It enables the digital representation of the physical and functional characteristics of places such as the internal structure, the furniture, building size, material type, wall colors and so on. Its files can be exchanged or networked to support decision-making about a place, since building under development can be adopted using features from existing buildings. The use of BIM can be extended beyond the planning and design phase of the project, throughout the building life cycle.

BIM is a used in a collaboration with Industry Foundation Classes (IFC). The IFC model specification is open and available. It is registered by ISO and is an official International Standard ISO 16739:2013. IFC follows an entity-relationship model organized into an object-based inheritance hierarchy. Entities of the model describe building elements, geometry and constructs. IFC has emerged from the International Alliance for Interoperability (IAI). IAI defines explicit shape representation between 12 US companies and is responsible for the organization of developing IFC.

The various subsets of BIM are commonly described in terms of dimensions:

- 3D (object model): Enables design and visualization of the 3D structure in a construction level of detail.
- 4D (time): The planning process is linked with the construction activities enabling the simulation of construction progress and enabling to communicate problems regarding spatial or temporal characteristics.
- 5D (cost): Since BIM is often connected to seller databases, it enables the instant estimation of the cost of the model against time.
- 6D (operation): BIM provides a detailed description of the building elements and engineering services. This information can be used to specify how the building facilities can be managed.
- 7D (sustainability): BIM can accurately estimate carbon emissions for specific elements of a project and validate the design in such a way that alternative options are always available.
- 8D (safety): BIM enables an accurate simulation. In this way, safety aspects in both design and construction, can be addressed and the project performance can be predicted before they are being build.

Its drawback is that it is designed to model a single building, limiting any geospatial relation with other buildings.

CityGML

CityGML [69] is a set of classes designed for describing types of objects within a 3D virtual city mode. It consists of two modules, the core module and the extension module. The core module must be implemented by any system, since it comprises the basic concepts and components of the CityGML. The extensions cover specific thematic (i.e. bridge, building, tunnel etc.).

Different Level of Details (LoD) are supported by CityGML. This is necessary in order to reflect independent data with different requirements. An object can be represented in different LoD at the same time. The different Levels of Details are:

- LoD0: Buildings are represented by footprints or roof edge polygons.
- LoD1: Building are represented as prismatic objects with flat rood structures.
- LoD2: Roof structures can be differentiated and different thematic surfaces can also be presented.
- LoD3: Architectural models can be represented including detailed wall structures, doors and windows.
- LoD4: Interior structure can be modeled. Such as interior doors, rooms, stairs or furniture.

Accuracy is the main difference between the different LoD and concerns positional of objects or their height. As a result, the LoD1 is accurate by 5m, the LoD2 by 2m, LoD3 is accurate by 0.5m and the LoD4 is accurate by 0.2m. CityGML models Semantic, Geometric and Topologic characteristics. At the Semantic level, walls, windows and rooms are represented as features, while their relationships are modeled as aggregated hierarchies. At the spatial level their location is represented by features and their links are modeled by the corresponding relationships. All objects need to be virtual closed in order their volume or surface to be computable. This is important especially for simulations.

Keyhole Markup Language (KML)

KML is an international standard maintained by the Open Geospatial Consortium, Inc. (OGC). It is a file format used to display geographic data in an Earth browser such as Google Earth. One can create KML files to pinpoint locations, to add image overlays, or to expose rich data in many ways. In general, KML can be used to augment existing maps with additional information or arbitrary objects. Additionally, it can be embedded in a Web page enabling the creation of interactive mash-ups. It is not restricted only on the visualization of graphical data on the globe, but also the control of the user's navigation in the sense of where to go and where to look. Hence, KML is complementary to most of the key existing OGC standards including GML (Geography Markup Language). Finally, it supports geometry elements derived from GML such as point, line string, linear ring, and polygon.

5.5 Comparison of the Models

This section provides a comparison of the different models analyzed in the previous sub-chapters. The models have been compared against geometry, topology, layer structure, standardization and the effort for transmitting them.

The criteria that were used for the comparison of the models for indoor information are:

- Geometry: If the model can represent geometric characteristics of the space. The metrics of this category are:
 1. 1D (i.e. node) representation of places.
 2. 2D geometrical representation of places.
 3. 3D representation of places.
 4. Geometry of places, objects and animation.
- Standard: Whether or not is supported by a recognized organization. The metrics are:
 1. By OGC but not specific for indoors.
 2. By ISO but not specific for indoors.
 3. By OGC for indoors.
 4. By ISO for Indoors.

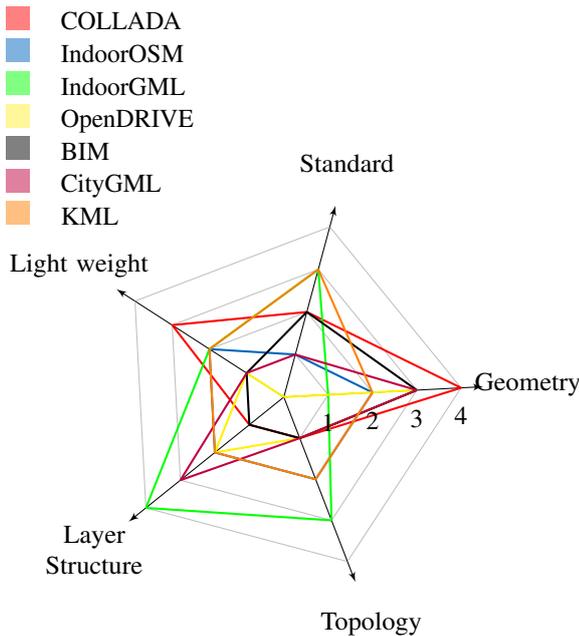


Figure 1. A caption for the diagram

- **Lightweight:** Whether it requires high throughput for transmitting the specified information. The metrics are:
 1. XML structured document.
 2. Compressible XML.
 3. JSON structured document.
 4. Compressible JSON.
- **Layer Structure:** Whether the model supports layer structure for its features. The metrics are:
 1. Abstract layer structure.
 2. Basic layer structure.
 3. Point of interest layer structure.
 4. Different transportation modes can be visualized or hidden.
- **Topology:** Whether topological relationships are described by the model.
 1. Adjacency: adjacent places are explicitly defined or can be retrieved via simple computations.
 2. Connectivity: Can presents graph based relations between cells.
 3. Vertical connections: The model is able to provide navigation over different levels of a structure.

Regarding the comparison, COLLADA supports animating characteristics and JSON format, this is the reason it scores high on the geometry and lightweight. From the other side, it lacks of topology representation and layer structure representation, since it has not been designed for navigation.

IndoorOSM is not precisely designed for indoor information, considering the fact that the features are used for representing discrete concepts of buildings, such as floors, are simple enhanced from outdoor locations such as bridges. This is the main reason it is weak in all perspectives.

IndoorGML on the other hand, it is designed for indoor places, does not have its own defined way for representing geometry, but instead it provides via referencing to another. This is the reason it scores low in geometry. It is XML based and not lightweight. In topology, it scores higher than any other, since it has been designed for addressing even vertical topological relationships. It enables layer structure, even for different use cases (i.e. navigation for pedestrians and navigation from paths accessible with wheelchairs).

OpenDRIVE, even-though essential for autonomous driving, it is not supported by any big organizations. It is not light-weighted, since it is XML based and encodes a lot of information. It is not designed for describing cellular topology with vertical topological relationships.

BIM, since it is based on IFC (which is designed for representing geometry of architectural objects, indoors and outdoor), scores high on Geometry. IFC is also an ISO standard, but not for indoor information. Neither of both is designed for representing topology. It does not provide a layer structure that could help on navigation. It is not lightweight, since it carries a lot of information.

CityGML LoD4 can model indoor information. It is an OGC standard, but not for indoor information. It is not designed for providing navigation, as a result it scores low on layer structure and topology, while it carries a lot of information, hence it cannot be considered as a lightweight model.

Finally, KML is an ISO standard for outdoor information; it is supported by most of the mapping browsers but is not optimized for modeling indoor information and vertical topology.

6. Major Market Players Analysis

Location-based services have accounted for a revenue of USD 2.8 billion in 2010. The LBS market in Western Europe is expected to grow at a compound annual growth rate (CAGR) of 56.24 percent over the period 2014-2019 [70]. Additionally, the global indoor location-based services market is expected to grow at a CAGR of 29.7 percent over the period 2014-2019 [71]. Moreover, the global market for mobile mapping will exhibit significant growth during the forecast period, and will grow at a CAGR of over 14% until 2020 [72]. Taken into consideration that, even though today's location-based services target mostly outdoor users, studies find that people spend some 80% of their time indoors [2], [3]. Hence, many firms in the fields of chipset manufacturers, mobile OS manufacturers, map providers, handset manufacturers and network equipment manufacturers have focused their resources on indoor LBS. In this section some of them have been described.

6.1 Chip set Manufacturers

1. **Broadcom:** acquired by Avago in 2016, it belonged to the wireless and broadband communication business. It introduced the BMC43462 system on chip (SoC), which integrates the AccuLocate technology on an 802.11ac Wi-Fi chip. AccuLocate technology relies upon fine timing measurement (FTM) technology which is independent of influencing environmental factors, enabling it of providing sub-meter accuracy.
2. **Qualcomm:** is a semiconductor manufacture company, in the wireless and broadband communication business. It has developed the IZat location technology. This technology, fuses satellite, WLAN, cellular networks, embedded sensors, a network of cloud-based assistance servers and object recognition to pinpoint the location of the user.
3. **InvenSense:** provides sensor platform solutions. They also provide the InvenSense Positioning Library (IPL). This library, besides GPS, it fuses gyroscope, accelerometer, magnetometer and barometric pressure sensors for tracking the position of a pedestrian or a vehicle.
4. **STMicroelectronics:** is the world's biggest supplier of consumer microelectromechanical systems. It has introduced the first dual-core gyroscope for managing user-motion while stabilizing the smartphone camera. Their indoor navigation system uses data from Wi-Fi access points, motion sensors and satellite-based positioning.
5. **CSR:** It was recently acquired by Qualcomm. Its main products were connectivity, audio, imaging and location chips. It developed a technology called SiRFusion where a Database with WiFi and magnetic fingerprints are extracted via crowdsourcing. After the learning period the technology promises accurate localization.

6.2 Mobile OS Manufacturers and Map providers

1. **Apple:** has recently acquired a startup (WiFiSlam) which provides technology that enables dynamically localization of people, based on WiFi RSS. They have also acquired a company that provides augmented reality solutions (Metaio). Services provide by this company can emerge alternative ways of localization, mapping and navigation. Additionally, a recent patent [73] strongly indicates that Apple is progressing on indoor localization. Finally, Apple has launched a dedicated indoor mapping app on iOS, letting business owners map out their venues using just their iPhones [74].
2. **Google:** Google Maps argues to have 10000 venues mapped, while Google Tango Project, a tablet equipped with depth sensor and accurate inertial motion sensors promises accurate indoor mapping for the crowd.

3. **Microsoft:** Bing Maps argue 3000 venues mapped, while HoloLens, a new project that provides Augmented Reality (the first device with a Holographic Processing Unit HPU) promises localization and mapping on the fly.
4. **HERE:** HERE Maps claim to possess more than 49.000 unique building maps in 45 countries.

6.3 Handset Manufacturers

1. **Motorola:** has introduced the TRX Indoor Localization System, which tracks and monitors location of persons in indoor settings. It can also model buildings in 3D in real time, while it can also identify activities. It is equipped with a gyroscope, accelerometer, pressure sensor, compass, and ranging sensors. Additionally, the model ATRIX™ HD MB886 supports the Indoor Location Manager (ILM). ILM provides mechanisms for determine the user's location indoor with dead reckoning, Wi-Fi or Hybrid positioning. It also provides mechanism for Self-Calibrating Wi-Fi using RSS. Finally, Motorola provides also Bluetooth Low Energy beacons for indoor localization.
2. **Nokia:** has introduced on 2011 the indoor location technology HAIP (High Accuracy Indoor Positioning). This technology claims to provide 0.3m of position accuracy using directional positioning beacons installed in covered areas.
3. **Sony Ericson:** provides two applications for indoor localization and mapping. The first is the SemcMap, it provides an indoor map service which allows the user to create his own indoor map by built up polygons and it supports panning and zooming on different floors, as well as searching. The second is the Indoor Finder app which provides walking directions, voice guidance, an augmented reality view and an advanced way to handle the map creation as well as a local positioning engine using Wi-Fi and Bluetooth via trilateration positioning and RSS.

6.4 Network Equipment Manufacturers

1. **Cisco:** has introduced the Mobility Services Engine platform. It uses Wi-Fi to increase visibility into the network, it deploys location-based mobile services, and can strengthen security. It can locate any Wi-Fi device in a venue, including smartphones and tablets, Wi-Fi tags, and Wi-Fi interferers, while location data can be exported to other applications using REpresentational State Transfer (REST) APIs. It also provides graphical user interface for Analytics in a venue-specific, location-based mobile service.
2. **Aruba Networks:** has introduced Aruba Beacons. They provide indoor location for mobile devices using Bluetooth Low-Energy (BLE) technology (Bluetooth 4.0).

6.5 InLocation Alliance

The InLocation Alliance aims to accelerate the indoor localization technologies. It is founded in August 2012 by: Broadcom, CSR, Dialog Semiconductor, Eptisa, Geomobile, Genasys, Indra, Insiteo, Nokia, Nomadic Solutions, Nordic Semiconductor, Nordic Technology Group, NowOn, Primax Electronics, Qualcomm, RapidBlue Solutions, Samsung Electronics, Seolane Innovation, Sony Mobile Communications, TamperSeal AB, Team Action Zone and Visioglobbe.

7. Conclusions

We can safely conclude from this state-of-the-art, that even though there is a plethora of localization technologies, there has none yet been established. The main reason is due to their high dependency on the building infrastructure (i.e. existing WiFi AP, magnetic disturbances or dedicate hardware). Hence, there is not a localization technology that works on every environment. Additionally, it can be concluded that the increase of precision in localization techniques implies an increase on the cost (i.e. BLE Beacons), while decreasing the infrastructure dependency, implies reduction on precision (i.e. dead reckoning).

Another conclusion of this state-of-the-art is that existing maps cannot be used for localization without being enhanced with semantic information (i.e. WiFi fingerprint, light or even sound fingerprint of particular areas) and topological relations. Additionally, retrieving adjacency and connectivity is a challenging procedure. Alternative strategies of storing and querying spatial information need to be emerged. A well upon agreed model for representing indoor spatial information has to be established.

Crowd-sourcing indoor spatial information seems to be a promising procedure for mapping. Pedestrian Dead Reckoning seems a promising infrastructure independent localization technique. The large accumulation of error can be reseted using unique identified locations. This procedure requires established mechanisms that manage accuracy, uncertainty and ambiguity.

Finally, keeping spatial databases for localization always updated alternative technologies need to be researched, while IndoorGML can be a promising model for representing indoor information.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] Axel Küpper. *Location-based services: fundamentals and operation*. John Wiley, Chichester, England ; Hoboken, NJ, 2005.
- [2] Buildings and the Environment: A Statistical Summary. Technical report, U.S. Environmental Protection Agency Green Building Workgroup, December 2004.
- [3] The National Human Activity Pattern Survey (NHAPS): A Resource for Assessing Exposure to Environmental Pollutants | Exposure Science.
- [4] Ki-Joune Li and Jiyeong Lee. Indoor spatial awareness initiative and standard for indoor spatial data. In *Proceedings of IROS 2010 Workshop on Standardization for Service Robot*, volume 18, 2010.
- [5] Moustafa Elhamshary and Moustafa Youssef. SemSense: Automatic construction of semantic indoor floorplans. In *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, pages 1–11. IEEE, 2015.
- [6] Imad Afyouni, Cyril Ray, and Christophe Claramunt. A fine-grained context-dependent model for indoor spaces. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, pages 33–38. ACM, 2010.
- [7] April Berthene. Why one airport chooses Wi-Fi for mobile navigation over beacons. <https://goo.gl/V3koXv>. accessed online 05/2016.
- [8] Robin Henniges. Current approaches of Wifi Positioning.pdf, 2012.
- [9] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, MobiSys '12*, pages 197–210, New York, NY, USA, 2012. ACM.
- [10] Lenin Ravindranath, Calvin Newport, Hari Balakrishnan, and Samuel Madden. Improving wireless network performance using sensor hints. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, NSDI'11*, pages 281–294, Berkeley, CA, USA, 2011. USENIX Association.
- [11] U. Lipowezky and I. Vol. Indoor-outdoor detector for mobile phone cameras using gentle boosting. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 31–38, June 2010.
- [12] Pengfei Zhou, Yuanqing Zheng, Zhenjiang Li, Mo Li, and Guobin Shen. Iodetector: A generic service for indoor outdoor detection. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, SenSys '12*, pages 113–126, New York, NY, USA, 2012. ACM.
- [13] Valentin Radu, Panagiota Katsikouli, Rik Sarkar, and Mahesh K. Marina. A semi-supervised learning approach for robust indoor-outdoor detection with smartphones. In *Proceedings of the 12th ACM Conference on Embedded*

- Network Sensor Systems*, SenSys '14, pages 280–294, New York, NY, USA, 2014. ACM.
- [14] Clemens Nylandsted Klokmose, Matthias Korn, and Henrik Blunck. WiFi proximity detection in mobile web applications. pages 123–128. ACM Press, 2014.
- [15] M. Llombart, M. Ciurana, and F. Barcelo-Arroyo. On the scalability of a novel WLAN positioning system based on time of arrival measurements. In *5th Workshop on Positioning, Navigation and Communication, 2008. WPNC 2008*, pages 15–21, March 2008.
- [16] Zdeněk NĚMEC and Pavel BEZOUŠEK. The Time Difference of Arrival Estimation of Wi-Fi Signals. *Radio-engineering*, 17(4):51, 2008.
- [17] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan. Received-Signal-Strength-Based Indoor Positioning Using Compressive Sensing. *IEEE Transactions on Mobile Computing*, 11(12):1983–1993, December 2012.
- [18] Mahnaz Roshanaei and Mina Maleki. Dynamic-KNN: A novel locating method in WLAN based on Angle of Arrival. In *IEEE Symposium on Industrial Electronics and Applications (ISIEA)*, volume 2, pages 722–726, 2009.
- [19] Matteo Cypriani, Frédéric Lassabe, Philippe Canalda, and François Spies. Wi-Fi-based indoor positioning: Basic techniques, hybrid algorithms and open software platform. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–10. IEEE, 2010.
- [20] Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman. Indoor location sensing using geo-magnetism. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 141–154. ACM, 2011.
- [21] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.
- [22] S. De Lausnay, L. De Strycker, J. P. Goemaere, N. Stevens, and B. Nauwelaers. Influence of MAI in a CDMA VLP system. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–9, October 2015.
- [23] Zeyu Wang, Zhice Yang, Jiansong Zhang, Chenyu Huang, and Qian Zhang. Demo: Lightweight Visible Light Communication for Indoor Positioning. pages 465–465. ACM Press, 2015.
- [24] M. Kourogi and T. Kurata. A method of pedestrian dead reckoning for smartphones using frequency domain analysis on patterns of acceleration and angular velocity. pages 164–168, May 2014.
- [25] Seyed Amir Hoseinitabatabaei, Alexander Gluhak, Rahim Tafazolli, and William Headley. Design, Realization, and Evaluation of uDirect-An Approach for Pervasive Observation of User Facing Direction on Mobile Phones. *IEEE Transactions on Mobile Computing*, 13(9):1981–1994, September 2014.
- [26] C. Combettes and V. Renaudin. Comparison of misalignment estimation techniques between handheld device and walking directions. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, October 2015.
- [27] Jun-geun Park, Ami Patel, Dorothy Curtis, Seth Teller, and Jonathan Ledlie. Online pose classification and walking speed estimation using handheld devices. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 113–122. ACM, 2012.
- [28] Tanusri Bhattacharya Kushani Perera. Trajectory Inference for Mobile Devices Using Connected Cell Towers. 2015.
- [29] Daniel Hahnel, Wolfram Burgard, Dieter Fox, Ken Fishkin, and Matthai Philipose. Mapping and localization with RFID technology. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 1015–1020. IEEE, 2004.
- [30] Ruoxi Jia, Ming Jin, and Costas J. Spanos. SoundLoc: Acoustic Method for Indoor Localization without Infrastructure. *arXiv preprint arXiv:1407.4409*, 2014.
- [31] Adopted Specifications | Bluetooth Technology Website. <https://www.bluetooth.com/specifications/adopted-specifications>. accessed online 05/2016.
- [32] S. Gezici, Zhi Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE Signal Processing Magazine*, 22(4):70–84, July 2005.
- [33] Guoqiang Mao and Baris Fidan, editors. *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking*. IGI Global, 2009.
- [34] Gregory Levitin. Evaluating correct classification probability for weighted voting classifiers with plurality voting. *European journal of operational research*, 141(3):596–607, 2002.
- [35] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm, 1996.
- [36] João Gama and Pavel Brazdil. Cascade generalization. *Machine Learning*, 41(3):315–343, 2000.
- [37] Joseph Sill, Gábor Takács, Lester W. Mackey, and David Lin. Feature-weighted linear stacking. *CoRR*, abs/0911.0460, 2009.
- [38] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

- [39] Moustafa Alzantot and Moustafa Youssef. CrowdInside: Automatic Construction of Indoor Floorplans. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems, SIGSPATIAL '12*, pages 99–108, New York, NY, USA, 2012. ACM.
- [40] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics Automation Magazine*, 13(2):99–110, June 2006.
- [41] Hang Chu, Dong Ki Kim, and Tsuhan Chen. You Are Here: Mimicking the Human Thinking Process in Reading Floor-Plans. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2210–2218, 2015.
- [42] G. A. Einicke and L. B. White. Robust extended kalman filtering. *IEEE Transactions on Signal Processing*, 47(9):2596–2599, Sep 1999.
- [43] Pierre Del Moral. Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, 2(4):555–581, 1996.
- [44] Sabry F El-Hakim and Pierre Boulanger. Mobile system for indoor 3-d mapping and creating virtual environments, December 28 1999. US Patent 6,009,359.
- [45] Joon-Seok Kim, Sung-Jae Yoo, and Ki-Joune Li. Integrating IndoorGML and CityGML for Indoor Space. In Dieter Pfoser and Ki-Joune Li, editors, *Web and Wireless Geographical Information Systems*, number 8470 in Lecture Notes in Computer Science, pages 184–196. Springer Berlin Heidelberg, May 2014. DOI: 10.1007/978-3-642-55334-9_12.
- [46] Hao Liu, Ruoming Shi, Ling Zhu, and Changfeng Jing. Conversion of model file information from IFC to GML. In *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, pages 3133–3136. IEEE, 2014.
- [47] Mikkel Boysen, Christian de Haas, Hua Lu, and Xike Xie. A Journey from IFC Files to Indoor Navigation. In *Web and Wireless Geographical Information Systems*, pages 148–165. Springer, 2014.
- [48] T. Eaglin, K. Subramanian, and J. Payton. 3d modeling by the masses: A mobile app for modeling buildings. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 315–317, March 2013.
- [49] Ruipeng Gao, Mingmin Zhao, Tao Ye, Fan Ye, Yizhou Wang, Kaigui Bian, Tao Wang, and Xiaoming Li. Jigsaw: indoor floor plan reconstruction via mobile crowdsensing. pages 249–260. ACM Press, 2014.
- [50] Zhengyou Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):892–899, July 2004.
- [51] S. Leutenegger, M. Chli, and R. Y. Siegwart. In *O11*.
- [52] Robert C. Bolles, H. Harlyn Baker, and David H. Mount. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.
- [53] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986.
- [54] H.P.V. C. Method and means for recognizing complex patterns, December 18 1962. <https://www.google.com/patents/US3069654>.
- [55] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663, 2012.
- [56] J Christopher Dainty. *Laser speckle and related phenomena*, volume 9. Springer Science & Business Media, 2013.
- [57] Wenyu Guo, Nick P. Filer, and Stephen K. Barton. 2d indoor mapping and location-sensing using an impulse radio network. In *Ultra-Wideband, 2005. ICU 2005. 2005 IEEE International Conference on*, pages 296–301. IEEE, 2005.
- [58] Slawomir Grzonka, Frederic Dijoux, Andreas Karwath, and Wolfram Burgard. Mapping indoor environments based on human activity. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 476–481. IEEE, 2010.
- [59] M. Alzantot and M. Youssef. UPTIME: Ubiquitous pedestrian tracking using mobile phones. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 3204–3209, April 2012.
- [60] Seth Teller, Jonathan Battat, Ben Charrow, Dorothy Curtis, Russell Ryan, Jonathan Ledlie, and Jamey Hicks. Organic indoor location discovery. *Computer Science and Artificial Intelligence Laboratory Technical Report*, 75:16, 2008.
- [61] Jun-geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth Teller, and Jonathan Ledlie. Growing an organic indoor location system. page 271. ACM Press, 2010.
- [62] Laura Radaelli and Christian S. Jensen. Towards fully organic indoor positioning. pages 16–20. ACM Press, 2013.
- [63] Management Association Resources, Information. *Geographic Information Systems: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications*. IGI Global, September 2012.
- [64] ISO/PAS 17506:2012 - Industrial automation systems and integration - COLLADA digital asset schema specification for 3d visualization of industrial data. http://www.iso.org/iso/catalogue_detail.htm?csnumber=59902/.

- [65] Marcus Goetz and Alexander Zipf. *Extending OpenStreetMap to indoor environments: bringing volunteered geographic information to the next level*. CRC Press: Delft, The Netherlands, 2011.
- [66] OGC® IndoorGML | OGC. <http://www.opengeospatial.org/standards/indoorgml>.
- [67] OpenDRIVE - Downloads. <http://www.opendrive.org/download.html>.
- [68] Charles Eastman and And Others. An Outline of the Building Description System. Research Report No. 50. September 1974.
- [69] CityGML | OGC. <http://www.opengeospatial.org/standards/citygml>.
- [70] Indoor LBS Market in Western Europe 2015-2019 | Technavio - Discover Market Opportunities.
- [71] Global Indoor LBS Market 2015-2019 | Technavio - Discover Market Opportunities.
- [72] Global Mobile Mapping Market 2016-2020 | Technavio - Discover Market Opportunities. <http://www.technavio.com/report/global-machine-machine-m2m-and-co\nnected-devices-mobile-mapping-market>.
- [73] Ronald Keryuan Huang, Rob Mayor, Isabel Mahe, and Patrick Piemonte. Interactive gaming with co-located, networked direction and location aware devices, May 10 2016. US Patent 9,333,424.
- [74] Mikey Campbell Sunday, November 01, 2015, and 09:47 pm PT. Apple indoor positioning app 'Indoor Survey' spotted on iOS App Store.

