

## Capabilities for value co-creation and value capture in emergent platform ecosystems: A longitudinal case study of SAP's cloud platform

Journal of Information Technology  
2021, Vol. 36(4) 365–390  
© Association for Information  
Technology Trust 2021



Article reuse guidelines:  
sagepub.com/journals-permissions  
DOI: 10.1177/02683962211023780  
journals.sagepub.com/jinf



Maximilian Schreieck<sup>1</sup>, Manuel Wiesche<sup>2</sup>   
and Helmut Krccmar<sup>1</sup>

### Abstract

Companies across industries are shifting toward a platform ecosystem strategy. By leveraging cloud computing technologies, companies aim to benefit from collaboration with a wide range of third-party developers within emergent platform ecosystems. To succeed, these companies need to develop new organizational capabilities to co-create and capture value in platform ecosystems. To understand what capabilities are crucial to establish platform ecosystems and how they contribute to value co-creation and value capture, we conducted a multi-year, in-depth case study of SAP's cloud platform project. We identified (1) technology-related capabilities (cloud-based platformization, open IT landscape management) and (2) relationship-driven capabilities (ecosystem orchestration, platform evangelism, platform co-selling) and illustrate how these capabilities help the platform owner to enable and balance value co-creation and value capture in an emergent platform ecosystem. With our findings, we contribute to the discussion on how companies can overcome the challenging emergent phase of platform ecosystems. We thereby bridge literature on value creation in platform ecosystems and on organizational capabilities. Though we conducted our study in the context of the enterprise software industry, we discuss how our findings apply to prospective platform owners from different contexts.

### Keywords

Platform ecosystem, IT platform, digital platform, enterprise software, platform ecosystem capabilities, platform emergence, value co-creation, value capture, SAP

### Introduction

In recent years, companies across industries have started to build platform ecosystems to leverage broad networks of third-party developers for value co-creation (Ceccagnoli et al., 2014; Förderer et al., 2018; Sandberg et al., 2020). For example, banks provide interfaces to their core banking systems to create an ecosystem for complementary digital services; car manufacturers open their infotainment systems to third-party developers; and enterprise software vendors create an ecosystem for business applications complementary to their enterprise software core (Sebastian et al., 2017). These companies leverage cloud computing technologies that allow third-party developers to implement complementary applications and customers to quickly deploy these applications (Lawton, 2008). While in the past, some platform ecosystems such as Google's Android

or Apple's iOS have become hugely successful, today, many companies in different contexts struggle to establish platform ecosystems (Bossert and Desmet, 2019; Yoffie et al., 2019).

To create successful platform ecosystems, platform owners need to simultaneously enable value co-creation in the platform ecosystem and capture a sufficient share of the co-created value (Tiwana et al., 2010). Hence, platform owners have to balance value co-creation and value capture

<sup>1</sup>Technical University of Munich, Germany

<sup>2</sup>TU Dortmund University, Germany

### Corresponding author:

Maximilian Schreieck, Technical University of Munich, Boltzmannstraße  
3, 85748 Garching, Germany.  
Email: maximilian.schreieck@tum.de

because they affect each other: on one hand, enabling value co-creation requires giving up some degree of control, which in turn weakens the platform owner's position to capture value. On the other hand, excessive value capture can harm value co-creation because it reduces incentives for third-party developers to join the platform ecosystem (Ceccagnoli et al., 2012; Cennamo and Santaló, 2019). Balancing value co-creation and value capture is crucial for emergent platform ecosystems (De Reuver et al., 2018): with insufficient value co-creation, the platform ecosystem is unattractive for third-party developers and positive network effects do not kick in. With insufficient value capture opportunities, the platform ecosystem might grow yet not yield sustainable profits for the platform owner.

One lens to study value creation in organizations is organizational capabilities (Mithas et al., 2011, 2012; Ravichandran and Lertwongsatien, 2005), which build on resources such as technology and relationships to support the value creation process (Bharadwaj, 2000; Ross et al., 1996). In platform ecosystems, technology and relationship resources change compared to traditional organizations and their supply chains: IT platforms introduce new technological resources, such as cloud computing technologies (Cusumano, 2010), and arm's length relations between third-party developers and the platform owner, replacing traditional principal-agent relationships in supply chains and outsourcing relationships (De Reuver et al., 2018). Thus, we need to reevaluate capabilities required for platform ecosystems. First capabilities for platform ecosystems have been discussed, but they are either described on a high level of abstraction as part of strategic management challenges (Helfat and Raubitschek, 2018; Teece, 2018) or they do not establish a link between capabilities and value co-creation and value capture (Tan et al., 2015). We, therefore, pose the research question: *What capabilities do companies need to enable and balance value co-creation and value capture in emergent platform ecosystems?*

To answer this question, we conducted a multi-year, in-depth case study of SAP, a leading multinational enterprise software vendor. With the latest version of its core enterprise resource planning (ERP) software, SAP has established a platform ecosystem for third-party developers, who create complementary software-as-a-service applications. We show that SAP developed crucial platform ecosystem capabilities that are technology-related (cloud-based platformization, open IT landscape management) and relationship-driven (ecosystem orchestration, platform evangelism, platform co-selling). A rich description of the capabilities' manifestations in organizational practice allows us to interpret how they helped SAP to enable and balance value co-creation and value capture in the emergent platform ecosystem.

By generalizing from the context of the enterprise software industry, we contribute to the literature on platform ecosystems and organizational capabilities in information system (IS). We first identify organizational capabilities that platform owners need in emergent platform ecosystems.

We then suggest that these platform ecosystem capabilities contribute to enabling and balancing value co-creation and value capture in emergent platform ecosystems, leading to sustainable success. Thus, by bridging the literature streams on organizational capabilities and platform ecosystems, we show how platform owners can overcome the critical emergent phase of platform ecosystems.

## Theoretical background

This section forms the foundation for theorizing platform ecosystem capabilities as an outcome of our study and provides a review of the relevant literature. The relevant theoretical domains include: (1) value co-creation and value capture in platform ecosystems, and (2) organizational capabilities in IS. We thereby consider both IS and management literature because the phenomenon of platform ecosystems is situated at the boundary of both disciplines (De Reuver et al., 2018). For greater readability, we describe the theoretical background before reporting our findings, even though the focus on organizational capabilities only emerged during the analysis of the first series of interviews in our empirical study.

### *Value co-creation and value capture in platform ecosystems*

With technological progress in IT, the locus of value creation has shifted from the single firm to supply chains and, more recently, to ecosystems that are complex and fragmented (Bitran et al., 2007; Pagani, 2013; Peppard and Rylander, 2006). Global connectivity through standard protocols and the advance of cloud computing technologies enable digital interconnection between products and processes within and across industries (Bharadwaj et al., 2013; Rai and Tang, 2014). An increasing number of companies in various sectors are striving to leverage this interconnectivity to create platform ecosystems.

Platform ecosystems build on an IT platform that can be defined as “[. . .] the extensible codebase of a software-based system that provides core functionality shared by the applications that interoperate with it and the interfaces through which they interoperate” (Tiwana et al., 2010: 676). The underlying switch from monolithic to modular software architectures facilitates collaboration between the platform owner and third-party developers, who create applications complementary to the IT platform (Tiwana et al., 2010; cf. Henfridsson et al., 2014). The platform owner, third-party developers, and customers form a platform ecosystem around the IT platform. Table 1 summarizes key terms related to platform ecosystems and illustrates them with an example from SAP.

Within platform ecosystems, the mode of collaboration changes as arm's length relations between the platform owner and third-party developers replace traditional principal-agent relationships with partners (De Reuver et al.,

**Table 1.** Definitions of key terms related to platform ecosystems.

Term	Definition	Illustration (based on SAP)	Sources
IT platform	"[. . .] the extensible codebase of a software-based system that provides core functionality shared by the applications that interoperate with it and the interfaces through which they interoperate"; also referred to as "digital platform," "software platform," or "cloud platform."	SAP's ERP system together with SAP's cloud platform that separates the ERP core from the periphery of applications	Tiwana et al. (2010: 676); Baldwin and Woodard (2009)
Application (app)	An add-on software subsystem or service that connects to the platform to add functionality; also referred to as "module," "extension," "plug-in," "complement," or "add-on."	Applications that enhance the functionality of SAP's ERP core, typically provided by SAP's ecosystem partners (e.g. an application for archiving ERP data)	Parker et al. (2017); Tiwana (2014)
Interfaces	Specifications and design rules that describe how the platform and applications interact and exchange information; also referred to as "application programming interfaces" (APIs)	The APIs that SAP's cloud platform offers for ecosystem partners to develop applications (e.g. to access ERP data in real-time)	Tiwana (2014)
Platform owner	An individual or organization representing the legal entity that owns the platform; also referred to as "platform operator."	The company SAP as a legal entity	Tiwana (2014); Evans et al. (2006)
Third-party developers	Individuals or organizations that develop one or more applications for the IT platform; also referred to as "complementors," or "(ecosystem) partners."	Companies that develop applications complementary to SAP's ERP system that are made available on the platform <sup>a</sup>	Tiwana (2014); Sarker et al. (2012)
Customers	Individuals or organizations that use the applications available on the IT platform; also referred to as "end-users."	Customers of SAP's ERP system who also use or consider using additional applications from the cloud platform	Tiwana (2014); Wareham et al. (2015)
Platform ecosystem	The platform and the applications specific to it as well as the stakeholders of the platform; also referred to as "digital platform ecosystem," "platform-based software ecosystem," or "software ecosystem."	SAP's ERP system with complementary applications, along with the ecosystem actors (SAP, ecosystem partners, and customers)	Cusumano and Gawer (2002); Tiwana (2014)

ERP: enterprise resource planning.

<sup>a</sup>In the empirical part of this article, we use the term "ecosystem partners" to refer to companies that develop applications on SAP's cloud platform. We still interpret these ecosystem partners as third-party developers in the theoretical sense.

2018). Thus, value creation takes on the form of value co-creation between the platform owner and third-party developers (Fuentelsaz et al., 2015; Sarker et al., 2012). For the platform owner, this creates the additional challenge to capture a sufficient share of the co-created value (Bharadwaj et al., 2013). Consequently, platform owners have to balance value co-creation and value capture. For example, enabling value co-creation requires that the platform owner relinquishes some degree of control, weakening its position to capture value. At the same time, excessive value capture can harm value co-creation because it reduces incentives for third-party developers to join the ecosystem (Ceccagnoli et al., 2012; Cennamo and Santaló, 2019).

To study value co-creation and value capture in platform ecosystems, IS scholars have focused on platform governance, that is, the platform owner's management of the

collaboration with third-party developers in the platform ecosystem (Ghazawneh and Henfridsson, 2013; Tiwana, 2014). For example, studies have been published on the optimal degree of openness of IT platforms (Benlian et al., 2015; Ondrus et al., 2015), the balance of openness and control (Boudreau, 2010; Ghazawneh and Henfridsson, 2013; Parker and Van Alstyne, 2018), the role of boundary resources (i.e. resources that support third-party development on platforms) to facilitate value co-creation on IT platforms (Eaton et al., 2015; Ghazawneh and Henfridsson, 2013), and optimal revenue sharing between platform owners and third-party developers (Oh et al., 2015).

Most studies focus on the growth and maturity phases of platform ecosystems rather than the phase of emergence—thus, it is unclear how IT platforms emerge in the first place (De Reuver et al., 2018). For example, Google's Android

platform and Apple's iOS platform have been studied intensely (e.g. Eaton et al., 2015; Karhu et al., 2018; Liu et al., 2014), but the circumstances of their emergence and associated challenges are rarely examined. In particular, for companies that establish IT platforms that transform their existing business, insights in the early phases of platform ecosystems would be valuable, as this transition is challenging (Altman and Tripsas, 2015).

The lens of organizational capabilities and their role for value co-creation and value capture provides one path to understand how emergent platform ecosystems can become successful (Tan et al., 2015; Venkatraman et al., 2014).

### *Organizational capabilities in IS*

With the shift toward an information society and the advance of IT, intangible resources (e.g. capabilities, skills, and knowledge) are becoming more important than physical resources. Accordingly, the knowledge-based view of the firm has emerged, demoting the resource-based view (Dosi et al., 2000; Grant, 1996; Kogut and Zander, 1992). According to the knowledge-based view, organizational capabilities can lead to a competitive advantage (Teece and Pisano, 1994). These capabilities can be broadly defined as an organization's ability to conceive, implement, and exploit its resources to perform a particular productive activity (Amit and Schoemaker, 1993; Mata et al., 1995), or more broadly speaking, the ability of a company to reach an intended outcome (Dosi et al., 2000). There is considerable theoretical support for asserting that an organization's performance is directly linked to its capabilities (Mithas et al., 2011, 2012; Ravichandran and Lertwongsatien, 2005).

IS literature focuses on understanding the role of IT in organizational capabilities. On one hand, IT can act as a resource that supports organizational capabilities. On the other hand, organizational capabilities that rely on human and relationship resources are necessary to benefit from IT (El Sawy et al., 2010; Pavlou and El Sawy, 2006). Table 2 provides an overview of organizational capabilities in IS synthesized from disparate literature, along with their underlying resources.

IS technical skills have been discussed as a capability that leverages human resources (e.g. Ravichandran and Lertwongsatien, 2005), that is, employees' up-to-date skills related to hardware and software. With regard to technology as an underlying resource, IS infrastructure capability, IS development capability, and IS planning and change management capability have all been highlighted in the literature (Ross et al., 1996; Wade and Hulland, 2004). These results show that technological capabilities not only focus on the use of software, hardware, and connectivity but also on processes, such as the development of IT systems. Capabilities that rely on relationships as an underlying resource focus either on relationships within companies or on relationships with external partners: structural and process governance

capabilities relate to internal governance activities such as exchanges between business and IT, and decision-making and monitoring (Kude et al., 2017; Peterson, 2004). Relational IT governance capabilities are outward-facing, incorporating the ability to organize and optimize the relationship between the IS function and external stakeholders (Wade and Hulland, 2004) and are embedded in the organization's general relational capabilities (Gulati, 1999; Lorenzoni and Lipparini, 1999). Recently, a number of studies have considered the wider relationships in ecosystems as resources for capabilities: digital business innovation capability refers to collaborations with independent ecosystem actors to drive innovation (Helfat and Raubitschek, 2018; Tan et al., 2015; Venkatraman et al., 2014).

