Technische Universität München
TUM School of Management

Blockchains' Potential to Reshape the Economy: An Examination of Trust and Decentralization in Smart Contract-Based Applications

Daniel Quirin Obermeier

Vollständiger Abdruck der von der TUM School of Management der Technischen Universität München zur Erlangung eines

Doktors der Wirtschafts- und Sozialwissenschaften (Dr. rer. pol)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Oliver Alexy

Prüfer*innen der Dissertation:

1. Prof. Dr. Joachim Henkel
2. Prof. Aija Leiponen Ph.D.
3. Prof. Jean-Philippe Vergne Ph.D.

Die Dissertation wurde am 24.04.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Management am 15.07.2023.

angenommen.

Table of contents

List of abbreviations

API	Application programming interface
BFT	Byzantine fault tolerance
DAO	Decentralized autonomous organization
dApp	Decentralized application
DeFi	Decentralized finance
Defi	Decentralized finance
DPoS	Delegated proof of stake
EIP	Ethereum improvement proposal
EOA	Externally owned account
ERC	Ethereum request for comments
ETH	Ether
EVM	Ethereum virtual machine
ICO	Initial Coin Offering
IoT	Internet of things
IPFS	Inter planetary file system
LISREL	Linear structural relations
MB	Mega bytes
NFT	Non-fungible token
PLS	Partial least squares
PoS	Proof of stake
PoW	Proof of work
SHA	Secure Hash Algorithm
URL	Uniform resource locator

List of figures

List of tables

Zusammenfassung

Der Blockchain-Technologie wird nachgesagt, dass sie einen Paradigmenwechsel in unserer digitalen Wirtschaft einleitet, indem sie die Notwendigkeit des Vertrauens in Transaktionen beseitigt und digitale Plattformen durch ein dezentrales und verteiltes Konsensprotokoll ersetzt und somit zu einer neuen, transparenteren, integrativeren und demokratischeren Ära des Internets führt, die oft als Web 3.0 bezeichnet wird. Doch so vielversprechend diese neue Ära auch sein mag, die jüngste Abkühlung des Kryptowährungsmarktes, die ständige Verzögerung der versprochenen bahnbrechenden Updates und die andauernde Suche nach einer "Killer"-Anwendung haben die Blockchain-Technologie in das Tal der Desillusionierung getrieben. Um dabei zu helfen zwischen Hype und tatsächlichem Potenzial zu differenzieren und die neue Technologie dabei zu unterstützen dieses Tal zu verlassen, untersucht diese Dissertation in drei Studien zwei zentrale Versprechungen der Blockchain-Technology: Die Schaffung eines vertrauen freien Systems und die Disintermediation von Plattformen.

Die erste Studie argumentiert, dass Smart Contracts die Notwendigkeit des Vertrauens in Transaktionen nur theoretisch beseitigen können, dies aber in der Praxis unwahrscheinlich ist, da es voraussetzen würde, dass die Nutzer den Quellcode des Smart Contracts lesen und vollständig verstehen. Stattdessen ist es wahrscheinlicher, das Smart Contracts eine neue Form des Vertrauens ermöglichen, die auf der Möglichkeit beruht, den Quellcode zu lesen. Anhand einer Stichprobe von 526 Smart-Contract-basierten Anwendungen auf Ethereum zeigt diese Studie, dass diese neue Form des Vertrauens die traditionelle Vertrauensbildung ergänzt und es den Anwendungen ermöglicht, mehr Nutzer anzuziehen.

Aufbauend auf den theoretischen Erkenntnissen der ersten Studie wechselt die zweite Studie zur Nutzerperspektive und untersucht, wie dispositionelle und institutionelle Faktoren das Vertrauen der Nutzer in Smart-Contract-basierte Anwendungen beeinflussen. Die Studie entwickelt ein neues Modell zur Vertrauensbildung und testet dieses Modell mit Hilfe einer neuartigen Umfrage-app, die speziell für diese Studie entwickelt wurde.

Die dritte Studie untersucht, wie der Ersatz einer zentralen Plattformautorität durch einen dezentralen Marktmechanismus die Nutzung von dApps beeinflusst. Die Ergebnisse dieser Studie zeigen, dass der derzeitige auktionsbasierte Mechanismus zur Zuteilung des begrenzten Transaktionsangebots Finanz-DApps gegenüber anderen DApps bevorzugt und langfristig zu einer Verringerung der Heterogenität von DApps führt. Diese Verringerung ist besonders problematisch, da sie das Ziel von Ethereum, eine breite Vielfalt von dApps zu hosten, entgegenwirkt und in Frage stellt, ob ähnliche Plattformen als Infrastruktur des Web 3.0 dienen können.

Abstract

Blockchain technology is hailed for inducing a paradigm shift in our digital economy by removing the need for trust in transactions and disintermediating digital platforms by substituting a central authority with a decentralized and distributed consensus protocol. This paradigm shift has given rise to the hope that blockchain technology will lead the way to a new, more transparent, inclusive, and democratic era of the internet, often referred to as Web 3.0. But however promising this new era might be, the recent cooldown of the cryptocurrency market, the perpetual delay of promised breakthrough updates, and the ongoing search for a ‘killer’ application have forced blockchain technology into the trough of disillusionment. To help lift blockchain technology out of this phase, in three studies, I investigate two of blockchain technology’s key claims: creating a supposed trust system and substituting a central platform authority with a decentralized market mechanism. With these studies, I aim to contribute to a better understanding of what is just hype and what is the real potential of this novel technology.

In the first study, I theorize about smart contracts' potential to remove the need for trust in transactions by predefining all rules and triggering transaction conditions in immutable computer code. I argue that it is unlikely smart contracts will do away with the need for trust in transactions as this would require users to read and fully understand the smart contract’s source code. Instead, smart contracts enable a new and distinct form of trust based on the possibility to read the source code. Using a sample of 526 smart contract-based applications on Ethereum, I show that this new type of trust complements traditional trust formation and allows apps to attract more users.

Building on the theoretical insights from the first study, I switch to the user perspective in the second study to investigate in greater detail whether dispositional and institutional factors influence how users form trust in smart contract-based applications. I develop a new trust formation model and test this by leveraging a decentralized survey application specifically developed for this study.

The third study investigates how substituting a central platform authority ensuring the correct execution of transactions with a decentralized market mechanism influences the use of dApps on the blockchain platform. This study’s findings suggest that the current auction-based mechanism for allocating the limited supply of transactions favors finance dApps over others and in the long run reduces dApp heterogeneity. This reduction is particularly problematic as it thwarts Ethereum’s goal to host a broad scope of dApps and questions whether similar platforms can serve as backbone for Web 3.0.

1 Introduction

1.1 Blockchain technology and its potential to shape the future of the digital economy

Launched in 2009 by a developer (or a team of developers) under the pseudonym Satoshi Nakamoto, Bitcoin was the first implementation of a blockchain. What started out as a decentralized, peer-to-peer, and disintermediated payment system to compete with traditional centralized financial institutions, today has developed into a thriving startup ecosystem. It includes diverse flagship projects by big companies spanning various industries, for example healthcare (Pfizer), insurance (AIG), energy (Siemens), government (the government of Dubai), logistics (Maersk), and travel (British Airways),[1] and its own cryptocurrency industry with a market capitalization currently exceeding $1.07 trillion.[2]

At its core, a blockchain is a distributed transactional database secured by cryptography and a decentralized consensus mechanism (Catalini & Gans, 2020; Halaburda, 2018; Werbach, 2018). As it distributes numerous copies of the same database across a peer-to-peer network and only allows a new entry if all network parties reach a consensus on its validity, a blockchain enables the disintermediation of centralized systems by replacing the central authority with a previously unseen class of validators called “miners” while remaining extraordinarily tamper-resistant (Hsieh, Vergne, Anderson, Lakhani, & Reitzig, 2018).