Eco-Sensitive Traffic Management

Nihan Celikkaya, Eftychios Papapanagiotou and Fritz Busch

Chair of Traffic Engineering and Control, Technical University of Munich, Munich
{nihan.celikkaya; eftychios.papapanagiotou; fritz.busch}@tum.de

Abstract

Road transportation has several negative external effects and emissions are one of the most important ones. This chapter focuses on two emission types: air pollutants and greenhouse gases which cause several environmental and health problems and where road transportation is one of the main contributors. In order to reduce these emissions, specific standards and restrictions are introduced in last decades which resulted in a decrease in annual mean air pollutant emission levels to a degree in applied countries. However, the limits for air pollutants are often still exceeded in hotspots where highest concentration levels are seen. Urban emission hotspots are mostly located in central areas close to the local emission sources like road traffic. Among many different policy instruments and measures, eco-sensitive traffic management (ETM), which is a dynamic traffic management application, aims to reduce road traffic related emissions while using the network as efficiently as possible and without creating new hotspots in the long run. In ETM, traffic management measures are activated in case of a limit excess of pollutants (current and/or projected). This requires constant air quality assessment and projection which are performed today with the help of traffic and air quality related detectors and modelling tools with different scopes. ETM systems are being studied in last decade and first results show that there is room for improvements in several areas. Critical points for analysis phase are availability and quality of the input data, aggregation levels and accuracy of the models, while for application phase informing and acceptance of users as well as coherence with other long-term and short-term measures are essential. Example studies commonly focus on reduction of NO_x and PM₁₀ emissions (which are the two air pollutants mainly caused by road transportation and problematic in urban areas) and indicate that these air pollutants can be reduced by ETM. Yet, they also point out that impacts of the measures should be carefully monitored and comprehensively evaluated in order to avoid creating other traffic or environmental problems like relocating the hotspots.

Keywords

Dynamic Traffic Management; Road Transportation Related Emissions; Air Quality Assessment

FUNDAMENTALS OF TRAFFIC MANAGEMENT

The continuous urbanization in all parts of the world generates numerous challenges for all cities, from small urban settlements to mega-cities, since the available space and infrastructure is unable to fully handle the population growth. In 2014 more than half (54%) of the world's population resided in urban areas (increased from 30% in 1950's) and this number is projected to reach 66% by 2050 [1]. Despite the great variety in urban environments and the different challenges that they face, urban mobility is a common issue with high priority. Moreover, the balance between travel demand and transport supply defines the level of mobility provided by the urban transportation system to its users. Congestion appears because demand at a specific part of the network in a certain point in time exceeds the capacity (i.e. transport supply). This can happen either due to a sudden increase in the demand (e.g. commuter traffic) or because of a drop of the capacity (e.g. construction site).

Introduction

Traffic management aims to mitigate the negative impacts of traffic on safety, environment, traffic flow and economic efficiency [2] by influencing and balancing traffic demand and supply through sets of appropriate short-, medium- or long-term measures [3]. Typically, traffic management measures look to reduce (or redistribute) demand and increase the capacity. Boltze [4] suggests that the term capacity should not be defined only by traffic characteristics, but it must also consider other aspects, such as noise and air pollution levels. This allows traffic management to follow certain policies and objectives by adding certain limitations to the capacity of the network (e.g. limit for CO₂ emissions). Figure 1 gives an overview of the main strategies for traffic management based on [5]. The strategy of avoiding traffic aims to reduce the travel demand, shifting traffic intends to redistribute traffic in time, space and among different traffic modes while controlling traffic aims to optimize current traffic flows by influencing mainly the supply through traffic control actuators (e.g. traffic lights).

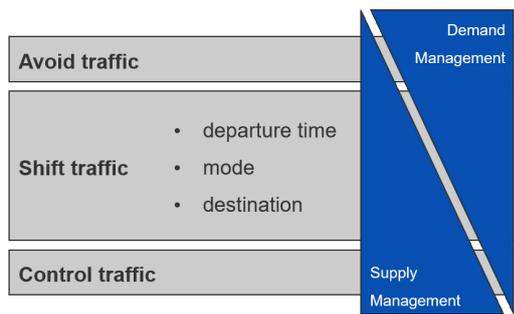


Figure 1. Traffic management strategies [4]

Based on the working mechanisms of its measures, traffic management can be separated into main categories as static traffic management, where long-term measures are in focus (e.g. introducing a reduced-speed area in the city center) and dynamic traffic management that emphasizes on short-term measures for specific traffic situations (e.g. variable message signs that show different speed limits according to the traffic situation) [3]. The field of avoiding traffic is almost completely served by measures of strategic traffic management while controlling traffic mostly relies on short-term measures provided by the dynamic traffic management. Shifting traffic however, is the field in which both strategic and dynamic measures have to cooperate closely. With respect to the focus of the LLCM project, targeting innovative and real-time technological solutions for transportation, the dynamic traffic management is the basis for the case studies described in this paper.

Dynamic Traffic Management

According to [5] dynamic traffic management consists of influencing the current traffic demand and the available transport supply through coordination of measures according to the situation, in order to achieve the best possible level of mobility for the specific time span. For every traffic situation that may occur, a specific strategy has to be developed in advance and be ready for implementation. A means that allows traffic managers to react quickly to a currently identified situation and to utilize past experiences is the so called “scenario approach” (Figure 2).

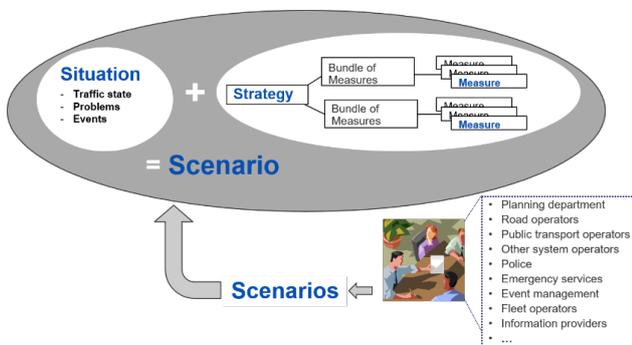


Figure 2. Definition of situation, strategy and scenario in dynamic traffic management, adapted from [5]

In Figure 2, the word situation depicts the current traffic state including problems, events and other relevant situations. Strategy is a predefined action plan for taking measures in order to improve a predefined situation. The scenario represents the combination of a situation and the corresponding strategy [5]. Since the time to select the traffic measures is limited in real-time, dynamic traffic management strategies are developed offline using mainly traffic simulation to reproduce the situation and evaluate the impacts of each strategy. The plausible measures are then listed to be used later by the operators at Traffic Management Center (TMC).

The implementation of the measures can be summarized in 6 steps (Figure 3): the essential step is to observe the network condition which is necessary to identify the problems in real-time. When a problem is identified, the provided list of possible strategies is evaluated (through simulation of one or more scenarios) and the best one will be implemented. The impact of the implemented strategy is also monitored to make the necessary changes if needed. The following figure illustrates the system architecture of dynamic traffic management strategy planning and implementation [5].

Because of the wide range of dynamic traffic management measures that can be implemented, a number of categories can be defined in order to distinguish and choose between them. A measure can influence the movement of travelers before (pre-trip) or during (on-trip) the trip. Moreover, measures can be compulsory (regulation and control measures) or voluntary, where just information or a recommendation is provided. Measures can be sent to all travelers (collective measures) or to individual users (e.g. dynamic navigation systems). Depending on the traffic mode that the dynamic traffic management measures focus on, they can be divided in following groups [6]:

- Private transportation measures
- Public transportation measures
- Multi-modal and inter-modal transportation measures
- Non-motorized private transportation measures

Table 1 shows a categorization of dynamic traffic management measures based on [5] and [6]. Private transportation measures focus typically on optimizing the current traffic flow and reduce congestion, while public transportation measures focus on improving the overall public transport capacity and reliability. In addition, traffic management typically aims to give incentives to the travelers to use alternative modes (e.g. use car-sharing instead of private car). Furthermore, measures in favor of cyclists and pedestrians can be adopted to make these sustainable modes more attractive and safe.

The implementation of the selected measures requires the application of intelligent systems. Intelligent Transportation Systems (ITS) are systems that use telematics (**Telecommunication & Informatics**) and communication in vehicles, between vehicles and between vehicles and infrastructure to achieve the goals of transport and mobility management. ITS applications are not limited with road transportation, it is

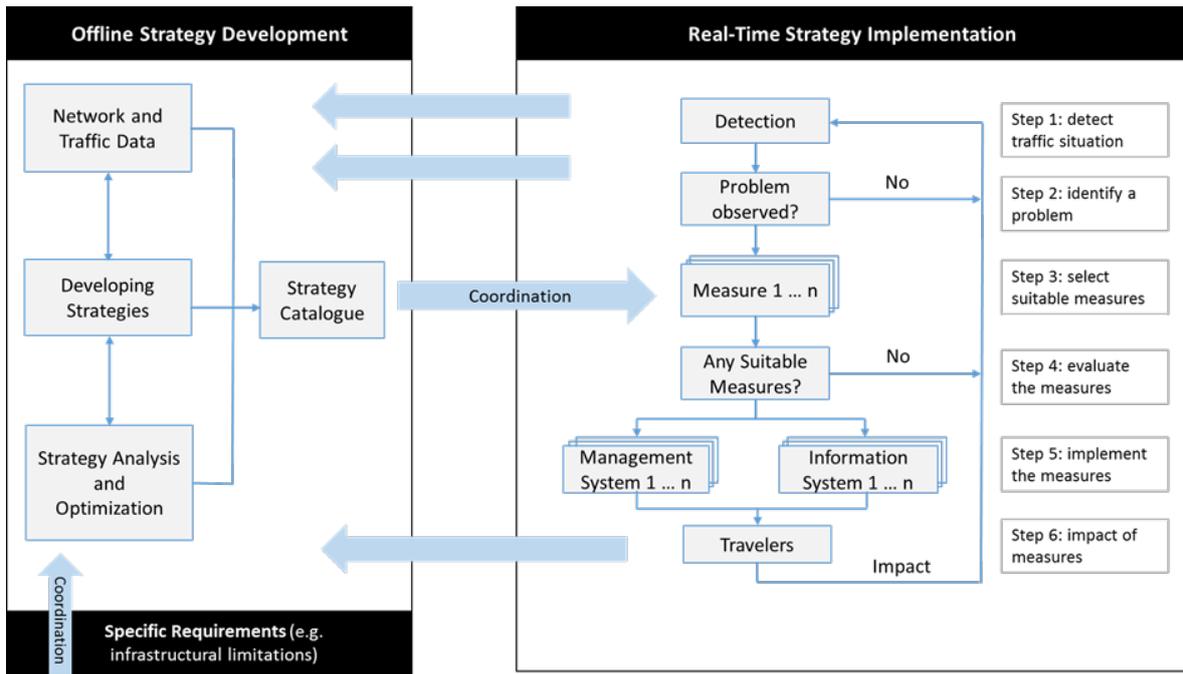


Figure 3. System architecture of traffic strategy management and implementation, translated and adapted from [5]

Table 1. Possible measures for dynamic traffic management, adapted from [5, 6]

Private Transport Measures	Public Transport Measures
Re-routing of traffic streams	Redistributing passengers inside public transport
Increase of capacity (through e.g. traffic signal control)	Rerouting public transport vehicles
Allow temporary special lanes	Public transport prioritization
Variable speed limits	Increase of capacity of a line
Ramp metering	Introduce special lines, lanes and stops
Restriction of overtaking	Secure (inter-) connections between lines
Dynamic adjustment of available parking spaces	Increase of accessibility and attractiveness
Dynamic toll system	Adjust ticket prices
Pre-emption of emergency vehicles	
Multi-modal and Inter-modal Measures	Non-motorized Transport Measures
Information about all modes and diverting measures	Prioritization of cyclists at intersections
Influencing the mode choice	Prioritization of pedestrians at intersections
Shifting the start of the trip	Temporary special lanes for cyclists
Changing the use of transport areas	Temporary pedestrian areas
Mobility pricing	Information about available bike-sharing systems
Information about available car-sharing vehicles	

applied also for rail, air, water transportation as well as multi-modal trip planning. ITS combine the necessary data collection, processing and simulation technologies to support short-term decision making and tactical actions. Additionally, they provide the technological means to deploy dynamic measures in the transportation system and communicate them towards the stakeholders. In today’s connected and digitalized world, ITS are a vital part of modern traffic management.

The European Commission supports the extended use of ITS solutions as part of the Digital Single Market Strategy for a more efficient management of transportation network. At the same time, already the next generation of ITS, the so-called Cooperative ITS (C-ITS), is promoted from the European Commission in order to take full advantage of the advances in connected vehicles technology [7, 8].

ROAD TRANSPORTATION AND EMISSIONS

Transportation is a derived demand; it takes place as a result of the need for delivering goods (i.e. freight transportation) and for reaching a destination to access a service or to do an activity (i.e. passenger transportation). While this accessibility is a great benefit that contributes considerably to the economic and social development; transportation has several negative effects like accidents, congestion and environmental impacts. These effects are called external effects (or external costs when they are considered in monetary terms) of transportation since they are not the primary concern of users' transportation decisions, rather a consequence which at the end affect other people or the whole society - in some cases even the future generations [9, 10, 11]. Therefore external effects of transport should be considered carefully during planning, implementation, evaluation and monitoring of transportation services.

According to Maibach, et.al. [9] external impacts of transportation can be categorized according to the problem areas as scarce infrastructure, safety and environment. While there had been some improvements, especially in developed countries, on issues of scarce infrastructure and safety, environmental effects gained significant importance globally as a result of unignorable visible effects and higher awareness in the society.

Negative impacts of transportation occur over different segments in environment. Some are directly linked to landscape, mostly caused by the infrastructure itself; like land consumption, land sealing, separation effect on habitats while others are caused by the use of the vehicles and the fuel like air pollution, climate change and noise [9, 10, 11].

This paper focuses mostly on air pollutants and partially on greenhouse gasses (which are, together with noise, light and vibration, referred to as environmental emissions of the road transportation) since existing ETM applications and studies focus on air pollution excess. Besides, it concentrates on traffic management for urban road networks, discusses the problems based on the data from Europe and through several examples from Germany.

Air Pollution, Greenhouse Gases and Road Transport

Atmosphere is an important part of the environment (i.e. biosphere) as it covers and protects the other layers [10]. Air pollutants and other gases in atmosphere may be transported over long distances and may exist longer in environment by diffusing in other layers (i.e. soil, water) [12]. Therefore emissions do not only have direct local effects and this makes the problem one of the most important negative environmental effects of transport. Air pollutants cause several health problems from irritations to respiratory or cardiovascular problems to cancer [13] and greenhouse gases contribute to the climate change which results in extreme weather conditions (e.g. heat waves, droughts, storms or floods) and "climate induced water and food shortages" [14, 15].

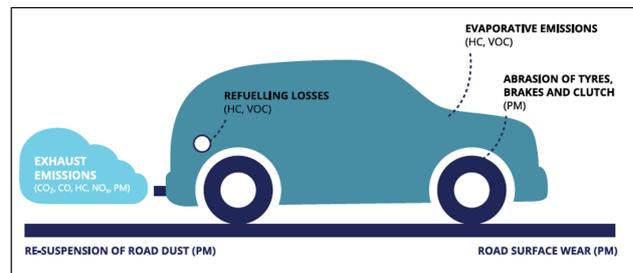


Figure 4. Different road transport emissions and sources [20]

There are two types of air pollutants: primary air pollutants which are emitted directly from sources and secondary pollutants which are formed in atmosphere later due to the existence of the primary pollutants [16, 17]. According to the World Health Organization (WHO), the six principal pollutants (i.e. classical pollutants) are oxides of nitrogen (NO_x), Carbon monoxide (CO), Particulate Matters (PMs), Lead (Pb), Sulfur dioxide (SO_2), and Ozone (O_3) [16]. NO_x , CO, Pb and SO_2 are primary pollutants, particulate matters can be primary or secondary according to their source and O_3 is a secondary pollutant [17].

Greenhouse gases (GHGs) absorb and emit radiation in atmosphere and cause changes on earth's climate, especially global warming [18]. Unlike pollutants, not all greenhouse gases directly threaten health and they are considered often separately. The three primary GHGs are carbon dioxide (CO_2), methane (CH_4) and nitrous oxide (N_2O) [16, 18]. It is today well known that GHGs are the major cause of the climate change and CO_2 is the dominating greenhouse gas with highest contribution [19]. A reduction in anthropogenic GHG emissions is crucial and in case of increasing concentrations, there is a danger that the whole climate system will change [19].

Predictably, transportation is not the only source of air pollutants or greenhouse gases. There are different other contributing sectors like industry, energy production and agriculture. Emission sources can be categorized not only according to the related sector but also according to the type of the source: point sources (e.g. factories), mobile or line sources (e.g. vehicles) and area sources (e.g. waste deposit areas) [17, 13].

Figure 4 explains and sums up the types of emissions that are emitted by road vehicles. They are categorized by the mechanism of production: exhaust and non-exhaust emissions. In addition to major pollutants (CO, NO_x and PM) and the major GHG (CO_2), road transport vehicles also emit hydrocarbons (HCs) and volatile organic compounds (VOCs). While CO_2 , CO, NO_x emitted only from exhaust, HC can also be emitted during refueling or evaporation. Particulate matters (PMs) have several different sources like exhaust, abrasion of car parts and the wear from the road surface [20]. Particulate matters that are emitted from exhaust contributes to smaller particles (i.e. fine particulate matters) while particulate matters from other sources contribute to $\text{PM}_{2.5}$ and PM_{10} [21]. It

is also important to note that different engine types have different exhaust emissions. To illustrate, while an Euro 6 diesel vehicle emit more NO_x and PM, an Euro 6 petrol vehicle emit more CO [20]. The figure can be associated more with the emissions from road transport in Europe. It does not include lead (Pb), sulfur dioxide (SO₂) which are listed as classical air pollutants by WHO as SO₂ is mostly emitted by non-transport sectors (e.g. burning coal) and sulphur together with lead is removed from the fuel in Europe (Figure 6) [22].

Trends and Critical Air Pollutants

There is a decreasing trend for annual average concentrations of major air pollutants from transport in Europe since 1990 [23]. However, when the GHGs are considered, emissions from transport sector (around 20%), in contrast to all other sectors, has increased [20, 24]. This escalation is not only caused by the increase in emissions from road transport (around 70% of the total transportation emissions) but also by the remarkable increase in air and sea transportation emissions over the last decades [24].

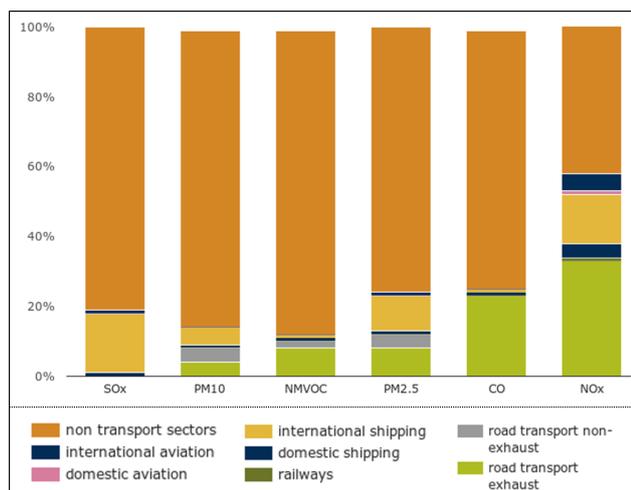


Figure 5. Contribution of transport sector to total emissions in Europe – data from 2015 [25]

In European cities road transportation is contributing majorly to NO_x and PMs as air pollutants [14]. In Figure 5 contribution of road transport (exhaust and non-exhaust) to total emissions in EU-28 member states can be seen in detail. The share of road transportation contributing to NO_x emissions is around 40 percent, to CO is around 20 percent and to PMs and NMVOC (Non-methane volatile organic compound) is around 10 percent [23]. Although CO emissions has the second highest share, CO emissions caused by road transport has been decreased by more than 80% since 1990 [23] in Europe, as a result of introduction of the first emission standards in 1992 [12].

Figure 6 shows the trends of some of the mentioned emissions from road transport in Europe between years 1990 and 2015. The graph indicates that all air pollutants show a decreasing trend. It can also be seen that SO_x, Pb, CO and

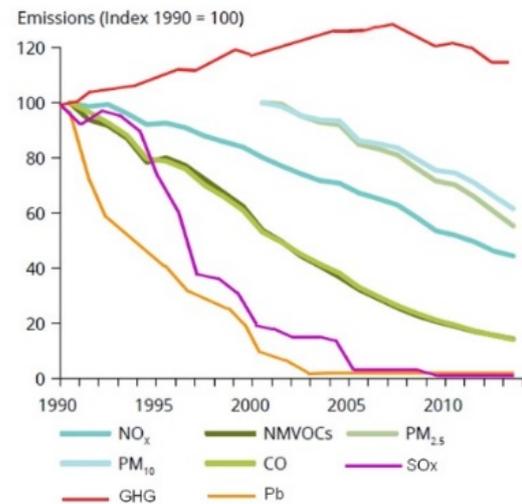


Figure 6. Trends of selected air pollutants and greenhouse gasses in Europe between 1990 and 2015, adapted from [23, 26]

NMVOC emissions have been reduced remarkably. Evidently NO_x, PM₁₀ and PM_{2.5} emissions were also reduced, by 56, 41 and 50 percent respectively, in this time period. There is however room for improvement, since the concentration of those pollutants in urban areas is still not meeting the required thresholds. In contrast to air pollutants, greenhouse gas emissions from road transport, similar to the overall transport sector, has been increased since 1990 and further reductions are crucial.

Regulations and Policies

In Europe, air pollutants are regulated by limit values that are set by European Commission (Directive 2008/50/EC) and the values are being updated regularly [27]. CO₂ emissions, on the other hand, are regulated by emission performance standards which regulate the average CO₂ emissions from new vehicles and mainly concerns car manufacturers [28]. Since urban traffic management targets GHG emissions only indirectly, the respective regulations will not be covered here.

Given the fact that the severity of health damages of emissions depends on the type of air pollutant (i.e. toxicity of the pollutant), the amount and duration of exposure [16]; different pollutants have different regulations defined by the concentration, averaging period and permitted exceedance. Permitted exceedance is the maximum number of observed limit excess (for given averaging period) per year. Example limit values for NO₂, PM₁₀ and PM_{2.5} can be seen in Table 2.

Member states have to assess air quality regularly and in case of a limit excess of regulated air pollutants, long term air quality plans as well as action plans that consider short term measures should be developed [27]. In addition, it is required that information about the amount and location of the excess pollution, type of area and possible origins of the pollution

Table 2. Air quality standards for NO₂, PM₁₀ and PM_{2.5} in Europe [27]

Pollutant	Concentration (µg/m ³)	Averaging period	Permitted Exceedance each year	Limit value entered into force
NO ₂	40	1 year	n/a	01.01.2010
	200	1 hour	18	
PM ₁₀	50	24 hour	35	01.01.2005
	40	1 year	n/a	
PM _{2.5}	25	1 year	n/a	01.01.2015

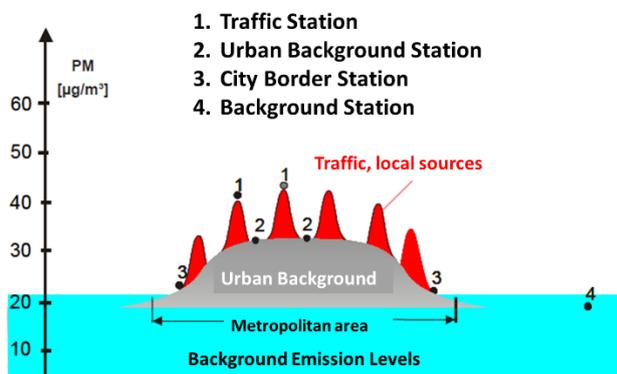


Figure 7. Types of emission stations, translated from (M. Lutz cited in [29])

are included in air quality plans [27]. In order to be able to provide this information, measurement stations are located in different areas: urban traffic locations, urban residential locations as well as suburban and rural locations [27].

Figure 7 illustrates that the measured emission levels at central locations do not only originate from local traffic but are composed by long range sources (background emissions), other urban sources (urban background) and other local sources. Measurement stations situated close to these local sources are the peak points (e.g. hotspots) where highest concentration levels, due to this cumulateness, are seen. Thus it should be kept in mind that traffic management measures can influence the emission levels at a certain hotspot only to a certain extent.

Although average air pollutant emissions from road transportation are decreasing, the above given limits are still being exceeded in these hotspots in Europe. Figure 8 illustrates the NO₂ emission observations from different measurement stations. It can be seen that while NO₂ emissions observed in background stations are mostly below limits, emissions measured at traffic stations largely show a limit excess. To illustrate with an example, in Germany 60% to 70% of limit excess of NO₂ emissions occurred between 2000 and 2015 was observed in urban traffic stations [30].

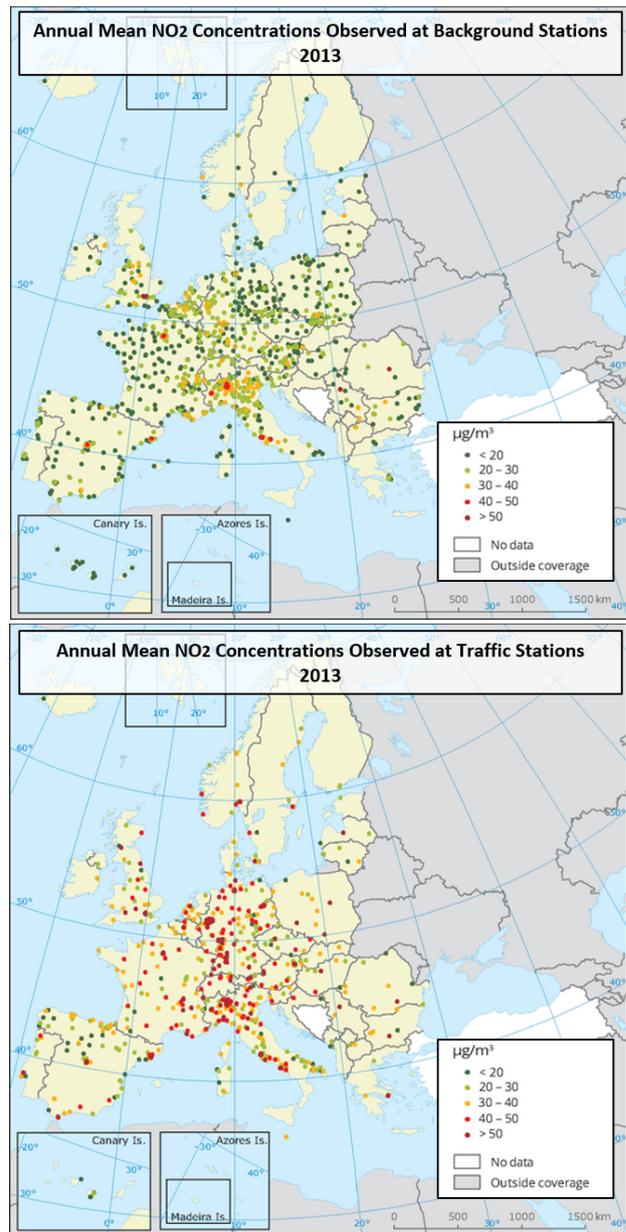


Figure 8. Comparison of annual mean NO₂ emissions from background stations and traffic stations [31]

There are several policies and policy instruments to reduce emissions of road transportation in urban areas. Policies can be categorized under four main focus areas as planning and regulation, infrastructure and operation, vehicle and fuel technology and information and awareness. Table 3 represents example policy instruments [10, 22, 32, 33, 16] for each focus area. It is important to mention that commonly strategies do not utilize only one policy instrument but combines many. For instance, if promotion of non-motorized modes (i.e. walking and biking) is set as one strategy to reduce emissions from the road transportation, this can be achieved by provision of attractive infrastructure, introduction of supportive regulations and raising awareness about advantages of these modes.

Table 3. Focus areas and example policy instruments for reduction of road transport emissions

Planning and Regulation	Infrastructure and Operation	Vehicle and Fuel Technology	Information and Awareness
Integrated land-use and transportation plans, Air quality action plans, Taxations on road, fuel or vehicle use, Demand management, congestion charging...	Improved public transportation services, Improved infrastructure for non-motorized transportation modes, Traffic management...	Vehicle inspection and maintenance, Promotion of cleaner vehicles, Promotion of cleaner fuels and energy production, Emission standards...	Eco driving behavior education, Promotion of environmental organisations, Awareness campaigns for emission free transportation modes...