This overview shows that most organizational capabilities in IS focus on how technology and relationship resources contribute to value creation within an organization or within close partnerships. The role of organizational capabilities for platform ecosystems has received little attention thus far (Vial, 2019). It, therefore, remains unclear what specific capabilities are helpful for platform owners and how these capabilities contribute to value co-creation and value capture. This understanding is particularly crucial for emergent platform ecosystems. Without the required capabilities, platform owners will struggle to either spark value co-creation in the first place or to capture a share of the value that is sufficient to maintain the platform ecosystem.

### **Research method**

To shed light on the capabilities that organizations need in the emergent phase of platform ecosystems, we conducted a multi-year, in-depth case study on the enterprise software vendor SAP and its cloud platform project. We take on an interpretivist stance (Sarker et al., 2018; Walsham, 1995, 2006) because the subject of our study—capabilities in an emergent platform ecosystem—is a recent phenomenon that is dynamically evolving. The best option for us to examine the phenomenon is to collect data through interviews while considering that our data are a construction of our interview participants' interpretation of what they observe and do (Klein and Myers, 1999; Walsham, 2006). We thereby have to take into account the context of the phenomenon to discuss the generalizability of the result (Goldkuhl, 2012; Klein and Myers, 1999). To do so, we provide a rich case overview before we describe our findings and discuss our study's contributions, boundary conditions, and limitations in the end (Davison and Martinsons, 2016; Sarker, 2016).

### *Case selection and overview of case data*

Based on theoretical sampling considerations (Eisenhardt and Graebner, 2007), we selected the case of SAP's cloud platform for our exploratory case study for the following reasons. First, the cloud platform has been emerging over



**Table 2.** Overview of organizational capabilities in IS.

Underlying resource	Capability	Description	References
Human	IS technical skills	A company's ability to leverage employees' up-to-date skills related to hardware and software	Bharadwaj (2000); Bhatt and Grover (2005); Wade and Hulland (2004); Ravichandran and Lertwongsatien (2005)
Technology	IS infrastructure capability	A company's ability to mobilize and deploy IT assets including hardware, software, and networking technologies in combination with other resources and capabilities	Bharadwaj (2000); Bhatt and Grover (2005); Wade and Hulland (2004)
	IS development capability	A company's ability to implement solutions based on new technologies along with alertness toward new technologies and trends	Pavlou and El Sawy (2006); Wade and Hulland (2004); Ross et al. (1996); see also Zhou and Wu (2010)
	IS planning and change management capability	A company's ability to anticipate technological changes and plan the usage of IS accordingly	Feeny and Ives (1990); Wade and Hulland (2004)
Relationship	Structural IT governance capability	A company's ability to connect and enable exchange between "business and IT management (decision making) functions."	Peterson (2004: 14); Kude et al. (2017)
	Process IT governance capability	A company's ability to "formaliz[e] and institutionaliz[e] strategic IT decision making or IT monitoring procedures."	Peterson (2004: 15); Kude et al. (2017)
	Relational IT governance capability	A company's ability to "coordinate competencies and combine knowledge across corporate boundaries."	Lorenzoni and Lipparini (1999: 317); Gulati (1999); Bharadwaj et al. (1999); Wade and Hulland (2004); Tan et al. (2015); Rai et al. (2012); Rai and Tang (2014)
	Digital business innovation capability	The ability of independent ecosystem participants to set up and coordinate to jointly explore new avenues of business model innovations supported by IT	Venkatraman et al. (2014); Helfat and Raubitschek (2018); Tan et al. (2015)

IS: information system.

several years, allowing us to study capabilities relevant for emergent platform ecosystems. Second, the cloud platform project is central to SAP's overall competitive strategy—the shift to cloud computing technologies has been referred to as the most significant change in the company's history. Thus, SAP provided the necessary financial resources to support the project and did everything to develop the required capabilities. This effort led to an increasing success of the cloud platform, making it an adequate case to study capabilities for emergent platform ecosystems.

Our case data include both primary and secondary data for the period from October 2015 to January 2019 (Figure 1). Although SAP's cloud platform project was initiated in 2012, it started to gain traction in 2015. This suggests that the period of our study is suited to analyze the capabilities SAP developed while establishing the platform ecosystem. Furthermore, our interviews allowed for a retrospective analysis of the period before 2015 given that most interview participants from SAP had a long-standing history at the company.

To gather primary data, we iteratively conducted semi-structured interviews that included participants' interpretations of the phenomenon (Walsham, 1995). In total, we

conducted 35 interviews with 36 interview participants between February 2016 and January 2019, 24 of the interviews were with SAP employees and 11 with ecosystem partners (Table 3). The interview participants included the company's vice president responsible for the platform project, the chief architect, and the product owner of the platform as well as other employees of SAP and ecosystem partners of different sizes. The interviews lasted 61 min on average. Twenty-nine interviews were conducted in German, six in English. We translated quotes from German-speaking interview participants that are included in this article. The interview questions covered the platform ecosystem strategy, the challenges and benefits associated with the shift toward that strategy, and the way SAP developed new capabilities. All interviews were recorded and transcribed.

During our study, we took on the role of an outside researcher (Walsham, 1995). By conducting interviews in several iterations with interview participants that had different perspectives on the cloud platform project, we were better able to critically reflect on our own interpretations, our prejudices, and the interpretations and biases of the interview participants (Klein and Myers, 1999). For

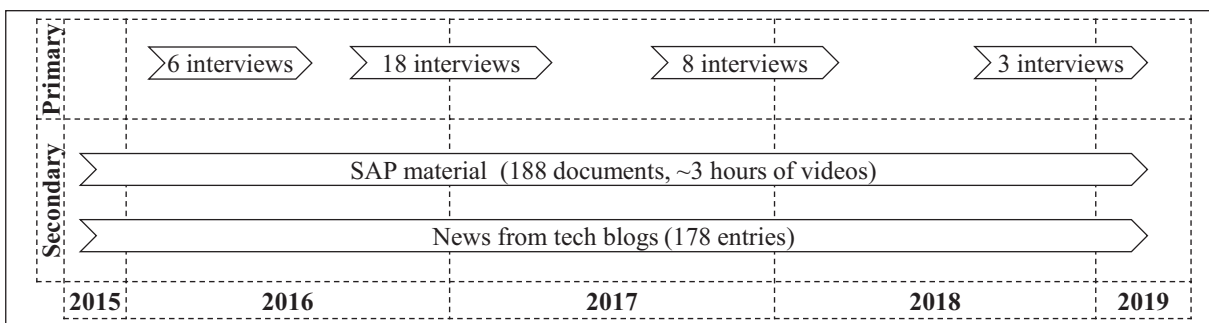


Figure 1. Overview of case data.

Table 3. Data collected for the case study.

Interviews—SAP (24 interviews; 25 interview participants)	
<p>Interview participants</p> <ul style="list-style-type: none"> <li>High-level managers responsible for the cloud platform (e.g. project lead, chief architect, product owner)</li> <li>Employees that worked with the ERP suite before the introduction of the cloud platform and could thus report on the changes inflicted by the platform ecosystem strategy</li> <li>Relatively new employees that had gathered experience in platform projects at other companies</li> </ul>	<p>Exemplary interview questions</p> <ul style="list-style-type: none"> <li>“What are the core features of SAP’s cloud platform?”</li> <li>“What challenges arose in the process of implementing the cloud platform?”</li> <li>“How did the interaction with partners change with the introduction of the cloud platform?”</li> <li>“What practices needed to change when the cloud platform was introduced?”</li> <li>“What is SAP’s business model behind the platform?”</li> </ul>
Interviews—ecosystem partners (11 interviews)	
<p>Interview participants</p> <p>High-level counterparts of SAP within 10 different ecosystem partner companies. These companies either offer complementary applications to the cloud platform or consult other companies on how to develop and market such applications</p>	<p>Exemplary interview questions</p> <ul style="list-style-type: none"> <li>“Can you describe the collaboration with SAP?”</li> <li>“What is your company’s motivation to contribute to SAP’s cloud platform ecosystem?”</li> <li>“What resources does SAP provide to support your development of complementary applications?”</li> <li>“What is your company’s business model behind the collaboration with SAP?”</li> </ul>
Secondary data	
Type of data	Material
<ul style="list-style-type: none"> <li>Documentation of SAP’s established enterprise software solution (technical documents, FAQ documents, customer presentations)</li> </ul>	5 documents
<ul style="list-style-type: none"> <li>Documentation of SAP’s cloud platform (technical documents such as API documentation, FAQ documents, customer presentations)</li> </ul>	163 documents
<ul style="list-style-type: none"> <li>Videos from SAP developer conferences with a focus on the cloud platform</li> </ul>	6 videos (~ 3 h)
<ul style="list-style-type: none"> <li>SAP’s investor relations (annual and interim reports 2015–2018)</li> </ul>	21 documents
<ul style="list-style-type: none"> <li>Acquisitions and acquired ecosystem partners (crunchbase data, ecosystem partner websites, and developer documentation)</li> </ul>	6 documents
<ul style="list-style-type: none"> <li>News and tech blog entries related to SAP</li> </ul>	178 entries

ERP: enterprise resource planning; FAQ: frequently asked questions; API: application programming interface.

example, by talking to ecosystem partners, we were able to limit the bias of SAP employees reporting positively on the company’s projects.

We collected secondary data to further account for biases in our own as well as our interview participants’ interpretations, and to provide a rich description of the case context.

Secondary data covered material that SAP had made available for ecosystem partners in the developer portal, internal presentations on the platform project, investor relations documents, reports on recent acquisitions performed by SAP, press releases, and videos of presentations at important SAP community events. We enhanced these data with entries from different technology blogs. These data were obtained by crawling the technology blogs based on keywords related to SAP's cloud platform. In total, the secondary data cover 195 documents, approximately 3 hours of video material, and 178 entries in tech blogs (Table 3).

### *Data analysis and interpretation*

For the analysis of our data, we followed an iterative approach that included coding of our data with an increasing degree of abstraction and reflecting our findings in view of our theoretical preconceptions (Klein and Myers, 1999; Walsham, 2006; Walsham and Sahay, 1999). From the start, we were interested in how SAP successfully established its cloud platform. We thus relied on theoretical preconceptions on value co-creation and value capture in platform ecosystems. In the course of our analysis, organizational capabilities emerged as a suitable lens to understand how SAP implemented the cloud platform successfully.

We started our analysis with open, *in vivo* coding of both primary and secondary data, relying on coding procedures used in the grounded theory methodology (Glaser, 1978, 2005; Wiesche et al., 2017). We created 489 codes associated with 673 quotes from interviews and other documents. In the process of open coding, the concept of capabilities emerged early on as interview participants reported on the changes SAP implemented when introducing the cloud platform. We thus went back to literature on organizational capabilities in IS, which informed our next step of coding (Suddaby, 2006; cf. Berente and Youngjin, 2012): we applied selective coding to cluster open codes into subcategories and categories to identify capabilities that SAP developed. While categories represent these capabilities, subcategories include the manifestations of these capabilities in organizational practice. In total, we identified 18 manifestations linked to five capabilities (see Table 5 in the Appendix; the full code book was provided to the review team). Not all open codes were linked to one of the manifestations but rather helped us to understand the case and its context.

As next step of our analysis, we interpreted the findings based on our theoretical understanding that had evolved along the analysis. We first differentiated technology-related and relationship-driven platform ecosystem capabilities based on the capabilities' underlying key resources. The detailed empirical manifestations of these capabilities allowed us to interpret how the capabilities impacted value co-creation and value capture in the platform ecosystem.

In the next sections, we first provide an overview on the SAP case to illustrate the background and context of the

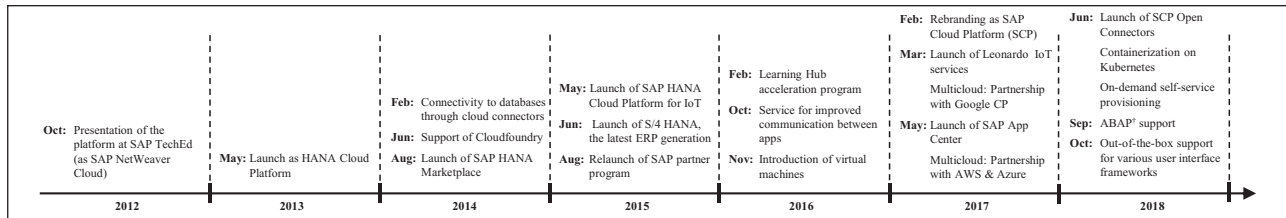
case. We then summarize our findings on the platform ecosystem capabilities and interpret how they helped SAP to enable and balance value co-creation and value capture.

### **Case overview: establishing a platform ecosystem**

SAP is a multinational software company focusing on enterprise software such as ERP systems. SAP's biggest success was its on-premises ERP suite that became widely adopted in the 1990s. The software was customizable so that customers could adapt it to their own needs and that ecosystem partners of SAP could add specialized functionality. For example, as customers expected end-to-end solutions, the ERP suite needed to be able to handle industry-specific processes as well as country-specific regulations such as fiscal or data protection laws. To address the resulting heterogeneous customer needs, SAP relied on ecosystem partners to enhance the product portfolio with extensions that could, for example, offer additional functionality or localization. Thereby, ecosystem partners developed on-premises extensions to SAP's ERP core that typically were adapted to the customers' specific IT landscape. With many ecosystem partners, SAP maintained close relationships that not only included support for the partners' development and sales activities but also joint portfolio planning to avoid overlap.