The hype around this new technology reached its breakthrough when Ethereum cofounder Vitalek Buterin published the Ethereum white paper in 2014, thereby initiating the second wave of innovation in blockchain technology. This wave expanded the functionality of blockchains beyond mere “record-keeping.” It introduced a second generation of blockchains hosting self-enforcing computer programs that are immutably stored on the blockchain and run without risk of downtime or censorship. To honor Nick Szabo (1994), the pioneer who first envisioned computerized transaction protocols that automatically execute terms of contracts without the need for trusted intermediaries, these programs are referred to with his term: *smart contracts*.

Empowered by such smart contracts, platforms like Ethereum grew beyond conducting simple cryptocurrency transfers to fully-fledged multi-sided marketplaces where any party can offer arbitrary services accessible to everyone with a web browser, in the form of decentralized applications, or “dApps” (Wu, Ma, Huang, & Liu, 2021). These dApps visually resemble ordinary web applications but instead of running on a centralized platform, they connect to the blockchain via smart contracts and use blockchain records as their transaction data

[1] https://101blockchains.com/companies-using-blockchain-technology/, accessed September 15, 2022.
[2] https://coinmarketcap.com/, accessed September 15, 2022.

(Leiponen, Thomas, & Wang, 2021). Currently, more than 3,475 dApps are running on Ethereum alone, offering services like advertising (e.g., basicattentiontoken.org), cloud-storage (e.g., storj.io), collectible games (e.g., cryptokitties.co), encrypted messengers (e.g., status.im), insurance (e.g., insurancefi.io), online casinos (e.g., fairspin.io), or prediction markets (e.g., augur.net).[3]

Though still in their infancy, these dApp platforms ultimately aim to provide the infrastructure for a new version of the internet—a fully decentralized, democratized, and fair version where users control their own data and identity. Gavin Wood, another Ethereum co-founder, coined this version of the internet *Web3*.[4] Web3 promises to empower a new blueprint of decentralized and distributed digital platforms that take up the battle with their centralized counterparts currently dominating the digital economy.[5] This shift towards decentralized and distributed digital platforms is desirable from an antitrust perceptive as it might be the only effective way to prevent a dystopian oligopoly of a few unaccountable platform behemoths with almost unlimited market power (Vergne, 2020). However, it also implies that the new platforms must compete with powerful platform players such as Amazon, Airbnb, Apple, Facebook, Google, and Uber.

To survive the competition, blockchain platforms need to offer users distinct benefits. In recent years, public debates have praised these benefits time and time again. For instance, the Economist has twice advocated these new platforms' potential: by introducing blockchain as the "the trust machine" that removes the need for trust in transactions (Economist, 2015); and later by presaging the disintermediation of various industries and speculating about the redundancy of organizations and even governments in a world run by blockchain technology (Economist, 2017).

Researchers have also jumped on the bandwagon to investigate this new technology's potential. Led by scholars in computer sciences and information systems, researchers have assessed the security of blockchain platforms (e.g., Kosba, Miller, Shi, Wen, & Papamanthou, 2016) examined promises such as the immutability and tamper-resistance of smart contracts (e.g., Fröwis & Böhme, 2017), and built first proofs-of-concept (e.g., Beck, Czepluch, Lollike, & Malone, 2016). These studies sparked researchers in economics and management to join in the efforts to recognize blockchain technology's potential. Economists, for example, theorized that blockchains reduce transaction costs (Catalini & Gans, 2020), discussed how smart contracts might reshape firms' boundaries (Halaburda, Levina, & Min, 2019), or used game theory to analyze the stability of the mining process and its implications for miners and

[3] https://dappradar.com/rankings/protocol/ethereum, accessed September 15, 2022.

[4] https://www.wired.com/story/web3-gavin-wood-interview/, accessed September 15, 2022.

[5] According to Jenifer Schenker (https://innovator.news/the-platform-economy-3c09439b56), nearly 30 percent of the global economy is mediated by centralized digital platforms.

users (Basu, Easley, O'Hara, & Sirer, 2019; Easley, O'Hara, & Basu, 2019). Management researchers questioned the need for centralized organizations when transactions are conducted on blockchains (Seidel, 2018), described Bitcoin as a new paradigm for organization design (Hsieh et al., 2018), and theorized that blockchains might reshape the organization of collaboration by providing an alternative to contractual and relational governance (Lumineau, Wang, & Schilke, 2020; Murray, Kuban, Josefy, & Anderson, 2019).

Although these efforts greatly helped to understand the technology and its implications, with the exception of Hsieh and Vergne (2022), most were only theoretical and turned a blind eye to how this new technology's potential has already materialized and is reshaping the digital economy. Empirical evidence of how blockchain technology has reshaped the strategies of companies providing services on blockchain platforms, and their users' behavior, is currently scarce. Without this evidence, however, it is difficult to gauge whether the new technology can live up to the high expectations.

This dissertation aims to fill this void by scrutinizing blockchain technology's main claims and providing empirical evidence of the implications for companies offering their services on blockchain platforms as well as for their users.

1.2 Research objectives, context, and designs

The central theme of this dissertation is to investigate the implications of blockchain technology's acclaimed promises: (1) create supposedly trust-free systems by removing the necessity of trust from transactions; (2) disintermediate digital platforms (substitute a centralized platform intermediary with a decentralized transaction verification mechanism).

The second unifying element of this dissertation is the research topic linking all three studies. For my investigation, I focus on dApps, the complements offered on blockchain platforms. They are provided by third-party developers who use them to implement arbitrary use cases. Without dApps, the usability of blockchain platforms would be restricted to simple money transfers. Hence, today, dApps mediate almost all transactions on a blockchain that are not direct transfers of crypto currencies between users. As a platform's success highly depends on successful complements (Rietveld & Schilling, 2020), attracting effective dApps is crucial for blockchain platforms to succeed. Yet, despite their importance, very little research has focused on dApps. For example, Wu et al. (2021) took a first look at dApps, describing their key characteristics and providing an overview of the various dApps deployed on Ethereum and usage indicators. Leiponen et al. (2021) described in more detail the dApp ecosystem and its general architecture, discussing how it could foster distributed innovation. Beyond these initial accounts, there has been scarcely any research looking into the impact of blockchain technology on the complements offered on these novel platforms. If these platforms are as novel as claimed, dApp providers really need to understand the implications in order to align their strategy accordingly. This dissertation therefore investigates the implications of the two paradigm changes that blockchain technology promises for dApps and their providers.

To achieve this overarching goal, each of the three studies focuses on a research objective addressing how blockchain technology's key promises impact dApps. The first (Chapter 3) and second (Chapter 4) study address the impact of a supposedly trust-free system on dApps. The third study (Chapter 5) investigates the impact of disintermediation on dApps heterogeneity on blockchain platforms.

Next, I describe in more detail blockchain technology's promises and how these relate to my research objectives.