To sum up, in order to be successful in reducing emissions of road transportation, it is important to integrate environmental effects of road transportation in planning processes, to control or limit use of undesired transportation modes, vehicles or fuels by regulations, to promote improvements in vehicle and fuel technology and to ensure a long term effect by informing transportation users and awareness raising.

ECO SENSITIVE TRAFFIC MANAGEMENT

Eco-sensitive traffic management (ETM) is a dynamic traffic management application which focuses more on one specific goal of traffic management: reducing negative environmental effects of the road transportation. However, there is no single terminology for these applications. Some other terminologies used are environmentally sensitive traffic management [34], environment responsive traffic control [35] environment-oriented traffic management [36] emission minimizing traffic control [37] or dynamic emission-dependent traffic control [38].

In ETM specific measures that aim to control or influence the traffic for emission reduction are activated, according to the current or projected emission levels, for a specific location and for a defined time period [39]. Thus, it aims to offer a midway solution by contributing to emission reduction and helping to comply with limits while still using the road network as efficiently as possible [35, 39].

It is important to note that other traffic management measures and vehicle technologies which primarily focus on increasing traffic efficiency often contribute to a reduction in emissions as well since they reduce the congestion and the number of stops. To illustrate, results of the ICT-Emissions Project which is focusing on CO₂ emissions of road transportation, shows that variable speed limits can reduce CO₂ emissions by 1.5% in congested and not congested areas. In addition, traffic adaptive urban traffic control can cause a decrease by 8% and adaptive cruise control (ACC) by 11% under normal traffic conditions [40]. The main difference between these technologies and ETM is that these measures are not activated due to high emission levels but based on traffic related indicators.

According to German Road and Transportation Research Association (FGSV), requirements of an ETM system are representation of the current emission levels, prognosis of

expected emission levels, evaluation of effectiveness of measures and monitoring of the impacts. Furthermore, documentation of data and results should be assured for future planning decisions[39]. According to this and other studies, the main steps of ETM can be summarized as [39, 35]:

- Assessment of the existing air quality
- Identification of the problematic areas and contributing sources
- Projection of expected emission levels
- Activation of selected measure or measures
- Assessment of effectiveness and impacts of applied measures

Examples of traffic management measures that can be applied in ETM are dynamic re-routing of traffic, restrictions for certain vehicle types (e.g. heavy duty vehicles), speed limitations and coordination of signal control, on certain routes (e.g. near hotspots) for defined time periods [39, 35]. In addition to traffic detectors, traffic network data, traffic simulation models and traffic control devices (e.g. adaptive signal control and dynamic traffic information) which are needed for any traffic management application, ETM requires meteorological detectors, air pollutant detectors, emission maps and models for meteorological prognosis and air quality assessment [39]. The reason is that air pollutant concentrations are influenced not only by the type and source of the air pollutant but also by atmospheric conditions like humidity, rainfall, temperature, wind direction, wind speed as well as locational surrounding conditions like building density, traffic condition and road surface [41, 35, 13, 42].

Figure 9 summarizes a generic approach where the main steps of ETM together with basic tools/systems and their interactions can be seen. As it can be perceived, one key component of ETM is modelling tools as they contribute to assessment of the air quality, enable projection of expected emission levels and help to conduct a pre-analysis of measures.

FGSV [39] points out that an important factor affecting the complexity of the ETM systems, required level of detail and accuracy of models is whether the measures are activated solely according to existing conditions or additionally due to projected conditions. Additional factors can be activation of one single measure or combination of measures as well as the size of the considered area (i.e. scope of the ETM).

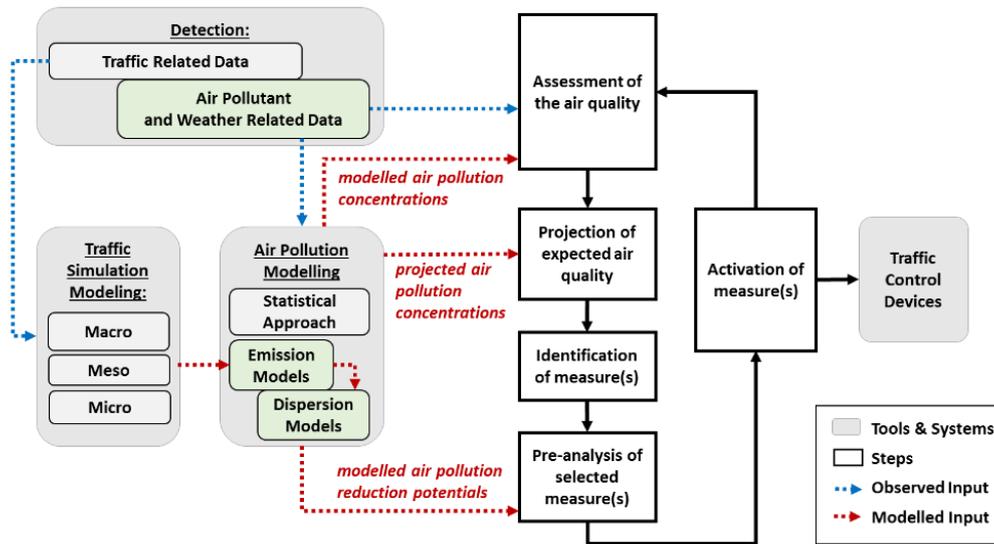


Figure 9. Generic approach to ETM, adapted from [35, 39, 43, 44]

Air Quality Assessment

As observation of traffic conditions is very important for all traffic management applications, assessment of the air quality is an additional crucial step for eco-sensitive traffic management. It helps to understand the air pollution problem, to find the influencing traffic related factors which contributes later to making projections, to define related measures and to activate or deactivate them. As it can be seen in Figure 9, air quality assessment can be done by using different methods. Emission levels can be collected directly from detectors, can be modelled by using air pollution models or can be determined by using both which is a more comprehensive and commonly used method [43].

In Europe, there are possible air pollution detection methods which are defined by European Commission [45]. Reference measurement norms for NO₂, NO_x, PM₁₀ and PM_{2.5} can be found in Table 4. The directive also points out that in addition to air pollution measurements, modelling techniques should be applied for assessment of the air quality when possible because measurement stations provide only a “point data” and this should be interpreted to a more comprehensive area to be able to understand the effects of the air pollution better [27].

For modeling of the air pollution for traffic related studies, there are two main approaches. Air pollution can be estimated by using empirical data and statistical approaches where factors influencing pollutant concentration are defined according to observed data and used for estimation of future air pollution levels [35]. While this approach can explain these factors well and offer a good prediction quality, it does not provide detailed information on spatial distribution [35]. Another way is to use air pollution models [43, 35]. Air pollution models should not only estimate the emissions from the sources

Table 4. EC Reference measurement methods [45]

Air Pollutant	List of Reference Measurement Methods
NO ₂ & NO _x	EN 14211:2005
PM ₁₀	EN 12341:1999
PM _{2.5}	EN 14907:2005

(e.g. downwind concentrations) but also describe important aspects of the dispersion process [46]. In this context, emission models are used to determine the emitted pollutants from the sources while dispersion models are used to estimate the distribution of emissions and herewith to estimate the actual air pollution concentrations (i.e. ground-level concentrations) in a specific area. Dispersion models simulate the emissions by considering atmospheric conditions, built environment and background emissions [35]. These models can also be used for understanding the sources and the influencing geophysical factors [47]. Existing ETM systems use one or a combination of these air quality assessment methods and modelling approaches [39, 35].

There are different emission calculation models used for road transportation related emissions with different levels of detail. Aggregated emission models give one emission factor per vehicle type without considering different driving behaviors; average speed models (e.g. COPERT¹, TREMOD²) additionally include different average trip speeds per vehicle type; traffic situation based emission models (e.g. HBEFA³) assign emission factors according to vehicle type and several predefined traffic conditions [43]. The most detailed emission models are instantaneous emission models which calculate emissions for every second and for each individual vehicle

¹Computer Programme to Calculate Emissions from Road Transport, EEA

²Transport Emission Model, UBA

³The Handbook Emission Factors for Road Transport, Infrac

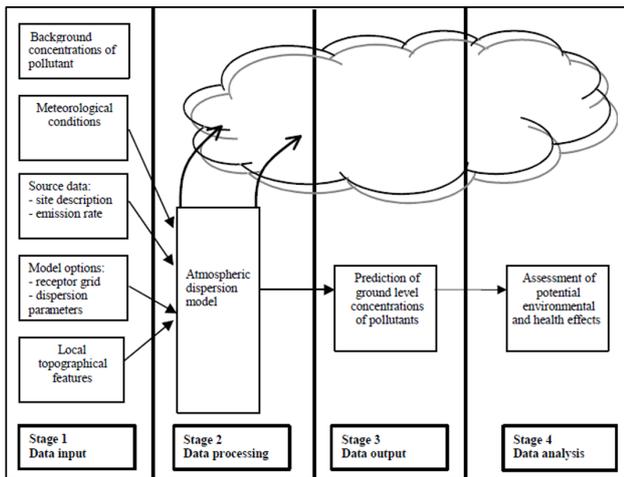


Figure 10. Overview of the air quality modelling procedure [47]

considering instantaneous engine power that is being used and the speed [43, 44]. Two commonly used examples for instantaneous models are PHEM⁴ and MOVES⁵.

To illustrate with some examples, COPERT is mostly used by many national governments to calculate road transportation emissions [23] while HBEFA mostly used by local authorities [43] as a result of the level of detail. On the other hand, emission factors in less detailed models (i.e. emission inventory models) are mostly aggregated from detailed vehicle based models (e.g. HBEFA and CORPERT databases are based on PHEM). Today, there are also modelling packages available which use emission calculation models (e.g. HBEFA) and combines with dispersion models (e.g. IMMIS).

Air pollution concentrations are influenced by road transportation related factors such as traffic composition which is described by vehicle types (e.g. trucks, cars and motorbikes), engine types (e.g. petrol, diesel and electric) and emission classes of vehicles as well as traffic conditions (e.g. average speeds, number of cold or warm stops) [35, 13]. Therefore, for an accurate emission calculation for road transportation, information on vehicles and traffic conditions (e.g. traffic volume, composition and state) are needed. However, data needed can change depending on the detail level of the emission model used. As it is for air quality assessment, traffic related data can be gathered from sensors, can be modelled or can be estimated by using both. Vehicle information and traffic conditions can be gathered from the field by detectors (e.g. induction loop detectors, video detectors, ultrasound detectors, radar detectors) or by mobile data collection sources (e.g. Floating Car Data - FCD). Depending on the scale, current and expected traffic conditions can be modelled by using macroscopic, mesoscopic or microscopic traffic simulation models. Especially for the accuracy of the prognosis, local traffic conditions at hotspots should be realistically projected [39].

⁴Passenger Car and Heavy Duty Emission Model, TU Graz

⁵Motor Vehicle Emission Simulator; EPA

Application of ETM and Effectiveness from Other Case Studies

In this section some ETM examples and studies from Germany will be presented in order to illustrate how these systems are applied, evaluated, to which degree ETM contributes to emission reductions and what lessons can be learned. For each study a short background information, steps used, data detection methods, measures and their impacts will be explained briefly.

Hagen

One of the first implementations of ETM in Germany is in Hagen [35]. The system is implemented as a measure under the air pollution action plan (2002) which is set after studies showed that there are high NO₂ and PM₁₀ emission levels at the main streets of inner city caused by the road traffic [38]. The methodology applied was analysis of existing traffic, weather and emission trends, defining influencing parameters, implementing a control algorithm, simulation of possible impacts using recorded data, optimization of the algorithm, test implementation on the field, evaluation and re-optimization of the algorithm followed by the praxis application [38].

Detection is done for a whole year to be able to eliminate seasonal effects [38]. Air pollution measurements are collected from three traffic and one urban background stations; atmospheric data (wind direction, wind speed and global radiation) was collected from three different stations. Traffic data was gathered from one inner city location using automatic traffic counters [38]. After the first analysis, "Markischer Ring" which was found to be the most critical street with frequent excess of NO₂ and PM₁₀ levels was selected as ETM application area.

For emission measurements an emission model which uses HBEFA for emission factors was used [38]. Emission models showed that in "Markischer Ring" 55% of the NO_x emissions was caused by heavy duty vehicles (HDVs), although they were only 4,5% of the total traffic volume [38]. Consequently, the implemented and assessed ETM measure was restriction of HDVs on this part of the ring road when NO_x levels are high. It was expected from the application to reduce NO_x levels and partly also to help reducing PM₁₀ emissions [38].

During the pre-analysis, it is also seen that NO₂ emission concentrations were not only affected by emissions from road traffic but also due to wind and radiation intensity on the measurement day [38]. For emission dependent traffic control, a prognosis algorithm was developed and different activation criteria were defined which take wind, radiation and emission levels in consideration [38]. A model based potential analysis with different HDV reduction rates (i.e. acceptance rates) showed that ETM does not help much for reducing the yearly average values but helps to reduce the number of limit exceeding for NO₂ and PM₁₀ (Figure 11). It is also seen that for NO₂ limit excess, dynamic ETM was as effective as static restrictions and all measures were highly dependent on acceptance of the measures by HDV drivers [38].

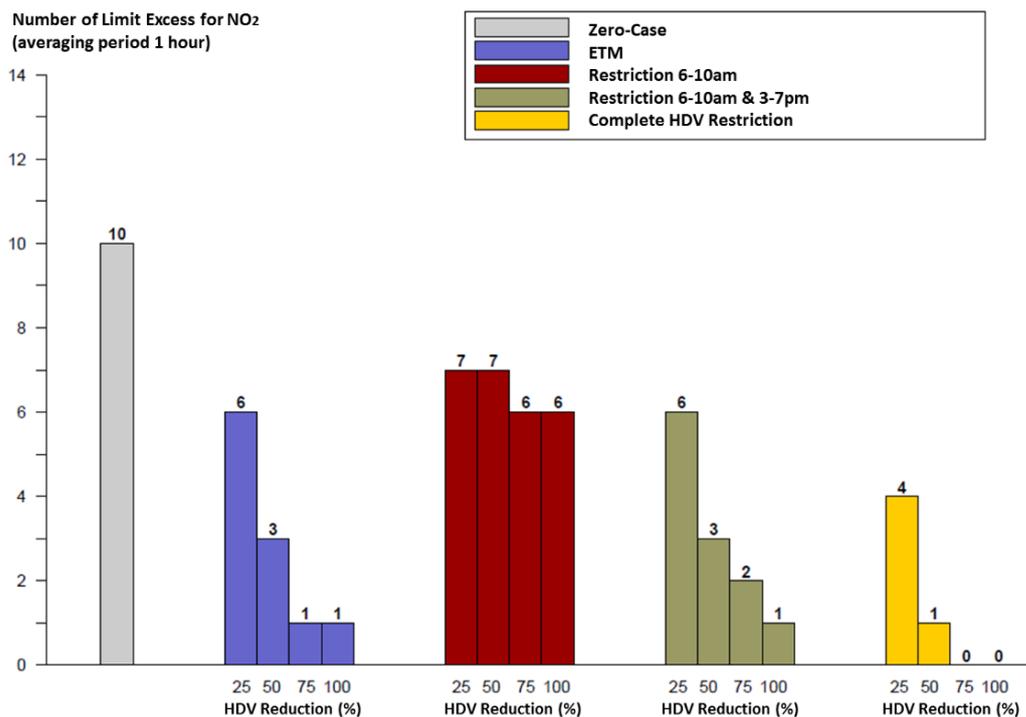


Figure 11. Effects of different measures on limit excess frequency, translated from [38]

Later, the ETM system was tested (2007) on the field for two days but the results did not give a significant comparison of air pollution concentrations since acceptance of the drivers (i.e. number of drivers that comply with the restriction) was low and it was not possible to directly compare to a zero-case since the zero scenario values were from another day with different background emissions and weather conditions [38]. However new expected emissions were modelled by using measured volumes from the test days [38]. The results showed that a static measure (closing the road from 6:00 to 10:00) and dynamic ETM (activated between 6:00am to 20:00), with good compliance rate, can both reduce NO_x emissions around 20% and PM_{10} emissions around 16% [38]. In order to check the impacts (i.e. relocation of emissions on other routes) "IMMIS^{luft}" was used in the assessment phase.

The study shows that for defined time periods dynamic control strategies can be as effective as static measures against emissions, depending on the compliance rates. It also points out that long term restrictions can increase emissions on alternative routes and dynamic measures can reduce this relocation effect. Although ETM system was found to be useful to reduce emission concentrations and excess frequencies, they were not very effective on reducing yearly average emission levels alone. To achieve a long-term reduction, more comprehensive larger scale measures (e.g. emission zones) are suggested. In addition, the study recommends to give proper and early information to users (e.g. information already on highway) in order to increase the compliance with measures.

Braunschweig

The air quality and action plan of Braunschweig was prepared in 2007 as a result of detected NO_2 and PM_{10} limit excesses [48]. The plan covers different road traffic related measures and addresses the importance of traffic management measures. In connection with the plan, the project ETM-Braunschweig was developed where the goals were analysis of flexible, short-term traffic management measures for reduction of air pollutants as well as development and testing of an ETM system [49]. Main steps of the study were setting of measures, system development and testing followed by test operation and evaluation [49].

Firstly, possible test areas were selected and data was being collected [49]. Air quality data (PM_{10} , NO_x , NO_2) were collected with a measurement container, traffic data were gathered from passive infrared detectors, test rides and videos. Meteorological data (wind direction, wind speed, dispersion class) were obtained from airport stations [49]. Pre-analysis showed that there was no PM_{10} excess during the analysis period but NO_2 limits were exceeded and "Altewiekring" (Figure 12) was one hotspot [49]. It is also noted that absence of an urban background station may lead to some underestimations of emissions. NO_x and NO_2 values showed a high correlation with traffic volumes, while no correlation for PM_{10} levels could be detected since measurement station provided only average daily values instead of hourly values [49]. Later, possible traffic management measures were defined and tested with a macroscopic traffic simulation model (PTV VISUM) to understand their emission reduction potentials [49].

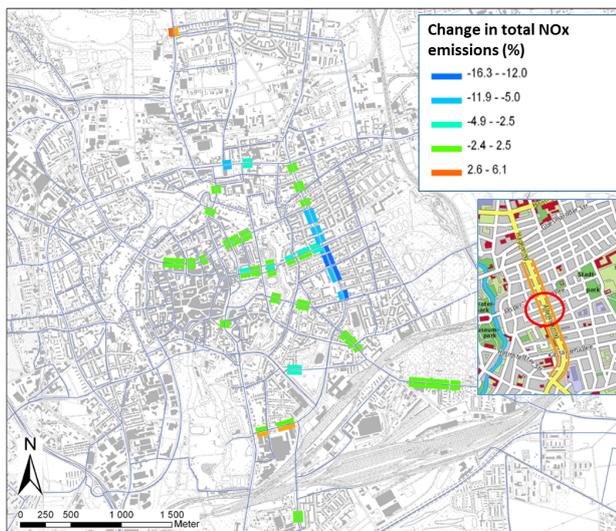


Figure 12. Change in total emission levels in Altewiekring comparing to one year before testing of the ETM [49]

Results indicated that highest reduction was achieved by restriction of HDVs and the second highest was obtained from ramp metering where traffic volumes entering to the area was reduced by 70% [49]. However, it is stated that there was a risk of increasing emissions on other parts of the network with the first option due to redirecting of HDVs. The final measure to test in the field was reduction of entering traffic volumes (by 20% and by 40%) and tests took place in 2009 (p.20-26). For air quality modelling, IMMIS^{mt} was used and validated with real data from the detectors [49]. Test results showed that there was a reduction in road transportation related NO_x and PM₁₀ emissions, respectively 12% and 13% which was reflected to overall emission reduction as 10% for NO_x and 4% for PM₁₀ [49]. It is seen in the whole study area that emission levels were mostly improved. No serious (more than 2%) relocation of emissions was detected for PM₁₀, while for NO_x two other points in network showed increased levels [49]

Afterwards, a second phase of the study was initiated in order to implement the results from the first phase on other hotspots, to have a more comprehensive picture and to be able to evaluate the interactions [49]. For the second phase, monitoring was improved, a prognosis function was introduced, activation with limit excess and improvement of information systems were aimed [49]. Firstly, air pollution monitoring has been done with IMMIS systems, validated and updated [50]. For traffic related data 130 infrared and 90 passive infrared detectors as well as video records and a traffic computer system (SCALA) were used [50]. For traffic modeling, macroscopic traffic simulation from first phase (PTV VISUM) was used and calibrated with the detector data [50]. As activation criteria, NO₂ concentration levels are selected and a tool for activation mechanism was developed [50].

Annual mean NO₂ emission levels are modelled and five hotspots, where values were higher than the limits, were de-

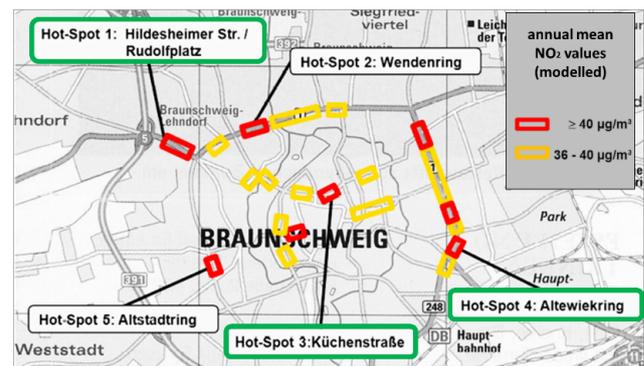


Figure 13. Location of the detected hotspots and their modelled annual mean NO₂ values

tected (Figure 13). For these hotspots detailed demand and network analysis has been done and possible traffic management measures were defined [50]. Finally, effectiveness and impacts of the measures were tested by using models for three of these areas: hotspot 1, 3 and 4 (Hildesheimerstraße, Küchenstraße, and Altewiekring). The results can be found in Table 5.

Assessment of measures which is done by using models indicated that all measures contributed to the desired reduction in hotspots. However, they all, more or less, lead to an increase in NO_x concentrations on alternative routes [50]. For example, in Hotspot 1 first measure that causes higher reduction in traffic volume was almost two times more effective on reduction of total NO_x concentrations than the second measure but it also caused higher NO_x increase in alternative routes. While this additional increase was within limits for some areas (in case of hotspot 1 and 4), for some areas relocation of emissions risked a limit access in other already critical spots (in case of hotspot 3). In last example, higher reductions were seen in other streets than the reductions on target streets [50].

Potsdam

Potsdam introduced an air quality and action plan in 2007 as a result of detected PM₁₀ and NO₂ limit excesses and proposed many strategies to overcome this problem [36]. One of the measures was to develop ETM strategies (control and information) and integrate an ETM system to the existing traffic management system [36]. The ETM is operational in Potsdam since 2012 for the reduction of NO_x and PM₁₀ limit excesses. System covers several hotspots and several measures are activated when traffic volumes and/or air pollution concentration levels are above the limits [39]. Measures cover different actions like improvement of the traffic flow by green waves, short-term traffic volume control at traffic signals (i.e. ramp metering) on the border of hotspots as well as informing users about traffic conditions, emission levels and related changes in the network [39]. In this example, some defined measures were adjusted after a pre-analysis as well. For example, it is seen that for one hotspot traffic signal control would not be

Table 5. Analyzed measures and reduction effects [50]

Hotspot	Measure	Modelled reduction in traffic volume (PC&HDV)	Reduction in total NOx concentration with permanent activation	Reduction in total NOx reduction with temporary activation
Hotspot 1	Reduction of traffic volumes into the city center from the highway (A391) by 30%	27%	20-25%	7-15%
	Reduction of traffic volumes into the city center from the main road (B1) by 30%	14%	10%	3-6%
Hotspot 2	Closing a street section before the hotspot	17%	8-9%	3-5%
	Left turn restriction from one street to the hotspot	16%	7-8%	3-5%
Hotspot 3	Reduction of green times on roads leading to hotspot and re-routing of traffic (in total four different measures on different spots)	9%	5-7%	3-5%

effective enough and additional speed reduction is introduced. Figure 14 represents the ETM System in Potsdam: pink areas show the hotspots, blue traffic lights represent the ramp metering points and green traffic lights indicate the sections with traffic signal control coordination.



Figure 14. ETM Potsdam [51]

First results show that there was a reduction of traffic volumes and queue lengths which resulted in a reduction of air pollutant concentrations and number of limit excesses [52]. To illustrate, at one traffic station in one of the hotspots, a yearly average NO₂ emission level that did not exceed the limit was observed for the first time [52]. The results from the streets where speed reduction measures were applied showed not only a decrease of air pollutant levels but also reduced traffic noise [53].

Conclusion and Discussion

Road transportation related air pollutant emissions, which is a current critical problem for many developing countries, solved to a degree in Europe in last decades mostly due to the new strict standards and regulations. However, some air pollutant limits, especially NO_x and PM₁₀, are often exceeded in urban hotspots near road transportation. Greenhouse gases show a slightly decreasing trend in last decade but still considerably high when compared to the levels in 1990. In order to be able to avoid climate change and its effects, reductions of GHG emissions from the road transportation should continue.

Eco-sensitive traffic management offers an opportunity to reduce road transportation related emissions in urban areas together with other policy instruments. In contrast to static traffic management, ETM offers some advantages by not limiting the accessibility completely and not causing a long term relocation of emissions on alternative routes. Hagen example shows that dynamic traffic management can improve emissions levels as good as static measures in some cases. However, the results also indicates that ETM systems help reducing short-term concentrations and number of limit excess but as a result of being not static, it is not that effective in reducing average air pollutant concentrations. In addition, although it does not cause a long term relocation effect, ETM measures can cause short term emission concentration levels to increase in other areas. Consequently, it is important to consider ETM as a part of the overall air quality planning and management.

Assessment of current emissions and projection of future emissions are two key steps of ETM which are mostly conducted by using models as the data gathered from real detectors are often spatially or temporally limited and provide

a point information. While models help to solve this problem, they have limitations and/or uncertainties due to aggregation levels and inaccuracies of the input data. Studies point out that emission concentrations are highly influenced by several factors like wind conditions, temperatures, radiation and background emissions that the same traffic conditions may result in different emission levels on different days due to different conditions. This shows the importance of the input data, modelling techniques; especially for the ETM systems that are activated according to expected emission levels. It implies that expected emission levels cannot be properly estimated solely by expected traffic conditions. Studies supports the idea that reduction of overall traffic volumes and restriction of certain vehicle types (e.g.heavy duty vehicles) are more effective than other measures. This emphasizes the importance of travel demand and the vehicle compositions (e.g.vehicle types and emission classes) on emission levels.