With the advance of cloud computing technologies and the emergence of competitors that relied mainly on cloud computing technologies (e.g. Salesforce), SAP underwent a paradigm shift with regard to its extensibility strategy. In 2012, SAP announced a cloud platform project to shift extensibility of the ERP core to the cloud and to create a more open and dynamic ecosystem for business applications (for a timeline of events see Figure 2). After a beta phase, the cloud platform was launched publicly in May 2013 as HANA Cloud Platform (HCP). The platform was named after the HANA database, SAP's proprietary in-memory database, which initially was the cloud platform's only underlying database. In an early blog entry, SAP described how the cloud platform addresses the challenge to enable fast development of business applications in the cloud that could be run as software-as-a-service applications:

Cloud platforms such as HCP address this challenge by providing a standardized design and runtime environment to develop, extend and run applications in the cloud. They provide an abstraction layer for the underlying infrastructure and take away the burden of managing all the lower-level infrastructure complexity such as load-balancing, disaster recovery (DR), high-availability (HA), fail-over scenarios (FO) and elasticity . . . just to name a few. This way, development organisations can focus on developing their business solution based on this standardized environment and the comprehensive set of capabilities, services and APIs offered by the platform. (SAP Blog, 10 October 2014)



**Figure 2.** Timeline of SAP's cloud platform.

†ABAP: "Advanced Business Application Programming," a proprietary programming language of SAP remotely similar to COBOL.

SAP's goal was to enable generativity in the cloud platform ecosystem, that is, to leverage the skills, expertise, and innovative capacity of its existing partner network along with new ecosystem partners to create a variety of business applications on the cloud platform:

Based on [the cloud platform], new applications, as well as extensions of existing applications can be built in the cloud. [...] Somewhat like an innovation layer for established, rather slow-ticking systems of SAP. [...] I think this is the benefit one could see, because we not only enable customers to do this but we also enable partners to develop such applications on the platform and this in turn creates an ecosystem. Thus, [the cloud platform] is an enabler of innovation [...]. (product manager of the cloud platform, SAP; translated)

On the cloud platform, ecosystem partners can collaborate with SAP through standardized channels, building on APIs and further resources such as documentation, sample applications, and video tutorials. To make the cloud platform attractive for ecosystem partners and customers, SAP increasingly opened the platform. In July 2014, SAP started to support Cloudfoundry, an open source cloud platform framework that later replaced SAP's proprietary cloud platform framework. In August 2015, SAP relaunched its partner program to allow partners to join the ecosystem and explore the platform with little upfront costs. In 2017, the platform was rebranded as SAP Cloud Platform to show that it had become the one and only extensibility platform in the SAP ecosystem and that HANA was no longer the sole underlying database. This move was followed by increasing compatibility to infrastructure offerings of competitors such as the Google Cloud Platform, Microsoft Azure, and Amazon Web Services. Technological openness was further increased in 2018 with the introduction of Kubernetes as open source technology for containerization and the SAP Open Connectors that allowed connectivity to various third-party solutions (e.g. Slack, Dropbox, Microsoft Office).

In parallel to increasing the openness of the platform ecosystem, SAP strived to make the ecosystem more attractive for partners. On one hand, SAP made it more convenient for ecosystem partners to interact with the underlying ERP system with the launch of the latest version of the ERP suite in 2014. The new ERP suite—even though still being deployed on-premises by a majority of the customers—was

designed to be extended through cloud applications rather than through on-premises extensions. On the other hand, SAP established a marketplace for the applications on the platform in 2014 (Figure 3), which was relaunched as SAP's general application store in 2017. The marketplace allowed ecosystem partners to reach SAP's global customer base without investments in a sales network.

While it took SAP some time to attract a significant number of customers and ecosystem partners for the platform, the number of customers has increased steadily since 2014. The number of ecosystem partners reached 500 in September 2016 and continued to rise to more than 3700 in December 2018. Accordingly, the number of customers rose from 1400 in June 2015 to 4000 in September 2016 to more than 10,000 in September 2018, according to SAP's official announcements. These numbers show that SAP struggled in the emergent phase of the platform but ultimately succeeded to establish the cloud platform as foundation for a new platform ecosystem. The case of SAP's cloud platform is, therefore, suited to study the capabilities a platform owner has to develop in the emergent phase of a platform ecosystem.

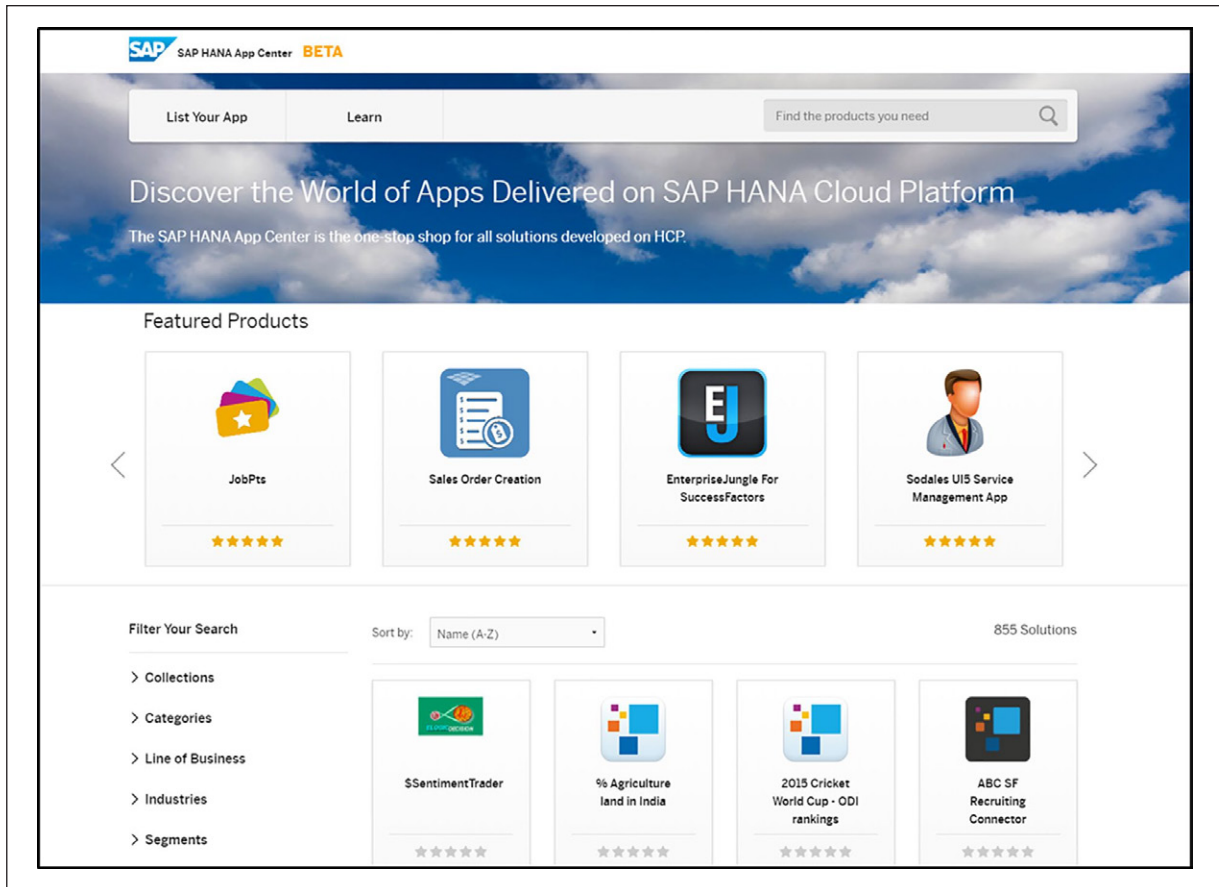
## Capabilities for emergent platform ecosystems

We discovered five platform ecosystem capabilities: two technology-related capabilities (cloud-based platformization, open IT landscape management) and three relationship-driven capabilities (ecosystem orchestration, platform evangelism, platform co-selling). Each capability can be described as a set of manifestations that we observed throughout the emergence of SAP's cloud platform ecosystem; and for each capability, we interpret how it impacted SAP's value co-creation and value capture in the platform ecosystem.

### Cloud-based platformization

In the past, partner extensions to SAP's on-premises ERP system had been deeply integrated into the ERP core, making maintenance and upgrades costly and slow.<sup>1</sup> With the introduction of the cloud platform, SAP developed the technology-related capability cloud-based platformization, which we describe as the ability to enable the development and





**Figure 3.** Screenshot of the SAP HANA Marketplace (5 September 2015).

deployment of modular, cloud-based third-party applications. Thus, the capability refers to SAP leveraging state of the art cloud computing technologies to modularize the platform's architecture. We discovered four manifestations of cloud-based platformization (see also Table 5 in the Appendix) that, in sum, led to a positive impact of cloud-based platformization on value co-creation and a mixed impact on value capture.

The first manifestation refers to SAP seizing control of the code of the ERP core. It was no longer possible for ecosystem partners to integrate their solutions directly into the core, that is, to merge code into the core. Instead, they had to use the cloud platform as an extensibility layer that abstracted from the core. Any interaction with the core was channeled through the platform and thus under the control of SAP. Making the ERP core available to third parties directly would have been risky as a misuse could affect the core's integrity and security. To enable third parties to build applications, SAP instead developed APIs that granted access to pre-defined functionality and data of the ERP core. SAP was able to enforce modular applications resulting in a scalable platform ecosystem that can incorporate a large number of third-party applications. The product owner of the cloud platform's software development kit (SDK) highlighted:

With [the cloud platform] we no longer allow [customized extensions to the ERP core] because as we are offering a software-as-a-service solution, we have to be able to operate the software continuously, we always have to be able to upgrade it. This is why there is only code of SAP in [the ERP core], code that SAP can maintain and upgrade, etc. Everything that is to be added as an extension must then be integrated on the cloud platform via standard interfaces [...] for reasons of maintainability. (product owner, cloud platform SDK, SAP; translated)

As mentioned by the product owner of the cloud platform's SDK, the second manifestation comprises connecting modular third-party applications and services through standardized APIs. For SAP's ecosystem partners, it is crucial that their applications can access the ERP core because it includes their customers' operating data. These data build the foundation for numerous use cases that are not included in SAP's standard ERP suite. When SAP seized control over the code of the ERP core, SAP had to provide alternative, standardized ways for ecosystem partners to consume ERP data from the core. The core thereby included not only SAP's proprietary ERP suite but also several acquisitions SAP performed over the years. The product owner of the platform explained that it took SAP some time to provide these APIs to the required extent:

When you say extensions, that means of course [...] extensions via APIs. This requires that the SAP cloud solutions, such as SuccessFactors, Ariba, Concur, the S4 Cloud Edition as they are all called, also offer APIs. It was a problem at first because they didn't have enough APIs; all of that has evolved over time. (product owner, cloud platform, SAP; translated)

The introduction of the cloud platform marked the shift from on-premises deployment to cloud-based deployment. As third manifestation, SAP enabled the deployment of applications in the cloud via virtualization, containerization, and microservice architecture. Virtualization and containerization represent two methods used to abstract operating systems and applications from the underlying hardware. They thus facilitate provision of applications via the cloud. A microservice architecture further helps to leverage virtualization and containerization because applications can be broken down in a set of microservices that can be deployed in their own containers. This helps to automate and speed up deployment:

Each application gets a small virtual cage in which it can run and is isolated from the other applications. The second aspect is that you have a standardized deployment format—especially with Docker<sup>2</sup>—[...] and of course, this makes deploying anything a lot easier. (product owner, cloud platform, SAP, translated)

As last manifestation, we discovered SAP's learning to develop and operate platform-as-a-service in the cloud. While in the past, SAP had sold their software to customers who then operated the software on their premises, SAP now had to develop and operate the cloud platform in a platform-as-a-service offering. This resulted in faster development cycles, increased needs for scalability of the underlying infrastructure, and higher requirements for security and availability. As a program and partner manager at SAP illustrated, this created additional efforts for SAP:

[The cloud platform] means cloud and cloud means operations. [...] While with [an on-premises extension] you basically sell to the customer and therefore no longer have much to do, in the cloud, you have the operational part and you need to make sure it works. That means the cloud parameters and the SLAs [service level agreements] for cloud are of course completely different and so you have a much higher effort. (program and partner management, SAP; translated)

While this increased the effort for SAP, it came along with benefits for the ecosystem. SAP could update the cloud platform simultaneously for all ecosystem partners and customers, thus, ecosystem partners were always able to offer their applications to all customers in the ecosystem.

The manifestations show that cloud-based platformization contributed to value co-creation in SAP's cloud platform ecosystem but created additional challenges for value

capture. As technology-related capability, cloud-based platformization built on technology resources such as standardized APIs, development and deployment tools such as containerization, and scalable infrastructure to enable value co-creation in the platform ecosystem. Simplified development of third-party applications led to a wider range of applications than in the old, on-premises ecosystem as illustrated by SAP's growing application marketplace. Cloud-based platformization also entailed standardized access to the ERP core. This standardization further contributed to value co-creation because it helped ecosystem partners to build applications that they could sell to different customers or at least to reuse building blocks of applications across different use cases. However, cloud-based platformization led to the emergence of small applications that entailed less opportunity for value capture through SAP. Instead of high up-front revenues through licensing required for on-premises extensions, SAP had to settle for a revenue share of the application sales, often through a subscription model or pay-per-use contracts as illustrated by most entries in the cloud platform's application marketplace. While, in the long run, these revenues could outperform revenue generated through on-premises licenses, SAP's initial value capture was reduced. Furthermore, cloud-based platformization required high upfront investments in the cloud platform architecture and came along with significant costs to run, maintain, and secure the cloud platform, further reducing SAP's potential value capture.

### *Open IT landscape management*

With the introduction of the cloud platform, SAP changed its strategy from building on mostly proprietary technology (e.g. SAP ABAP as programming language for extensions or SAP HANA as in-memory database) to leveraging a broad range of technologies from different sources. During this shift, SAP developed the capability of open IT landscape management, which we define as the ability to select and support compatibility with technologies from various origins to increase the platform ecosystem's technological openness. Open IT landscape management is a technology-related capability for which we observed three major manifestations (see also Table 5 in the Appendix).