Project 1: Smart contracts on a blockchain: Transaction governance with the potential of deductive certainty

The first promise I investigate is that blockchain technology creates a supposedly *trust-free* system by removing the need for trust in transactions. This characteristic of blockchain platforms will remove the burden of searching for latent cues that allow us to gauge the trustworthiness of our transaction partners and expand our transaction activities to people we usually would not trust (Greiner & Wang, 2015). This claim was particularly hyped by the Economist, which called blockchain "the trust machine" (Economist, 2015). Researchers also picked up the notion of a trust-free blockchain-based system and investigated its characteristics. Beck et al. (2016) developed a proof-of-concept for a supposedly trust-free coffee shop payment solution, concluding that it could potentially change many existing trust-based transaction systems. Hawlitschek, Notheisen, and Teubner (2018) conducted a literature review on blockchain technology and trust in the sharing economy to study the limitations of supposedly trust-free systems. They introduced the notion of a trust frontier, arguing that only the core of blockchain-based transactions runs without trust, but whenever a connection is needed with the off-chain world and human behavior, trust is still necessary. More recently and with a slightly different perspective, Lumineau et al. (2020) proposed blockchain technology as a new governance mechanism to organize collaboration. They theorized that blockchains can act as both a substitute for or complement contractual and relational governance that relies on legal contracts or trust to achieve cooperation and coordination during a transaction. All this literature does not provide a definite and empirically grounded view regarding the role of trust on blockchain platforms. There is a lack of research on how companies actively using such platforms should operate and what trust-building strategies are required to attract users. Accordingly, the first study investigates the following research objective:

> **Research objective 1:** Investigate whether blockchain technology and smart contracts change how transactions can be governed and to what extent dApp providers could use this new technology to build trust in new exchange relationships.

Given the importance of trust for governing exchange relationships (Poppo & Cheng, 2018), particularly in electronic commerce (Gefen, Karahanna, & Straub, 2003), addressing this objective is not only highly relevant for management and information systems theory, but also for companies that currently offer or are considering offering their services on a blockchain platform. Furthermore, it is important to unravel the implications and limitations of a supposedly trust-free system and understand to what extent blockchain technology will foster more inclusive international commerce by allowing companies currently excluded due to their untrustworthy cultural background or legislation, to participate in international exchange relationships.

Working jointly with Joachim Henkel and building on the governance, trust formation, and information processing literature, I addressed this objective by first theorizing that smart contracts can enable the formation of a new type of *deduction-related trust* based on a purely deductive process (i.e., reading a smart contract's source code) instead of inductively processing traditional trust cues (e.g., reading information on the dApp's website). Then, I compared this new type of trust with traditional induction-based trust formation, before finally testing, based on a sample of 536 dApps on Ethereum and a moderated OLS regression, which trust-building strategy leads to more exchange relationships.

Project 2: How do you trust in a trust-free system? Exploring trust formation in dApps on blockchains

The second study is closely related to the first study and complements it by investigating the trust formation process from a user perspective. It studies what types of trust cues users are looking for, how they process and consider them when deciding to interact with a dApp. This study extends prior research developing trust-building models for traditional web applications (e.g., Gefen et al., 2003; McKnight, Choudhury, & Kacmar, 2002b) to the realm of dApps on a blockchain. This extension is important because the first study showed that dApps which allow the formation of deduction-related trust are more successful at establishing exchange relationships. As existing trust-building models do not account for this new type of trust, they fail to explain this finding and hence require updating. Accordingly, the second study addresses the following research objective:

> **Research objective 2:** Investigate how users form trust in applications running in a supposedly trust-free system, then develop a trust formation model that accounts for the possibility of deduction-related trust.

To tackle this objective, working together with Joachim Henkel, I used the insights generated in the first study, the theory of reasoned action (TRA) (Fishbein & Ajzen, 1975), and prior trust-building models (e.g., Gefen et al., 2003; McKnight et al., 2002b, 2002a; McKnight, Cummings, & Chervany, 1998) to propose a new trust-building model that describes how users form trust and decide to transact with a dApp. To test this model, I developed an online questionnaire and used a novel survey tool to administer the survey. The tool is a survey dApp which I developed together with Johannes Weiss specifically for this study (Weiss & Obermeier, 2021). The key feature of the dApp is that it enables you to send survey invitations to dApp users on the blockchain. As survey participants had to submit their answers by sending a transaction to the smart contract, I could pseudonymously link their responses to their past transaction history and thus their past trusting behavior. Linking survey research to real trusting behavior—instead of measuring the intention to trust—is novel to the trust formation literature and is thus an additional contribution.

Project 3: Competition in a Market for Transactions: The Effect of Ethereum's Gas Price Mechanism on dApp Heterogeneity

Blockchain technology's second promise is disintermediation, that as a logical consequence of a trust-free system, is therefore closely related to the first promise. If trust in the transacting party on blockchain platforms is no longer necessary, then the role of a centralized intermediary, typically to create an institutional environment where strangers can trust each other, changes dramatically, or even becomes entirely dispensable (Mehrwald, Treffers, Titze, & Welpe, 2019). Consequently, disintermediation promises to enable the creation of digital platforms that can leverage the advantage of centralized platforms (facilitate exchange between parties who are strangers and benefit from network effects), without the potential downsides of a centralized intermediary: excessive market power, hold-up problems in contracts with platform participants, and control over participants' data, (Catalini & Gans, 2020; Vergne, 2020). However, proponents of disintermediation often overlook that even blockchain platforms require intermediaries to verify and enforce transactions. These new intermediaries, called miners or verifiers, also need an incentive to provide their services. The only difference is that they are distributed and decentralized and thus have less power over platform participants than their centralized counterparts (Catalini & Tucker, 2018). Although desirable from an antitrust perspective, replacing the enforcement of transactions with a central authority and numerous decentralized verifiers has considerable disadvantages. As scholars have shown, even mining that started out fully decentralized gravitates towards centralized mining pools with considerable market power (Cong, He, & Li, 2021). Because these mining pools are not legally registered corporates, they are more difficult to regulate than centralized platform companies. Another disadvantage is that coordinating many decentralized verifiers increases costs. To keep these costs manageable and ensure a sufficient degree of decentralization, the number of transactions a blockchain platform can process is typically limited.[6] This limited transaction supply is often referred to as the 'scalability problem' with blockchains.[7] To resolve the limited supply issue and incentivize miners to verify transactions, blockchains use a market mechanism that allocates the transaction verification service to the transaction sender with the highest willingness to pay for the transaction. Especially in times of network congestion, this mechanism has led to skyrocketing transaction fees and what the public media bemoan as Ethereum's 'gas fee crisis.'[8] Scholars have thus recently empirically investigated the dynamics surrounding transaction fees as a result of decentralized transaction verification. They employed game theoretic models to assess the implications of Bitcoin's transaction fee mechanism for miners (Easley et al., 2019) and users (Basu et al., 2019). Or they applied a supply and demand perspective to study the long-run stability of Bitcoin's transaction fee mechanism (Ilk, Shang, Fan, & Zhao, 2021). In the context of Ethereum, Donmez and Karaivanov (2021) found that network congestion and urgency of transactions are the major drivers of transaction fees. There has been less empirical focus on the consequences of the decentralized verification of transactions and the accompanying transaction fees for dApps. Because all dApps on a blockchain platform must compete for the limited supply of transactions, it is not clear which applications can survive and thus what type of applications blockchain platforms will offer in the future. This issue is especially pressing as blockchain platforms put a strict limit on platform providers' strategic tools for protecting complements if necessary (e.g., through subsidies), thus hindering their ability to orchestrate an appealing ecosystem of platform complements. Therefore, I propose the following research objective:

[6] At the time of writing, Bitcoin processes 7 and Ethereum 15 transactions per second.