A final conclusion can be that ETM is a meaningful tool to reduce air pollutant emission concentrations that can be improved further by including other emission types, using denser/more dynamic input data and more accurate prediction tools, assuring higher compliance rates and by integrating with higher level measures to ensure a long-term effectiveness. Current developments in technology and mobility market is offering several opportunities for ETM. With recent developments in mobile detection devices, cellphones and wearables it can become possible to gather more detailed traffic and air quality data. Although it is not yet commonly used or proven, these technologies may offer a potential to be integrated in ETM and to fill the information gap in detection and monitoring phases and contribute to a better air quality assessment and projection. Emerging developments in e-mobility can substantially help to reduce emissions of road transportation in urban areas and to improve effectiveness of eco-sensitive traffic management. By integrating e-vehicles to ETM systems ambitious international, national and local goals concerning e-mobility can be also supported. In addition, it is important to consider greenhouse gases and noise emission levels for future ETM systems to speed up the reduction of greenhouse gases and their negative effects as well as to reduce noise pollution in urban areas.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] United Nations. *World Urbanization Prospects: The 2014 Revision : Highlights*. New York.
- [2] Friedrich Maier, Robert Braun, Fritz Busch, and Paul Mathias. Pattern-based short-term prediction of urban congestion propagation and automatic response: Predicting urban congestion patterns. *Traffic Engineering and Control (TEC) magazine*, June:227–232, 2008.
- [3] FGSV: Forschungsgesellschaft fuer Strassen-und Verkehrswesen. *Hinweise zur Strategieanwendung im dynamischen Verkehrsmanagement*, volume 381,1 : W1. FGSV, Koeln, 2011 edition, 2011.
- [4] Manfred Boltze and Vu Anh Tuan. Approaches to Achieve Sustainability in Traffic Management. *Procedia Engineering*, 142:205–212, 2016.
- [5] FGSV: Forschungsgesellschaft fuer Strassen-und Verkehrswesen. *Hinweise zur Strategieentwicklung im dynamischen Verkehrsmanagement*, volume 381. FGSV, Koeln, 2003 edition, 2003.
- [6] Leif Fornauf. *Entwicklung einer Methodik zur Bewertung von Strategien für das dynamische Straßenverkehrsmanagement*. PhD thesis, Technische Universität, Darmstadt, October 2015.
- [7] European Telecommunications Standards Institute.
- [8] ITS EduNet. Definition of ITS. <http://www.its-edunet.org/>.
- [9] M. Maibach, C. Schreyer, D. Sutter, H.P van Essen, B.H Boon, R. Smokers, A. Schroten, C. Doll, B. Pawlowska, and M. Bak. Handbook on estimation of external costs in the transport sector: Internalisation Measures and Policies for All external Cost of Transport (IMPACT): Publication number: 07.4288.52, 2008. http://ec.europa.eu/transport/themes/sustainable/internalisation_en.htm; Accessed: 22/06/2016.
- [10] Udo Becker. *Grundwissen Verkehrsökologie: Grundlagen, Handlungsfelder und Maßnahmen für die Verkehrswende*. Oekom Verlag, München, 2015.
- [11] Erik Verhoef. External Effects and Social Costs of Road Transport. *Transportation Research Part A: Policy and Practice*, 28(4):273–287, 1994.
- [12] Cristina Guerreiro, Frank de Leeuw, Valentin Foltescu, Alberto González Ortiz, and Jan Horálek. *Air quality in Europe: 2015 report*. Publications Office, Luxembourg, 2015.
- [13] Dietrich Schwela. *Air Quality Management: Revised October 2009*. Sustainable Transport: A sourcebook for policy-makers in developing cities. Deutsche Gesellschaft für Technische Zusammenarbeit (GTZ), Eschborn, 2009.
- [14] Carlos Dora, Hosking Jamie, Mudu Pierpaolo, and Fletcher Elaine. *Urban Transport and Health: Revised October 2009*. Sustainable transport: a sourcebook for policy-makers in developing cities. Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ), Eschborn, 2011.

- [15] IPCC (Intergovernmental Panel on Climate Change). *Climate change 2014: Synthesis report: Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, R.K. Pachauri and L.A. Meyer (eds.)]*. Geneva, Switzerland, 2014.
- [16] Ken Gwilliam, Masami Kojima, and Todd Johnsonn. *Reducing Air Pollution from Urban Transport: Companion, 2005*.
- [17] WHO (World Health Organization). *Air Quality Guidelines: Global Update 2005 : particulate matter, ozone, nitrogen dioxide, and sulfur dioxide*. Copenhagen and Denmark, 2006.
- [18] IPCC (Intergovernmental Panel on Climate Change). *Climate change 2014: Mitigation of Climate Change : Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change: [Edenhofer, O., R. Pichs-Madruga, Y. Sokona, E. Farahani, S. Kadner, K. Seyboth, A. Adler, I. Baum, S. Brunner, P. Eickemeier, B. Kriemann, J. Savolainen, S. Schlömer, C. von Stechow, T. Zwickel and J.C. Minx (eds.)]*. Cambridge University Press, Cambridge and United Kingdom and New York and NY and USA, 2014.
- [19] IPCC (Intergovernmental Panel on Climate Change). *Summary for Policy Makers in Climate Change 2013: The Physical Science Basis: Contribution of Working Groups I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (eds.)]*. Cambridge University Press, Cambridge and United Kingdom and New York and NY and USA, 2013.
- [20] EEA (European Environment Agency). *Explaining road transport emissions: A non technical guide, 2016*.
- [21] P. Pallavi and Roy M. Harrison. Estimation of the contribution of road traffic emissions to particulate matter concentrations from field measurements: A review. *Atmospheric Environment*, 77:78–97, 2013.
- [22] Ken Gwilliam, Masami Kojima, and Todd Johnsonn. *Reducing Air Pollution from Urban Transport, 2004*.
- [23] EEA (European Environment Agency). *European Union emission inventory report 1990-2013 under the UNECE Convention on Long-range Transboundary Air Pollution (LRTAP)*, volume 08/2015 of *EEA Technical report*. Publications Office, Luxembourg, 2015.
- [24] EC (European Commission). *EU transport in figures. Statistical pocketbook 2014*. Publications Office of the European Union, Luxembourg, 2014.
- [25] EEA (European Environment Agency). *Contribution of the transport sector to total emissions of the main air pollutants, 2015*.
- [26] EUROBASE. Air Pollution Database 1990-2013, 2016 June. <http://ec.europa.eu/eurostat/web/environmental-data-centre-on-natural-resources-old/data/database>.
- [27] 2008/50/EC. Directive 2008/50/EC of the European Parliament and of the Council of 21 May 2008 on ambient air quality and cleaner air for Europe. http://ec.europa.eu/environment/air/quality/legislation/existing_leg.htm; Accessed: 16/06/2016.
- [28] 443/2009. Regulation (EC) No 443/2009 of the European Parliament and of the Council of 23 April 2009 setting emission performance standards for new passenger cars as part of the Community's integrated approach to reduce CO2 emissions from light-duty vehicles.
- [29] Regierung von Oberbayern (District Government of Upper Bavaria). *Luftreinhalteplan Für die Stadt München (Air Quality Plan Munich)*, September 2004.
- [30] A. Minkos, U. Dauert, G. Schütze, S. Feigenspan, T. Himpel, and S. Kessinger. *Luftqualität 2015: Vorläufige Auswertung (Air Quality 2015: Preliminary Evaluation): Hintergrund // Januar 2016*. Umweltbundesamt (German Environment Agency), Dessau-Roßlau, 2016. <https://www.umweltbundesamt.de/publikationen/luftqualitaet-2015>; Accessed: 19/06/2016.
- [31] EEA (European Environment Agency). *Exceedances of air quality objectives due to traffic measured in traffic stations and background stations, 2016*.
- [32] S. Rodt, B. Georgi, B. Huckestein, L. Mönch, R. Herbener, H. Jahn, K. Koppe, and Lindmaier J. *CO2-Emissionsminderung im Verkehr in Deutschland (Reduction of CO2 emissions from Transport in Germany): Mögliche Maßnahmen und ihre Minderungspotenziale (Possible measures and their reduction potentials)*. Umweltbundesamt (German Environment Agency), Dessau-Roßlau, 2010.
- [33] OECD: Organization for Economic Co-operation and Development. *Policy Instruments for Achieving Environmentally Sustainable Transport*. OECD Publishing, Paris, 2002.
- [34] LRP Berlin. *Air Quality Plan for Berlin 2011-2017*. Senatsverwaltung für Stadtentwicklung und Umwelt (Senate Department for Urban Development and the Environment), December 2014. <http://www.stadtentwicklung.berlin.de/umwelt/luftqualitaet/de/luftreinhalteplan/download.shtml>; Accessed: 28/06/2016.
- [35] Manfred Boltze and Sven Kohoutek. *Environment-Responsive Traffic Control, 2010*.
- [36] LRP Potsdam. *Luftreinhalte- und Qualitätsplan für die Landeshauptstadt Potsdam: Fortschreibung 2010-2015 (Air pollution Control and Air Quality Plan for the*

- State Capital Potsdam): [VMZ, IVU, LK ARGUS, Landeshauptstadt Potsdam]. Ministerium für Ländliche Entwicklung, Umwelt und Landwirtschaft des Landes Brandenburg (Ministry of Rural Development, Environment and Agriculture of the Federal State of Brandenburg), Juni 2012.
- [37] Karin Hirschmann and Fellendorf Martin. Emission minimizing traffic control – simulation and measurements, 2009. mobil.TUM 2009 12&13 May.
- [38] G. Ludes, B. Siebers, T. Kuhlbusch, U. Quass, M. Beyer, and F. Weber. *Feinstaub und NO₂: Entwicklung und Validierung einer Methode zur immissionsabhängigen dynamischen Verkehrssteuerung (Fine particles and NO₂: Development and validation of a method for the immission-relevant, dynamic re-routing of traffic)*. Umweltbundesamt (German Environment Agency), Dessau-Roßlau, 2010.
- [39] FGSV: Forschungsgesellschaft fuer Strassen-und Verkehrswesen. Wirkung von Maßnahmen zur Umweltbelastung Teil 3: Umweltsensitives Verkehrsmanagement (UVM), Zwischenstand (Effects of Measures to reduce environmental impacts Part 3: Environment Sensitive Traffic Management (ETM), Preliminary results, 2014.
- [40] ICT-Emissions. ICT-Emissions Project Handbook: The Wise Way to Cut Down on CO₂: THE REAL-LIFE IMPACT OF THE INTELLIGENT TRAFFIC AND IN-VEHICLE SYSTEMS ON CO₂ EMISSIONS AND. <http://www.ict-emissions.eu/deliverables-results/results/>; Accessed 21/06/2016.
- [41] F. Hülsmann, R. Gerike, B. Kickhöfer, K. Nagel, and R. Luz. Towards a multi-agent based modeling approach for air pollutants in urban regions, 2011. In Proceedings of the Conference on Luftqualität an Straßen; pp: 144–166.
- [42] Alex de Visscher. *Air Dispersion Modeling: Foundations and Applications*. John Wiley & Sons, Inc., New Jersey, 2013.
- [43] Friederike Hülsmann. *Integrated agent-based transport simulation and air pollution modelling in urban areas - the example of Munich*. Dissertation, Technische Universität München, München, 2014.
- [44] Robin North and Simon Hu. CARBOTRAF - Report on Emission Models: A Decision Support System for Reducing CO₂ and Black Carbon Emissions by Adaptive Traffic Management: D4.1 Emission Models, 2012. <http://www.carbotraf.eu/deliverables>; Accessed: 20/06/2016.
- [45] EC (European Commission). Implementation of Ambient Air Quality Legislation: Assessment: Methods, June 2016. <http://ec.europa.eu/environment/air/quality/legislation/assessment.htm>; Accessed: 20/06/2016.
- [46] Robert Macdonald. Theory and Objectives of Air Dispersion Modelling. *Modelling Air Emissions for Compliance MME 474A Wind Engineering*, 2003.
- [47] *Good practice guide for atmospheric dispersion modelling*. Ministry for the Environment, New Zealand, Wellington and N.Z, 2004.
- [48] LRP Braunschweig. Luftreinhalte- und Aktionsplan Braunschweig (Air Pollution Control and Action Plan Braunschweig). Stadt Braunschweig (City of Braunschweig), 2007. https://www.braunschweig.de/leben/umwelt_naturschutz/luft/luftreinhalteplanung.html; LastUpdate: 2015; Accessed: 20/06/2016.
- [49] BLIC. Umwelterorientiertes Verkehrsmanagement Braunschweig (Environment Oriented Traffic Management Braunschweig): Gemeinsamer Ergebnisbericht (Joint Summary Report), 2010.
- [50] UVM-BS Projektkonsortium. Umwelterorientiertes Verkehrsmanagement Braunschweig Stufe 2 (Environment Oriented Traffic Management Braunschweig Stage 2): Gemeinsamer Ergebnisbericht (Joint Summary Report), 2012.
- [51] Landeshauptstadt Potsdam (State Capital Potsdam). Umwelterorientierte verkehrssteuerung: Gesünder, sauberer und mobiler für potsdam. http://www.mobil-potsdam.de/fileadmin/user_upload/UVS/downloadfassung_allgemeiner_flyer_20120314_2.pdf. Accessed: 23/06/2016.
- [52] Landeshauptstadt Potsdam (State Capital Potsdam). Zwischenbilanz Umwelterorientierte Verkehrssteuerung: Pressemitteilung Nr. 173: Schadstoffbelastung gesunken / Stickstoffdioxid auch in der Großbeerenstraße erstmals unter zulässigem Grenzwert, 13.03.2014. <https://www.potsdam.de/173-zwischenbilanz-umwelterorientierte-verkehrssteuerung>.
- [53] Landeshauptstadt Potsdam (State Capital Potsdam). Potsdamer Luftqualität verbessert: Information 042/2013: Erste Ergebnisse der Evaluierung der Umwelterorientierten Verkehrssteuerung, 23.01.2013. <https://www.potsdam.de/content/042-potsdamer-luftqualitaet-verbessert>.

Traffic Management for Major Events

Sasan Amini, Eftychios Papapanagiotou and Fritz Busch

Chair of Traffic Engineering and control, Technical University of Munich
{sasan.amini; eftychios.papapanagiotou; fritz.busch}@tum.de

Abstract

Transportation system is a critical infrastructure for the movement of people and goods. However, major events such as unexpected incidents and planned special events put its reliability at high risk. Recently, the number of studies on the resilience of transportation infrastructure has been growing, but the operational strategies are still adopted from manuals and checklists. Despite the advances in computation and traffic simulation models, which have made real-time short-term traffic prediction possible, application of such tools for large-scale urban road network remains a challenge. This report gives an overview of the state-of-the-art and state-of-the-practice for traffic management strategy implementation in case of major events. Additionally, it discusses the new opportunities created by the recent findings in field of urban traffic modelling and the availability of connected-vehicles. The key conclusion is that large-scale traffic simulation is beneficial to evaluate management strategies in real-time and to expedite the recovery of the transportation network, e.g. through dynamic route planning for road users and emergency response, but on-line calibration is necessary to have a representative model.

Keywords

Major events; Traffic management; Transportation resilience; Intelligent transportation systems

Major Events and their Impact on the Transportation System

Transportation infrastructure is designed for movement of goods and people under predictable travel demand conditions. Although during the planning phase peak conditions and some expected variations are taken into account, “*it is not financially, environmentally and physically practical to construct systems that take into account every possible incident and event*” [1]. Admittedly, it has been accepted that congestion happens when unpredictable disruptions occur such as accidents, road closure, extreme demand to reach an event venue, etc. Since unexpected incidents are the dominant source of travel time unreliability [2], it is crucial to predict the performance of the transportation network during unusual conditions and plan a set of actions to enhance the mobility and safety of travelers. Before investigating the possible management strategies, it is necessary to identify a major non-recurring condition. To do so, a brief overview on characteristics of congestion is given in the next section.

Recurring and Non-Recurring Congestion

Most of the transport professionals [3, 4, 5, 6] divide the components of congestion into recurring and non-recurring; however, there is no unique definition of each type. Recurring congestion refers mostly to a congestion that is relatively predictable and happens periodically at certain hours due to presence of large number of vehicles on the road network. On the other hand, non-recurring congestion is defined as unusual congestion caused by unpredictable incidents such as accidents, vehicle breakdown, adverse weather condition or

planned special events, work zones, etc. In this report, we call the overall traffic situation in case of former condition routine, and the latter case non-routine. Detecting unusual conditions is a complex task, especially in urban environments, but significantly contributes to a better use of the existing infrastructure and the implementation of management measures. Generally, data sources for traffic incident detection can be categorized in three different groups [7]:

- **Traffic surveillance:** loop detectors, CCTV camera, etc.
- **Non-transportation related reports:** e.g. from the police, fire protection departments, etc.
- **Crowd sourcing:** social networks, volunteer reports, etc.

For urban road networks, travel time (and indirectly delay) is the most commonly used indicator to decide whether the congestion is recurring [4]. In most studies, non-routine situations are not directly defined. Instead, a routine situation based on the expected travel time using the historical data is defined and the excessive delay is labeled as non-recurring. For example, in one of the first attempts, Dowling et al. [3] used Annual Average Daily Traffic (AADT) together with the corresponding Peak-Hour Factor from Highway Capacity Manual (HCM 2000) [8] to define routine travel times. In a more recent study, Anbaruglo et al. [9] measured the expected travel time in an urban road network using Automated Number Plate Recognition (ANPR) cameras and considered a static congestion factor of 1.2 as a threshold for non-recurring congestion. This implies that if the observed travel time on a link is higher than 20% of the expected travel time, the excessive congestion is categorized as non-recurring. Hojati

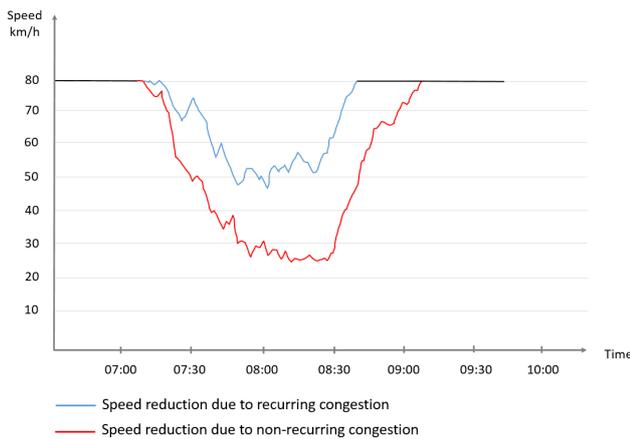


Figure 1. Differences of recurring and non-recurring congestion impact on a highway segment

et al. [2] extracted the recurrent speed profiles from loop detectors to investigate unusual changes during the reported events. Despite the promising findings, there are many limitations on the proposed methodologies: first, the only source of information is the volume and speed measurements from loop detectors and the reported events by Traffic Management Center (TMC). Second, these methods are reactive in their nature and cannot be applied for real-time and proactive applications. Moreover, the static threshold may lead to a false detection of non-recurring congestion especially in crowded parts of the network. Figure 1 depicts the idea of detecting non-routine situation by comparing the differences in speed on a motorway cross-section (reduced capacity due to work zone).

Obviously, the impact of major events on the performance of the transportation system is not limited to road network; public transport is also faced with extreme delays and overcrowded fleet. Pereira et al. [10] proposed a similar approach to detect unexpected demand for public transport. In their study, the 90th percentile of the regular demand is considered as the threshold to mark the situation as non-routine; Figure 2 illustrates their approach and shows how the number of passengers evolve through time and the time windows in which the demand exceeds the 90th percentile is marked with red color. The novelty of their research is to use machine learning algorithms to explore the Internet in order to find an explanation for the excessive demand, e.g. if there is a planned special event which people are attending.

Predicting the space-time development of an urban activity is another new method proposed by Scholz and Lu [11] in which large-scale trajectory data are analyzed in order to describe the evolution of activity "hotspot". The real-time activity patterns are linked with the historical data to predict how the patterns may evolve in the near future, which is indeed valuable for transportation management.

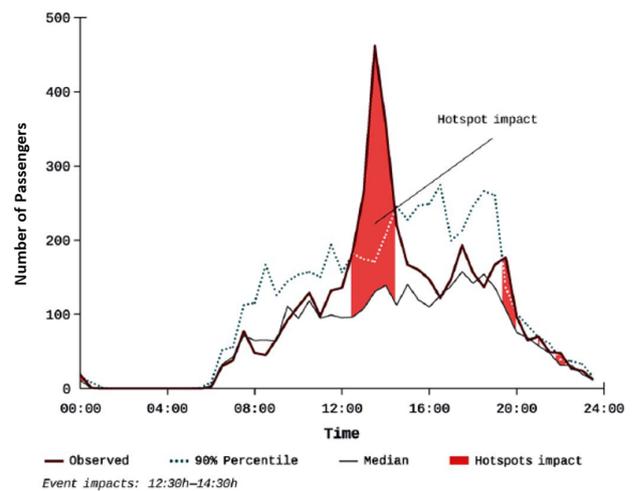


Figure 2. Non-routine demand for public transport [10]

Definition of major Events

Although events are different from each other in many ways, but they all have one feature in common: imposing a non-routine stress on the network in terms of safety, capacity reduction or demand surge [1]. Major events are discussed in many studies, but are rarely defined. They are distinguished from routine congestion by their spatio-temporal size. But the question is when the event is considered to be major? Mueller [12] has proposed a methodology to define major-, mega- and giga-events using 4 indicators: number of visitors, media coverage, costs and urban transformation. Handbook for Event Transportation (Handbuch Eventverkehr) [13] follows a similar path and categorizes events according to an extensive list of factors and elements including but not limited to the number of expected visitors, relative size, open or closed access, location, weather dependent event or not, duration and financing.

Recently, the proceedings of STADIUM (Smart Transport Applications Designed for large events with Impacts on Urban Mobility) research project [7], a European project as part of FP7 program, was published which provides two ways of defining an event: first, selecting from a predefined list, and second, by giving a number of required characteristics. Figure 3 illustrates the approach for event definition. However, the study focuses only on planned special events and does not include unplanned emergency events. Such events are usually investigated under evacuation situations, which occur during natural or man-made disasters (e.g. flood, wildfire, nuclear power plant failure, etc.) and happen with no- or short-notice.

A categorization suitable for the purpose of dynamic traffic management is introduced in Traffic Engineering Handbook [1] as depicted in Figure 4. Such categorizations are normally done by considering characteristics of events such as spatio-temporal scale, probability of occurrence, cause of the event (man-made or natural) and possibility to predict in advance. Table 1 brings some examples for each type of event.

Table 1. Different events and their characteristics defined by [1]

Event	Event Type	Advance Notice	Duration	Hazard	Impact Area	Frequency
Vehicle Crash	Unplanned/Sometimes emergency	None	Minutes to hours	Low	Local to several miles	Frequent
Concert/Sport event	Planned	Months/Years	1+Days	None	Several miles	Seasonal frequent
Olympics/One-time event	Planned	Years	1+Days to weeks	None	Several miles	Infrequent
Parades	Planned	Months/Years	Hours	Low	Few miles	Occasional
Snow/Ice storm	Unplanned	Hours to days	hours to days	medium	Regional	Seasonal
Flooding	Unplanned/Emergency	hours to days	Hours to months	Varies	Local to regional	Seasonal
Hurricane	Unplanned/Emergency	Hours to days	Days	High	Regional	Seasonal
Wildfire	Unplanned/Emergency	Minutes to days	Hours to weeks	Medium to high	Regional	Seasonal
Bridge collapse	Unplanned/Emergency	None	Months	High	Several miles	Infrequent

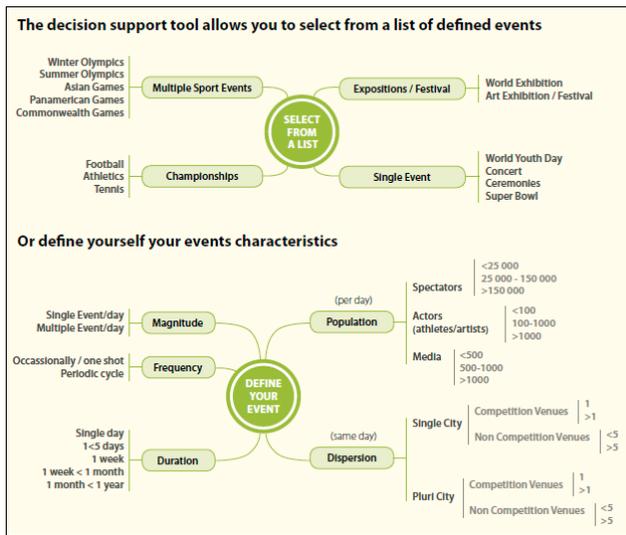


Figure 3. Defining an event according to STADIUM handbook [14]

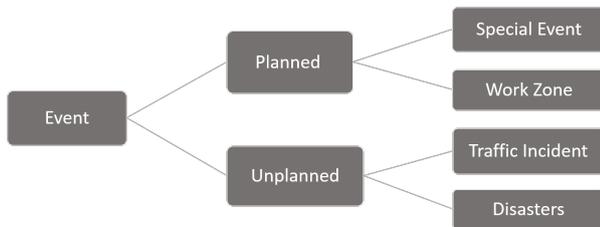


Figure 4. Event categories according to [1]

One of the key concepts to understand and measure the performance of transportation system infrastructure during disturbances is resilience. It has gained much interest only within the last few years. Recently, new policies, methods and technologies have been promoted to enhance the resiliency of transportation networks [1]. Thus, it is necessary to grasp a better understanding of transportation resilience concept and its terminology.

Transportation Resilience

The ability to provide and maintain a certain level of service in the face of disruptions to the normal operation is defined as resilience [15]. It relies on the network structure and the strate-

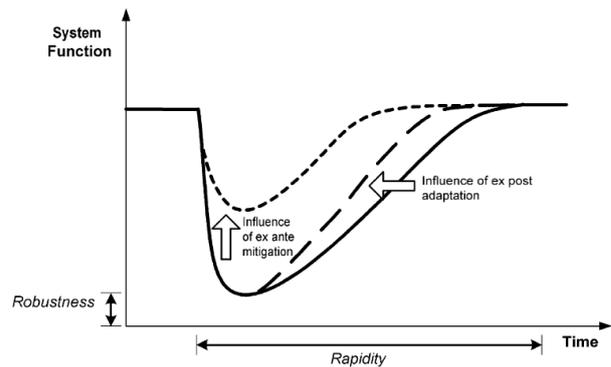


Figure 5. Effect of decision-making on resilience [19]

gies to preserve and restore the serviceability in case of an incident. Resilience measures are therefore also related to interventions that assist the system to return to pre-incident levels. These interventions may be pre-incident or post-incident. Although there is no single measure of resilience, Murray-Tuite [16] has introduced ten dimensions for transportation resilience which are defined in Table 2.

Robustness (or strength) is one of the most vital properties of resilience in major events. It is defined as “the ability of a network to cope with variations in demand or network capacity without much influence on travel times” [17]. Offering users highly reliable travel time prediction requires a robust network, especially in over-saturated situations [18]. Figure 5 depicts the concept of “resilience triangle” in which robustness and the effect of management measures on resilience both before the event and after its occurrence is depicted [19]:

Figure 5 implies that pre-event measures increase the robustness of the network which leads to more reliable travel time predictions during major disruptions while post-event measures expedite the recovery. Hence, the ability to forecast the impact of an incident immediately after its occurrence is crucial to advanced traffic management and significantly improves the system’s performance [20].

Hoogendoorn et al. [15] have further developed the concept of Macroscopic Fundamental Diagram (MFD) and proposed a generalized MFD (g-MFD) to analyze traffic dynamics in a network for both recurring and non-recurring congestion. MFD was first introduced in 2008 in [21] and explains the relationship between the average density (accumulation) and the

Table 2. Transportation Resilience dimensions [16]

	Dimension	Definition
Transportation Resilience	Redundancy	indicates that multiple, components serve the same function
	Diversity	the components are functionally different
	Strength/Robustness	indicates the system's, ability to withstand an event
	Efficiency	indicates input-output ratio optimization
	Autonomous Components	the ability to operate independently
	Safety	the system does not harm its users
	Mobility	indicates that travelers, are able to reach their chosen destinations at an acceptable level of service
	Recovery	acceptable level of service can be restored rapidly and with minimal outside assistance after an event occurs
	Adaptability	implies that the system is flexible and elements are capable of learning from past experience
	Collaboration	indicates that information and resources are shared among components or stakeholders

weighted average traffic flow (production) in a road network. Based on the g-MFD, Hoogendoorn et al. have proposed an alternative definition for network resilience by taking the partial derivative of the level of service to the spatial density variation: “the rate in which the level of service drops when the spatial variation in density increases” [15]. In this method, the average speed in the network is selected as a proxy for level of service (which could be the network production as well). However, the methodology has been only tested on a ring motorway in Netherlands and has to be further investigated on different types of network structure and management strategies.