The first manifestation, leveraging open source software and established languages, refers to SAP integrating open source software into the cloud platform and enabling compatibility with various programming languages. Initially, SAP had launched the platform based on a proprietary cloud platform framework that supported a limited set of programming languages for cloud applications. But the cloud platform team soon realized that the platform would benefit from a more open technological foundation. SAP thus opted for Cloudfoundry, an open source cloud platform framework that supported various popular programming languages, such as Java, Python, Node.js, and

HTML5. SAP not only built on Cloudfoundry but also contributed back to the open source project with both code and financial support. In return, SAP benefited from the constant improvements that the open source community contributed to the project. Furthermore, ecosystem partners were more likely to be familiar with the relevant technologies used in Cloudfoundry than in SAP's earlier proprietary framework. The large community was also helpful in solving ecosystem partners' issues. One of SAP's software engineers from the cloud platform team summarized the increased technological openness:

We are more open with [the cloud platform] and that is also the general path that SAP wants to follow because we know we cannot deliver best of breed in every aspect and there are a lot of strong open source communities developing simple things like a syntax highlighting editor [ . . . ] but also complex things that allow you to do machine learning and NLP [natural language processing] [ . . . ]. And [the cloud platform] really offers you the capability to deploy such modules—sometimes written in node [node.js], sometimes written in Java. [ . . . ] [The cloud platform] is really opening up and moving away from the trend of just allowing proprietary languages. (software engineer, cloud platform, SAP)

As part of open IT landscape management, SAP further enhanced openness by making the cloud platform compatible with different underlying database technologies and infrastructure-as-a-service offerings even though most of them were offered by competitors. While the cloud platform initially only supported SAP's proprietary in-memory database HANA, SAP better separated the platform from the underlying database technology and became open to databases from competitors such as Oracle. This was also reflected in the cloud platforms rebranding from HCP to SAP Cloud Platform in early 2017. SAP also opened its cloud platform with regard to the infrastructure it could be deployed on, which was again simplified by the Cloudfoundry cloud platform framework:

You can also just put [the cloud platform] on Amazon [Web Services] or Microsoft Azure, which is one of the great advantages of Cloudfoundry. Cloudfoundry—we are in the foundation—so there's an open source community and Cloudfoundry can also be operated by us, on hyper-scale providers like Amazon Web Services, Microsoft Azure and Google Cloud Platform. [ . . . ] That is multicloud. (product owner, cloud platform, SAP, translated)

As third manifestation, we identified that SAP increasingly supported the integration of the cloud platform with SAP on-premises legacy, other SAP cloud solutions, and third-party solutions. The cloud platform is an extensibility layer to the underlying ERP core no matter whether the core was deployed on-premises or in the cloud. At the time of this study, most customers still ran their ERP core as older on-premises versions. Thereby, the installations at the customers' premises differed greatly from one customer to

the next—a heterogeneity the cloud platform had to compensate for. In addition, SAP increased the cloud platform's compatibility with other cloud solutions it had acquired over time, for example, with SuccessFactors, a cloud-based human resources management system:

What the advantage [of the cloud platform] always is, where we have invested from the beginning, is a close integration with the on-premise landscapes that a customer already has and I think that's the biggest selling point. This, and a close integration with other cloud solutions. We have made acquisitions, SuccessFactors is the most important of them, it is a solution for human resources management. And of course, we make sure that the SuccessFactors solution can work closely with [the cloud platform]. (vice president, cloud platform, SAP; translated)

At later stages, the cloud platform's compatibility went beyond the SAP stack. In 2018, SAP introduced the Open Connectors that allowed ecosystem partners to easily integrate third-party cloud solutions such as Dropbox, Slack, or Microsoft Office into applications on the cloud platform. In sum, ecosystem partners had a greater choice of infrastructure, tools, and services to use when developing on the cloud platform.

In line with these manifestations, we observed that open IT landscape management supported value co-creation but weakened SAP's position for value capture. Open IT landscape management as technology-related capability leveraged technologies such as open source frameworks, connectors, or interfaces to legacy systems. The contribution to value co-creation is straightforward: technological openness fueled generativity and innovation on the platform because ecosystem partners could build on the skills and landscape they have rather than having to shift to a completely new technology stack. In addition, by using open source solutions, SAP leveraged synergies with the communities behind these solutions for value co-creation because the community members contributed code or solved issues. However, when relying on open IT landscape management, SAP had to consider its potential negative effect on value capture. SAP weakened its position in the ecosystem by giving up proprietary solutions that would lock in ecosystem partners on the platform. For example, when SAP granted ecosystem partners a choice of what database to use, SAP gave up license fees for its HANA database and made it easier for both ecosystem partners and customers to migrate data to competing ERP systems.

### *Ecosystem orchestration*

The introduction of the cloud platform led to new challenges for SAP with regard to collaborating within the ecosystem. SAP aimed to migrate its existing ecosystem partners to the cloud platform while also attracting new ecosystem partners. To combine both, SAP developed a capability we refer to as ecosystem orchestration, that is,



the ability to enable and manage high-quality third-party contributions to the platform ecosystem. This capability had its roots in SAP's long-standing experience in managing a partner ecosystem but goes beyond the traditional partnership approach for close collaboration. We identify five manifestations of the ecosystem orchestration capability (see also Table 5 in the Appendix).

The first manifestation—enabling ecosystem partners with standardized boundary resources—was crucial to provide ecosystem partners with the tools and the information they needed to develop applications on the cloud platform. In the past, SAP had provided ecosystem partners with boundary resources as well, but it was now crucial that these resources were standardized to reduce the costs. With channels such as the community network, video tutorials, and documentation, SAP aimed to address the majority of questions and issues ecosystem partners had with regard to the cloud platform:

Either available in the online help or on the SCN, the SAP Community Network, there is actually everything [developers need]. Then there is the SAP HANA Academy, there are tons of videos and then there are also the Open SAP courses. All the information is available and it is also possible to get a trial account for the [cloud platform] to just start developing. Of course, there is also access to documentation. (manager for partner certification, SAP; translated)

To put these resources into use as effectively as possible, SAP also worked on fast onboarding of new ecosystem partners and their applications. As part of this manifestation, SAP relaunched its partner program to include a novel category of ecosystem partners. Partners of this group were able to become part of the ecosystem with minimal upfront costs and were able to directly start developing on the cloud platform with trial accounts. In addition, SAP streamlined the process to enable development and deployment of an application on the cloud platform, resulting in a timeline of hours rather than days or weeks. The main goal was to enable rapid prototyping rather than allowing only market-ready applications because, with rapid prototyping options, ecosystem partners were more likely to come up with innovative applications. One of SAP's software engineers highlighted:

I think the latest tutorial shows [developing an app] to you in half an hour, including full connection to a database, and even previewing what you just built. [. . .] And that is also a strong point I'd say, that it is so open, in terms of quick and fast prototyping. (software engineer, cloud platform, SAP)

Along with enabling and accelerating the development of third-party applications on the platform, SAP had to enforce ecosystem quality standards with formal control. Given that the applications on the cloud platform were used in a business context, their quality was a critical factor for

customers. In the past, SAP had conducted extensive quality checks down to the level of code reviews for the extensions of their ERP suite. Such a level of detail was no longer possible for a potentially large number of third-party applications. SAP had to come up with formal control processes that balanced the effort and the outcome in terms of quality:

There is a vetting process ensuring that the applications are relatively debugged and we have an SAP service verification process so some of the applications are SAP certified. [. . .] Although it is not SAP in terms of legal reasons, we don't take responsibility for things that might happen from a partner application. But there are certain standards for a partner to meet in order to be in the app center. (software engineer, cloud platform, SAP)

In addition to boundary resources and fast onboarding, SAP increasingly leveraged their existing customer base as incentive to attract new ecosystem partners or to convince existing ecosystem partners to build applications on the cloud platform. Given that SAP had a global reach, this was a key selling point, in particular, for small ecosystem partners that were not able to set up global sales activities.

If you look at the reasons to join the ecosystem, of course it's the market and customer access [. . .] I have the general market access to the SAP market so to speak, I am in the SAP context, I'm already in there, I'm in the app store, that's also a certain reason, a background noise that sparks the partners' interest. (product manager, cloud platform, SAP)

Besides a global reach, ecosystem partners gained access to key customers through SAP. Large SAP customers typically would not consider small companies as their software providers. Being part of the SAP ecosystem in many cases was a door opener because customers trusted the quality that SAP vouched for.

As last manifestation of ecosystem orchestration, SAP enabled strategic ecosystem partners with individualized support. SAP realized that onboarding existing, strategic ecosystem partners was an important signaling effect for the ecosystem as a whole. SAP was, therefore, in continuous exchange with its existing ecosystem partners, provided individual support in addition to standardized boundary resources, and incorporated the ecosystem partners' feedback in the development of the cloud platform and its boundary resources. A project manager at an ecosystem partner company summarized:

What we also got is the close partnership with SAP. We have regular sync calls [. . .]. We also have the opportunity to convene expert rounds and then advise or coordinate on architectural issues, clarify our requirements and receive feedback from SAP experts, for example, on the topic of



asynchronous messaging [. . .] and how we can best implement this. (project manager for cloud platform, SAP ecosystem partner; translated)

This company also relied on the cloud platform's SDK and provided detailed feedback on how the SDK can be improved by SAP. The close collaboration led to this company being one of the first major ecosystem partners to offer solutions on the cloud platform.

The manifestations highlight that ecosystem orchestration as relationship-driven capability built on the relationships to existing and new ecosystem partners to foster both value co-creation and value capture. Orchestrating these relationships by providing boundary resources to existing ecosystem partners and new third-party providers, and by ensuring a sufficient level of quality through control mechanisms supported value co-creation in the ecosystem because it led to high-quality applications. Thereby, SAP was able to differentiate between ecosystem partners of low priority that received standardized boundary resources and ecosystem partners of strategic importance that received individualized support. This tailoring of boundary resources further increased value co-creation in the platform ecosystem because it allowed SAP to leverage its existing partner network for the platform ecosystem while incentivizing new ecosystem partners to join. With regard to value capture, ecosystem orchestration helped SAP to position itself as strategic bottleneck in the platform ecosystem. Boundary resources—in particular individualized support—tied ecosystem partners to the platform despite increasing technological openness. In addition, activities and solutions of ecosystem partners were observable for SAP because they were all happening on the platform rather than at customer sites. This way, SAP not only observed trends in the platform ecosystem but also absorbed innovative solutions by either acquiring or imitating them. For example, SAP acquired the Internet of things (IoT) platform vendor Plat.ONE in 2016 after observing that ecosystem partners had integrated Plat.ONE's IoT solutions with SAP's cloud platform.

### *Platform evangelism*

With the introduction of the cloud platform, SAP established a new ecosystem that not only included existing ecosystem partners but also heterogeneous new ecosystem partners. To make the ecosystem a success, SAP not only had to fulfill the ecosystem partners' technical requirements but also had to convince them on an emotional level. SAP addressed this with the capability of platform evangelism, a relationship-driven capability we define as the ability to create a joint vision for the platform ecosystem to incentivize third-party contributions. The capability builds on several manifestations that we observed during our study of the cloud platform ecosystem (see also Table 5 in the Appendix).

The first manifestation refers to SAP inspiring ecosystem members with a shared vision. Even though ecosystem partners on the platform are companies, it is individuals who make the decision to onboard the platform or to suggest this decision to their superiors. A shared vision can be a powerful trigger for individuals to push such a decision. To establish a shared vision, SAP engaged with the SAP community relying on employees that had become renowned members of the community—so-called evangelists. These evangelists also worked with further influencers and bloggers to spread their messages, as illustrated by one of the evangelists in a blog entry:

The early days were all about pushing the first wave of adoption among community influencers and bloggers. The first push is always the hardest (see Derek Silver's "How to start a movement") and hence we've been focusing on engaging with thought leaders & multipliers. We shared our vision for the road forward including how to gain adoption and stating the platform's unique selling proposition. (SAP blog entry, 12 December 2016)

Furthermore, SAP rebranded its platform with a campaign that focused on the openness of the platform and the ease-of-use for ecosystem partners—characteristics that ecosystem partners and customers not necessarily had associated with the platform before the rebranding. The increasing use of open source technology by SAP strengthened this rebranding and was underlined by the name change of the platform from HCP to SAP Cloud Platform. Along with these efforts to change the platform's image, SAP also worked hard to convince existing ecosystem partners to onboard the cloud platform. For example, SAP organized partner events to showcase successful partner applications on the platform. A vice president at SAP highlighted the importance of existing ecosystem partners:

We have to convince them! There is no choice, we have to convince them. We have to help them. We have to enable them. But it is absolutely critical that we manage to migrate them. (vice president, business development, SAP)

These manifestations show that platform evangelism complemented ecosystem orchestration by focusing on the relationships to potential future ecosystem partners and the wider developer community. Platform evangelism enhanced traditional marketing approaches by creating a shared vision for the platform within the developer community. This increased both awareness for and commitment to the platform ecosystem, contributing to immediate and future value co-creation. Furthermore, the synergetic exchange among community members—that can be observed, for example, in SAP's online community with more than 25,000 entries related to the cloud platform—strengthened value co-creation on the platform. Platform evangelism activities thereby increased the brand value of SAP and its

platform, an asset that translated to value capture, for example, through higher prices than those of competitors. While some ecosystem partners expressed concerns about the relatively high prices for the cloud platform and related components, this did not seem to affect the growth of the platform ecosystem.

### *Platform co-selling*

SAP had a strong history in selling directly to its customers, in particular, to large customers. As part of creating strategic alliances with ecosystem partners, SAP granted selected ecosystem partners access to these direct sales channels by including partner extensions in the sales activities. With the introduction of the cloud platform, new sales channels emerged, in particular, a marketplace for third-party applications. However, given the complexity of the enterprise software industry, the marketplace did not replace existing sales channels but complemented them. SAP developed the capability of platform co-selling, that is, the ability to establish and utilize the ecosystem's sales channels through collaboration with ecosystem partners. We observed three manifestations for this capability (see also Table 5 in the Appendix).