[7] https://www.gemini.com/cryptopedia/blockchain-trilemma-decentralization-scalability-definition, accessed September 15, 2022.

[8] https://www.coindesk.com/markets/2018/07/06/ethereums-growing-gas-crisis-and-whats-being-done-to-stop-it/, accessed September 15, 2022.

> **Research objective 3:** Explore how Ethereum's transaction verification mechanism impacts both the use of dApps on a blockchain platform and the heterogeneity of dApps offered on Ethereum in the long run.

In collaboration with Hanna Halaburda, I addressed this research objective by conceptualizing Ethereum as a market for transactions where dApps compete for the limited supply of transactions and where the platform provider's strategic tools to orchestrate an appealing ecosystem of complements are limited. Based on this conceptualization, I then analyzed how the diverse characteristics of a dApp affect its sensitivity towards transaction fees. Empirically, I leveraged daily transaction records of 1,590 dApps on Ethereum covering three years, together with a novel instrumental variable to estimate the different demand curves and price elasticities. Theoretically, this study is important because it shows that the decentralized verification of transactions aggravates competition on platforms by introducing an additional externality to using dApps. It also provides insight into what types of dApps this form of competition favors. From a practical perspective, this study has important implications for dApp providers, platform providers, and policymakers: it helps dApp providers better assess the competitive landscape on blockchain platforms and informs their entry or exit decisions; for blockchain platform providers, it highlights the transaction verification mechanism as an important tool for complement orchestration; and finally for policymakers, thanks to this new form of competition, blockchain platforms might be less imperiled by "winner-takes-it-all" dynamics. Given the current decentralized transaction verification mechanism, it is unlikely that one blockchain platform can become a general purpose platform hosting the full spectrum of dApps.

Beyond the three objectives discussed above, a further goal is to showcase the richness and usefulness of blockchain data. To this end, each of the three studies applies a different theoretical perspective, a different methodological approach, and a different way to complement blockchain data with meaningful off-chain information. I chose these different approaches not only to learn and extend my skillset but also to inspire other scholars in the fields of management, information systems, and economics to follow my example and leverage the abundant information in public transaction records stored on a blockchain to study new platform phenomena and push our frontiers of knowledge.

1.3 Structure of this dissertation

The structure of this dissertation is as follows: Chapter 2 outlines the foundations of blockchain technology, thus equipping readers who are new to the field with the basic knowledge required to read the subsequent chapters. Readers already familiar with the basics of blockchain technology may want to skip this chapter or only read the sections that are new to them. To ensure chapters are all self-contained, in addition to this general overview, each chapter outlines specific and relevant elements of blockchain technology. The background in Chapter 2 comprises the following subchapters: blockchain basics (2.1), smart contract basics (2.2), a primer on tokens, coins, cryptocurrencies (2.3), an introduction to decentralized applications (2.4), fundamental features of the Ethereum blockchain, the context for all three studies (2.5), and an illustrative example of an interaction with a dApp on Ethereum (2.6).

Chapter 3 theorizes and provides initial empirical evidence that blockchain technology, particularly smart contracts, will not remove the need for trust in transactions with dApps but offers a novel way to govern transactions, resulting in a new form of trust. It shows that successful dApp providers are already considering this in their trust-building efforts by offering trust cues that allow the formation of this new type of trust (Research objective 1). This chapter is based on joint work with Joachim Henkel. The introductory Section (3.1) outlines the new mechanism's smart contracts provided to form trust and explains how this study contributes to the pertinent literature. Then Section (3.2) reviews the fundamental features of blockchain technology and smart contracts and discusses how they can lead to a new form of transaction governance. Section 3.3 theorizes how governing a transaction with a smart contract can lead to a new type of belief about the reliability of an exchange relationship and introduces the notions of deductive certainty and deduction-related trust. Section 3.4 describes our methodological approach and data, while Section 3.5 presents our results, including robustness checks. The concluding Section 3.6 discusses the implications of this study and outlines avenues for further research.

Chapter 4, also based on joint work with Joachim Henkel, switches from the dApp providers' perspective to the users' perspective. It describes the new trust formation model we developed that seeks to explain how users form trust in decentralized applications (Research objective 2). This is linked to Chapter 3 as it also builds on the notions of deductive certainty and deduction-related trust. Unlike in Chapter 3, however, this perspective is not rooted in the literature on governance mechanisms but in the trust formation literature and provides more detailed insights into users' thought processes. After the introduction (4.1), I define trust, review existing trust formation models, and explain how the context of a dApp transaction likely differs from forming trust in normal web applications (4.2). Based on this knowledge, I develop a new model of trust formation that accounts for dApps' novel trust cues (4.3). To test this model, I use a survey sent directly to dApp users through my survey dApp that I developed for this project. This dApp allowed me to collect data from real dApp users and pseudonymously link their survey responses to their transaction history. Section 4.4

describes how I compiled the survey and used the dApp to collect the data. Section 4.5 presents the outcome of the survey analysis and Sections 4.6 and 4.7 conclude by discussing the limitations and implications of this study.

Chapter 5 examines the impact of replacing a centralized platform authority with a decentralized transaction verification mechanism on the use of complements offered on a blockchain platform (Research objective 3). This chapter, based on joint work with Hanna Halaburda, leans more towards the field of economics, and therefore has a slightly different structure. After the introduction (5.1), I outline previous studies investigating the dynamics surrounding transaction fees on blockchain platforms and the literature on platform competition (5.2). Section 5.3 provides background information on Ethereum's transaction verification mechanism and the resulting market for transactions. In Section 5.4, I propose a conceptual framework that details the consequences of Ethereum's transaction verification mechanism for the heterogeneity of dApps on the platform and provides intuition for the subsequent empirical investigation. I then describe the sample and data sets (5.5) and explain the identification strategy underlying my empirical analysis (5.6). Section 5.7 presents the results of different demand curve estimation models along with robustness checks. My survival analysis in Section 5.8 assesses how Ethereum's gas price impacts the hazard ratio of different types of dApps. I end by discussing these studies' limitations and implications for platform complementors (dApp providers), platform providers, and policymakers.

Chapter 6 concludes by reflecting on our key findings and contributions to theory and practice. I propose fruitful avenues for future research that build on the studies presented in this dissertation and can advance our understanding of blockchain technology's merits.

2 Foundations of Blockchain Technology

This chapter aims to give readers who have little or no experience in the field of blockchain technology the knowledge they need to understand the subsequent chapters. Since the information is aimed at general management and economics scholars, I compromise on technical details in favor of understandability where necessary and only focus on the major concepts. For a more technical review, see (Antonopoulos, 2018) or Bashir, (2020). It is important to note that the blockchain space is a fast-evolving field where knowledge can quickly become outdated. As this thesis aims to document the current state of the technology, future readers will have to keep this in mind and take this chapter rather as a snapshot that requires updating than an all-time valid description. Furthermore, although more blockchains are built around the same principles, they can differ in their technical implementation. As the main context here is the Ethereum blockchain, which serves as a role model for many other smart contract-enabling blockchain platforms, I focus this review on Ethereum.