Traffic Management for Major Events

Managing major events usually includes stakeholders beyond the typical transportation agencies and, therefore, requires continuous and effective coordination and collaboration. The travel patterns of event attendees (or evacuees) differ from their daily mobility and are much more complex for planners to predict. Therefore, authorities tend to rely on trial-error approaches, checklists and recommendations provided in handbooks (e.g. [22, 23]) and undertake reactive measures based on their previous experiences rather than planning [24]. Based on these manuals Decision Support Systems (DSS) are designed to help the local authorities to plan and implement the suitable ITS measures to cope with transport requirements for a major event. One of the recently developed DSS as part of the STADIUM project is described in the following section together with the corresponding ITS measures. Implementing dynamic traffic management measures is described in Figure 3 chapter TP4.2; when an incident is detected, possible strategies from a library of measures will be evaluated in real-time (using traffic simulation tools) and the best one will be implemented. This library of measures is

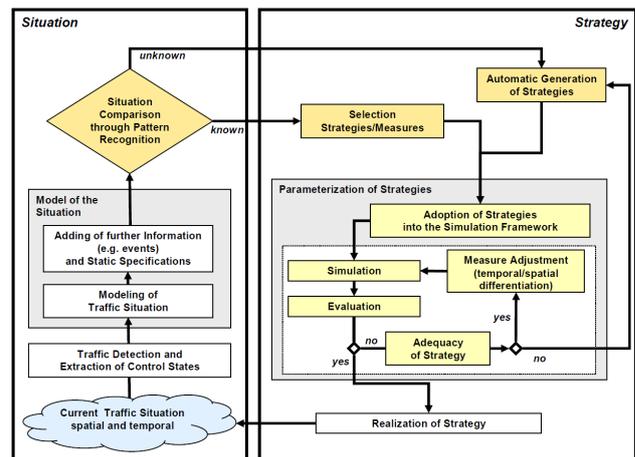


Figure 6. A general framework for automatic traffic management strategy generation [25]

normally developed off-line and in case of major events, it is likely to not find a very effective management strategy due to uniqueness of the situation. The alternative solution is to develop management strategies also in real-time. For example, in [25] a framework for automatic generation of management strategies have been provided as depicted in Figure 6.

However, due to several difficulties (i.e. regulations for implementing the selected measures, high levels of complexity of incident detection on large networks, technical limitations, etc.) the idea of automatic generation of management strategies has been limited to a few number of measures on isolated intersections or a group of intersections, on a short section of motorways, etc. Admittedly, there is a need for deep research in this field to automatically develop management strategies on large-scale networks.

Table 3. List of ITS measures for large events adapted from [14]

Category	Goal	Practical Example
Demand management applications	reducing the traffic congestion on demand side	Ramp closure, Turn restriction at intersections, Re-routing, ...
Traffic management systems	enhancing the capacity of the transport network	Demand-responsive signal control, Extra public transport supply, Contraflow streets, ...
Collective and alternative transport application	aim at bridging the gap between public and private transport modes	Car-sharing and car-pooling, Park & Ride, On-demand transport, ...
Integrated payment systems	involves technologies for congestion pricing and payment methods	Automated gates, Contactless payment, Single payment for multi-purpose bookings, ...
Integrated platforms	tools which rely on large-scale data acquisition and exchange to improve the system performance	Smartphone applications, Service monitoring, Public transport coordination
Traveler information services	designed to provide real-time information to users and fleet managers, and promote use of alternative transport modes	Navigation and routing, Real-time parking information and public transport time table

Traffic Modelling and Simulation of Major Events

Modelling and simulation can illustrate the demand and operation on the network and help to improve the management strategies with a sufficient level of accuracy before their implementation [1]. However, difficulties in demand prediction for a major event remain a challenge to employ such tools [24]. Moreover, route choice and driving behavior are different from routine conditions as well, which makes the calibration task much more complex. The level of details in simulation has been proven to significantly affect the accuracy of the results. It is even more important in case of major events where usually large-scale networks are modelled and some features are neglected in order to save computation time. For instance, Fellendorf [28] has tested three levels of traffic simulation (micro-, meso- and macroscopic) on three different events in Germany to evaluate the effectiveness of traffic simulation to be used as a tool for assessing management strategies. He concludes that models to forecast traffic at large events are satisfying, but the road network has to have a high level-of-detail to capture any changes in infrastructure and control measures. In another study [17] the impact of properly modelling spillback has been investigated and the outcome of the study indicates that the affected links due to an incident cannot be found if spillback is not simulated. The study also implies that without modeling spillback the affected links due to a road closure cannot be found and the road network is considered more robust.

Simulation is not only useful for understanding and predicting the behavior of people, but it is also a useful tool for planning dynamic routes for emergency responses. Murray-Tuite [16] uses simulation to compare the effect of system optimum against user equilibrium traffic assignment during evacuation, and the results show that system optimum performs better in terms of recovery and mobility.

Despite the results of these studies and other similar ones,

some argue that even if data from very similar events are used, real-time calibration and validation of the models are required, as the behavior of the road users could be different in the same repeated event [29]. Recent efforts, hence, have been towards real-time short-term traffic prediction using Dynamic Traffic Assignment (DTA) method to model large-scale networks at an aggregated level: with this approach it is possible to evaluate various management strategies in real-time before their implementation on a network. However, one of main limitations of DTA models is that they have a macroscopic approach and do not consider the inhomogeneity of individuals [30]. There are several model-based (e.g. DynaMIT [31], VISTA [32], PTV Optima [33] and data-driven approaches (K-Nearest Neighbor Model [34], Hidden Markov Model [35], Adaptive Kalman Filter [36]) to predict the short-term situation of the road network. However, most of these tools and methods have been so far tested on motorway networks with limited number of ITS measures. Partitioning the road network to sub-networks in order to reduce the level of complexity has been mostly used to overcome this problem. Network decomposition can be done in many ways; by using the notion of MFD to dynamically define areas with a homogeneous congestion [37], by determining static congestion clusters using historical Floating Car Data (FCD) [38] or by using Artificial Neural Network (ANN) to identify the affected road segments [39] are among the most recent ones. However, such methodologies are not easily applicable to most of the major events due to lack of historical data and difficulties in conducting experiments to collect the required data [40].

Application of Active Traffic Management Tools for Major Events

Traffic management based on short-term prediction is mainly composed of two cores: first a short-term traffic prediction

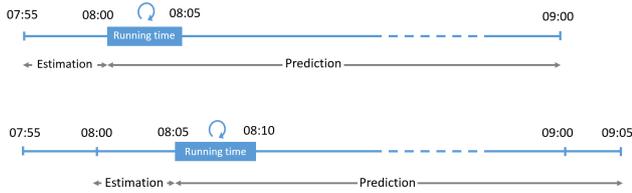


Figure 9. Illustration of rolling horizon methodology in traffic state prediction

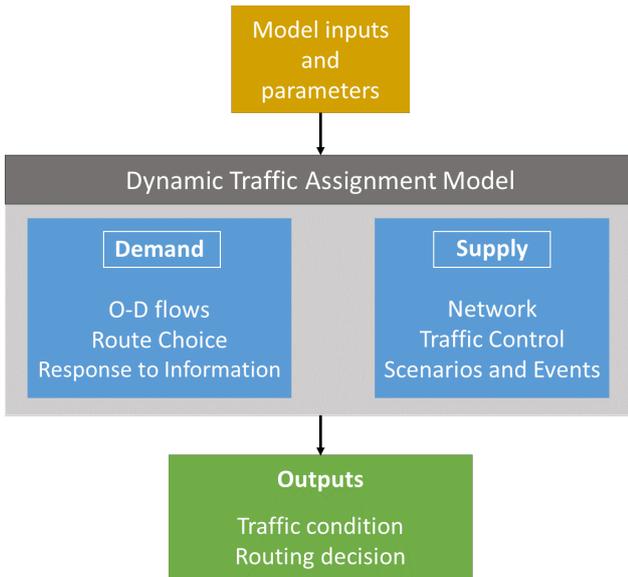


Figure 10. Dynamic traffic assignment framework [42]

model, and second, a strategy analyzer which quantifies the impact of each management scenario and recommends the best one to be implemented. Rolling horizon is the general approach employed by such tools, in which first, the current state of the network is estimated and then DTA models are used to predict the short-term conditions for a given horizon (e.g. one hour). The procedure is repeated periodically to get frequent actuations of the prediction. Taking advantage of parallel computing, advanced traffic simulations execute such predictions usually in less than 10 minutes using real-time data [41].

In these tools, demand and supply are separately simulated and their complex interrelations are represented by large-scale mesoscopic or macroscopic traffic simulations as depicted in Figure 10 [42]. Below, some of the recently developed traffic management tools are briefly introduced and their applicability for major events is discussed.

DynaMIT

DynaMIT (**D**ynamic **N**etwork Assignment for the **M**anagement of **I**nformation to **T**ravelers) [31] is a simulation based DTA which captures the effect of the delivered information to the drivers. However, it requires extensive amount of traffic

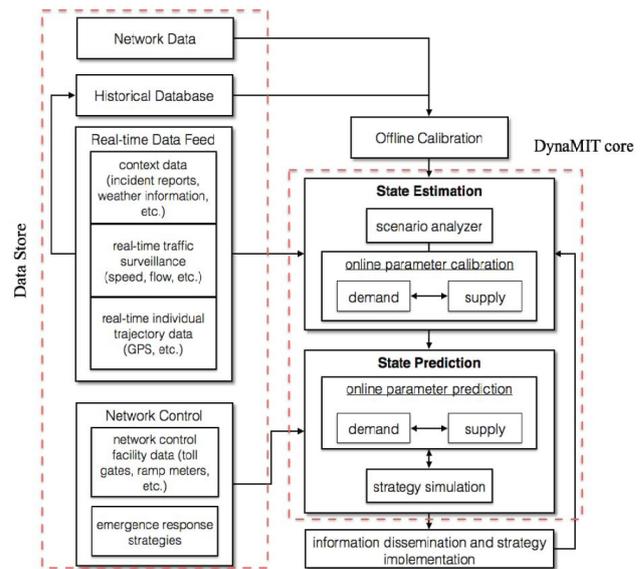


Figure 11. Architecture of DynaMIT2.0 [41]

surveillance data and incident information to generate reliable outputs. For example, in a case study in New York, DynaMIT was employed to evaluate incident diversion strategies through Variable Message Sign (VMS), but there was a need for manual calibration of 6470 parameters including O-D flows, segment capacities, speed-density relationship parameters. Recently, a strategy simulator has been integrated in DynaMIT2.0 [41] which predicts incident duration (using topic modelling technique) and special event demand (by employing a Bayesian additive linear model). Strategy simulation module (see Figure 11) analyzes the impact of different management strategies in real-time; at the moment the objective function is to either minimize the total travel time in the network or maximize total traveler welfare using Generic Algorithm. DynaMIT-E [43] is an extension of DynaMIT for emergencies which has a similar framework, and therefore, is not discussed in details. The only difference is that in DynaMIT-E scenarios are developed differently and the situation characteristics are included (e.g. changes in network topology due to the event).

VABENE++

VABENE++ [44] is a traffic management tool developed exclusively for major events. The advantage of VABENE++ in comparison to other tools is that German Center for Aerospace (DLR) is aiming to provide satellite image and real-time aerial images in addition to ground-based sensor data. The collected data from these sources are integrated into EmerT portal (Emergency mobility of rescue forces and regular traffic), which is a web-based decision support application. VABENE++ uses microscopic traffic simulation (SUMO: Simulation Urban Mobility) to forecast the condition of the network to deliver the travel time on various routes. However, it is not

clear what kind of ITS measures are used in VABENE++ and how are the strategies selected. Currently, VABENE++ is available for three demonstration areas in Germany and is accessible only for authorized users.



Figure 12. Car detection and speed measurement using aerial image [44]

Siemens Sitraffic

Siemens offers a modular Sitraffic software package [45] for a wide range of control and management purposes. Sitraffic Concert is an integrated traffic management platform that offers TMC a solution for strategic coordination of traffic control and traffic information systems. It is a tool for everyday use and is able to react to the disturbances in the traffic flow through incident and strategy management modules. Siemens has also integrated PTV VISSIM (microscopic traffic simulation software developed by PTV Group) in Sitraffic packages in order to estimate the traffic state and evaluate the impact of the control strategies prior to their implementation. Sitraffic Concert was used during Athens Olympic games in back in 2004 [46] in two operating centers and was able to visualize the traffic condition based on the delivered information. In addition, it was capable of automatic control of Variable Message Signs (VMSs), adjusting the traffic signal phase and cycle time and to alert the police officers on the scene if needed.

PTV Optima

PTV Optima [33] follows a similar path; it uses model-based approach to predict the traffic state for a time period of up to one hour. The dynamic traffic assignments are derived from PTV VISUM (macroscopic transport modelling software developed by PTV Group) and integrated together with online data (e.g. loop detectors, FCD, ANPR, etc.) into PTV Optima. It runs traffic simulations in background to evaluate the impact of management strategies using Key Performance Indexes (KPIs), which are usually in accordance to TMC’s objectives (the most frequent KPIs can be seen in Figure 12). The recommended strategy is then available for operators at TMC through a web-based GUI and could be also automatically disseminated among the travelers. The operators at

TMC are able to implement the recommended strategy via a web-based GUI and observe the state of the network through a map-based monitoring tool.

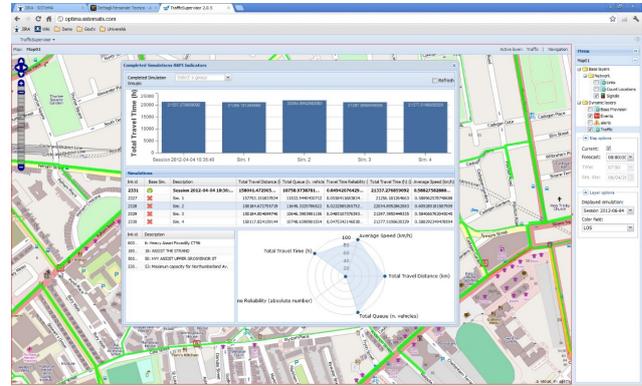


Figure 13. PTV Optima compares strategies by calculating KPIs and recommends the best one [33]

Discussion and conclusion

In this report, first, recurring and non-recurring congestion together with their characteristics were reviewed and their key differences were scrutinized. Many studies try to distinguish major events by bringing examples for planned and unplanned events. This implies lack of a widely accepted methodology to define whether an event is major, especially for unplanned incidents. In the scope of TUM LLCM TP4.3 sub-project, major events are unpredicted incidents or planned special events which lead to non-recurring congestion beyond the spatio-temporal size of routine congestions. In working toward this goal, a measurement index will be developed in order to classify the event intensity and the level of necessary management actions.

Recently, research in field of major (or extreme) events increasingly focuses on transportation resilience as a key element of dealing with the events. However, most of the studies have only considered infrastructure damage and not short-term disruption on urban network. Thus, there is a need for further research to design coordinated traffic management measures to support decision-making process especially for ex post actions in order to expedite the recovery to pre-event level of service. Network decomposition and online parameter calibration for a large-scale network are the obstacles to implement the best management measures in comparison with long-term transportation planning actions. Recent findings in aggregated traffic dynamics, especially the notion of MFD, have significantly contributed to advance the traffic modelling on network scale. In addition, exploiting parallel computation has significantly contributed to enhance the performance of real-time short-term traffic predictions to evaluate different management scenarios. Nevertheless, modelling the behavior of road users at aggregated levels cannot capture the inhomogeneity and preferences of individuals. Thus, in this project

the possibilities to use agent-based approaches to rectify this limitation will be investigated.

Big data in transport, especially real-time in-vehicle data, and the existence of technologies (e.g. Car-to-X communications, Bluetooth identification, etc.) has created new opportunities and challenges to re-engineer the implementation of traffic management measures, which are potentially a replacement for manuals and checklists that responsible stakeholders go through before and during events.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] Deborah Matherly, Pamela Murray-Tuite, and Brian Wolshon. Traffic management for planned, unplanned, and emergency events. In *Traffic Engineering Handbook*, pages 599–636. John Wiley & Sons, Inc, Hoboken, NJ, USA, 2015.
- [2] Ahmad Tavassoli Hojati, Luis Ferreira, Simon Washington, Phil Charles, and Ameneh Shobeirinejad. Modelling the impact of traffic incidents on travel time reliability. *Transportation Research Part C: Emerging Technologies*, 65:49–60, 2016.
- [3] Richard Dowling, Alexander Skabardonis, Michael Carroll, and Zhongren Wang. Methodology for measuring recurrent and nonrecurrent traffic congestion. *Transportation Research Record: Journal of the Transportation Research Board*, 1867:60–68, 2004.
- [4] OECD. *Managing Urban Traffic Congestion*. OECD Publishing, Paris, 2007. ISBN:9789282101285.
- [5] K. Ozbay, and P. Kachroo. Incident management in intelligent transportation systems. http://digitalscholarship.unlv.edu/ece_fac_articles/103, 1999. Accessed: 22/06/2016.
- [6] Daniela Bremmer, Keith Cotton, Dan Cotey, Charles Prestud, and Gary Westby. Measuring congestion: Learning from operational data. *Transportation Research Record: Journal of the Transportation Research Board*, 1895:188–196, 2004.
- [7] STADIUM. ITS for large events. <http://www.largeevents.eu/>, 2013. Accessed: 25/05/2016.
- [8] *Highway Capacity Manual*. Transportation Research Board, National Research Council, Washington D.C., 2000. ISBN:0-309-06681-6.
- [9] Berk Anbaroglu, Benjamin Heydecker, and Tao Cheng. Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. *Transportation Research Part C: Emerging Technologies*, 48:47–65, 2014.
- [10] Francisco C. Pereira, Filipe Rodrigues, Evgheni Polisciuc, and Moshe Ben-Akiva. Why so many people? explaining nonhabitual transport overcrowding with internet data. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1370–1379, 2015.
- [11] Ruoqing W. Scholz and Yongmei Lu. Detection of dynamic activity patterns at a collective level from large-volume trajectory data. *International Journal of Geographical Information Science*, 28(5):946–963, 2014.
- [12] Martin Müller. What makes an event a mega-event? definitions and sizes. *Leisure Studies*, 34(6):627–642, 2015.
- [13] G. Wolfgang Heinze. Grundlagen der verkehrsplanung von event. In *Handbuch Eventverkehr*, volume 9 of *KulturKommerz*, pages 25–64. Erich Schmidt Verlag GmbH & Co., Berlin, 2004.
- [14] STADIUM. ITS for large events: Incident management. http://www.largeevents.eu/wp/wp-content/uploads/2012/10/incident_management.pdf, 2013. Accessed: 5/6/2016.
- [15] Serge P. Hoogendoorn, Victor L. Knoop, Hans van Lint, and Hai L. Vu. Applications of the generalized macroscopic fundamental diagram. In Mohcine Chraïbi, Maik Boltes, Andreas Schadschneider, and Armin Seyfried, editors, *Traffic and Granular Flow '13*, pages 577–583. Springer International Publishing, Cham, 2015.
- [16] Pamela Murray-Tuite. A comparison of transportation network resilience under simulated system optimum and user equilibrium conditions. In *2006 Winter Simulation Conference*, pages 1398–1405, 2006.
- [17] V. Knoop, H. van Zuylen, and S. Hoogendoorn. The influence of spillback modelling when assessing consequences of blockings in a road network. *European Journal of Transport and Infrastructure Research*, 8(4):287–300, 2008.
- [18] B. Immers, A. Bleukx, J. Stada, C. Tampere, and I. Yperman. Robustness and resilience of road network structures, 2004.
- [19] Timothy McDaniels, Stephanie Chang, Darren Cole, Joseph Mikawoz, and Holly Longstaff. Fostering resilience to extreme events within infrastructure systems: Characterizing decision contexts for mitigation and adaptation. *Global Environmental Change*, 18(2):310–318, 2008.
- [20] Richard Margiotta, Rick Dowling, and Jawad Paracha. Analysis, modeling, and simulation for traffic incident management applications.

- [21] Nikolas Geroliminis and Carlos F Daganzo. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, 42(9):759–770, 2008.
- [22] U.S. Department of Transportation. Planned special events: Checklists for practitioners, 2006.
- [23] Federal Highway Administration. *A Guide to Regional Transportation Planning for Disasters, Emergencies, and Significant Events: NCHRP Report 777*. Washington, D.C., 2014.
- [24] Francisco C. Pereira, Filipe Rodrigues, and Moshe Ben-Akiva. Using data from the web to predict public transport arrivals under special events scenarios. *Journal of Intelligent Transportation Systems*, 19(3):273–288, 2014.
- [25] Paul Mathias, Robert Braun, and Fritz Busch. Automatic generation of traffic management strategies. In *12th World Congress on Intelligent Transport Systems*, 2005.
- [26] FGSV: Forschungsgesellschaft fuer Strassen-und Verkehrswesen. *Hinweise zur Strategieentwicklung im dynamischen Verkehrsmanagement*, volume 381. FGSV, Koeln, 2003 edition, 2003. ISBN:3-937356-14-2.
- [27] FGSV: Forschungsgesellschaft fuer Strassen-und Verkehrswesen. *Hinweise zur Strategieranwendung im dynamischen Verkehrsmanagement*, volume 381,1 : W1. FGSV, Koeln, 2011 edition, 2011. ISBN:978-3-941790-84-1.
- [28] Martin Fellendorf. Traffic modelling of large events – a summary of selected german examples. *IFAC Proceedings Volumes*, 39(12):17–24, 2006.
- [29] E. Prionisti and C. Antoniou. Sensitivity analysis of driver behavior under emergency conditions. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1263–1268, Sept 2012.
- [30] Gregor Lämmel, Marcel Rieser, and Kai Nagel. *Large Scale Microscopic Evacuation Simulation*, pages 547–553. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [31] Moshe Ben-Akiva, Haris N. Koutsopoulos, Constantinos Antoniou, and Ramachandran Balakrishna. Traffic simulation with dynamit. In Jaume Barceló, editor, *Fundamentals of Traffic Simulation*, volume 145 of *International Series in Operations Research & Management Science*, pages 363–398. Springer New York, New York, NY, 2010.
- [32] Athanasios K. Ziliaskopoulos and S.Travis Waller. An internet-based geographic information system that integrates data, models and users for transportation applications. *Transportation Research Part C: Emerging Technologies*, 8(1-6):427–444, 2000.
- [33] PTV Group. PTV Optima. <http://vision-traffic.ptvgroup.com/en-us/products/ptv-optima/>, 2016. Accessed: 25/05/2016.
- [34] Lun Zhang, Qiuchen Liu, Wenchen Yang, Nai Wei, and Decun Dong. An improved k-nearest neighbor model for short-term traffic flow prediction. *Procedia - Social and Behavioral Sciences*, 96:653–662, 2013.
- [35] Yan Qi and Sherif Ishak. A hidden markov model for short term prediction of traffic conditions on freeways. *Transportation Research Part C: Emerging Technologies*, 43:95–111, 2014.
- [36] Jianhua Guo, Wei Huang, and Billy M. Williams. Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transportation Research Part C: Emerging Technologies*, 43:50–64, 2014.
- [37] Mohammadreza Saeedmanesh and Nikolas Geroliminis. Optimization-based clustering of traffic networks using distinct local components. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems - (ITSC 2015)*, pages 2135–2140, 2015.
- [38] Felix Rempe, Gerhard Huber, and Klaus Bogenberger. Spatio-temporal congestion patterns in urban traffic networks. *Transportation Research Procedia*, 15:513–524, 2016.
- [39] Simon Kwoczek, Sergio Di Martino, and Wolfgang Nejdl. Stuck around the stadium? an approach to identify road segments affected by planned special events. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems - (ITSC 2015)*, pages 1255–1260.
- [40] Giuseppe Musolino and Antonino Vitetta. Calibration and validation of a dynamic assignment model in emergency conditions from real-world experimentation. *Procedia - Social and Behavioral Sciences*, 111:498–507, 2014.
- [41] Yang Lu, Ravi Seshadri, Francisco Pereira, Aidan OSullivan, Constantinos Antoniou, and Moshe Ben-Akiva. Dynamit2.0: Architecture design and preliminary results on real-time data fusion for traffic prediction and crisis management. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems - (ITSC 2015)*, pages 2250–2255, 2015.
- [42] Yi-Chang Chiu, Jon Bottom, Michael Mahut, Alexander Paz, Ramachandra Balakrishna, Travis Waller, and Jim Hicks. Dynamic traffic assignment: A primer.
- [43] Ramachandran Balakrishna, Yang Wen, Moshe Ben-Akiva, and Constantinos Antoniou. Simulation-based framework for transportation network management in emergencies. *Transportation Research Record: Journal of the Transportation Research Board*, (2041):80–88, 2008.
- [44] Veronika Gstaiger, Franz Kurz, Hohloch., and Marc Hohloch. Vabene++ multi-sensor approach to support crisis management. *International Disaster and Risk Conference (IDRC)*, 2014.

- [45] Siemens. Sitraffic concert, sitraffic scala and sitraffic guide. <https://www.mobility.siemens.com/mobility/global/SiteCollectionDocuments/en/road-solutions/urban/traffic-control-center/three-complex-tasks-en.pdf>. Accessed: 25/05/2016.
- [46] Road Traffic Technology. Athens traffic management system. <http://www.roadtraffic-technology.com/projects/athens-traffic-management/>.

Collaborative and Social Mobility Services

Daniel Herzog and Wolfgang Wörndl

Department of Informatics, Technical University of Munich, Munich
{herzogd, woerndl}@in.tum.de

Abstract

Innovative mobility services are necessary to face the challenges of urbanization. In this paper, we review the state of the art of collaborative and social mobility services by introducing a new classification, outlining scientific work and presenting example applications. The classification considers all important aspects of individual transport, public transport and intermodal passenger transport: Recommending destinations, planning and organizing a trip, finding or sharing vehicles and protecting humans and the environment. In future works, our classification and overview of existing services can be used to develop new and innovative mobility services.

Keywords

Mobility Service; Social; Collaborative; Mobile application; Recommendation

1. Introduction

The trend towards urbanization is accelerating. Today, more than half of the world's population is living in urban areas and this value is supposed to increase to 66% by 2050 [1]. At the same time, cars continue to be the most preferred means of transport in many places of the world. In Europe, half of the population uses a car at least once a day while public transport is used by only 16% every day. In contrast, 29% of Europe's population never uses public transport [2]. Even though bikes are the most used means of transport in some cities, half of the population in Europe never uses a bike as an alternative means of transport. While the population and thus the number of car drivers in urban areas is increasing, a car's average occupancy with 1.5 passengers per vehicle remains to be low [3].

The combination of these facts leads to a lack of space in cities, more congestion and higher pollution. Innovative mobility services are essential when facing the challenges of future mobility. They are supposed to make transport within and between cities more convenient and sustainable.