The first manifestation refers to running an online marketplace. Soon after the launch of the cloud platform, SAP launched the first marketplace for third-party applications. At first, the marketplace did not play a role in SAP's direct sales processes. As a result, customers would not visit the marketplace and it was not attractive for ecosystem partners. This changed over time when SAP made the marketplace one of the central elements of its cloud strategy and included the marketplace in the portfolio of its sales teams. A product and innovation manager highlighted:

“Cloud” is a special topic and it is contingent on having a store and which conditionally has an extension accessibility framework, which allows small partners to quickly build apps and make them available to a large customer base, otherwise it doesn't pay off. (product and innovation manager, SAP; translated)

However, the marketplace did not replace existing direct sales channels because customers were used to buying through their account executives at SAP. Also, some ecosystem partners that provided applications on the marketplace had their own sales networks and did not want to move their customers on an SAP-branded marketplace. SAP thus combined platform, direct, and partner sales channels, which we identified as one manifestation of the platform co-selling capability. In essence, SAP developed a high degree of flexibility with regard to the channel that an application on the cloud platform would be sold through:

The big [partners], they have their own sales track, [. . .] so for example, I would say they have almost as many salespeople as we do. They sell their applications or their integrations, which

they have built, themselves. [. . .] It's like that with the big partners, with the small partners it's more that they want to be sold through us, because of course they don't have a big sales track. (product owner, IoT, SAP; translated)

Given the heterogeneous sales processes, SAP also had to identify optimal prices across ecosystem participants. Again, SAP had to be flexible with regard to pricing models. Large ecosystem partners often wanted to purchase resources on the cloud platform upfront, while small ecosystem partners preferred pay-per-use pricing models or a revenue share based on the sales of their applications toward their customers. With regard to pay-per-use pricing, SAP had to come up with the right metrics to measure consumption. In different use cases, different metrics had to be applied because customers expected a metric that is somehow related to their revenue:

The whole market and entire pricing models will move towards usage-based billing. Why? Because it gives you flexibility and that's actually the point. You always want economies of scale, flexibility, more speed, digital transformation in the cloud. You want to test something for a short time without tying yourself to someone for three years, but these offers are actually interesting. Whether you have more revenue after that, time has to tell. (product manager, cloud platform, SAP; translated)

While ecosystem orchestration and platform evangelism focused on the partner side of the ecosystem, platform co-selling leveraged the trilateral relationships between SAP, ecosystem partners, and customers. By combining the sales channels historically available to SAP and its ecosystem partners along with the sales channels created by the cloud platform, SAP reached more customers. By offering bundle deals, certifying applications, or offering applications under its own brand, SAP significantly enhanced ecosystem partners' sales through SAP sales channels in addition to the sales on the platform's online marketplace. This created additional incentives for ecosystem partners to create new applications, thus leading to more value co-creation. Platform co-selling allowed SAP to stay in control of most of the sales channels as well as to remain visible to the customers. In addition, many ecosystem partners depended on SAP's sales channels because they were not able to set up and maintain a global sales network by themselves. Those factors strengthened SAP's position to negotiate value capture in the relationship with ecosystem partners.

In sum, the two technology-related capabilities (cloud-based platformization, open IT landscape management) contributed to value co-creation but created challenges for value capture. The three relationship-driven capabilities (ecosystem orchestration, platform evangelism, platform co-selling) showed positive impact on both value co-creation and value capture (Table 4). Thus, only the combination of technology-related and relationship-driven capabilities ensured the success of SAP's cloud platform ecosystem.

## Discussion

We have identified five capabilities for emerging platform ecosystems and described their impact on value co-creation and value capture from the platform owner's perspective. Given that not all capabilities support both value co-creation and value capture, platform owners face the challenge of striking a balance between the two. Based on our findings, we now discuss how platform owners can approach this challenge. We then summarize our contributions to theory and practice along with the limitations of our study.

### *Balancing value co-creation and value capture in emergent platform ecosystems*

While both technology-related capabilities and relationship-driven capabilities contribute to value co-creation, technology-related capabilities weaken opportunities for value capture. We thus confirm that platform owners have to balance value co-creation and value capture (Bharadwaj et al., 2013), and we show that the set of platform owner's capabilities can contribute to striking this balance (Figure 4). Finding the right balance is particularly crucial for emergent platform ecosystems; otherwise, they run the risk of either withering or not providing a sustainable revenue to the platform owner (Ceccagnoli et al., 2012; Cennamo and Santaló, 2019).

In line with the literature on platform launches, establishing value co-creation is important in the first phase of an emergent platform ecosystem (Evans and Schmalensee, 2010; Parker et al., 2016). A combination of technology-related and relationship-driven capabilities helps to kick off value co-creation, given that both types of capabilities have a positive impact. For example, cloud-based platformization simplifies implementation of applications on the platform hence increasing incentives for third-party developers to join the platform ecosystem (cf. Benlian et al., 2018; Henfridsson et al., 2014; Tiwana et al., 2010). This effect is complemented by standardized and individualized boundary resources as part of ecosystem orchestration (cf. Eaton et al., 2015; Karhu et al., 2018; Nielsen and Aanestad, 2006; Parker and Van Alstyne, 2018). Similarly, open IT landscape management increases the freedom of choice for third-party developers, fueling generativity and innovation on the platform (cf. Ondrus et al., 2015; Thomas et al., 2014), which can be highlighted as part of platform evangelism activities.

However, value co-creation alone is insufficient; the platform owner has to capture value or at least establish a good position for future value capture to guarantee the sustainable success of the emergent platform ecosystem (Bharadwaj et al., 2013). Given that technology-related capabilities impede value capture, platform owners need to focus on relationship-driven capabilities to ensure value capture. For example, open IT landscape management reduces technological lock-in because third-party

developers can rely on commonly used technologies rather than the platform owner's proprietary technologies. The platform owner has to compensate for the weakened lock-in effect on the technological level by increasing the interdependency on the relationship level. At the same time, excessive value capture can harm value co-creation if it alienates ecosystem partners. For example, absorbing complementary applications as part of ecosystem orchestration discourages innovation in some cases (Huang et al., 2013), and strongly limiting the visibility of ecosystem partners in the market in favor of the platform owner's visibility can make ecosystem partners leave or switch to competing platforms.

When platform owners leverage relationship-driven capabilities to capture value, it may be necessary to adjust the approach to value co-creation, creating an iterative loop to balance value co-creation and value capture, as depicted in Figure 4 above. For example, to benefit from the relationship-driven capability of platform co-selling, the platform owner might need to further invest in its cloud-based platformization capability to enable provisioning and billing at the microservice level.

In sum, the platform owner needs to both enable and balance value co-creation and value capture from the start of the platform ecosystem. Even if value co-creation is a prerequisite for the success of a platform ecosystem, platform owners who neglect value capture will struggle to establish a sustainable platform ecosystem. Only by combining technology-related and relationship-driven capabilities can platform owners create a successful platform ecosystem. This finding shows that established companies such as SAP, with strong expertise in orchestrating their partner ecosystem, are in a good position to launch platform ecosystems, even if they do not implement the newest technologies as fast as startup competitors. They can leverage the relationship resources that are embedded in their existing ecosystem to fuel interaction within the newly created platform ecosystem.

### *Contributions to theory and practice*

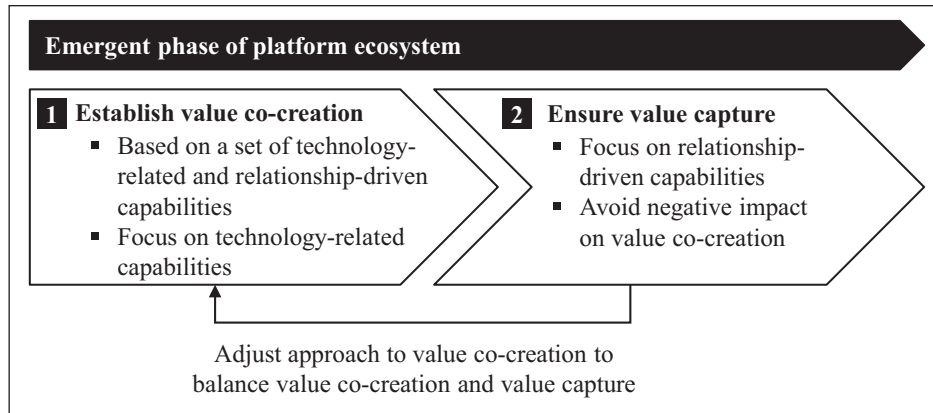
Our findings on platform ecosystem capabilities and their impact on value co-creation and value capture contribute to theory on platform ecosystems and add to the literature on organizational capabilities in IS.

With regard to theory on platform ecosystems, we conceptualize how platform owners can establish a successful platform ecosystem that enables and balances value co-creation and value capture. First, we highlight that considering both value co-creation and value capture is crucial, adding to IS literature that has focused predominantly on value co-creation (Grover and Kohli, 2012; Sarker et al., 2012). Management research has long pointed out the need to consider value co-creation and value capture as distinct

**Table 4.** Summary of platform ecosystem capabilities.

Capability and manifestations	Key underlying resource	Impact on value co-creation and capture
<p>Cloud-based platformization: The ability to enable development and deployment of modular, cloud-based third-party applications</p> <p>Manifestations</p> <ul style="list-style-type: none"> <li>■ Seizing control of the code of the ERP core</li> <li>■ Connecting modular third-party applications and services through standardized APIs</li> <li>■ Enabling deployment of applications in the cloud via virtualization, containerization, and microservice architecture</li> <li>■ Developing and operating platform-as-a-service in the cloud</li> </ul>	<p>Technology (e.g. APIs, development and deployment tools, scalable infrastructure)</p>	<p>Value co-creation</p> <ul style="list-style-type: none"> <li>■ Simplified development of third-party applications enables wider range of applications</li> <li>■ Standardized access to ERP core increases value of applications for customers</li> <li>■ Value capture</li> <li>■ Smaller size of applications on the platform initially reduces value capture</li> <li>■ High investments in architecture and running costs diminish value capture</li> </ul>
<p>Open IT landscape management: The ability to select and support compatibility with technologies from various origins to increase the platform ecosystem's technological openness</p> <p>Manifestations</p> <ul style="list-style-type: none"> <li>■ Leveraging open source software and established languages</li> <li>■ Enabling compatibility with technology stacks of competitors</li> <li>■ Supporting integration with SAP on-premises legacy, other SAP cloud solutions, and third-party solutions</li> </ul>	<p>Technology (e.g. open source frameworks, connectors, interfaces with legacy systems)</p>	<p>Value co-creation</p> <ul style="list-style-type: none"> <li>■ Technological openness allows ecosystem partners use the skills they have</li> <li>■ Activities in open source communities (e.g. issue solving) create synergies</li> <li>■ Value capture</li> <li>■ Reduced technological lock-in reduces value capture</li> </ul>
<p>Ecosystem orchestration: The ability to enable and manage high-quality third-party contributions to the platform ecosystem</p> <p>Manifestations</p> <ul style="list-style-type: none"> <li>■ Enabling ecosystem partners with standardized boundary resources</li> <li>■ Fast onboarding of new ecosystem partners and applications</li> <li>■ Enforcing ecosystem quality standards with formal control</li> <li>■ Leveraging the existing customer base</li> <li>■ Enabling strategic ecosystem partners with individualized support</li> </ul>	<p>Relationship (e.g. relationships to existing and new ecosystem partners)</p>	<p>Value co-creation</p> <ul style="list-style-type: none"> <li>■ Boundary resources adapted to the type of ecosystem partner facilitate development of applications</li> <li>■ Quality control increases value for customers</li> </ul> <p>Value capture</p> <ul style="list-style-type: none"> <li>■ Relational lock-in in the platform ecosystem improves position for value capture</li> <li>■ Observation and absorption increase value capture</li> </ul>
<p>Platform evangelism: The ability to create a joint vision for the platform ecosystem to incentivize third-party contributions</p> <p>Manifestations</p> <ul style="list-style-type: none"> <li>■ Inspiring with a shared vision</li> <li>■ Establishing platform brand image as an open platform</li> <li>■ Convincing existing ecosystem partners to onboard the cloud platform</li> </ul>	<p>Relationship (e.g. relationships to potential future ecosystem partners, wider developer community)</p>	<p>Value co-creation</p> <ul style="list-style-type: none"> <li>■ Increasing awareness for cloud platform attracts ecosystem partners</li> <li>■ Evangelism fuels synergistic exchange within developer community</li> </ul> <p>Value capture</p> <ul style="list-style-type: none"> <li>■ Brand value increases value capture</li> </ul>
<p>Platform co-selling: The ability to establish and utilize the platform ecosystem's sales channels through collaboration with ecosystem partners</p> <p>Manifestations</p> <ul style="list-style-type: none"> <li>■ Running an online marketplace</li> <li>■ Combining platform and partner sales channels</li> <li>■ Identifying optimal pricing across ecosystem participants</li> </ul>	<p>Relationship (e.g. relationships to ecosystem partners and customers within sales channels)</p>	<p>Value co-creation</p> <ul style="list-style-type: none"> <li>■ Shared access to customers increases sales opportunities</li> </ul> <p>Value capture</p> <ul style="list-style-type: none"> <li>■ Control over sales channels and visibility for customers improve position for value capture</li> </ul>





**Figure 4.** Balancing value co-creation and value capture.

mechanisms (Lepak et al., 2007; Priem, 2007), and our findings suggest that this is particularly crucial for emergent platform ecosystems. Value co-creation without sufficient value capture for the platform owner leads to an unsustainable platform ecosystem, but excessive value capture can stifle the positive network effects in the ecosystem. Thus, platform owners also need to consider how they can both enable and balance value co-creation and value capture.