2.1 Blockchain basics

In its generic form, a blockchain is one possible representation of decentralized ledger technology (Beck, Avital, Rossi, & Thatcher, 2017). Decentralized ledgers are databases in the form of append-only event logs shared between networked parties (Rückeshäuser, 2017). Adding to this event log requires consent from all networked parties. Consensually updated and stored by all participating parties, a blockchain ensures the integrity of its data by making it prohibitively costly for any party to change the data unilaterally. In this way, a blockchain provides a shared ground truth that everyone can rely on at the same time (Risius & Spohrer, 2017).

Blockchain technology is not an innovation but rather a recombination of existing technologies (Halaburda, 2018). Therefore the term blockchain is used here as shorthand for a combination of technologies (e.g., cryptography, peer-to-peer networks) comprising certain properties (Antonopoulos & Wood, 2019). Although all blockchains vary regarding their technical implementation, they usually share the properties depicted in Figure 1.[9]

[9] Here, the term blockchain refers to the most common representation of a blockchain comprising these properties.

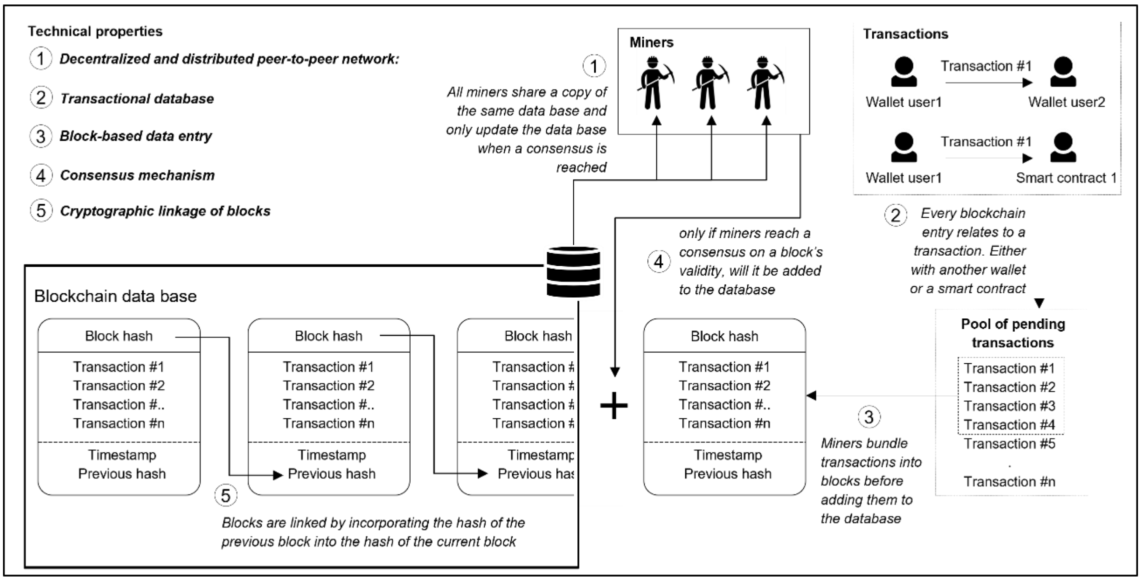

Technical properties
1 Decentralized and distributed peer-to-peer network:
2 Transactional database
3 Block-based data entry
4 Consensus mechanism
5 Cryptographic linkage of blocks
1 All miners share a copy of the same data base and only update the data base when a consensus is reached
Miners
Transactions
Transaction #1
Wallet user1
Wallet user2
Transaction #1
Wallet user1
Smart contract 1
2 Every blockchain entry relates to a transaction. Either with another wallet or a smart contract
4 only if miners reach a consensus on a block's validity, will it be added to the database
Blockchain data base
Block hash
Transaction #1
Transaction #2
Transaction #..
Transaction #n
Timestamp
Previous hash
5 Blocks are linked by incorporating the hash of the previous block into the hash of the current block
Pool of pending transactions
Transaction #1
Transaction #2
Transaction #3
Transaction #4
Transaction #5
.
Transaction #n
3 Miners bundle transactions into blocks before adding them to the database

Figure 1: Technical properties of a blockchain

- **Technical property 1: Decentralized and distributed peer-to-peer network.** A blockchain is a peer-to-peer network where nodes (any device such as a computer or server running the required blockchain software) are interconnected. All nodes maintain their own copy of the data. They keep their data updated by propagating transactions through the network (Bashir, 2020). The network is decentralized as the propagation of transactions is dispersed across all nodes. This means that each node can initiate the propagation of a new transaction and there is no hierarchical reporting. Blockchains, in their original sense, are part of a distributed decision-making process (Vergne, 2020). This means that no individual node determines the validity of transactions, but the network jointly tries to reach a consensus on this. Consequently, as blockchains are built on the principle of redundancy and replaceability, there is no single point of failure and a blockchain's reliability does not depend on one or a few nodes but on all nodes equally.
- **Technical property 2: Transactional database.** Data added to a blockchain originates from transactions between two entities identified by their address, called *public key* (Halaburda, 2018; Werbach, 2018). Compared to common relational databases, where entries can be changed by directly modifying the memory reserved for the respective entity, the blockchain entity can only be changed by sending a transaction to or from its address. Its current state can be determined by tracing all previous transactions to or from this address, and thus all changes to a blockchain's state can be traced by analyzing the transaction records. To initiate a transaction, the sender needs to indicate the recipient's address (public key), the amount they want to transfer, and sign with their *private key* (a secret code that only the wallet owner knows)

to authorize the transaction. Depending on the specific blockchain, a transaction can comprise further data. With Ethereum, a transaction must indicate the fee the sender is willing to pay for the validation of the transaction and can optionally include an amount of a specific token that should be transferred or general data that serves as input for smart contracts (Antonopoulos & Wood, 2019).

- **Technical property 3: Block-based data entry.** To increase their throughput, transactions are not processed one at a time but bundled into blocks. The number of transactions bundled is limited by the block's storage space. While Bitcoin has a fixed block size of 1MB, Ethereum's block size can vary. Before a transaction is included in a block, it is added to a pool of pending transactions.[10] Transaction validators, so-called 'miners' select a set of pending transactions and compute various validation tasks. Only once all a block's transactions have been validated and the network has reached a consensus about their validity, can this new block be added to the blockchain. Note that it is up to the miners to decide which transactions are bundled together in a block. Each miner can compose blocks differently as long as the network agrees on the transactions' validity (Daian et al., 2020) Typically, blockchains only allow adding blocks linearly,[11] which creates a natural bottleneck and limits the transaction throughput and thus the scalability (Vukolić, 2016).
- **Technical property 4: Consensus mechanism.** A consensus mechanism allows a set of distributed and decentralized nodes to work together and mitigate the risk of individual nodes' opportunistic behavior. Blockchains use consensus mechanisms to reach an agreement on the network's current state and on how to update this (Bashir, 2020). In non-technical terms, a consensus mechanism is the set of rules all nodes use to agree on the correct history of transactions (selecting which chain is 'correct') and how all transactions comprising a new block add to this history. Consensus mechanisms have been used for decades in peer-to-peer databases to establish agreements among nodes but the intersection of blockchain technology has led to the creation of multiple new consensus mechanisms. The most popular consensus mechanisms are: *proof of work (PoW), proof of stake (PoS), delegated proof of stake (DPoS), Byzantine fault tolerance (BFT),* or a combination of these. All are designed in such a way that at least 51 percent of the voting power is required to reach a consensus. What

[10] The pool of pending transactions has different names that even differ within a blockchain network, depending on the client. TX-queue or TX-pool are the most popular with Ethereum clients, while the term "mempool" commonly used across multiple blockchain networks is originally from Bitcoin.