Such mobility services are not solely limited to individual transport. They can promote public transport and facilitate the planning of intermodal passenger transport which combines different kinds of means of transport within one trip. Furthermore, pedestrians represent another target group of mobility services. One example is a tourist who receives recommendations for points of interest (POIs) like restaurants, museums or monuments while exploring a city.

Recommender systems (RSs) filter large amounts of data to present services or products to users which best satisfy their needs. Due to the widespread usage of mobile devices like smartphones, RSs are more and more accessed in mobile environments [4]. Recommendations can be improved when taking context into account such as the user's current location to recommend, for example, interesting spots nearby [5]. Furthermore, RSs can incorporate social or collaborative

aspects like the opinions of friends or people with similar preferences to generate personalized recommendations. They enrich innovative mobility services by supporting road users, passengers and pedestrians in various scenarios, for example, when recommending POIs, intermodal routes or interesting events.

The main goal of this work is to provide an overview of existing collaborative and social mobility services and relevant scientific approaches. For this purpose, we introduce a classification for mobility services. We present released applications and related work in research for each mobility service category. This work terminates with a short conclusion. In future works, this survey and the introduced classification can be used as a basis for developing new and innovative mobility services.

2. Classification of Collaborative and Social Mobility Services

In this work, we call services and applications mobility services if they support the user in finding destinations, routes and relevant information for required means of transport, if they make transport within and between cities more convenient, faster, safer or sustainable or if they motivate people to move in a way that fulfills these requirements. A mobility service is called collaborative and social if the service improves when used by multiple users or if somehow information, personal data or hardware is shared between users or the user and the service.

In the following, we introduce a new classification of connected mobility services by dividing the topic into different categories. For each category, we present examples of existing research projects and released applications. These examples show how digitalization shapes the future of connected mobility and they highlight the benefits for road users and passengers when receiving personalized information. The

presented state of the art can be used as a basis for identifying new and innovative mobility services.

Figure 1 illustrates the classification. To the best of our knowledge, no similar classification for collaborative and social mobility services exists. We developed this classification based on the existing mobility services we found. The classification represents all important aspects of mobility: First of all, the user has to plan and execute her or his trip. This includes either using individual transport, public transport or combining both in intermodal trips. Innovative mobility services promote the sharing of vehicles. Furthermore, interesting POIs can be identified and added to the trip by using mobile RSs. In addition, sustainable services that protect humans and the environment are steadily gaining popularity. The classification considers further mobility services that do not fit in one of the presented categories, for example, services that motivate people to use other services or selected means of transport.

3. Trip Planning and Organizing

Existing mobility services facilitate each step of planning and organizing a trip. The range of services covers not only individual transport, moreover they support the usage of public transport and intermodal passenger transport as well.

3.1 Individual Transport

Important steps of a trip using individual transport are the creation of a route or finding a destination, navigating to the destination, finding parking spots, gas stations or charging stations. Most of the existing services support the user in solving one or a small number of these tasks. A few comprehensive services exist. Google Maps¹ is one example of such a comprehensive service as it combines route planning, traffic data and information about gas services and POIs. Google Maps is available as a web application and mobile application for different operating services. Another similar service is BayernInfo², which provides traffic information, a route planner for car drivers or cyclists and information about parking spots with a focus on Bavaria. BayernInfo is also available as a web application and for Android and iOS devices.

3.1.1 Route Creation

Various route planner such as Roadtrippers³, Furkot⁴ and myscenicdrives⁵ for car drivers or Bikely⁶ for cyclists are available. These applications allow to create individual routes, to export them and to share them with friends.

A few research projects and released applications support collaborative route creation. Cheng et al. [6] developed CozyMaps, a multi-display system using tablets to create

route sections of a planned trip. Updates are instantly sent to all other users and shared on a large display providing an overview of the complete route and the areas the other users are working on. Furthermore, the users can share their current work by sending it to the large screen. In a user study, the participants called this solution useful and funny. Holone et al. [7] presented OurWay, a collaborative route planning system. It uses community ratings of route segments to provide routes adapted to the users' abilities and needs. An indoor experiment where users in wheelchairs solved navigational tasks shows that the approach of OurWay leads to promising results. Nevertheless, the authors found out that ratings were mainly produced by the individuals to accomplish their personal goal rather than intentionally providing support to the community. Wörndl and Hefele [8] developed a recommender system for city trip planning. Users can enter a starting and end point and express their preferences in six different categories like sights, nightlife or food on a scale from 0 (no places are suggested) to 5 (places in this category are preferred if possible). Optionally, the user can enter a time and budget limit. The recommended POIs are selected by taking Foursquare user ratings into account, and combined to a trip.

CityTripPlanner⁷ is a similar, released web-based service but offers route creation only for a set of pre-determined cities. UMapper⁸ is a website allowing users to create embeddable online maps together. In a Wiki-like collaboration, the users can add markers, shapes or routes and share their results.

3.1.2 Navigation

Some existing mobility services focus on navigation and providing real-time traffic data to improve routing. Figure 2 shows Waze⁹, an application available for Android, iOS, Windows and as a web application. Its real-time navigation adapts to current warnings shared by the community. Users can notify others about accidents, threats, road blocks or cheap gas stations. The incentive is a Gamification method which awards users sharing information points. In addition, Waze can navigate the user to Facebook events or calendar entries.

Nunav¹⁰ is an Android application which automatically distributes car drivers on the street to minimize travel time for all road users using an intelligent swarm algorithm. The system updates the driver's route every 15 seconds to ensure an optimized routing. The user can see the saved travel time and is able to share the estimated time of arrival (ETA) with friends.

3.1.3 Parking

A large number of basic parking garage search applications exists. Some operators of parking garages enrich their offer by real-time information about available parking spots. Examples of parking spot and garage applications are Pango Mobile

¹<http://maps.google.com>

²<http://www.bayerninfo.de>

³<https://roadtrippers.com>

⁴<https://trips.furkot.com>

⁵<https://www.myscenicdrives.com>

⁶<http://www.bikely.com>

⁷<http://www.citytripplanner.com>

⁸<http://www.umapper.com>

⁹<https://www.waze.com>

¹⁰<http://nunav.net>

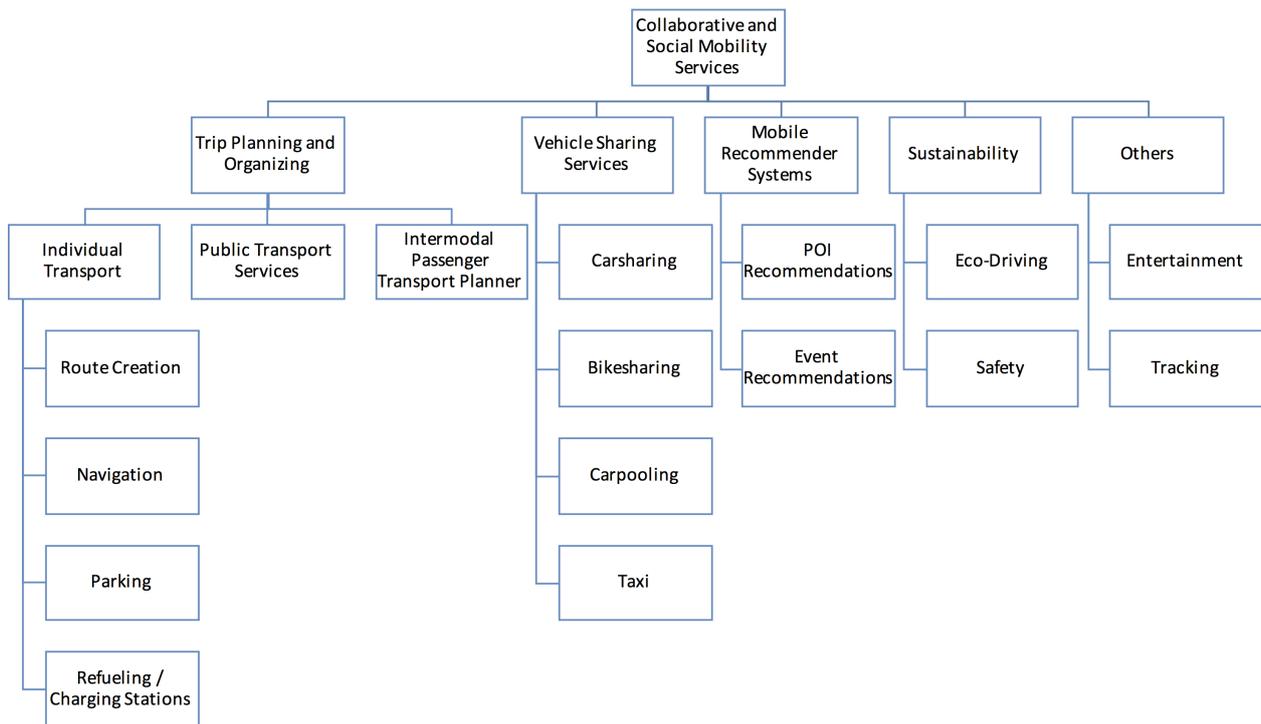


Figure 1. A classification of existing collaborative and social mobility services

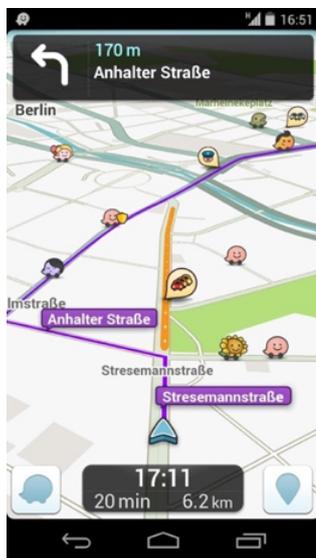


Figure 2. Waze Android Application

Parking¹¹, Best Parking¹² and ParkNow¹³.

Some research projects try to improve parking spot search

¹¹<http://www.mypango.com>

¹²<http://www.bestparking.com>

¹³<http://de.park-now.com>

by taking real-time data of other vehicles into account. ParkSense is a smartphone based sensing system that detects if a driver has released a parking spot using Wi-Fi beacons in urban areas [9]. The authors showed that this approach leads to a smaller energy footprint than traditional location sensors like GPS and Wi-Fi based positioning but still maintains a sufficient accuracy. PocketParker is a crowdsourcing system using smartphones to predict parking spot availability [10]. The implemented sourcing technique requires no explicit user input or additional infrastructure. The service detects arrivals and departures automatically. An evaluation of the system showed that PocketParker can detect parking events quickly and correctly. ParkNet, however, is a mobile system comprising vehicles that collect parking space occupancy using a GPS receiver and a passenger-side-facing ultrasonic rangefinder while driving by potential parking spots [11]. The results of 500 miles of road-side parking data collected over two months shows high accuracy. Furthermore, the authors claim that their solution would be more cost-effective by an estimated factor of roughly 10-15 compared to a sensor network with a dedicated sensor at every parking space.

A few released applications implement similar ideas and focus on collaboration between drivers. ParkNav¹⁴ calculates the probability of free parking spots by taking forecasts, statistics and real-time data into account. It is available for

¹⁴<http://www.parknav.com>

Android, iOS and as a web application. Parkopedia¹⁵ is a web and mobile application for Android, iOS and Windows. It contains a database of existing parking spots and garages. It is an example of a service whose content is created manually by a community. JustPark¹⁶ (previously: Park at my house, available for iOS and as a web application) and MonkeyParking¹⁷ (iOS) focus on private parking spots, e.g., on residential driveways. People offering their private parking spots to earn money when their spots are booked.

3.1.4 Refueling / Charging Stations

In Germany, changes in fuel prices have to be sent to the “Markttransparenzstelle für Kraftstoffe” since August 31, 2013 [12]. Since then, various websites and mobile applications are available to find gas stations and to determine the lowest price.

A few mobile applications enhance this offer by some social features. GasBuddy¹⁸ is a service available in the US and Canada. It supports Android, iOS, Windows and Blackberry and is available as a web application. GasBuddy motivates users to report current gas prizes and awards them points for a leaderboard. In addition, GasBuddy holds a drawing for a \$100 coupon every day. SmartTanken¹⁹ is a German application available for iOS and as a web application. It allows users to comment and rate charging stations. Furthermore, the users can add missing stations. PlugShare²⁰ is a similar service for electric vehicles. It is available for Android, iOS and as a web application. PlugShare offers the same features as SmartTanken but users can also coordinate with others when they want to charge their vehicle.

3.2 Public Transport Services

Moovit²¹ is a journey planner application available for Android, iOS and Windows. Besides calculating routes for public transport, the application can calculate the ETA, discover arrival and departure times of stops nearby and store the user’s favorite routes. It differentiates in its offer of social functions. Users can notify the community about every kind of event or incident as well as the crowdedness or cleanliness of stations. Committed users are awarded points. Tiramisu²² is an application available for Android, iOS and as a web application. It is associated with research at the RERC on Accessible Public Transportation at Carnegie Mellon University and available as a beta release. Tiramisu is a real-time bus tracking which calculates arrival information and calculates delays automatically. Users can share the current location of a bus and its fullness. Various Facebook groups exist to find passengers who are willing to share the Bayern-Ticket of Deutsche Bahn to decrease the price per person. Deutsche Bahn launched a mobile

application for Android and iOS called DB Mitfahrer²³ which supports the search. Users can create a new ticket group for a trip or find existing ones. Furthermore, users can rate other passengers and add them to their personal favorites.

3.3 Intermodal Passenger Transport Planner

Applications like Moovel²⁴ (available for Android, iOS and as a web application) provide an intermodal route planner to combine different means of transport within one journey. Moovel enhances this offers by in-app booking of, for example, Car2Go vehicles and mobile payment.

Some of the comprehensive solutions such as Bayern-Info and Google Maps, presented in section 3.1, also support intermodal passenger transport planning. They allow to incorporate further means of transport such as bikes or public transport into generated routes.

4. Vehicle Sharing Services

In the last years, new mobility services that promote the sharing of vehicles were released. Sharing a vehicle can either mean car- or bikesharing but services that promote sharing a ride or a taxi are also considered in this section.

4.1 Carsharing

Carsharing can be defined “as the organized collective use of passenger cars. It can reduce car ownership while ensuring a high level of mobility for urban residents” [13]. Carsharing gained popularity during the last years due to services like DriveNow²⁵ (offered by BMW and Sixt), Car2Go²⁶ (Daimler, Europcar) and the American company Zipcar²⁷.

Private carsharing services allow users to share their own vehicles with the community. Hence, they are an example of a mobility service not solely focusing on sharing data. Instead, hardware is shared within a community. Tamyca²⁸ and Drivy²⁹ are two services available as web applications and for Android and iOS allowing users to offer their private vehicles for rent. Users can rate the vehicle they rented.

4.2 Bikesharing

Bikesharing services work like carsharing services but target cyclists. Bikesharing services exist in many cities and are either operated by a company or by the city government. Examples are MVG Rad³⁰ in Munich and Vélib³¹ in Paris.

Private Bikesharing services allow users to share their own bike with others. BitLock³² offers a bicycle lock that can be closed and opened by an Android or iOS app. It uses

¹⁵<http://www.parkopedia.de>

¹⁶<https://www.justpark.com>

¹⁷<http://monkeyparking.strikingly.com>

¹⁸<http://www.gasbuddy.com>

¹⁹<https://www.smarttanken.de>

²⁰<http://www.plugshare.com>

²¹<http://moovitapp.com>

²²<http://www.tiramisutransit.com>

²³<http://www.bahn.de/wmedia/view/mdb/media/intern/mitfahrer-app>

²⁴<https://www.moovel.com>

²⁵<https://www.drive-now.com>

²⁶<https://www.car2go.com>

²⁷<http://www.zipcar.com>

²⁸<https://www.tamyca.de>

²⁹<https://en.drivy.com>

³⁰<https://www.mvg.de/services/mobile-services/mvg-rad.html>

³¹<http://www.velib.paris>

³²<http://bitlock.co>

Bluetooth Low Energy for communication between phone and bike and its battery can last over five years. Bike owners can share the position of their bike and grant permissions to others who are allowed to use the bike. Spinlister³³ is a website and application for Android and iOS devices for renting private bikes as well as other sport equipment like skis or snowboard. In-app payment is available and Spinlister insures the rented products.

4.3 Carpooling

Carpooling or ridesharing “exists when two or more trips are executed simultaneously, in a single vehicle” [14]. Blablacar³⁴ and Flinc³⁵ are two examples of carpooling services which allow users to find passengers when they travel from one city to another. The price for a city reduces the driver’s expenses for costs and compensates her or him for wear. The basic idea is that all persons in the car pay around the same amount for a trip. The service operator keeps a share of each transaction. Passengers can rate the driver after the ride.

Uber³⁶ (Android, iOS, Windows), as illustrated in Figure 3, and Lyft³⁷ (Android, iOS, web application) are services which differ in their purpose. Instead of offering carpooling for trips from one city to another, these two services can rather be called a taxi alternative. They focus on short trips within a city and the main incentive for drivers is profit [15]. Uber is already called “the world’s largest taxi company” [16] without owning a single vehicle which shows how such connected services shape the future of mobility.



Figure 3. Uber Android Application

³³ <https://www.spinlister.com>

³⁴ <https://www.blablacar.com>

³⁵ <https://flinc.org>

³⁶ <https://www.uber.com>

³⁷ <http://lyft.com/>

La’Zooz³⁸ is a project for real-time social ridesharing. Its goal is to synchronize empty seats with people traveling in the same direction. For this purpose, a location-based mining app is available for Android devices. Users should share their movement data to reach a critical mass of movement necessary for the real-time synchronizing.

4.4 Taxi

A few mobile applications for traditional taxi services exist as well. One example is MyTaxi³⁹, available for Android, iOS, Windows and Blackberry. Users can track taxis around them in real-time, book a taxi and pay via app. Shäre-a-taxi⁴⁰ is an Android and iOS application for taxis enhanced by a carpooling feature. A user can order a taxi and allow other users on the way to join the taxi. This reduces the costs for all passengers as everybody has only to pay her or his share. The payment is done via app.

5. Mobile Recommender Systems

Recommender systems support users to overcome the information overload problem by filtering a large amount of data to identify the information or products which best satisfy their needs. Smartphones and tablets allow to access web content in a mobile context, i.e., in different locations. Recommendations in mobile environments need to be more precise because users cannot browse through large lists of results to find suitable items. On the other side, mobile recommender systems promise more accurate recommendations because they can identify the context of a recommendation in a more detailed manner, e.g., the user’s current location or the means of transport she or he is using. [4]

In general, context can be described as “any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves” [17]. Research shows that context-aware recommender systems can generate more accurate recommendations than systems which do not take context into account [18]. Nevertheless, recommender systems should not provide recommendations solely based on the current context as the user does not want to lose the power of decision [19]. Furthermore, recommender systems can incorporate social or collaborative aspects like the opinions of friends or people with similar preferences to improve the recommendations.

The following examples show how recommender systems can support the users to get access to relevant information while moving.

³⁸ <http://lazooz.org>

³⁹ <https://www.mytaxi.com>

⁴⁰ www.share-a-taxi.com

5.1 POI Recommendations

Today, mobile recommender systems are used in various domains. A lot of research has been done in the field of mobile tourist guides and POI recommendations. Ricci and Nguyen [20] developed MobyRek, an on-tour support for travelers allowing them to complement their pre-travel plans by getting personalized travel product recommendations. The users can criticize the recommended products to adapt the recommendations to their needs. Averjanova et al. [21] extended the MobyRek critique-based system with a map interface. The map-based visualization of recommendations and the thereby offered means of interaction improve the system's effectiveness and increase the user satisfaction. Woerndl et al. [22] developed a hybrid POI recommender which allows the user to choose between different recommendation algorithms. Brauhofner et al. [23] present South Tyrol Suggests (STS)⁴¹, an Android-based mobile application that recommends POIs in South Tyrol. STS is context aware as well, it considers, for example, the weather when recommending tourist activities. STS is able to personalize recommendations even for new users by learning the user's preference model using a simple questionnaire. Tumas and Ricci [24] developed a personalized mobile city transport advisory system (PECITAS) for the citizens and city guests of Bolzano, Italy. Using PECITAS the user can obtain, directly on her or his mobile phone, recommendations for personalized paths between two arbitrary points in the city. ReRex is an iPhone application that allows users to obtain POI recommendations adapted to the current context [25]. Cena et al. [26] developed UbiquiTO, a tourist guide for users in Turin, Italy, which adapts the content provided to the user's interests, the physical location, the used devices and further context conditions. Park et al. use Bayesian Networks to model user preferences. Their proposed system collects context information like the location, the time and the weather to provide map-based personalized recommendations. This approach allows to overcome certain limitations like a small display and limited resources of mobile devices.

Research also pays a lot of attention to social POI recommendations. Brown et al. [27] developed George Square, a system for sharing leisure. Users can share their current location, browsed web pages and pictures. Furthermore, they can communicate with each other. Collaborative filtering recommends pages and places to others. A user study showed that the users like this approach of sharing their visits. SPETA is a social pervasive e-Tourism advisor. It takes the user's profile and the context (e.g., location, weather) into account. Further information can be extracted from social networks. Contacts nearby can be detected and considered for recommendations [28]. COMPASS is an example for a context-aware tourist recommender which recommends POIs like monuments but also travel buddies [19]. The application I'm feeling LoCo uses social networks to learn user profiles. Furthermore, constraints like the user's location, the means of transport she or he is using, physical constraints and the user's mood are taken

⁴¹<https://play.google.com/store/apps/details?id=it.unibz.sts.android>

into account for recommendations [29].

A large number of released applications exists that provide lists with POIs like restaurants, hotels or museums to tourists and locals but only a few of them provide personalized recommendations of POIs. Such POI RS benefit from a large community as users can rate locations and support others with reviews or recommendations. Examples of such services are TripAdvisor⁴² and Yelp⁴³ (Figure 4) which are available as web applications and for various mobile devices.

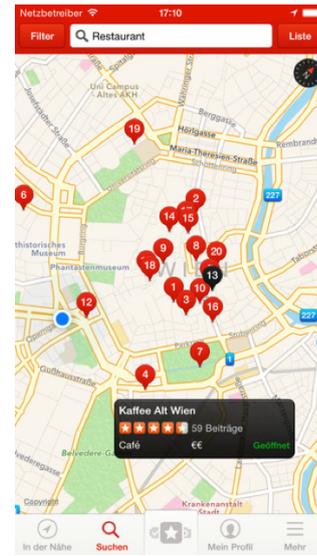


Figure 4. Yelp iOS Application

5.2 Event Recommendations

Event recommender systems suggest cultural or social events, amongst others, to the user. Event recommendations pose a special challenge in the field of recommender systems as user ratings are not available before the event takes place [30]. Hence, event recommenders have to consider further aspects such as planned visits of friends.

Herzog and Wörndl [31] developed a mobile application recommending all kinds of cultural events. The recommendation algorithm considers the context (e.g., the distance to a venue), the user's profile as well as interests of users with similar profiles. Users can login via Facebook to see Facebook contacts who are planning to attend an event. The application has been released as München Ticket Eventempfehlung⁴⁴ for Android and iOS.

Bandsintown⁴⁵ is an application for concerts. It is available as an Android, iOS, web and Facebook application. It recommends concerts based on the user's profile. In addition, the user can integrate external sources like Facebook or Spotify to improve the recommendations. Users can show

⁴²<https://www.tripadvisor.com>

⁴³<http://www.yelp.com>

⁴⁴<https://play.google.com/store/apps/details?id=de.x4a.eveapp.mt>

⁴⁵<http://bandsintown.com>

their planned attendance and see which other Facebook users are going to attend the event. XING EVENTS⁴⁶ is a business event application provided by the German social network XING. It is available for Android and iOS. The application recommends business events that might be interesting based on the user's profile and business network. Furthermore, potentially interesting contacts are recommended.

6. Sustainability

Many of the presented services have the positive effect of reducing pollution or making transport safer. In the last years, some services motivating people to drive more carefully or to protect people that travel alone at night were released. In this work, we allocate every service that mainly focuses on protecting humans or the environment to the category sustainability.

6.1 Eco-Driving

Mobility services offering guidance to drive more efficiently and to save fuel are already provided by some car manufacturers. One example of such a service is BMW's ECO PRO Analyser⁴⁷. This service is part of the BMW Connected App for Android and iOS devices. The app can be used via the in-car navigation system when the user's device is connected to the vehicle. It analyzes the driving style and provides the user with advices for a more efficient driving style.

A few research projects and released applications extend the idea of a driving style analyzer by implementing social incentives. Magana and Munoz-Organero [32] present GAFU, a training tool for efficient driving incorporating Gamification methods to motivate for participation. Users are awarded points and can compare their results with others. An experiment with 36 participants shows that the Gamification approach helps drivers not to lose interest for fuel saving and to avoid returning back to previous driving habits. Geco⁴⁸ is a similar service available for Android, iOS and Blackberry. The app uses the smartphone's sensors to analyze the user's driving style and to provide feedback. The users can compare their results with the community.

6.2 Safety

Walkly⁴⁹ (previously known as WalkSafe) was founded by a group of Computer Science students and will be available for Android, iOS and Windows. Its goal is to create a Global Safety Network where users look out for one another. Users can notify their safety networks about their walks and arrival times. If a user fails in arriving at the specified time and location or if she or he sends out a distress signal, the safety network is notified automatically and can prompt further actions. Life 360⁵⁰ is a location sharing application available

⁴⁶<https://play.google.com/store/apps/details?id=com.xing.me&hl=de>

⁴⁷<http://www.bmw.de/de/footer/publications-links/technology-guide/eco-pro-analyser.html>

⁴⁸<http://geco-drive.fr>

⁴⁹<http://www.walklyapp.com>

⁵⁰<https://www.life360.com>

for Android, iOS and Windows. It allows to communicate with and to share locations of users within a community, for example, family members. Additional features are a 24/7 live advisor, a roadside assistance and a non-smartphone tracking functionality which allows to track phones without GPS sensors.

7. Others

A few mobility services we found do not fit in one of the presented categories such as games related to mobility. In this section, we present services that entertain people and that allow to track movements.

7.1 Entertainment

In 2010, Yahoo launched the two-month citywide challenge Yahoo! Bus Stop Derby⁵¹ turning bus stops into social gaming hubs. 20 bus stops were equipped with interactive touch screens. People waiting for their bus could chose between four games and challenge other users at other bus stops. The goal was to make public transport more fun and to reduce boredom while waiting for a bus.

7.2 Tracking

Glympse⁵² (Figure 5) (Android, iOS, Windows), Track⁵³ (iOS) and RouteShare⁵⁴ (iOS) are examples for tracking applications which allow the user to share her or his route or current location. They track the user's movement using the device's GPS sensor. Users can share their locations dynamically with selected contacts. The contacts then are able to follow the user via web browser or mobile application. Many applications can also provide an ETA.

8. Conclusion

In this paper, we introduced a new classification for collaborative and social mobility services. The presented classification and the examples of research projects and released applications underline the importance of innovative use cases to promote smart mobility. Many mobility services support users in solving one or more tasks in their everyday mobility. The success of some of these services depends on a high number of users. Hence, incentives are necessary to motivate people to use such services. First mobility services introduced Gamification methods to reach this goal but future work should also examine alternative methods to increase the number of users and to bind customers.