Second, we suggest that the lens of organizational capabilities is helpful to understand how platform owners can cope with the double-sided challenge of value co-creation and value capture, enhancing IS literature on platform governance (e.g. Ghazawneh and Henfridsson, 2013; Tiwana, 2014). The literature on platform governance has considered some of the concepts that we identified as part of relationship-driven capabilities, such as boundary resources (Eaton et al., 2015; Ghazawneh and Henfridsson, 2013) and a balance of openness and control (Boudreau, 2010; Ghazawneh and Henfridsson, 2013; Parker and Van Alstyne, 2018). We link these concepts of platform governance to relationship-driven capabilities that interact with technology-related capabilities, creating a more complete picture of value co-creation and value capture in platform ecosystems. Accordingly, our findings apply to emergent platform ecosystems—a phase of platform evolution that remains understudied (De Reuver et al., 2018)—and contribute to recent calls to illuminate the role of capabilities in digital transformation (Vial, 2019). Platform ecosystem capabilities have to be developed early on if a platform ecosystem is to be successful. Furthermore, while value co-creation is more important in the emergent phase of a platform ecosystem, neglecting value capture in this phase will make it more difficult for platform owners to capture value at later stages.

With regard to the literature on organizational capabilities in IS, we add specific platform ecosystem capabilities. The effect of most capabilities—in particular, technology-related capabilities—has been analyzed within organizations or within strategic partnerships and supply chains

rather than within platform ecosystems (e.g. Bharadwaj, 2000; Bhatt and Grover, 2005; Wade and Hulland, 2004). For relationship-driven capabilities, in recent years, studies have considered platform ecosystems (Helfat and Raubitschek, 2018; Tan et al., 2015; Venkatraman et al., 2014). Venkatraman et al. (2014) take on a value perspective and use the high-level conceptualization digital business innovation capability to describe how platform owners become fit to co-create and capture value in platform ecosystems. We enhance their work by providing a broader portfolio of capabilities, supported by rich empirical insights on the capabilities' manifestations in organizational practice. Helfat and Raubitschek (2018) highlight the need for capabilities that enable innovation and ecosystem orchestration in platform ecosystems. We add technology-related capabilities and their interplay with relationship-driven capabilities, contributing to a contextualization of findings from management research in IS (cf. Hong et al., 2014). Furthermore, we confirm the importance of capabilities in early stages of platform ecosystems (Tan et al., 2015); however, we add to the work by Tan et al. (2015) by establishing the link between platform ecosystem capabilities and value creation in emergent platform ecosystems.

Our findings also contribute to practice. We provide a set of platform ecosystem capabilities with empirical details on their manifestation in organizational practice. Companies that aim to establish digital platforms can use them to evaluate whether they “have what it takes” to be successful or what they need to do to increase their chances of success. The detailed manifestations we provide help practitioners to derive strategies to develop these capabilities. Platform ecosystems are increasingly important for established companies as part of their digital transformation (Sebastian et al., 2017). Thus, our study provides timely insights for numerous companies from traditional industries. In addition, we demonstrated the need to balance value co-creation and value capture in emergent platform ecosystems and show the value an existing partner ecosystem can have. These insights will help practitioners to avoid common mistakes such as measuring the success of a platform

ecosystem solely based on short-term profits or relying on growth through positive network effects without thinking about future value capture (cf. Alstynne et al., 2016; Yoffie et al., 2019).

### Limitations

The theoretical insights from our study go beyond the case of SAP but are subject to boundary conditions and limitations. While the single case study approach poses a challenge for the generalizability of our insights (cf. Corley and Gioia, 2004), we argue that the platform ecosystem capabilities we have identified are relevant for emergent platform ecosystems beyond the enterprise software industry.

We described the context of the case in detail in the case overview section to facilitate discussion of the generalizability of our findings (Klein and Myers, 1999). The context of an enterprise software vendor that introduces a platform ecosystem is similar to companies from other industries. On the one hand, established companies, such as manufacturers, banks, insurance companies, or energy utilities are increasingly striving to unlock the generativity of IT platforms (Parker et al., 2016; Sebastian et al., 2017) and face similar challenges to enable and balance value co-creation and value capture in emergent platform ecosystems. The industrial IoT, for example, allows production equipment manufacturers to establish an IT platform for managing complex production setups with an ecosystem of third-party applications. A manufacturer might first need to build up technology-related capabilities, such as cloud-based platformization and open IT landscape management, before launching a successful platform ecosystem. Future research could target platform projects in other industries to verify our findings. A cross-industry study could help to control for industry-specific contextual factors; and with a configurational approach, researchers may help identify patterns for successful platform ecosystems (El Sawy et al., 2010).

On the other hand, our findings provide a starting point to understand the success of the major platform ecosystems, such as Google's Android ecosystem. We note that some of the manifestations we identified are specific to the context of the enterprise software industry with its history of on-premises businesses. For example, the manifestation that refers to convincing existing ecosystem partners to onboard the cloud platform would not apply to digital platforms launched with a greenfield approach without a strong existing partner ecosystem. However, we argue that we can generalize the capabilities to such greenfield platform launches, which should improve our understanding of platform emergences in general because research has mostly focused on the mature phases of these platforms. While the platform ecosystem capabilities we identified might also be of relevance in more mature platform ecosystems, their contribution to enabling and balancing value co-creation

and value capture is critical in the emergent phase of platform ecosystems. Capabilities to maintain the success of an established platform ecosystem might differ and would benefit from further research.

Other limitations of our study include the retrospective bias inherent to interviews and the time frame of our study. The interview participants' statements on the development of the cloud platform since its launch in 2012 and the growth of the number of customers since 2015 suggest that the platform ecosystem strategy was successful. However, our evaluation of how sustainable SAP's platform strategy was is limited to the time frame of our study. While we addressed the issue of retrospective bias by triangulating interview data with other data sources, continued observation of the platform ecosystem would contribute to further clarity on the impact of the capabilities on the long-term success of the platform.

### Conclusion

Our in-depth case study of SAP's cloud platform project provides a rich empirical understanding of capabilities that platform owners need in the emergent phase of their platform ecosystem and how these capabilities impact value co-creation and value capture. We have identified two technology-related capabilities (cloud-based platformization, open IT landscape management) and three relationship-driven capabilities (ecosystem orchestration, platform evangelism, platform co-selling). The technology-related capabilities contribute to value co-creation but may weaken the platform owner's position to capture value. Relationship-driven capabilities compensate for that by supporting both value co-creation and value capture. Platform owners thus have to combine capabilities to both enable and balance value co-creation and value capture in emergent platform ecosystems. Considering that many platform ecosystems fail early on, we contribute to the discussion of what it takes for platform ecosystems to survive the emergent phase.

### Authors' note

Maximilian Schreieck is also affiliated with The Wharton School, University of Pennsylvania, Philadelphia, USA.

### Acknowledgement

We thank the interview partners at SAP and partner companies for their valuable time and insights. Earlier versions of this paper benefited from constructive feedback by Philip W. Yetton, T. Ravichandran, and Stefan Henningsson. We also thank participants of the AIS SIG GTM workshop at ICIS 2017, hosted by Suprateek Sarker, as well as the participants of research seminars at Ludwig Maximilian University of Munich and Technical University of Munich for valuable input. Finally, we are grateful for the guidance by the senior editor and the anonymous reviewers who helped us focus and improve the paper.

## Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

## ORCID iD

Manuel Wiesche  <https://orcid.org/0000-0003-0401-287X>

## Notes

1. The issue of high costs for maintaining and upgrading on-premises ERP systems has also been discussed in IS literature (Ng and Gable, 2010).
2. Docker is an open-source platform-as-a-service offering that enables software delivery in packages called containers.

## References

- Alstynne MWV, Parker GG and Choudary SP (2016) 6 reasons platforms fail. *Harvard Business Review*. Available at: <https://hbr.org/2016/03/6-reasons-platforms-fail> (accessed 12 August 2021).
- Altman EJ and Tripsas M (2015) Product to platform transitions: Implications of organizational identity. In: Shalley C, Hitt M and Zhou J (eds) *Oxford Handbook of Creativity, Innovation, and Entrepreneurship: Multilevel Linkages*. Oxford: Oxford University Press.
- Amit R and Schoemaker PJH (1993) Strategic assets and organizational rent. *Strategic Management Journal* 14(1): 33–46.
- Baldwin CY and Woodard CJ (2009) The architecture of platforms: A unified view. In: Gawer A (ed.) *Platforms, Markets and Innovation*. Cheltenham and Northampton, MA: Edward Elgar, pp. 19–44.
- Benlian A, Hilkert D and Hess T (2015) How open is this platform? The meaning and measurement of platform openness from the complementors' perspective. *Journal of Information Technology* 30(3): 209–228.
- Benlian A, Kettinger WJ, Sunyaev A, et al. (2018) The transformative value of cloud computing: A decoupling, platformization, and recombination theoretical framework. *Journal of Management Information Systems* 35(3): 719–739.
- Berente N and Youngjin Y (2012) Institutional contradictions and loose coupling: Postimplementation of NASA's enterprise information system. *Information Systems Research* 23(2): 376–396.
- Bharadwaj A, Sambamurthy V and Zmud RW (1999) IT capabilities: Theoretical perspectives and empirical operationalization. In: *20th international conference on information systems*, Charlotte, NC, USA, 1 January.
- Bharadwaj AS (2000) A resource-based perspective on information technology capability and firm performance: An empirical investigation. *MIS Quarterly* 24(1): 169–196.
- Bharadwaj AS, El Sawy OA, Pavlou PA, et al. (2013) Digital business strategy: Toward a next generation of insights. *MIS Quarterly* 37(2): 471–482.
- Bhatt GD and Grover V (2005) Types of information technology capabilities and their role in competitive advantage: An empirical study. *Journal of Management Information Systems* 22(2): 253–277.
- Bitran G, Gurumurthi S and Sam SL (2007) The need for third-party coordination in supply chain governance. *MIT Sloan Management Review* 48(3): 30–37.
- Bossert O and Desmet D (2019) The platform play: How to operate like a tech company. *McKinsey Digital*. Available at: <https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-platform-play-how-to-operate-like-a-tech-company> (accessed 12 August 2021).
- Boudreau KJ (2010) Open platform strategies and innovation: Granting access vs. devolving control. *Management Science* 56(10): 1849–1872.
- Ceccagnoli M, Forman C, Huang P, et al. (2012) Cocreation of value in a platform ecosystem: The case of enterprise software. *MIS Quarterly* 36(1): 263–290.
- Ceccagnoli M, Forman C, Huang P, et al. (2014) Digital platforms: When is participation valuable? *Communications of the ACM* 57(2): 38–39.
- Cennamo C and Santaló J (2019) Generativity tension and value creation in platform ecosystems. *Organization Science* 30(3): 617–641.
- Corley KG and Gioia DA (2004) Identity ambiguity and change in the wake of a corporate spin-off. *Administrative Science Quarterly* 49(2): 173–208.
- Cusumano MA (2010) Cloud computing and SaaS as new computing platforms. *Communications of the ACM* 53(4): 27–29.
- Cusumano MA and Gawer A (2002) The elements of platform leadership. *MIT Sloan Management Review* 31(1): 51–59.
- Davison RM and Martinsons MG (2016) Context is king! Considering particularism in research design and reporting. *Journal of Information Technology* 31(3): 241–249.
- De Reuver M, Sørensen C and Basole RC (2018) The digital platform: A research agenda. *Journal of Information Technology* 33(2): 124–135.
- Dosi G, Nelson RR and Winter SG (2000) Introduction: The nature and dynamics of organizational capabilities. In: Dosi G, Nelson RR and Winter SG (eds) *The Nature and Dynamics of Organizational Capabilities*. Oxford: Oxford University Press, p. 22.
- Eaton BD, Elaluf-Calderwood S, Sørensen C, et al. (2015) Distributed tuning of boundary resources: The case of Apple's iOS service system. *MIS Quarterly* 39(1): 217–243.
- Eisenhardt KM and Graebner ME (2007) Theory building from cases: Opportunities and challenges. *Academy of Management Journal* 50(1): 25–32.
- El Sawy OA, Malhotra A, Park Y, et al. (2010) Seeking the configurations of digital ecodynamics: It takes three to tango. *Information Systems Research* 21(4): 835–848.
- Evans DS and Schmalensee R (2010) Failure to launch: Critical mass in platform businesses. *Review of Network Economics* 9(4): 1–25.
- Evans DS, Hagiu A and Schmalensee R (2006) *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries*. Cambridge, MA: The MIT Press.
- Feeny DF and Ives B (1990) In search of sustainability: Reaping long-term advantage from investments in information tech-