[11] IOTA (iota.com) is an exception. In its network, blocks can be added to arbitrary previous blocks. Consequently, the IOTA ledger resembles a network rather than a linear chain.

differentiates these consensus mechanisms is how they protect against malicious attacks. For example, PoW achieves this by making miners expend a lot of energy, while PoS forces miners to lock up a lot of collateral. As two miners may find a valid but differing block and propagate it to different nodes at exactly the same time, a blockchain might fork into two competing versions. To maintain only one original version and prevent the creation of many different forks, the most prominent networks such as Bitcoin and Ethereum rely on the *longest chain rule*. In PoW chains, this rule determines that the chain with the highest cumulative PoW difficulty is the legitimate chain. Thus when miners synchronize their database with peers, they update their database to the chain with the most blocks and start mining on the most current block.

- **Technical property 5: Cryptographic linkage of blocks.** A cryptographic linkage turns a sequence of blocks into a chain. All the blocks in a blockchain are chained together by their hash (Yermack, 2017). A hash is a fixed size string (256 bits in Ethereum's case) that encrypts information. It is created by applying a hash function to some information. Hash functions (e.g., SHA-256, SHA-3, or Keccak) are designed in such a way that small changes to the input lead to a substantial change in the hash. This ensures on one hand that it is easy to detect changes to the data, and on the other hand prevents the data input being guessed based on the hash. Common blockchains use a block's data together with the previous block's hash to create a block's hash. Through this linkage, a change to one block would change all the hashes in every subsequent block and could be easily detected. If a malicious party wants to change a transaction in the past, not only does it have to recalculate all the hashes in every block, but then also convince the entire network that the chain with the updated hashes is the legitimate one. This design implies that while it is possible to change data on a blockchain, the economic costs very likely exceed the benefits.

The above technical properties allow blockchains to achieve desirable functional characteristics. Transaction records stored on a blockchain can be perceived as immutable (Fröwis & Böhme, 2017) as it is prohibitively costly to change past transactions unilaterally (*immutable transactions*). Immutability and the chronological recording of transactions imply that all past transactions can be traced to their origin (*traceable transactions*). It is thus easy to prevent not only double spending of a payment unit but also discordance about the state of the network. Human agency or any third-party conflict resolution and enforcement of transactions therefore become dispensable (*disintermediation*) and allow a blockchain network to run as an automated machine system (*automatic enforcement of transactions by machines*) (Hsieh et al., 2018). In addition, for machines to execute every transaction automatically, all input triggers and outcomes must be predefined and deterministic *(predefined and deterministic transactions)*. This feature excludes opportunistic behavior and renegotiation by design (Halaburda et al., 2019) and has led proponents of blockchain technology to suggest that blockchains are 'trust-free' systems (Hawlitschek et al., 2018). Further fueling this suggestion is the fact that most popular blockchain platforms run completely pseudonymously without limiting access to their network (*pseudonymity and permissionless*). Whereas nearly all blockchain platforms share other features, *pseudonymity* and *permissionless* are design choices traded by a company or consortium-owned blockchain network to achieve better scalability (e.g., Hyperledger Fabric, Quorum, Corda). However, there is a recurring debate about whether such platforms should be called 'blockchains.' Andreas Antonopoulos, a leading blockchain expert insists that if it is not "open, borderless, censorship-resistant, decentralized, publicly verifiable and neutral [. . .], it's not a blockchain" (Antonopoulos, 2020).

In summary, there is a huge variety of blockchains with different properties. As the technology is constantly evolving, it is questionable if one standard definition of the term *blockchain* will prevail in the future. To minimize conceptual ambiguity, I try to specify what type of blockchain I am discussing and provide current implementations such as Bitcoin or Ethereum as reference point. Here, the term blockchain refers to the combination of technical setup and properties defined above.

2.2 Smart contract basics

The notion 'smart contracts' can be misleading, as they are neither smart—no artificial intelligence is involved—nor contracts in the legal sense. Though the concept has been popularized by the hype around blockchains, smart contracts are not limited to blockchains (Halaburda, 2018). Unrelated to the blockchain realm, in 1994 Nick Szabo[12] introduced the idea of smart contracts as secure computer protocols automatically enforcing contractual agreements over computer networks. He argued that many kinds of contractual clauses—in fact, every computable clause—can be expressed and encoded in computer code in the form of algorithms. The technology available at that time could not provide a suitable environment for decentralized, reliable self-executing contracts, a situation that pushed smart contracts almost into oblivion. The advent of blockchain technology, however, changed this: blockchain technology provided smart contracts with immutability, commonly validated data input, and a

[12] https://firstmonday.org/ojs/index.php/fm/article/view/548/469-publisher=, accessed September 15, 2022.

secure execution environment backed by the very same consensus mechanism that validates transactions. Technically speaking, a smart contract is no more than a computer program running on top of a blockchain (Antonopoulos & Wood, 2019). The misnomer 'smart contract' was introduced by Vitalik Buterin, who adopted the term in the Ethereum white paper (Buterin, 2014) to describe Ethereum's potential to run arbitrary automated scripts on top of the blockchain, but later regretted his choice of name.[13] The success of the Ethereum blockchain has further popularized the term that now commonly refers to automated scripts running on a blockchain (e.g., Halaburda et al., 2019; Murray et al., 2019). These scripts give a blockchain 'Turing complete' programmability. This means that smart contracts can define arbitrary transactions that exceed the complexity of simple cryptocurrency transfer transactions offered by the blockchain protocol. In this thesis, I follow the common convention and use the term smart contracts to refer to such scripts.

Pioneered by the Ethereum blockchain, today most blockchains offer the possibility to create and use smart contracts (e.g., Polkadot, Cardano, Binance smart chain). Although the smart contracts on these platforms vary regarding technical implementation, they also share common properties. First, smart contracts are *immutable*. Like any other data stored on a blockchain, a smart contract is stored in one of the chain's blocks (Antonopoulos & Wood, 2019). Changing a smart contract's byte code would thus require recomputing all hashes and consent from all other nodes that the new chain is legit. This would be extremely costly and hence is highly unlikely (Fröwis & Böhme, 2017).[14] Second, smart contracts are *predefined* and *deterministic*. Accordingly, all the conditions of a transaction have to be specified ex-ante and in such a way that the outcome is the same for every party executing a smart contract (Antonopoulos & Wood, 2019). Finally, smart contracts are *automatically and jointly enforced*. Once a smart contract is triggered, all the network nodes execute the predefined instructions and compare the transaction outcome. The smart contract transaction is only added to the blockchain if the majority of nodes achieve the same outcome. Similar to ordinary blockchain transactions, this ensures that smart contracts do not depend on human agency or third-party intermediaries, only on predefined data inputs (Halaburda, 2018; Murray et al., 2019).

A smart contract's lifecycle can be broken down into four steps that explain in detail how smart contracts work.

[13] https://twitter.com/VitalikButerin/status/1051160932699770882, accessed September 15, 2022.