Our findings show that combining well-known mobility use cases or extending them by own ideas is a promising approach to develop new and innovative mobility services applications. Thus, future work in the field of collaborative and social mobility use cases should combine the ideas and

⁵¹<http://www.busstopderby.com>

⁵²<https://www.glympse.com>

⁵³<https://track.gs>

⁵⁴<http://www.routeshare.com>

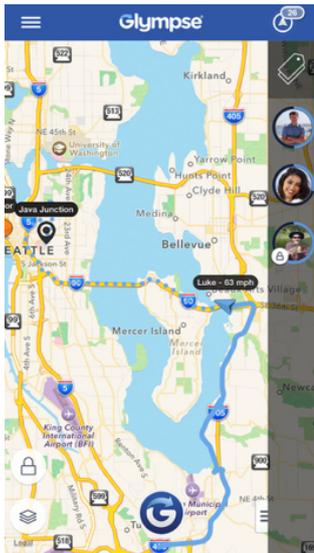


Figure 5. Glympse Android Application

strengths of existing mobility services to overcome limitations of single applications. One suggestion is the implementation of personalized recommendations into established mobility services such as POI finders to reduce the user effort and to improve the outcome of the service. Our classification and the presented example applications serve as a basis for this purpose.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] United Nations, Department of Economic and Social Affairs, Population Division. World Urbanization Prospects: The 2014 Revision, (ST/ESA/SER.A/366), 2015.
- [2] TNS Opinion & Social. Special Eurobarometer 406: Attitudes of Europeans towards urban mobility, December 2013.
- [3] Barbara Lenz, Claudia Nobis, Katja Köhler, Markus Mehlin, Robert Follmer, Dana Gruschwitz, Birgit Jesske, and Sylvia Quandt. *Mobilität in Deutschland 2008*. 2010.
- [4] Francesco Ricci. Mobile recommender systems. *Information Technology & Tourism*, 12(3):205–231, 2010.
- [5] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 191–226. Springer US, Boston, MA, second edition, 2015.
- [6] Kelvin Cheng, Liang He, Xiaojun Meng, David A. Shamma, Dung Nguyen, and Anbarasan Thangapalam. Cozymaps: Real-time collaboration on a shared map with multiple displays. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, pages 46–51, New York, NY, USA, 2015. ACM.
- [7] Harald Holone, Gunnar Misund, Håkon Tolsby, and Steinar Kristoffersen. Aspects of personal navigation with collaborative user feedback. In *Proceedings of the 5th Nordic Conference on Human-computer Interaction: Building Bridges*, NordiCHI '08, pages 182–191, New York, NY, USA, 2008. ACM.
- [8] Wolfgang Wörndl and Alexander Hefe. Generating paths through discovered places-of-interests for city trip planning. In Alessandro Inversini and Roland Schegg, editors, *Information and Communication Technologies in Tourism 2016: Proceedings of the International Conference in Bilbao, Spain, February 2-5, 2016*, pages 441–453. Springer International Publishing, Cham, 2016.
- [9] Sarfraz Nawaz, Christos Efstratiou, and Cecilia Mascolo. Parksense: A smartphone based sensing system for on-street parking. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, MobiCom '13, pages 75–86, New York, NY, USA, 2013. ACM.
- [10] Anandathirtha Nandugudi, Taeyeon Ki, Carl Nuessle, and Geoffrey Challen. Pocketparker: Pocketsourcing parking lot availability. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 963–973, New York, NY, USA, 2014. ACM.
- [11] Suhas Mathur, Tong Jin, Nikhil Kasturirangan, Janani Chandrasekaran, Wenzhi Xue, Marco Gruteser, and Wade Trappe. Parknet: Drive-by sensing of road-side parking statistics. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 123–136, New York, NY, USA, 2010. ACM.
- [12] Bundeskartellamt. Market Transparency Unit for Fuels, 2015. Retrieved May 20, 2016 from http://www.bundeskartellamt.de/EN/Economicsectors/MineralOil/MTU-Fuels/mtufuels_node.html.
- [13] Maike Gossen and Gerd Scholl. Latest trends in car-sharing. *Institute for Ecological Economy Research (IÖW)*, pages 1–5, 2011.
- [14] Catherine Morency. The ambivalence of ridesharing. *Transportation*, 34(2):239–253, 2007.
- [15] Catherine Clifford. How BlaBlaCar Is Different From Uber, 2015. Retrieved May 20, 2016 from <https://www.entrepreneur.com/article/250420>.

- [16] Tom Goodwin. The Battle Is For The Customer Interface, 2015. Retrieved May 20, 2016 from <http://techcrunch.com/2015/03/03/in-the-age-of-disintermediation-the-battle-is-all-for-the-customer-interface/>.
- [17] Anind K. Dey, Gregory D. Abowd, and Daniel Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.*, 16(2):97–166, December 2001.
- [18] Kenta Oku, Shinsuke Nakajima, Jun Miyazaki, and Shunsuke Uemura. Context-aware svm for context-dependent information recommendation. In *Proceedings of the 7th International Conference on Mobile Data Management*, MDM '06, page 109, Washington, DC, USA, 2006. IEEE Computer Society.
- [19] Mark Setten, Stanislav Pokraev, and Johan Koolwaaij. Context-aware recommendations in the mobile tourist application compass. In Paul M Bra and Wolfgang Nejdl, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 3137 of *Lecture Notes in Computer Science*, chapter 27, pages 235–244. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [20] Francesco Ricci and Quang Nhat Nguyen. Mobyrek: A conversational recommender system for on-the-move travellers. In D. R. Fesenmaier, H. Werthner, and K. W. Wober, editors, *Recommendation Systems: Behavioural Foundations and Applications*. CABI Publishing, 2006.
- [21] Olga Averjanova, Francesco Ricci, and Quang Nhat Nguyen. Map-based interaction with a conversational mobile recommender system. In *Proceedings of the 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, UBIKOMM '08, pages 212–218, Washington, DC, USA, 2008. IEEE Computer Society.
- [22] Wolfgang Woerndl, Christian Schueller, and Rolf Wöjtech. A hybrid recommender system for context-aware recommendations of mobile applications. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, ICDEW '07, pages 871–878, Washington, DC, USA, 2007. IEEE Computer Society.
- [23] Matthias Braunhofer, Mehdi Elahi, Mouzhi Ge, Francesco Ricci, and Thomas Schievenin. STS: design of weather-aware mobile recommender systems in tourism. In *Proceedings of the 1st Workshop on AI*HCI: Intelligent User Interfaces (AI*HCI 2013)*, 2013.
- [24] Gytis Tumas and Francesco Ricci. Personalized mobile city transport advisory system. In Wolfram Höpken, Ulrike Gretzel, and Rob Law, editors, *Information and Communication Technologies in Tourism 2009*, pages 173–183. Springer Vienna, Vienna, 2009.
- [25] Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci. Context relevance assessment and exploitation in mobile recommender systems. *Personal Ubiquitous Comput.*, 16(5):507–526, June 2012.
- [26] Federica Cena, Luca Console, Cristina Gena, Anna Goy, Guido Levi, Sonia Modeo, and Ilaria Torre. Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide. *AI Commun.*, 19(4):369–384, December 2006.
- [27] Barry Brown, Matthew Chalmers, Marek Bell, Malcolm Hall, Ian MacColl, and Paul Rudman. Sharing the square: Collaborative leisure in the city streets. In *Proceedings of the Ninth Conference on European Conference on Computer Supported Cooperative Work*, ECSCW'05, pages 427–447, New York, NY, USA, 2005. Springer-Verlag New York, Inc.
- [28] Angel García-Crespo, Javier Chamizo, Ismael Rivera, Myriam Mencke, Ricardo Colomo-Palacios, and Juan Miguel Gómez-Berbís. Speta: Social pervasive e-tourism advisor. *Telemat. Inf.*, 26(3):306–315, August 2009.
- [29] Norma Saiph Savage, Maciej Baranski, Norma Elva Chavez, and Tobias Höllerer. I'm feeling loco: A location based context aware recommendation system. In Georg Gartner and Felix Ortig, editors, *Advances in Location-Based Services: 8th International Symposium on Location-Based Services, Vienna 2011*, pages 37–54. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [30] Einat Minkov, Ben Charrow, Jonathan Ledlie, Seth Teller, and Tommi Jaakkola. Collaborative future event recommendation. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 819–828, New York, NY, USA, 2010. ACM.
- [31] Daniel Herzog and Wolfgang Wörndl. Extending content-boosted collaborative filtering for context-aware, mobile event recommendations. In *Proceedings of the 12th International Conference on Web Information Systems and Technologies*, pages 293–303, 2016.
- [32] V. C. Magana and M. Munoz-Organero. Gafu: Using a gamification tool to save fuel. *IEEE Intelligent Transportation Systems Magazine*, 7(2):58–70, Summer 2015.

An Integration Platform for Temporal Geospatial Data

Andreas Kipf and Alfons Kemper

Department of Informatics, Technical University of Munich, Munich
{kipf, kemper}@in.tum.de

Abstract

With the emergence of connected vehicles, data is increasingly annotated with temporal and geospatial attributes. At the same time, data is being produced at ever higher rates, taking state-of-the-art data processing systems to their limits. The aim of this sub project is to extend high-performance main-memory database systems, such as HyPer, with temporal and geospatial processing capabilities to tackle emerging mobility workloads.

Keywords

Streaming; Geospatial Data Processing

Introduction

Gartner recently forecasted that there will be more than 20 billion connected devices in use in 2020, a 400% increase compared to this year [1]. Driven by this trend and the enormous amount of data that these devices generate, many dedicated stream processing engines have been developed in recent years [2, 3, 4]. Nowadays, companies often create proprietary solutions to address workloads that these systems cannot handle out of the box, such as when high-throughput data streams need to be joined with traditional business data commonly stored in relational database systems. For example, in the context of connected vehicles, car sharing companies might want to compute the cost of rentals in real time in order to send notifications to customers when price limits set in user profiles are exceeded. These custom solutions are not only error-prone, but also highly inefficient since data is often transferred to distant systems or even joined at the application layer.

The goal of this sub project is to create the first *all-in-one* temporal and geospatial data processing system that holistically addresses emerging mobility workloads while outperforming combinations of dedicated systems. There are two important characteristics of connected vehicle data that we will address in this work: 1) It is continuously being produced. Thus, to allow for real-time analytics, it has to be processed in a stream-like fashion. Additionally, the data processing engine has to provide an efficient access to historical data to be joined with real-time data streams. 2) It comes with temporal and geospatial attributes. Thus, to allow for real-time insights, the data processing engine has to specifically optimize for these data types.

To achieve this goal, we will build on the main-memory database system HyPer [5]¹ developed at the Chair of Database Systems at the Technical University of Munich. It achieves

¹When saying HyPer, we are referring to the research version of HyPer developed at the Technical University of Munich.

an outstanding performance for both OLTP and OLAP workloads, even when they operate simultaneously on the same database. HyPer uses two different snapshotting mechanisms to avoid expensive synchronization. By leveraging the *copy on write* feature of the MMU, the *fork* mechanism [5] efficiently creates consistent copies of the database to enable analytical queries to run without interruptions. The second snapshotting mechanism [6] is based on *multi version concurrency control* (MVCC) and isolates transactions by versioning individual attributes. It further features data-centric LLVM code generation with just-in-time compilation. Finally, HyPer has an advanced dynamic programming-based optimizer including the ability to unnest arbitrary queries.

Streaming

To allow for an efficient processing of streaming data, we plan to integrate streams directly into HyPer's kernel and represent them as relational operators. This will allow us to compute optimized query plans that consider the characteristics of streams such as their data rate. Further, we can compute the selectivity of filters over time and thus derive better query plans at runtime. Additionally, we will allow users to specify data freshness (i.e., what tuples to consider in a joined table) and overall response time constraints (i.e., the rate at which it should produce results) on continuous queries. We will use this information to guide the database kernel in order to efficiently process continuous queries without wasting resources.

Related Work

Michael Stonebreaker, who recently received the prestigious Turing Award, once identified eight rules [7] that real-time stream processing engines should follow, including the support for SQL as a query language and the integration of stored and streamed data. However, up to today, these requirements

have not been fully addressed.

Apache Storm [2], Apache Spark Streaming [3], and Apache Flink [4] are widely used stream data processing systems, however, to further analyze streaming results and to combine streams with transactional data, they require users to ship the data to an external system.

Storm is a *tuple-at-a-time* low-latency stream processor that does not ensure state consistency. Storm keeps upstream backups of data and replays them if no acknowledgements from downstream nodes have been received (at least-once semantics). Trident [8] extends Storm with exactly-once semantics and consistent state support.

Spark Streaming extends Spark [9] with stream processing capabilities. It follows a *micro batching* approach, thus allowing users to use the same programming model as for Spark. In contrast to Storm, Spark Streaming is optimized for throughput.

Flink combines the best of both worlds by offering *tuple-at-a-time* processing semantics while employing a batch-based checkpointing mechanism that allows for superior throughput compared to Storm.

MemSQL [10] is a main-memory database system, which, similar to HyPer, compiles queries into native machine code at runtime. MemSQL itself does not specifically address streaming workloads, however, it offers a Spark connector (MemSQL Streamliner [11]), thereby easing the integration between MemSQL and Spark. Yet the two systems remain separated, which means that streams cannot be joined with data in the relational database system without materializing and transferring the streaming results.

PipelineDB [12] extends the database system PostgreSQL by integrating continuous queries into the database kernel. It introduces the concept of continuous views, an extension to the SQL standard, that allows users to continuously compute aggregates over streams. Users can insert stream tuples into a stream table in the same way as inserting tuples into a regular database table. The difference is that stream tuples will not be permanently stored and will only be used to update aggregates. The SQL code for the example of computing the average speed on a road segment within the last minute looks as follows:

```
CREATE CONTINUOUS VIEW events
WITH (max_age = '1 minute') AS
SELECT AVG(speed) FROM events_stream
WHERE coordinates = ...
```

However, it cannot simultaneously handle streaming and traditional OLTP and OLAP workloads as it lacks efficient snapshotting mechanisms. Additionally, it cannot reevaluate query plans at runtime and follows the inefficient interpreter approach.

The research prototype AIM [13] allows one to efficiently process high-throughput input streams and repeatedly update a materialized view using a specialized storage layout. In parallel to the stream processing, these views can continuously be joined with dimension tables. Isolation is guaranteed using

a delta-based approach, which is an alternative to HyPer's snapshotting mechanisms. The drawback of the AIM system is that it does not address updates on the dimension tables. Further, it is not a complete database system and is limited to a specific use case in the telecommunications domain.

[14] suggests to combine stream with transaction processing, since stream processing applications often require transaction guarantees such as consistency and isolation. Their proposed system called S-Store is based on a main-memory OLTP engine and integrates additional streaming functionality. However, S-Store does not specifically address analytical workloads, as it lacks efficient snapshotting mechanisms.

Objectives

First of all, we need to be able to efficiently ingest streaming data into the system. To achieve that, we plan to make use of low-latency InfiniBand technologies and user-space networking libraries such as MTCP².

Often it is not enough to analyze streaming data on its own. Instead, users want to combine it with existing data stored in tables or external files. As a first objective, we will integrate streaming functionality into the main-memory database system HyPer that already addresses OLTP and OLAP workloads and can ingest data stored in files at wire speed [15]. In particular, we will integrate the concept of continuous queries that specifically addresses streaming workloads in contrast to regular OLAP queries that would need to be reexecuted for each new batch of incoming tuples.

To join stream tuples with tuples in tables, there are essentially two approaches: 1) When a stream tuple arrives, we will join it with all the data that was present when the stream tuple arrived. This approach is only feasible with the attribute-based MVCC snapshotting technique, since forking the entire database for each incoming tuple or small batches of incoming tuples would be very expensive. 2) We will regularly create consistent snapshots of the database and join small batches of incoming tuples against the current snapshot. In that case, incoming stream tuples might not be joined with the most current data, however, this approach yields maximum throughput. The decision of which of the two approaches should be taken requires trading off between latency and throughput and depends on the workload's requirements. To inform this decision, we will integrate two new types of constraints into SQL: 1) Data freshness constraints will allow users to define how often individual pipelines in the query plan of a continuous query need to be rematerialized. In case a user wants maximum data freshness, we could employ a push-based mechanism. Whenever data in a base table is updated (e.g., triggered by a transaction), we will recompute all affected pipelines and update corresponding data structures such as hash tables. 2) Overall runtime (latency) constraints will allow one to specify the rate at which a query should produce results.

Besides guiding the snapshotting decision, both types of

²<https://github.com/eunyoung14/mtcp>

constraints will allow us to improve the resource efficiency of continuous queries. For example, we can measure the runtime of a query and in case the runtime easily satisfies the latency constraint, we can increase the batch size in which incoming tuples are processed.

A major objective is to tackle the problem of how often a query plan of continuous queries needs to be reevaluated based on the runtime properties of continuous queries such as the selectivity of filters. It is well-known that particularly join ordering can highly influence the runtime of a query. Since the query optimizer works on statistics (e.g., cardinality estimates), it cannot find the perfect plan upfront. Additionally, the workload may change over time and thus the query plan needs to be recomputed to exploit the current workload characteristics. For example, the number of groups in aggregations (which are not available upfront) may allow us to use smaller and thus more efficient hash tables. Thus, a runtime monitoring of continuous queries will allow us to iteratively compute better query plans.

When a continuous query first arrives, we need to evaluate whether there are multiple query plans and whether the stream influences these plans at all. Since queries will be compiled at runtime, there will be a compilation overhead, which needs to be considered. One might argue that the compilation overhead is amortized soon. However, this heavily depends on the variance of the workload characteristics. Another challenge is to migrate the current state (e.g., selectivity of filters, number of groups in hash tables) between the old and the new query plan.

Along these lines, a corresponding cost function needs to consider multiple aspects such as compilation time and state migration costs, expected performance and efficiency gains, and the variance of the selectivity of filters, which makes it a difficult optimization problem that has not been addressed yet.

Streaming data often needs to be aggregated over certain windows (e.g., average speed on a road segment within the last minute). To maintain these windowed aggregates with maximum efficiency, we plan to compile aggregation functions into efficient LLVM code at runtime.

Finally, we plan to answer the question of how to efficiently scale that system to multiple nodes. This question is twofold: 1) As the system should be able to handle a large number of streams, the processing capabilities of one node may be exceeded and thus streams need to be assigned to different nodes. This leads to the question of how to efficiently join streams being processed by different nodes. Instead of allowing window-based joins on the raw stream tuples (as covered in [16, 17, 18]), we will focus on joining the results of the stream processing with the results of other streams or tables. In particular, we will use InfiniBand RDMA operations to efficiently push new results to distant nodes. 2) To be able to process a stream with a data rate that cannot be handled by a single node, the stream needs to be partitioned across multiple nodes. When such a stream is joined with

one or multiple tables, the distribution scheme of individual tables (e.g., hash-partitioned by a certain column) needs to be considered by the query plan. For example, in case a stream is joined with two tables, it would be beneficial to partition the stream based on the same attribute as the table with the highest cardinality estimate in order to minimize network traffic. Additionally, data freshness constraints on individual tables need to be taken into account when choosing the partition attribute.

Summarizing, we will tackle the following research questions:

- How can streaming data be efficiently ingested into a main-memory database system?
- How can continuous queries be integrated into a main-memory database system while considering data freshness and latency constraints?
- How can continuous queries be iteratively optimized at runtime?
- How can windowed aggregations be performed with maximum efficiency?
- How can that system be scaled to multiple nodes using InfiniBand technologies?

Additionally, we plan to address the question of how that system can be combined with in-situ data processing on external files. For example, files could be reorganized based on the access patterns of continuous queries to speed up subsequent accesses.

The expected contributions will include *answers to all of these research questions* and a *prototype* that tightly integrates stream processing into a high-performance main-memory database system and achieves maximum performance for combined workloads.

Geospatial Data Processing

In the context of connected vehicles, data often contains geo locations. Traffic analysis/monitoring systems make use of this data, e.g., to optimize traffic flows in real time or to collect statistics on traffic peaks. Due to the complex nature of computations on geospatial data types (e.g., Point, Polygon) and the latency requirements in real-time scenarios, data processing engines have to be tuned for these workloads. A dominant use case for traffic analysis/monitoring systems is the aggregation of the vehicles' data (e.g., their speed) based on their geospatial location (e.g., the city area that they are currently in), which essentially boils down to a join between multiple polygons (the areas) and points (the vehicles' locations). To goal is to optimize this join using specialized index structures and by offloading computations to the GPU.

The challenge is to integrate geospatial data processing into a system that compiles queries into efficient machine code. Similar to our planned streaming extensions, we will base

our geospatial data processing efforts on the main-memory database system HyPer.

Related Work

There has been a variety of related work in this subject area. We will focus on joins between multiple polygons and points and corresponding indexing schemes as well as efforts to speed up geospatial data processing using GPUs.

[19] provides an overview over the basic approaches of indexing geospatial objects. A well-known indexing scheme for geospatial objects is the R tree. The R tree organizes points/polygons into non-disjoint bounding boxes. The drawback of this approach is that it leads to a non-disjoint decomposition of space, thus leading to a search of the entire space in the worst case. Another approach of indexing geospatial objects is using disjoint cells. The drawback of this approach is that objects may be reported multiple times. There are different variants of this approach:

uniform grid All cells have the same size. The drawback is the possibility of many sparse cells.

adaptive grid Cells may have different sizes. Quad tree data structures and the Google S2 library³ implement this variant.

partitions at arbitrary positions Cells do not follow a regular decomposition scheme. The R+ tree implements this variant.

In the Google S2 library, points are linearized using the Hilbert space-filling curve and mapped to 64bit unsigned integers, thus allowing for efficient contains checks using bitwise operations. This is possible since smaller cells share common prefixes with parent cells. We believe that the S2 library may yield very efficient geospatial join implementations.

GPUdb [20] and MapD [21] are two examples of geospatial processing engines on GPUs. Besides being able to process vast amounts of data within milliseconds, they also claim that they can visualize results faster than others, since the results already reside on the GPU being used to render the display output. However, rendering is only an argument when data processing and visualization happen on the same machine, which is often not the case, especially in enterprise environments where costly GPUs are usually only found in servers.

In addition to these GPU only solutions, there are multiple approaches of speeding up certain computationally-intensive geospatial computations by offloading them to GPUs at query runtime. [22] claims to achieve a 62-240x overall speedup over the CPU counterparts. According to them, the transfer cost between the main memory and the GPU's memory is amortized over the query execution time in most cases.

³<https://code.google.com/archive/p/s2-geometry-library/>

Objectives

First of all, we plan to integrate geospatial data processing capabilities into a main-memory database system that compiles queries at runtime. One objective is to utilize the functionality of the Google S2 library to speed up geospatial joins. Another objective is to leverage the compute power of GPUs to speed up geospatial data processing.

Summarizing, we will tackle the following research questions:

- How can geospatial data types/joins be integrated into a main-memory database system that compiles queries at runtime?
- How can joins between multiple polygons and points be optimized?
- How can geospatial computations be accelerated using GPUs?

The expected contributions will include *answers to all of these research questions* and a *prototype* that tightly integrates geospatial data processing into a high-performance main-memory database system.

The overall goal is to eventually integrate both the stream and the geospatial data processing extensions into a single prototype to fulfill our vision of an *all-in-one* temporal and geospatial data processing system for emerging mobility workloads.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

References

- [1] Gartner says 6.4 billion connected "things" will be in use in 2016, up 30 percent from 2015. <http://www.gartner.com/newsroom/id/3165317>.
- [2] Apache storm. <http://storm.apache.org/>.
- [3] Apache spark streaming. <http://spark.apache.org/streaming/>.
- [4] Apache flink. <http://flink.apache.org/>.
- [5] Alfons Kemper and Thomas Neumann. HyPer: A Hybrid OLTP & OLAP Main Memory Database System Based on Virtual Memory Snapshots. In *ICDE*, pages 195–206, April 2011.
- [6] Thomas Neumann, Tobias Mühlbauer, and Alfons Kemper. Fast Serializable Multi-Version Concurrency Control for Main-Memory Database Systems. In *SIGMOD, SIGMOD '15*, pages 677–689, New York, NY, USA, 2015. ACM.

- [7] Michael Stonebraker, Ugur Cetintemel, and Stan Zdonik. The 8 requirements of real-time stream processing. *ACM SIGMOD Record*, 34(4):42–47, 2005.
- [8] Apache storm trident. <http://storm.apache.org/documentation/Trident-state>.
- [9] Apache spark. <http://spark.apache.org/>.
- [10] Rajkumar Sen, Jack Chen, and Nika Jimsheleishvili. Query Optimization Time: The New Bottleneck in Real-time Analytics. In *IMDM*, *IMDM '15*, pages 8:1–8:6, New York, NY, USA, 2015. ACM.
- [11] Memsql streamliner. <http://blog.memsql.com/spark-streamliner/>.
- [12] Pipelinedb. <https://www.pipelinedb.com/>.
- [13] Lucas Braun, Thomas Etter, Georgios Gasparis, Martin Kaufmann, Donald Kossmann, Daniel Widmer, Aharon Avitzur, Anthony Iliopoulos, Eliezer Levy, and Ning Liang. Analytics in motion: High performance event-processing and real-time analytics in the same database. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 251–264. ACM, 2015.
- [14] John Meehan, Nesime Tatbul, Stan Zdonik, Cansu Aslantas, Ugur Cetintemel, Jiang Du, Tim Kraska, Samuel Madden, Andrew Pavlo, Michael Stonebraker, et al. S-store: Streaming meets transaction processing. *arXiv preprint arXiv:1503.01143*, 2015.
- [15] Tobias Mühlbauer, Wolf Rödiger, Robert Seilbeck, Angelika Reiser, Alfons Kemper, and Thomas Neumann. Instant loading for main memory databases. *Proceedings of the VLDB Endowment*, 6(14):1702–1713, 2013.
- [16] Buğra Gedik, Rajesh R Bordawekar, and S Yu Philip. Celljoin: a parallel stream join operator for the cell processor. *The VLDB journal*, 18(2):501–519, 2009.
- [17] Jens Teubner and Rene Mueller. How soccer players would do stream joins. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 625–636. ACM, 2011.
- [18] Rajagopal Ananthanarayanan, Venkatesh Basker, Sumit Das, Ashish Gupta, Haifeng Jiang, Tianhao Qiu, Alexey Reznichenko, Deomid Ryabkov, Manpreet Singh, and Shivakumar Venkataraman. Photon: Fault-tolerant and scalable joining of continuous data streams. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 577–588. ACM, 2013.
- [19] Hanan Samet. *Sorting in space*. 2008.
- [20] Gpudb. <http://www.gpudb.com/>.
- [21] Mapd. <http://www.mapd.com/>.
- [22] Bogdan Simion, Suprio Ray, and Angela Demke Brown. Speeding up spatial database query execution using gpus. *Procedia Computer Science*, 9:1870–1879, 2012.

Big Geospatial Data Exploration

Varun Pandey and Alfons Kemper

Department of Informatics, Technical University of Munich, Munich
{pandey, kemper}@in.tum.de

Abstract

In the past few years, massive amounts of location-based data has been captured. Numerous datasets containing user location information are readily available to the public. Analyzing such datasets can lead to fascinating insights into the mobility patterns and behaviors of users. Moreover, in recent times a number of geospatial data-driven companies like Uber, Lyft, and Foursquare have emerged. Real-time analysis of geospatial data is essential and enables an emerging class of applications. There has been a rapid advancement in research areas such as machine learning and data mining, which can be attributed to the growth in the database industry and advances in data analysis research. This has resulted in a need for systems that can extract useful information and knowledge from data. Data scientists use various data mining tools on top of databases for this purpose. To achieve lower latencies and minimize transmission costs between the database and external tools, it is necessary to move computation closer to the data. The current trend in database research is to integrate these various analytical functionalities that are useful for knowledge discovery into the database kernel. The goal is to have a full-fledged general-purpose database that allows big data analysis along with conventional transaction processing. Our aim is to integrate analytical functionalities for geospatial data into a main memory database system to facilitate big data analysis. In this report we carry out a survey of available state-of-the-art algorithms and tools for geospatial analytics on big data.