- nology. *Journal of Management Information Systems* 7(1): 27–46.
- Förderer J, Kude T, Schuetz SW, et al. (2018) Knowledge boundaries in enterprise software platform development: Antecedents and consequences for platform governance. *Information Systems Journal* 29(1): 119–144.
- Fuentelsaz L, Garrido E and Maicas JP (2015) Incumbents, technological change and institutions: How the value of complementary resources varies across markets. *Strategic Management Journal* 36(12): 1778–1801.
- Ghazawneh A and Henfridsson O (2013) Balancing platform control and external contribution in third-party development: The boundary resources model. *Information Systems Journal* 23(2): 173–192.
- Glaser BG (1978) *Theoretical Sensitivity: Advances in the Methodology of Grounded Theory*. Mill Valley, CA: Sociology Press.
- Glaser BG (2005) *The Grounded Theory Perspective III: Theoretical Coding*. Mill Valley, CA: Sociology Press.
- Goldkuhl G (2012) Pragmatism vs interpretivism in qualitative information systems research. *European Journal of Information Systems* 21(2): 135–146.
- Grant RM (1996) Prospering in dynamically-competitive environments: Organizational capability as knowledge integration. *Organization Science* 7(4): 375–387.
- Grover V and Kohli R (2012) Cocreating IT value: New capabilities and metrics for multifirm environments. *MIS Quarterly* 36(1): 225–232.
- Gulati R (1999) Network location and learning: The network location of network resources and firm influence on alliance capabilities. *Strategic Management Journal* 20(5): 397–420.
- Helfat CE and Raubitschek RS (2018) Dynamic and integrative capabilities for profiting from innovation in digital platform-based ecosystems. *Research Policy* 47(8): 1391–1399.
- Henfridsson O, Mathiassen L and Svahn F (2014) Managing technological change in the digital age: The role of architectural frames. *Journal of Information Technology* 29(1): 27–43.
- Hong W, Chan FKY, Thong JYL, et al. (2014) A framework and guidelines for context-specific theorizing in information systems research. *Information Systems Research* 25(1): 111–136.
- Huang P, Ceccagnoli M, Forman C, et al. (2013) Appropriability mechanisms and the platform partnership decision: Evidence from enterprise software. *Management Science* 59(1): 102–121.
- Karhu K, Gustafsson R and Lyytinen K (2018) Exploiting and defending open digital platforms with boundary resources: Android's five platform forks. *Information Systems Research* 29(2): 479–497.
- Klein HK and Myers MD (1999) A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly* 23(1): 67–94.
- Kogut B and Zander U (1992) Knowledge of the firm, combinative capabilities, and the replication of technology. *Organization Science* 3(3): 383–397.
- Kude T, Heinzl A and Neff A (2017) Achieving IT-based synergies through regulation-oriented and consensus-oriented IT governance capabilities. *Information Systems Journal* 28(5): 765–795.
- Lawton G (2008) Developing software online with platform-as-a-service technology. *Computer* 41(6): 13–15.
- Lepak DP, Smith KG and Taylor MS (2007) Introduction to special topic forum value creation and value capture: A multi-level perspective. *Academy of Management Review* 32(1): 180–194.
- Liu CZ, Au YA and Choi HS (2014) Effects of freemium strategy in the mobile app market: An empirical study of Google Play. *Journal of Management Information Systems* 31(3): 326–354.
- Lorenzoni G and Lipparini A (1999) The leveraging of interfirm relationships as a distinctive organizational capability: A longitudinal study. *Strategic Management Journal* 20(4): 317–338.
- Mata FJ, Fuers WL and Barney J (1995) Information technology and sustained competitive advantage: A resource-based analysis. *MIS Quarterly* 19(4): 487–505.
- Mithas S, Ramasubbu N and Sambamurthy V (2011) How information management capability influences firm performance. *MIS Quarterly* 35(1): 237–256.
- Mithas S, Tafti A, Bardhan I, et al. (2012) Information technology and firm profitability: Mechanisms and empirical evidence. *MIS Quarterly* 31(1): 205–224.
- Ng CS-P and Gable GG (2010) Maintaining ERP packaged software—A revelatory case study. *Journal of Information Technology* 25(1): 65–90.
- Nielsen P and Aanestad M (2006) Control devolution as information infrastructure design strategy: A case study of a content service platform for mobile phones in Norway. *Journal of Information Technology* 21(3): 185–194.
- Oh J, Koh B and Raghunathan S (2015) Value appropriation between the platform provider and app developers in mobile platform mediated networks. *Journal of Information Technology* 30(3): 245–259.
- Ondrus J, Gannamaneni A and Lyytinen K (2015) The impact of openness on the market potential of multi-sided platforms: A case study of mobile payment platforms. *Journal of Information Technology* 30(3): 260–275.
- Pagani M (2013) Digital business strategy and value creation: Framing the dynamic cycle of control points. *MIS Quarterly* 37(2): 617–632.
- Parker G and Van Alstyne MW (2018) Innovation, openness, and platform control. *Management Science* 64(7): 3015–3032.
- Parker G, Van Alstyne MW and Choudary SP (2016) *Platform Revolution*. New York: W. W. Norton & Company.
- Parker G, Van Alstyne MW and Jiang X (2017) Platform ecosystems: How developers invert the firm. *MIS Quarterly* 41(1): 255–266.
- Pavlou PA and El Sawy OA (2006) From IT leveraging competence to competitive advantage in turbulent environments: The case of new product development. *Information Systems Research* 17(3): 198–227.
- Peppard J and Rylander A (2006) From value chain to value network: Insights for mobile operators. *European Management Journal* 24(2–3): 128–141.
- Peterson R (2004) Crafting information technology governance. *Information Systems Management* 21(4): 7–22.
- Priem RL (2007) A consumer perspective on value creation. *Academy of Management Review* 32(1): 219–235.



- Rai A and Tang X (2014) Information technology-enabled business models: A conceptual framework and a coevolution perspective for future research. *Information Systems Research* 25(1): 1–14.
- Rai A, Pavlou PA and Du S (2012) Interfirm IT capability profiles and communications for cocreating relational value: Evidence from the logistics industry. *MIS Quarterly* 36(1): 233–262.
- Ravichandran T and Lertwongsatien C (2005) Effect of information systems resources and capabilities on firm performance: A resource-based perspective. *Journal of Management Information Systems* 21(4): 237–276.
- Ross JW, Beath CM and Goodhue DL (1996) Develop long-term competitiveness through IT assets. *MIT Sloan Management Review* 38(1). Available at: <https://sloanreview.mit.edu/article/develop-longterm-competitiveness-through-it-assets/> (accessed 12 August 2021).
- Sandberg J, Holmström J and Lyytinen K (2020) Digitization and phase transitions in platform organizing logics: Evidence from the Process Automation Industry. *MIS Quarterly* 44(1): 129–153.
- Sarker S (2016) Building on Davison and Martinsons' concerns: A call for balance between contextual specificity and generality in IS research. *Journal of Information Technology* 31(3): 250–253.
- Sarker S, Sarker S, Sahaym A, et al. (2012) Exploring value cocreation in relationship between an ERP vendor and its partners: A revelatory case study. *MIS Quarterly* 36(1): 317–338.
- Sarker S, Xiao X, Beaulieu T, et al. (2018) Learning from first-generation qualitative approaches in the IS discipline: An evolutionary view and some implications for authors and evaluators (part 2/2). *Journal of the Association for Information Systems* 19(8): 909–923.
- Sebastian IM, Moloney KG, Ross JW, et al. (2017) How big old companies navigate digital transformation. *MIS Quarterly Executive* 16(3): 197–213.
- Suddaby R (2006) From the editors: What grounded theory is not. *Academy of Management Journal* 49(4): 633–642.
- Tan B, Pan S, Lu X, et al. (2015) The role of IS capabilities in the development of multi-sided platforms: The digital ecosystem strategy of Alibaba.com. *Journal of the Association for Information Systems* 16(4): 248–280.
- Teece D and Pisano G (1994) The dynamic capabilities of firms: An introduction. *Industrial and Corporate Change* 3(3): 537–556.
- Teece DJ (2018) Profiting from innovation in the digital economy: Enabling technologies, standards, and licensing models in the wireless world. *Research Policy* 47(8): 1367–1387.
- Thomas DWL, Autio E and Gann DM (2014) Architectural leverage: Putting platforms in context. *Academy of Management Perspectives* 28(2): 198–219.
- Tiwana A (2014) *Platform Ecosystems: Aligning Architecture, Governance, and Strategy*. Burlington, MA: Morgan Kaufmann.
- Tiwana A, Konsynski B and Bush AA (2010) Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research* 21(4): 675–687.
- Venkatraman N, El Sawy OA, Pavlou P, et al. (2014) *Theorizing digital business innovation: Platforms and capabilities in ecosystems* (Fox School of Business Research Paper 15-080), pp. 1–36. Philadelphia, PA: Fox School of Business.
- Vial G (2019) Understanding digital transformation: A review and a research agenda. *Journal of Strategic Information Systems* 28(2): 118–144.
- Wade M and Hulland J (2004) Review: The resource-based view and information systems research: Review, extension, and suggestions for future research. *MIS Quarterly* 28(1): 107–142.
- Walsham G (1995) Interpretive case studies in IS research: Nature and method. *European Journal of Information Systems* 42: 74–81.
- Walsham G (2006) Doing interpretive research. *European Journal of Information Systems* 15(3): 320–330.
- Walsham G and Sahay S (1999) GIS for district-level administration in India: Problems and opportunities. *MIS Quarterly* 23(1): 39–66.
- Wareham J, Fox PB and Cano Giner JL (2015) Technology ecosystem governance. *Organization Science* 25(4): 1195–1215.
- Wiesche M, Jurisch MC, Yetton P, et al. (2017) Grounded theory methodology in information systems research. *MIS Quarterly* 41(3): 685–701.
- Yoffie DB, Gawer A and Cusumano MA (2019) A study of more than 250 platforms reveals why most fail. *Harvard Business Review*. Available at: <https://hbr.org/2019/05/a-study-of-more-than-250-platforms-reveals-why-most-fail> (accessed 12 August 2021).
- Zhou KZ and Wu F (2010) Technological capability, strategic flexibility, and product innovation. *Strategic Management Journal* 31(5): 547–561.

### Author biographies

**Maximilian Schrieck** is a postdoctoral researcher at the Chair for Information Systems, Technical University of Munich (TUM), Germany and a visiting scholar at the Wharton School, University of Pennsylvania. He studies governance of digital platforms and enterprise software. His research has been published in *IEEE TEM*, *Electronic Markets*, the *IT for Development Journal* and in conference proceedings such as ICIS, AOM, and ECIS.

**Manuel Wiesche** is a Full Professor and Chair of Digital Transformation at TU Dortmund University. He graduated in Information Systems from Westfälische Wilhelms-Universität, Münster, Germany and holds a doctoral degree and a habilitation degree from TUM School of Management, Technische Universität München, Munich, Germany. His current research interests include IT workforce, IT project management, digital platform ecosystems, and IT service innovation. His research has been published in *MISQ*, *EJIS*, *JMAR*, *CACM*, *I&M*, *EM*, and *MISQE*.

**Helmut Krcmar** is Full Professor of Information Systems and the Chair of Information Systems, Technical University of Munich (TUM), Germany. His research interests include information and knowledge management, service management, and business process management. Among others, his work has appeared in *MISQ*, *JMIS*, *JSIS*, *EJIS*, *JIT*, *ISJ*, and *Wirtschaftsinformatik*.

## Appendix

**Table 5.** Manifestations and illustrative evidence of platform ecosystem capabilities.

Cloud-based platformization	
Seizing control of the code of the ERP core	<p>“With [the cloud platform] we no longer allow [customized extensions to the ERP core] because as we are offering a software-as-a-service solution, we have to be able to operate the software continuously, we always have to be able to upgrade it. This is why there is only code of SAP in [the ERP core], code that SAP can maintain and upgrade, etc. Everything that is to be added as an extension must then be integrated on the cloud platform via standard interfaces [. . .] for reasons of maintainability.” (product owner, cloud platform SDK, SAP; translated)</p> <p>“That is, we still want extensions, but these extensions are no longer supposed to run directly on the [ERP core], but should be deployed on the [cloud platform], but still be sold via SAP.” (program and partner manager, SAP; translated)</p> <p>“We, as [ERP core] development, provide interfaces that are visible in the [cloud platform] and can be consumed, and then only these interfaces are usable and nothing else. Everything else is not visible and there we get a decoupling of the lifecycles, i.e. we can update our [ERP core], while we promise that we keep the interfaces stable.” (development, ERP core, SAP; translated)</p>
Connecting modular third-party applications and services through standardized APIs	<p>“When you say extensions, that means of course [. . .] extensions via APIs. This requires that the SAP cloud solutions, such as SuccessFactors, Ariba, Concur, the S4 Cloud Edition as they are all called, also offer APIs. It was a problem at first because they didn’t have enough APIs; all of that has evolved over time.” (product owner, cloud platform, SAP; translated)</p> <p>“In the past, we also had a huge monolith with a kind of spaghetti architecture [. . .] this decoupling and this modularization, that’s an important aspect that we have made with [the cloud platform] and that allows us to build even small apps and only now this possibility has been opened for extensions on the [cloud platform].” (product and partner governance, SAP; translated)</p> <p>“If partners want to develop extensions now, there is the question where they do it. And for that, [the cloud platform] comes into play. Because from [the cloud platform] they have—and we ensure that—API access to these acquired systems. When we acquire the companies, they sometimes have APIs, often these APIs are not usable, but ultimately, we always work with the acquired companies to get a stable and usable API that is easily accessible through [the cloud platform].” (chief architect, cloud platform, SAP; translated)</p>
Enabling deployment of applications in the cloud via virtualization, containerization, and microservice architecture	<p>“Each application gets a small virtual cage in which it can run and is isolated from the other applications. The second aspect is that you have a standardized deployment format—especially with Docker—[. . .] and of course, this makes deploying anything a lot easier.” (product owner, cloud platform, SAP; translated)</p> <p>“Surrounding these runtimes, we provide different services, for example execute services, we also have mobile and analytical services, we have the IoT services, [. . .] machine learning, deep learning, . . . those will be coming together in the platform. Basically, the many different kind of services that allow a customer or a partner to quickly create an application, go to market and capture the revenue.” (software engineer, cloud platform, SAP)</p> <p>“[the cloud platform] just gives you an entry point to your database, wherever that database is, but with web IDE as an entry point, it has this look and feel of a platform and dashboard where it can just deploy several services and leverage capabilities that [the cloud platform] gives me, like weather service, authentication service. These services are called microservices and a lot of teams are actually developing microservices that then developers can use in their applications. [. . .] And that is why it is a platform.” (software engineer, cloud platform, SAP)</p>
Developing and operating platform-as-a-service in the cloud	<p>“[The cloud platform] means cloud and cloud means operations. [. . .] While with a [on-premises extension] you basically sell to the customer and therefore no longer have much to do, in the cloud, you have the operational part and you need to make sure it works. That means the cloud parameters and the SLAs for cloud are of course completely different and so you have a much higher effort.” (program and partner management, SAP; translated)</p> <p>“Then, the whole topic of Cloudscale is important. At the moment, it’s not only the technical scalability, but the scalability of the teams and services, because we’re still doing some things by hand. [. . .] This simply needs to be more automated. We also need to delegate more to our users through self-services.” (product owner, cloud platform, SAP; translated)</p> <p>“Even when we run [the cloud platform] in a third-party data center, we still control the life cycle of our software there. This means that we update the landscapes at all partners or customers at exactly the same time. We upgrade at the same time as we upgrade our own landscapes. So that everywhere it’s just one software version.” (vice president, cloud platform, SAP; translated)</p>

(Continued)