[14] Changing the byte code stored on the blockchain is very costly. Fröwis and Böhme (2017) show it is possible to change a smart contract's control flow (order in which script statements are executed) by changing functions in external libraries used by the smart contract. Such changes can be detected or prevented by requiring specific versions of imported libraries.

- **Step 1: Writing and deploying a smart contract.** Smart contracts are typically written in human-readable (high-level) scripting languages such as Solidity, Vyper, or Marlowe. Before smart contracts can be executed by all networked nodes, they must be compiled into machine-readable (low-level) byte code, that is then deployed on the blockchain. For example, the Ethereum network uses a specific contract creation transaction to deploy the byte code (Antonopoulos & Wood, 2019). Any participant can initiate this transaction, which is sent to a dedicated address (0x0), and attaches the smart contract's byte code. If successfully executed, the network identifies this transaction as contract creation, stores the byte code in a block, and assigns it a new address. This address can then be used to transact with the smart contract. It is important to note that due to this process, only the byte code is stored on the blockchain and is visible to everyone. The smart contract's source code remains with its author and is thus not necessarily public. It is up to the author to decide whether to publish and verify the human-readable source code.
- **Step 2: Using a smart contract.** To interact with a smart contract, users must send a transaction to the smart contract address. This transaction can be a simple transfer of the native protocol token (e.g., Ether), or a call for a specific smart contract function. If it calls for a specific function, the requisite conditions must be specified in the data sent with the transaction. Usually, such transactions are wrapped in a user interface (front-end) to ease the use of the smart contract but can also be called directly. In that case, however, the user must know the smart contract's requisite conditions.
- **Step 3. Executing a smart contract.** Smart contracts are only executed if they are triggered by a transaction, and so they can never run independently. Even though a smart contract can call other smart contracts, the first trigger has to be initiated by a non-smart contract address (i.e., an externally owned wallet). When a smart contract is triggered, the transaction is added to the pool of pending transactions where it waits to be validated and added to the blockchain. To validate a smart contract transaction, the blockchain network nodes need to execute the byte code and agree on the outcome of the transaction. Only if this consensus is reached will the smart contract transaction be added to the blockchain. Regardless of how many smart contract functions are called in one transaction, the transaction will be only be executed in its entirety and recorded if all operations are executed successfully. If one operation fails or does not achieve consensus among the nodes, the entire transaction will fail, and all prior state transitions will be ignored as if the transaction never existed.

- **Step 4: Deleting a smart contract.** As explained above, smart contracts are immutable in the sense that the byte code running on the blockchain cannot be altered. The byte code can, however, be deleted or coded in such a way that the smart contract functions or even the entire smart contract cannot be disabled. Ethereum, for example, provides a ‘self-destruct’ function (previously called ‘suicide’) that allows the byte code to be deleted from the blockchain (Antonopoulos & Wood, 2019). This function, however, can only be called by the contract’s author and has to be explicitly added to the contract, otherwise the smart contract cannot be deleted. Once this function has been successfully called, the byte code is deleted and cannot be restored. Destroying the contract neither removes the contract address nor past transaction history, and no transaction with the contract address will result in a code execution. Money sent to such an address is lost forever.[15]

Equipped with smart contracts, blockchains have been hailed as implementing a broad array of use cases and even reshaping whole industries (e.g., Dutra, Tumasjan, & Welpe, 2018; Friedlmaier, Tumasjan, & Welpe, 2016; Mehrwald et al., 2019). However, even though smart contracts’ abilities are continuously expanding (Christidis & Devetsikiotis, 2016), they still face significant limitations yet to be overcome. First, the predefined and deterministic nature of the blockchain implies that a transaction can only modify the blockchain data in a calculable way with no uncertainty. Although this is desirable in most cases, it also implies that smart contracts lack a source of entropy or randomness. For applications like lotteries or gambling in general, randomness is a prerequisite as pseudorandom numbers leave them exposed to security issues (Bartoletti & Pompianu, 2017; Chatterjee, Goharshady, & Pourdamghani, 2019). Second, smart contracts run encapsulated in the blockchain’s environment. Consequently, they have no built-in capabilities to directly access data not stored on the blockchain (e.g., stock price data, weather data, IoT sensor data, or even the time). As this data is required as input for smart contracts expanding their area of use beyond pure cryptocurrency, smart contracts rely on so-called ‘oracles’ (Al-Breiki, Rehman, Salah, & Svetinovic, 2020). These oracles are interfaces that provide a smart contract with data from an external source (e.g., a stock market price or weather API, or an IoT sensor). Unlike the immutable transaction data stored on the blockchain, which all nodes can access, the validity of external data cannot be verified by all nodes during contract execution, thus making smart contracts vulnerable to opportunistic behavior. This limitation is also often referred to as an inherent smart contract ‘oracle problem’ (e.g., Caldarelli, 2020; Sheldon, 2021). Third, smart

[15] Plus money currently locked in those black holes.

contracts are not transparent by design. As explained above, only the byte code is stored on the blockchain. The human-readable source code resides with its authors and is only publicly available if they publish it and verify that this source code corresponds to the compiled version running on the blockchain. Without this verification, a party offering a smart contract on the blockchain can hide the contract's true functionality.[16] Fourth, smart contracts are not executed ad hoc once they are called. A transaction that triggers a smart contract, like all other transactions, enters the pool of pending transactions and is only executed if a miner decides to add it to a block. Depending on the network congestion, this can take several hours or even days.[17] As protocols continue to improve in terms of scalability, this time will be further reduced, but there will still be delays that are difficult to predict. Therefore, for the time being, smart contracts are less suitable for time-sensitive applications and remain vulnerable to frontrunning attacks (Daian et al., 2020). Developed by humans, smart contracts can exhibit a variety of bugs and negligently programmed security weaknesses. Such errors are particularly serious due to the immutability of smart contracts that precludes updates and patches. Although it is technically possible to outsource smart contract functionalities through libraries that can also be modified in retrospect, this could lead to severe security vulnerability: Fröwis and Böhme (2017) show that based on a sample of 194,332 smart contracts on the Ethereum blockchain, two out of five can be altered without changing the byte code on the blockchain. On the one hand, this gives developers the opportunity to update the smart contract's functionality. On the other hand, it also opens up loopholes for opportunistic behavior. The recommended development practice is to deploy a new smart contract rather than try to fix the old one (Antonopoulos & Wood, 2019).

To conclude, smart contracts are no panacea for enabling inherently secure transactions but require thoughtful consideration of their potential and limitations. It is important to bear in mind that a smart contract's functionality always depends on the implementation of the specific smart contract. Just because no one can hack and compromise a blockchain's security, does not mean that the smart contract running on it is secure. Multiple hacking attacks have shown this. Most notable is the notorious DAO hack, where hackers exploited a security weakness in a smart contract and stole 60 million dollars' worth of Ether.[18]

[16] https://medium.com/etherscan-blog/verifying-contracts-on-etherscan-f995ab772327, accessed September 15, 2022.

[17] The most well-known example of network congestion was caused by CryptoKitties (https://www.bbc.com/news/technology-42237162), a collectibles game where users could breed and exchange digital pictures of cats.

[18] https://www.gemini.com/cryptopedia/the-dao-hack-makerdao#section-the-response-to-the-dao-hack, accessed September 15, 2022.