Keywords

Geospatial data mining; Big data

Introduction

The most important goal of data exploration is to extract knowledge and make meaningful inferences [1]. Visualization is a powerful and intuitive way which helps in knowledge discovery and data mining [2]. Data exploration and visualization have become a major research area in the era of *Big data*. In 2007 Jim Gray, a pioneer in database industry, coined the term *Fourth Paradigm* which represents modern day data intensive scientific discovery [3]. He suggested to tackle *Big data* there is a need for a set of tools and technology that help in data visualization and exploration. Since then there has been plethora of research on such tools and technologies [4, 2, 5, 6, 7, 8, 9, 10, 11].

Big data is characterized by the 3V's: *Volume*, *Variety* and *Velocity*. The 3V's were first coined by Doug Laney in 2001 [12]. At that time the 3V's were not used to define or characterize *Big data*, but major enterprises such as McAfee, Microsoft, Gartner and Intel still used 3V's to define it in the following 15 years [13, 14, 15, 16]. In the 3V's model:

Volume refers to the size of data in magnitudes of terrabytes, petabytes and exabytes. Every day about 2.5 exabytes of data is created and this number doubles every 40 months [13].

Variety refers to the different sources of data such as messages, status updates and images on social networks,

readings from remote sensors, GPS signals from cell phones etc. The data can structured, semistructured or unstructured.

Velocity refers to speed at which the data is generated which can be real-time or nearly real-time. To utilize the commercial value of the data, it has to be processed and analyzed in a timely manner i.e. in real-time.

A report by the McKinsey Global Institute [17] called *Big Data* as the next frontier for innovation, competition and productivity. They researched *Big data* in 5 domains which will drive the global economy and generate value in each:

Healthcare in US If healthcare *Big data* is used creatively then the sector could create more than \$300 billion in value every year.

Public Sector in Europe In the developed economies of Europe, government administrators could save more than \$149 billion in operational efficiency improvements alone by using *Big data*, not including using *Big data* to reduce fraud and errors and boost the collection of tax revenues.

Retail in US a retailer using big data to the full could increase its operating margin by more than 60 percent.

Manufacturing are using data obtained from sensors embedded in products to create innovative after-sales service offerings such as proactive maintenance

Global Personal Location Data services that are enabled by personal-location data can allow consumers to capture \$600 billion in economic surplus.

Since then there has been a rapid advancement in research areas such as machine learning and data mining, which can be attributed to the growth in the database industry and advances in data analysis research. This has resulted in a need for systems that can extract useful information and knowledge from data. Data scientists use various data mining tools on top of databases for this purpose. To achieve lower latencies and minimize transmission costs between the database and external tools, it is necessary to move computation closer to the data. The current trend in database research is to integrate these various analytical functionalities that are useful for knowledge discovery into the database kernel. The goal is to have a full-fledged general-purpose database that allows big data analysis along with conventional transaction processing.

Big geospatial data exploration is an interesting field. In the past few years, massive amounts of location based data have been captured. A number of datasets containing user location information are readily available to the public these days. Analyzing such datasets can lead to fascinating insights into the mobility patterns and behaviors of users. It can help in planning urban cities leading to smarter cities. New York City has recently published a taxi data set containing about 1.1 billion rides taken across the city since 2009. Storing and analyzing this huge amount of spatial data is essential for many applications and a key component to geographical information systems. As McKinsey report [17] highlights global personal location data has a great commercial value and timely analysis of it is necessary to utilize its full value. Companies like Uber, Lyft, and Foursquare have a need to create real-time applications, including alerting systems, that consider the most current state of their data, enabling *real world awareness*. The emergence of these data-driven applications have been enabled by the advent of the Internet of Things and the massive amounts of geotagged sensor data it generates. In addition, there has been a surge in location data generated from the web. Popular internet services like Facebook, Twitter, Instagram, FourSquare, and Google have utilized this growth. They allow their users to geotag their posts, and the use of this feature has led to an exponential growth in data containing location information. It has been estimated that about 15 percent of the tweets per day are geotagged, which approximates to about 70 million posts per day. Similar figures are estimated for other services as well.

It is essential to understand the difference between spatial data and geospatial data. Spatial data can be any data that represents space with a certain frame of reference. For example in our solar system, sun is the frame of reference. Geospatial data is the spatial data where the underlying frame of refer-

ence is the earth's surface. Since most of the applications today generate or need to process geographical data, we focus on geospatial data.

Thus the focus of this report is on the research trends and state-of-the art survey of geospatial data mining and exploration techniques for datasets containing vehicular datasets. We intend to propose a general purpose main-memory database systems (MMDB) that can answer regular geospatial queries and that also incorporates commonly used geospatial data mining algorithms in the database kernel. We also present our preliminary results by showcasing *HyperSpace* [18], a geospatial processing module, in a state-of-the art MMDB HyPer [19].

Geospatial Data Mining

Geospatial data mining is a sub discipline of data mining. It is concerned with knowledge discovery and pattern recognition in geospatial datasets. Geospatial data can be complex that involve points, linestrings and polygons. Querying such data involve various geometric computations which can be expensive. In [20], the authors identify that classical data mining algorithms perform poorly on geospatial datasets. Classical data mining algorithms make the assumption that everything is related to everything. They violate Tobler's first law of Geography [21] which states that nearby things are more related than distant things. Chawla et al. also suggest that for geospatial data, data mining tasks need to extended so as to tackle the challenges associated with such datasets.

Data mining is usually structured as top-down or bottom up. Top-down approach involves forming a hypothesis in the beginning and then testing the hypothesis on the dataset, revising it if the tests do not confirm the hypothesis. On the other hand, bottom-up approach does not involve any hypothesis. Bottom-up approach involves examining the data and then come up with patterns. Most of geospatial data mining involve bottom-up data mining approaches. The geospatial data mining techniques can be divided into 3 categories:

Clustering Clustering is one of the most popular mining technique for geospatial data. Clustering partitions data into subclasses that group similar objects together. We know from Tobler's first law of geography [21] that nearby things are more related than distant things, clustering generally groups neighboring entities together thus making it a popular choice for mining geospatial datasets. Clustering has wide range of applications. It can be used for flow detection [22, 23, 24, 25], hotspot detection [26, 27, 28] and predicting variation of passengers in hotspots [29].

Association Rule Mining Association Rule Mining involves findings associations in a dataset where an event X leads to event Y. In terms of geospatial datasets it means finding whether an event X leads to event Y in spatial neighborhoods. Application of association rule

mining involve traffic accidents analysis [30], crime detection [31], and point-of-interest detection [32].

Classification Classification in mining means predicting an output based on certain input. To be able to predict the outcome, the algorithm is trained with an initial dataset containing a set of attributes and respective outcome. The algorithm then tries to discover relationships between attributes to be able to make the prediction. The algorithm is then given an input dataset with same attributes except the prediction attribute and it produces a prediction. Application of classification for geospatial data involves land usage classification [33], activity recognition [34] and social event recognition [35].

Visualization Visualizing geospatial data and allowing the user to explore the data seems to be the most intuitive way. It might not lead to deep insights or patterns but it is intuitive and an interactive way to explore geospatial data. Current research trend suggest that visualization could be the answer for most, if not all, geospatial exploration needs. [11, 18] show promise but there is a long way to achieve true geospatial exploration using visualization alone.

Current Database Technology

There are publicly available datasets that can help in geospatial exploration. The New York City (NYC) Taxi Rides [36] dataset is a good example, but is only a sample of what is captured by the aforementioned companies. The dataset contains approximately 1.1 billion taxi rides taken in the city since 2009. This represents about 470,000 taxi rides everyday in one of the most densely populated cities in the world. Uber, a popular on demand car service available via a mobile application, has also made a subset of the taxi rides available for the cities of San Francisco and NYC. For NYC, Uber published data containing around 19 million rides for the periods from April to September 2014 and from January to June 2015 [37]. Ever since the datasets were published, there have been multiple static analyses on these datasets [38, 39, 37]. The authors of [11] present a comprehensive system built from scratch for storing, querying, and visual exploration of geospatial data using kd-trees. Their system takes two seconds to execute a query that returns 100,000 taxi trips, which is too slow to address real-time workloads. MemSQL has some real-time capabilities [40] and is one of the first main-memory database systems (MMDBs) to deeply integrate geospatial support. The current database systems do not offer the performance required by real-time applications for analytics, and companies are often forced to build their own solutions [41]. PostGIS [42] is a spatial database extension for the PostgreSQL object-relational database system. It was used along with R in [37] for analysis of NYC dataset, and the whole process took 3 days on a general purpose laptop.

There is a need for a general purpose MMDB that is fast enough to answer regular geospatial queries and that also in-

corporates commonly used geospatial data mining algorithms in the database kernel. By implementing *HyperSpace* [18] we take a step in that direction. *HyperSpace* is a geospatial processing module in HyPer [19]. HyPer belongs to an emerging class of hybrid databases, which enable *real world awareness* in real time by evaluating OLAP queries directly in the transactional database. In HyPer, OLAP is decoupled from mission-critical OLTP either by using the *copy on write* feature of the virtual memory management or *multi version concurrency control* [43]. These snapshotting mechanisms enable *HyPerSpace* to evaluate geospatial predicates on rapidly changing datasets. *HyperSpace* achieves much better performance compared to an open-source database PostgreSQL, a commercially available MMDB, and a successful key-value store. *HyPerSpace* showcases that an interactive analysis of huge amounts of rapidly changing geospatial data is possible.

HyperSpace

Similar to what PostGIS is to PostgreSQL, *HyPerSpace* is a geospatial extension to HyPer. For geospatial data processing in *HyPerSpace*, we make use of the Google S2 geometry library¹. This is not novel, since System B also uses the S2 library for evaluating geospatial predicates. The novelty of our system is the integration of geospatial functionalities into a high-performance MMDB with snapshotting mechanisms which makes it possible to evaluate geospatial predicates on rapidly changing datasets.

At the moment, we support the three geospatial datatypes `Point`, `LineString`, and `Polygon`. Most of the geospatial processing is done using the S2 library.

S2 decomposes the earth into a hierarchy of cells. It considers earth of radius 1, and encloses it in a cube that completely covers it. S2 projects a point on the earth's surface onto one of the cube's faces and finds the cell that contains it. The faces of the cube are the top level cells, which can be recursively divided into four children to obtain lower level cells. There are 30 levels in total, and cells at the same level cover equivalent areas on earth (e.g., level 30 cells cover approximately 1cm^2 each). The cells are enumerated using the Hilbert space-filling curve. The Hilbert curve is hierarchical in nature and fits well with the decomposition of earth into cells. Hilbert space-filling curves are fast to encode/decode and they have a very desirable spatial property: they preserve spatial locality. This means that the points on earth that are close to each other are also close on the Hilbert curve. The enumeration of the cells gives a compact representation of each cell in a 64 bit integer called *CellId*. A *CellId* thus uniquely identifies a cell in the cell decomposition. Similarly, other spatial datatypes like `LineString` and `Polygon` can be approximated using cells.

The enumeration of cells in S2 is hierarchical, which means that a parent cell shares its prefix with its children. To check if a cell is contained in another, we simply need to

¹<https://code.google.com/archive/p/s2-geometry-library/>

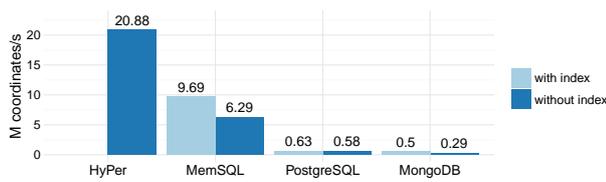


Figure 1. *HyPerSpace* vs. related systems: throughput of *ST_Covers* using lat/long co-ordinates

compare their prefixes, which is a bit operation. This enables one to index points based on their *CellIds* and thus be able to retrieve points contained in a certain cell by performing a prefix lookup on the index. B tree data structures are a good choice to index *CellIds*, since they support fast prefix lookups (essentially range scans). Additionally, B trees allow for high update rates, which is an essential requirement for real-time workloads.

For evaluation, we used the NYC Taxi Rides dataset consisting of approximately 1.1 billion rides taken in the city from January 2009 until June 2015. The dataset includes the pickup and dropoff locations (latitudes and longitudes), pickup and dropoff times, and various details about the trip, such as distance, payment type, number of passengers, various taxes, tolls, surcharge, tip amount, and total fare. For privacy reasons, it does not contain details about drivers or passengers. The exact route taken for the trip is also not available. We needed to clean the dataset as some of the pickup or dropoff locations did not make sense as they were way outside NYC. We cleaned such records from the dataset and only considered rides that originated between longitude values -70.00 and -80.00, and latitude values 35.00 and 45.00. For evaluation, we made use of the taxi data for the month of January 2015. The cleaned dataset for January 2015 contains a total of 12505344 records.

All experiments were run *single threaded* on an Ubuntu 15.04 machine with an Intel Xeon E5-2660 v2 CPU (2.20 GHz, 3.00 GHz maximum turbo boost) and 256 GB DDR3 RAM and all reported performance results are averages over ten runs.

We compared *HyPerSpace* with the following related systems: System A, System B, and PostgreSQL 9.4.5 (postgis-2.2.0). Since PostgreSQL does not support intra-query parallelism, we configured all systems to run single threaded. For evaluation purposes, we find how many rides originated from Midtown Manhattan in January 2015. In SQL notation, the following query is issued:

```
select count(*)
from nyc,pickups_jan_2015
where ST_Covers(nyc.geog,pickups_jan_2015.geog)
      and borough='Manhattan'
      and neighborhood='Midtown';
```

With the exception of System B, with NoSQL syntax, the query looks similar on all systems.

Figure 1 shows the throughput of the *ST_Covers* predicate for all of the systems. System A, System B, and PostgreSQL

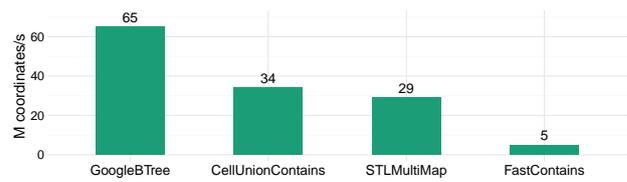


Figure 2. Microbenchmark results: throughput of *ST_Covers* using lat/long co-ordinates

achieve better performance when using appropriate index structures. Particularly System B, which also makes use of the Google S2 geometry library, benefits from its index on points. System B's index is basically a B tree on the 64bit *CellIds*. System B computes an exterior covering of the polygon using the S2 library. That covering consists of cells at various levels (i.e., of different sizes). For each cell of this covering, it then performs a prefix lookup in the B tree (essentially a range scan) and evaluates qualifying points for actual containment in the polygon. System B suffers heavily from its document-based storage layout, since it needs to parse GeoJSON documents at runtime.

HyPerSpace completes the query in 550ms and thus it achieves more than twice the performance of its closest competitor, which is System A with an index on points (1290ms). We have not evaluated *HyPerSpace* with an index on points yet, but ran multiple microbenchmarks outside of *HyPerSpace*. All microbenchmarks were implemented in C++11 and compiled with gcc 4.9.2 with `-O3` and `-march=native` settings. We compared the implementation *CellUnionContains* that we used in *HyPerSpace* as well as *FastContains*, which is a modified version of the *S2Loop.Contains* implementation that skips the initial bounding box check, to the two index-based implementations *GoogleBTree* and *STLMultiMap*.

Figure 2 shows the throughput of the *ST_Covers* predicate for the different implementations. *GoogleBTree*, which is an implementation similar to System B's index, completes the workload in 191ms. In the *GoogleBTree* implementation, we first compute exterior and interior coverings for the given polygon and then perform a range scan in a Google B tree² for each cell of the exterior covering. For each qualifying point, we check whether the point is contained in the interior covering, which is essentially a binary search on a sorted vector of *CellIds*. Only if a point qualifies the exterior, but not the interior covering, an exact containment check using our modified implementation of the *S2Loop.Contains* function needs to be performed. The other index-based implementation *STLMultiMap* takes twice as long (425ms) as *GoogleBTree* to complete the workload, even though it uses the same approach. In C++11, the `stl::multi_map` interface that we used in this case is implemented by a RB tree, which is less efficient for range scans. It is well known that a B+ tree would yield even higher rates for range scans than a B tree. However, for the sake of expediency and reproducibility of our measure-

²<https://code.google.com/archive/p/cpp-btree/>

ments, we have used the B tree implementation provided by Google instead of a custom B+ tree implementation. Once we integrate this approach into *HyPerSpace*, we will make use of an optimized B+ tree implementation. The difference in performance between the two implementations *GoogleBTree* and *STLMultiMap* shows that the overall runtime of this approach is heavily influenced by the actual index structure used.

The approach *CellUnionContains* completes the workload in 367ms, compared to 550ms when implemented within *HyPerSpace*. The overhead is mostly caused by function calls that are issued for each of the 12M points. *CellUnionContains* is a straightforward approach. It first computes the bounding box and exterior and interior coverings for the given polygon. For each of the points, *CellUnionContains* then performs the following steps: First, it checks whether the point is within the bounding box. If that is the case, it checks for containment in one of the cells of the exterior covering. Analogous to the containment check for the interior covering, this essentially comes down to a binary search. Then the *CellUnionContains* approach continues analogous to the *GoogleBTree* approach by checking the interior covering and performing the exact containment check if necessary. By properly using the S2 mechanisms, our *CellUnionContains* approach achieves a slightly better performance than the index-based *STLMultiMap* approach, even though we have to loop over all of the 12M points.

For visualization purpose we also created an interactive web interface, called *HyPerMaps*, that demonstrates the outstanding geospatial processing performance of *HyPerSpace* on the NYC Taxi Rides dataset. The user interaction concept of *HyPerMaps* is designed to minimize the requirement of users' expertise with the explored data. The ability of *HyPerSpace* to answer queries with typically sub-second latency enables tight feedback loops. It supports users during query formulation and encourages an iterative approach. During filtering of the data, users can rely on datatype dependent elements, which provide context-based information like value distributions or geographic locations in real time. Users can draw polygons on the map to filter points geographically. Subsequently, users can combine different graphical and textual representations to create an informative and intuitive visualization. During this data exploration process, *HyPerMaps* will automatically compute updated results reflecting the current state of the user interface as well as the underlying dataset.

Figure 3 shows *HyPerMaps* visualizing the taxi dataset. On the left, various tiles allow users to specify filters on the data, which will be immediately translated into SQL code as illustrated on the top. This binding works in both directions—manually written SQL code will be translated into corresponding tiles. Users can choose between a heat map and pins to display selected points on the map. On the right, *HyPerMaps* shows aggregated information about selected points in tabular or in chart form.

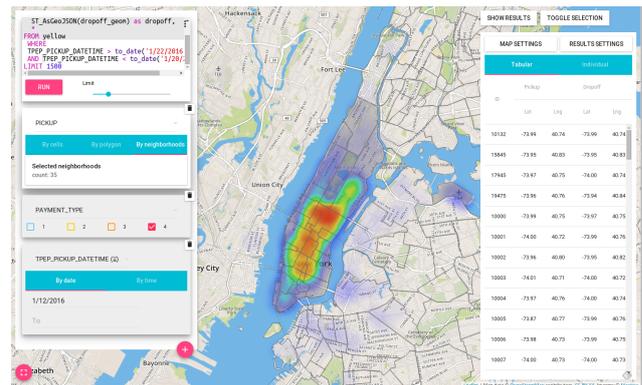


Figure 3. Interactive visualization of a real-time replay of NYC taxi rides using *HyPerMaps*

Conclusion

In this report we try to define what *Big data* means in context of geospatial data. We also carried a survey of available data mining algorithms for geospatial datasets especially focusing on vehicular dataset. The main goal of this subproject is exploration of big geospatial data and integrate data mining techniques, which help in analysis of such datasets, in an MMDB. This reduces the cost of moving the data from the data storage to the analytical tools which is expensive when the volume of the data is considered. We also presented preliminary results by integrating a geospatial module called *HyperSpace* in an MMDB *Hyper* to take a step in the direction of big geospatial exploration.

Acknowledgments

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation. Bavaria, an initiative of the Bavarian State Government.

References

- [1] Stratos Idreos, Olga Papaemmanouil, and Surajit Chaudhuri. Overview of data exploration techniques. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 277–281. ACM, 2015.
- [2] Jeffrey Heer and Sean Kandel. Interactive analysis of big data. *XRDS: Crossroads, The ACM Magazine for Students*, 19(1):50–54, 2012.
- [3] Anthony JG Hey, Stewart Tansley, Kristin Michele Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.
- [4] Ben Shneiderman. Extreme visualization: squeezing a billion records into a million pixels. In *Proceedings*

of the 2008 ACM SIGMOD international conference on Management of data, pages 3–12. ACM, 2008.

- [5] Kristi Morton, Magdalena Balazinska, Dan Grossman, and Jock Mackinlay. Support the data enthusiast: Challenges for next-generation data-analysis systems. *Proceedings of the VLDB Endowment*, 7(6):453–456, 2014.
- [6] Eugene Wu, Leilani Battle, and Samuel R Madden. The case for data visualization management systems: Vision paper. *Proceedings of the VLDB Endowment*, 7(10):903–906, 2014.
- [7] Parke Godfrey, Jarek Gryz, and Piotr Lasek. Interactive visualization of large data sets. Technical report, Technical Report EECs-2015-03 March 31 2015. Department of Electrical Engineering and Computer Science. York University. Toronto, Ontario. Canada, 2015.
- [8] Daniel A Keim. Exploring big data using visual analytics. In *EDBT/ICDT Workshops*, page 160, 2014.
- [9] Aditya Parameswaran, Neoklis Polyzotis, and Hector Garcia-Molina. Seedb: Visualizing database queries efficiently. *Proceedings of the VLDB Endowment*, 7(4):325–328, 2013.
- [10] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. immens: Real-time visual querying of big data. In *Computer Graphics Forum*, volume 32, pages 421–430. Wiley Online Library, 2013.
- [11] Nivan Ferreira Ferreira, Jorge Poco, Huy T. Vo, Juliana Freire, and Cláudio T Silva. Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips. In *IEEE Transactions on Visualization and Computer Graphics*, pages 2149–2158, December 2013.
- [12] Doug Laney. 3-D Data Management: Controlling Data Volume, Velocity and Variety. *META Group Research Note*, February 2001.
- [13] Andrew McAfee, Erik Brynjolfsson, Thomas H Davenport, DJ Patil, and Dominic Barton. Big data. *The management revolution. Harvard Bus Rev*, 90(10):61–67, 2012.
- [14] Planning Guide. Getting started with big data. *Intel*, 2013.
- [15] Erik Meijer. The world according to linq. *Queue*, 9(8):60, 2011.
- [16] Mark Beyer. Gartner says solving ‘big data’ challenge involves more than just managing volumes of data. *Gartner. Archived from the original on*, 10, 2011.
- [17] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.
- [18] Varun Pandey, Andreas Kipf, Dimitri Vorona, Tobias Mühlbauer, Thomas Neumann, and Alfons Kemper. High-Performance Geospatial Analytics in HyPerSpace. In *SIGMOD, SIGMOD ’16*, New York, NY, USA, 2016. ACM.
- [19] Alfons Kemper and Thomas Neumann. HyPer: A Hybrid OLTP & OLAP Main Memory Database System Based on Virtual Memory Snapshots. In *ICDE*, pages 195–206, April 2011.
- [20] Sanjay Chawla, Shashi Shekhar, Weili Wu, and Uygur Ozesmi. Modeling spatial dependencies for mining geospatial data. In *SDM*, pages 1–17. SIAM, 2001.
- [21] WR Tobler. Cellular geography. In *Philosophy in geography*, pages 379–386. Springer, 1979.
- [22] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, pages 99–108. ACM, 2010.
- [23] Wangsheng Zhang, Shijian Li, and Gang Pan. Mining the semantics of origin-destination flows using taxi traces. In *UbiComp*, pages 943–949. Citeseer, 2012.
- [24] Marco Veloso, Santi Phithakkitnukoon, and Carlos Bento. Sensing urban mobility with taxi flow. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, pages 41–44. ACM, 2011.
- [25] Xi Zhu and Diansheng Guo. Mapping large spatial flow data with hierarchical clustering. *Transactions in GIS*, 18(3):421–435, 2014.
- [26] Siyuan Liu, Yunhuai Liu, Lionel M Ni, Jianping Fan, and Minglu Li. Towards mobility-based clustering. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 919–928. ACM, 2010.
- [27] Markus Loecher and Tony Jebara. Citysense: Multiscale space time clustering of gps points and trajectories. In *Proceedings of the Joint Statistical Meeting*, 2009.
- [28] Bin Li, Daqing Zhang, Lin Sun, Chao Chen, Shijian Li, Guande Qi, and Qiang Yang. Hunting or waiting? discovering passenger-finding strategies from a large-scale real-world taxi dataset. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 63–68. IEEE, 2011.
- [29] Xiaolong Li, Gang Pan, Zhaohui Wu, Guande Qi, Shijian Li, Daqing Zhang, Wangsheng Zhang, and Zonghui Wang. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science*, 6(1):111–121, 2012.
- [30] Sachin Kumar and Durga Toshniwal. Analysing road accident data using association rule mining. In *Computing, Communication and Security (ICCCS), 2015 International Conference on*, pages 1–6. IEEE, 2015.

- [31] Andrey Bogomolov, Bruno Lepri, Jacopo Staiano, Nuria Oliver, Fabio Pianesi, and Alex Pentland. Once upon a crime: towards crime prediction from demographics and mobile data. In *Proceedings of the 16th international conference on multimodal interaction*, pages 427–434. ACM, 2014.
- [32] Ickjai Lee, Guochen Cai, and Kyungmi Lee. Mining points-of-interest association rules from geo-tagged photos. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 1580–1588. IEEE, 2013.
- [33] Gang Pan, Guande Qi, Zhaohui Wu, Daqing Zhang, and Shijian Li. Land-use classification using taxi gps traces. *Intelligent Transportation Systems, IEEE Transactions on*, 14(1):113–123, 2013.
- [34] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):13, 2010.
- [35] Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen, Lin Sun, and Shijian Li. ibat: detecting anomalous taxi trajectories from gps traces. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 99–108. ACM, 2011.
- [36] TLC Trip Record Data. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.
- [37] Todd Schneider. Analyzing 1.1 Billion NYC Taxi and Uber Trips, with a Vengeance. <http://toddschneider.com/posts/analyzing-1-1-billion-nyc-taxi-and-uber-trips-with-a-vengeance/>.
- [38] Chris Wong. FOILING NYC’s Taxi Trip Data. http://chriswhong.com/open-data/foil_nyc_taxi/.
- [39] Reuben Fischer-Baum and Carl Bialik. Uber Is Taking Millions Of Manhattan Rides Away From Taxis. <http://fivethirtyeight.com/features/uber-is-taking-millions-of-manhattan-rides-away-from-taxis/>.
- [40] Gary Orenstein. Real-Time Geospatial Intelligence with Supercar. <http://blog.memsql.com/real-time-geospatial-intelligence-with-supercar/>.
- [41] João Miranda. Uber Unveils its Realtime Market Platform. <http://www.infoq.com/news/2015/03/uber-realtime-market-platform/>.
- [42] PostGIS. <http://postgis.net/>.
- [43] Thomas Neumann, Tobias Mühlbauer, and Alfons Kemper. Fast Serializable Multi-Version Concurrency Control for Main-Memory Database Systems. In *SIGMOD, SIGMOD ’15*, pages 677–689, New York, NY, USA, 2015. ACM.