**Table 5.** (Continued)

## Open IT landscape management

Leveraging open source software and established languages

“We are more open with [the cloud platform] and that is also the general path that SAP wants to follow because we know we cannot deliver best of breed in every aspect and there are a lot of strong open source communities developing simple things like a syntax highlighting editor [ . . . ] but also complex things that allow you to do machine learning and NLP [natural language processing] [ . . . ]. And [the cloud platform] really offers you the capability to deploy such modules—sometimes written in node [node.js], sometimes written in Java. [ . . . ] [The cloud platform] is really opening up and moving away from the trend of just allowing proprietary languages.” (software engineer, cloud platform, SAP)

“[The cloud platform] now offers JS, Java, C++, so the variety of programming languages also increases by moving to [the cloud platform]. And that is the openness we give. Since all microservices are just endpoints you can leverage you don’t necessarily have to write your entire application just in one language.” (software engineering, cloud platform, SAP)

“[the cloud platform] includes a database, in this case HANA, which makes it possible to store data more compactly and also more performantly and this of course also includes a standardization with regard to operating concepts and with regard to the underlying cloud technology and SAP has cleverly used a lot of open source and also offers a lot of open source components, because if you build a pure proprietary solution it is difficult to convince companies that this has a future.” (CEO, ecosystem partner; translated)

Enabling compatibility with technology stacks of competitors

“You can also just put [the cloud platform] on Amazon [Web Services] or Microsoft Azure, which is one of the great advantages of Cloudfoundry. Cloudfoundry—we are in the foundation—so there’s an open source community and Cloudfoundry can also be operated by us, on hyper-scale providers like Amazon Web Services, Microsoft Azure and Google Cloud Platform. [ . . . ] That is multcloud.” (product owner, cloud platform, SAP; translated)

“Nevertheless, the infrastructure side needs to become more cost-effective. And there will always be the customer’s need to use different infrastructure layers or different ecosystems.” (product manager, cloud platform, SAP; translated)

“[In Cloudfoundry] there are so-called APIs, Cloud Provider Interfaces [ . . . ] and there are then for the various IaaS [infrastructure-as-a-service] stacks, there are APIs for vSphere, a API for Amazon AWS and for OpenStack, too. This allows you to move the Cloudfoundry platform itself, the application on top of it, relatively easily to a data center from the customer in the extreme variant.” (vice president, cloud platform, SAP; translated)

Supporting integration with SAP on-premises legacy, other SAP cloud solutions, and third-party solutions

“What the advantage [of the cloud platform] always is, where we have invested from the beginning, is a close integration to with on-premise landscapes that a customer already has and I think that’s the biggest selling point. This, and a close integration with other cloud solutions. We have made acquisitions, SuccessFactors is the most important of them, it is a solution for human resources management. And of course, we make sure that the SuccessFactors solution can work closely with [the cloud platform].” (vice president, cloud platform, SAP; translated)

“The [cloud platform’s] main feature is its good integration with the SAP ecosystem and existing customer base. [ . . . ] New applications can be built and extension applications for existing applications on the cloud or on-premise. So, I can build an app that builds on the cloud platform but accesses the on-premise system.” (product manager, cloud platform, SAP; translated)

“And then the third pillar that I mentioned is the integration pillar. We see people using [the cloud platform] as an integration platform. This essentially means taking, integrating, using [the cloud platform] as an integration layer between various systems whether it be cloud-to-cloud, from SuccessFactors to NetWeaver, to Fieldglass, or an on-premise system to the cloud.” (software engineer, cloud platform, SAP)

## Ecosystem orchestration

Enabling ecosystem partners with standardized boundary resources

“Either available in the online help or on the SCN, the SAP Community Network, there is actually everything [developers need]. Then there is the SAP HANA Academy, there are tons of videos and then there are also the Open SAP courses. All the information is available and it is also possible to get a trial account for the [cloud platform] to just start developing. Of course, there is also access to documentation.” (manager for partner certification, SAP)

“[With the cloud platform we go] more into the mass business, [ . . . ] there is standard support, there are standard resources, as already available in the [cloud platform], standard contracts or packages that you can buy, licenses and they are used accordingly.” (program and partner manager, SAP; translated)

(Continued)

Table 5. (Continued)

Fast onboarding of new ecosystem partners and applications	<p>“But when you decided on the integration on [the ERP core] using [the cloud platform], well then you need [the cloud platform] in order to test whether it works. So, through the PartnerEdge for App Developers we have created a set of tools, resources, access to websites and platform-as-a-service; and all these things are accessible, so when I am ready and want to test my application, I can do so rapidly and easily.” (vice president, business development, SAP)</p> <p>“So how can customers move fast but without doing too much investment on hardware and the answer is to use the cloud platform. Basically, I think the cloud platform is what we call the open platform-as-a-service, that allows customers to do fast developments, agile developments and ability to go to market in a quick time frame.” (software engineer, cloud platform, SAP)</p> <p>“I think the latest tutorial shows it to you in half an hour, including full connection to a database, and even previewing what you just built. [. . .] And that is also a strong point I’d say, that it is so open, in terms of quick and fast prototyping.” (software engineer, cloud platform, SAP)</p> <p>“In the context of [the cloud platform], potential partners have many simple ways to develop something. The entry barrier is low. I have an idea for a web app and get an account for free. I get all the information delivered free of charge. I can just start and just try it out.” (product and partner governance, SAP; translated)</p>
Enforcing ecosystem quality standards with formal control	<p>“There is a vetting process ensuring that the applications are relatively debugged and we have an SAP service verification process so some of the applications are SAP certified. [. . .] Although it is not SAP in terms of legal reasons, we don’t take responsibility for things that might happen from a partner application, but there are certain standards for a partner to meet in order to be in the app center.” (software engineer, cloud platform, SAP)</p> <p>“Then we show them: ‘here are your tools, so you can deploy your applications. If you show us that you have checked all the boxes [. . .] in this whole governance process—then we will make [your application] visible, in public, on the Internet.’” (chief architect, cloud platform, SAP; translated)</p> <p>“Then, partners get access to our cloud platform, documentation, individual contacts, certain guidelines, design guidelines and the like, how an application has to look like, also to have a consistent design of apps. After all, the customer does not want to have a smorgasbord of applications afterwards. So, there are various guidelines.” (product manager, cloud platform, SAP; translated)</p>
Leveraging the existing customer base	<p>“If you look at the reasons to join the ecosystem, of course it’s the market and customer access [. . .] I have the general market access to the SAP market so to speak, I am in the SAP context, I’m already in there, I’m in the app store, that’s also a certain reason, a background noise that sparks the partners’ interest.” (product manager, cloud platform, SAP; translated)</p> <p>“If you look at the markets today, especially in Germany, then SAP has a market penetration [. . .] of over 90% of the ‘relevant’ companies. I don’t think there’s a company that doesn’t have an SAP affinity in any form. So, when you are on this wave today, you naturally have a much easier customer access.” (business development, SAP ecosystem partner; translated)</p> <p>“You must never scare off your existing customer base and these are internal stakeholders or even external SAP customers. [. . .] I don’t think we’re going to turn 180 degrees. In the long term, we must try to get more innovation from the partner side, to generate more sales.” (product manager, cloud platform, SAP; translated)</p>
Enabling strategic ecosystem partners with individualized support	<p>“What we also got is the close partnership with SAP. We have regular sync calls [. . .]. We also have the opportunity to convene expert rounds and then advise or coordinate on architectural issues, clarify our requirements and receive feedback from SAP experts, for example, on the topic of asynchronous messaging [. . .] and how we can best implement this.” (project manager for cloud platform, SAP ecosystem partner; translated)</p> <p>“Basically, for the platform, it is the same as in the sales channel. Of course, there are partners with whom we cooperate more closely, and with whom we then integrate in different scenarios, either because their technology is good or because we have nothing or basically because it is not worth it [for SAP].” (product owner, IoT, SAP; translated)</p> <p>“We provide an entire enablement kit. It has got everything. Presentation on the process they are going through, the documents they need to fill out. We don’t have recordings because these things keep changing too often, so we don’t have them. Then we have a lot of enablement sessions where we prepare them for what to expect to cover the qualification.” (product manager certification, SAP)</p>
Platform evangelism	
Inspiring with a shared vision	<p>“The early days were all about pushing the first wave of adoption among community influencers and bloggers. The first push is always the hardest (see Derek Silver’s ‘How to start a movement’) and hence we’ve been focusing on engaging with thought leaders &amp; multipliers. We shared our vision for the road forward including how to gain adoption and stating the platform’s unique selling proposition.” (SAP blog entry, 12 December 2016)</p>

(Continued)



Table 5. (Continued)

Establishing platform brand image as an open platform	<p>“We have an entire team, called the developer relations team. They are really the ones to manage our relations between not just the cloud platform but various developer technologies and our external developer community. For example, we post these things or they post these things called code jams. They essentially are an event where the developer relations team will come to hosts [and] they will spend the whole day with you or them and go through a series of learning tutorials to take you and kind of show you the capabilities of these products.” (software engineer, cloud platform, SAP)</p> <p>“We are assembled here today to prepare you for your exams to become future Starfleet commanders of the United Federation of Planets. You and your crew will explore strange new worlds, you will seek out new civilizations and you will boldly go where no one has gone before. As part of your exam, there is the famous and feared Kobayashi Maru challenge. Please repeat after me: ‘Kobayashi Maru!’” (executive keynote, SAP TechEd 2017)<sup>a</sup></p> <p>“Bjoern Goerke, CTO at SAP and Cloud Platform President explains that Cloud Foundry is the best choice for SAP Cloud Platform’s new multicloud architecture, as it “is supported by all major cloud providers.” (Cloudfoundry blog entry, 22 May 2017)</p> <p>“We have a twitter channel which we use primarily as a way to farm out information, developer-centric information about the platform, answer some questions that people have, we more suggest that people put their questions on the community or submit a ticket, but we also answer questions on twitter. We farm out new information on product management, about new features coming out.” (software engineer, cloud platform, SAP)</p> <p>“The cloud platform keeps our community very developer-centric. If you go search for [cloud platform] on the developer community or the SAP community a lot of blogs are based on the resources that developers outside of SAP have created. Tutorials, update information and there is various learnings and findings by going through the process of developing on the platform.” (software engineer, cloud platform, SAP)</p>
Convincing existing partners to onboard the cloud platform	<p>“We have to convince them! There is no choice, we have to convince them. We have to help them. We have to enable them. But it is absolutely critical that we manage to migrate them.” (vice president, business development, SAP)</p> <p>“In the [cloud platform] ecosystem, the simplicity of the tools we provide are success factors. That we have a web portal and you can download information there and the efficiency of our marketing that you also reach out to the partners. There are many thousands and somehow you have to graze the market.” (product and partner governance, SAP; translated)</p> <p>“The [cloud platform] and microservices model only scales if you also convince appropriate partners. This is part of our strategy, which is already moving towards implementation, but here too, the partners are an important aspect.” (chief operating officer, SAP; translated)</p>
Platform co-selling	
Running an online marketplace	<p>“‘Cloud’ is a special topic and it is contingent on having a store and which conditionally has an extension accessibility framework, which allows small partners to quickly build apps and make them available to a large customer base, otherwise it doesn’t pay off.” (products and innovation management, SAP; translated)</p> <p>“We can get [the marketplace] and work a new audience. SAP as you know has been specifically good in selling directly to the largest companies for example in the Fortune 500. There is where we sell but there are many others where we don’t sell.” (vice president, business development, SAP)</p> <p>“The idea is to create an application marketplace and yeah build like a fully stocked marketplace of applications that you can not only try out but also then purchase directly from the app center.” (software engineer, cloud platform, SAP)</p>
Combining platform, ecosystem partner, and direct sales channels	<p>“The big [partners], they have their own sales track, [ . . . ] so for example, I would say they have almost as many salespeople as we do. They sell their applications or their integrations, which they have built, themselves. [ . . . ] It’s like that with the big partners, with the small partners it’s more that they want to be sold through us, because of course they don’t have a big sales track.” (product owner, IoT, SAP; translated)</p> <p>“Certain customers, I can only access via SAP. Let’s imagine a small app provider, 10-20 employees, who is not necessarily invited to a procurement project at Bosch and is probably not included in the selection process at all. If he sells together with SAP, then yes. This is the customer access.” (product manager, cloud platform, SAP; translated)</p> <p>“The partner ‘outbounds’ our platform. He sells our platform and he adds an application to it. That means, we have a second benefit. An [SAP ERP] reseller with its application on top of it.” (partner manager, SAP; translated)</p>

(Continued)

**Table 5.** (Continued)

Identifying optimal pricing across ecosystem participants	<p><i>“The whole market and entire pricing models will move towards usage-based billing. Why? Because it gives you flexibility and that’s actually the point. You always want economies of scale, flexibility, more speed, digital transformation in the cloud. You want to test something for a short time without tying yourself to someone for three years, but these offers are actually interesting. Whether you have more revenue after that, time has to tell.”</i> (product manager, cloud platform, SAP; translated)</p> <p><i>“There are also big partners, so you want to be flexible in discounting. In the past, it was strict and therefore was not flexible enough for the big partners. And for the small partners, we were too expensive because they were told if you have an application you pay us so much. The fact that they have costs because they have to pay for [the cloud platform license] was not taken into account at the time.”</i> (product owner, cloud platform, SAP; translated)</p> <p><i>“We need to keep the price as low as possible, because we simply have to establish the [cloud platform] as a platform along with the largest possible ecosystem. And if we just say, for small companies, you have to put down a lot of money, then you are left alone because you don’t get any support [. . .] this just won’t fly.”</i> (program director, sales and services, SAP; translated)</p>
---	--

ERP: enterprise resource planning; SDK: software development kit; API: application programming interface; NLP: natural language processing; ABAP: advanced business application programming; HANA: high performance analytic appliance; CEO: chief executive officer; IoT: internet of things; CTO: chief technology officer.

<sup>a</sup>Star Trek is an American science fiction franchise based on a TV series aired in the 1960s and written by Gene Roddenberry. Star Trek has inspired a cult phenomenon since then with numerous films, TV series, comics, and novels around the starship USS Enterprise and its captain, James T. Kirk. The Kobayashi Maru challenge that the platform evangelist refers to is part of the training of Starfleet cadets.