2.3 Tokens, coins, and cryptocurrencies

In the blockchain realm, the terms crypto token, crypto coin, and cryptocurrency are often confused as the same thing, despite conceptual differences. Although a detailed understanding of their differences is not necessary to grasp this dissertation, it can mitigate confusion. I will therefore briefly explain these terms.

The confusion around crypto coins and tokens arises because both are commonly traded as cryptocurrencies on cryptocurrency exchanges (e.g., the Dogecoin and Basic Attention Token are traded on the same exchange); some tokens are referred to as stable coins even though they technically resemble tokens (e.g., Dai); and a clear definition of cryptocurrencies is lacking (Maese, Avery, Naftalis, Wink, & Valdez, 2016). A feature they share is that they represent a value whose price increases with demand. Consequently, tokens and coins are used as payment methods and are subject to fluctuating prices and speculation. Despite their commonality, comparing their intended use and technical implementation reveals important differences.

Crypto coins (hereafter *coin* refers to crypto coins native to a blockchain) are only meant to be used for payment as a digital replication of money running on a blockchain (Nakamoto, 2008). All transactions with this coin are recorded on the blockchain and the blockchain's protocol defines how the coin is created, transferred, and deleted, which means there is typically only one coin per blockchain. In other words, a coin is 'native' to a blockchain (Voshmgir, 2020).

In contrast, crypto tokens (hereafter *token* refers to crypto tokens on a blockchain) are more ambiguous. Tokens are the digital representation of arbitrary things of value. They can represent voting rights, access rights, company shares, or even physical objects like a car or a painting. Their commonality is that they exist digitally on one or multiple blockchains and are typically managed by a smart contract. Unlike coins, they are not native to a specific blockchain protocol but to the smart contract. Representing an asset in the form of a digital token on a blockchain allows users to interact, buy, or exchange some of that asset's rights, for example ownership. Important to note is that the actual asset (especially true for physical assets) often does not reside on the blockchain as only the asset's business logic (e.g., voting mechanism, transfer ownership of rights) is coded into the token's smart contract. Tokens can have different functionalities and represent different things such as resources (e.g., digital storage space), physical assets (e.g., house ownership), access, company equity, voting rights, collectibles, identity, attestation, utility, currency, or often a combination of these. More generally, there are two distinct types of tokens: fungible and non-fungible. Fungible tokens are not unique, they can be reproduced, added up, or split into subunits (a currency token like wrapped Bitcoin). Non-fungible tokens (NFTs), on the other hand, are unique. Every NFT can only exist exactly once. NFTs can thus be used to add rarity to something that could be otherwise replicated. Importantly, however, even though an NFT cannot be replicated, the digital asset that exists outside the blockchain and is tied to the NFT (e.g., a digital photo) can be replicated without repercussions for the NFT. Therefore, we should see the NFT as a unique certificate representing certain rights about a good but not the digital good itself, unless it exists exclusively on the blockchain (e.g., a decentralized autonomous organization's share).

2.4 Decentralized applications

Decentralized application (dApp) is yet another term in the blockchain realm that no longer has its initial narrow definition and is currently used more widely. Today, we call applications dApps if they run based on a smart contract on the blockchain. To what extent the smart contract or application is decentralized at all is usually disregarded (Cai, Wang, Ernst, Hong, Feng, & Leung, 2018). Narrowly defined, a decentralized application is a web application that is not controlled by one central authority but by a smart contract. As Figure 2 illustrates, it differs from a traditional centralized web application in two main ways. First, it is connected to a smart contract running on a blockchain. The frontend provides a graphical user interface for a smart contract transaction and initiates transactions for the user. To complete a transaction, the user has to authorize it by signing with their private key. This is typically done by connecting a wallet to the dApp using a browser plug-in like MetaMask.[19] The execution of the business logic encoded in the smart contract is enforced automatically and without the possibility of interference by the dApp provider. Second, a truly decentralized application also hosts the frontend code and the backend server on a decentralized server. IPFS (Inter Planetary File System) is one protocol that allows hosting off-chain files on a decentralized database. A dApp is only truly decentralized if it hosts its frontend and backend on such a database and has mechanisms in place that do not allow an individual party to change the stored data or interfere with the communication between these modules.

[19] https://metamask.io/

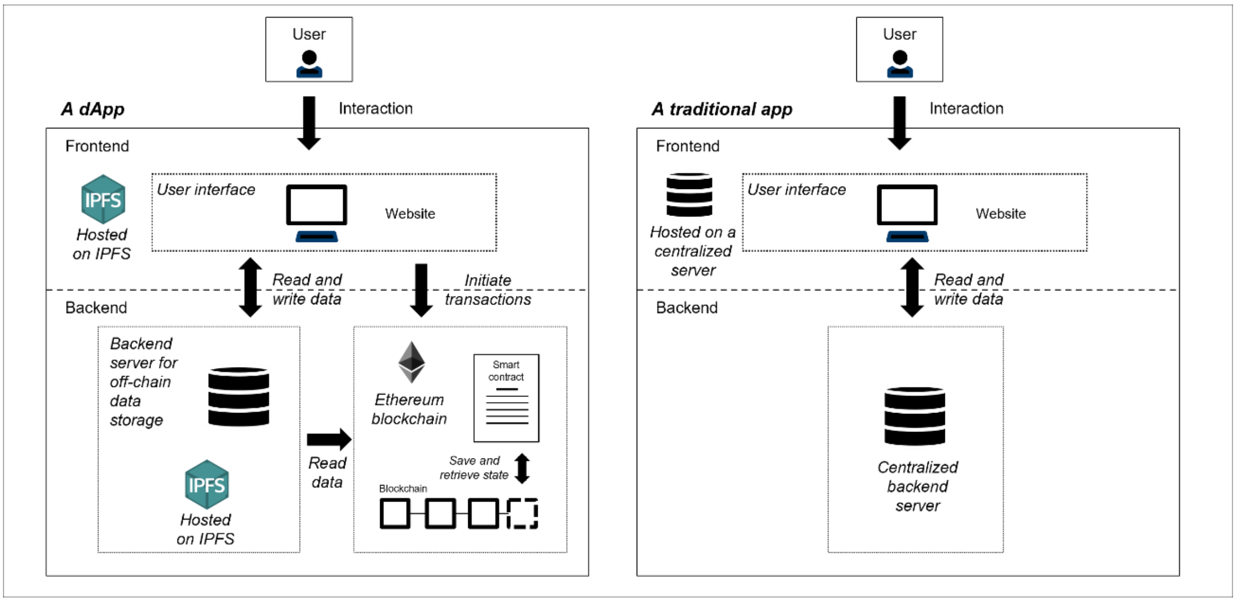


Figure 2: Comparison of a dApp with a traditional centralized app

If a dApp is designed in that way, it has no central point of authority. Hence, it is not possible for one party to change the functionality of the dApp without the consent of all other parties (Wu et al., 2021). A dApp can thus provide three properties that typical centralized applications (in the app store or other browser-based web applications) cannot provide (Antonopoulos & Wood, 2019). First, true dApps do not have any downtime. Unlike with the centralized applications usually deployed on a centralized server, a breakdown of one server does not shut down the dApp since it is running on several nodes. Second, interactions with a dApp are traceable and transparent because they are stored as transactions on the blockchain. Third, they are resistant to censorship. As the code is running on multiple systems, altering the dApp on one system does impact the dApp.

As discussed above, these properties need every part of the dApp to be decentralized. Due to performance limitations with decentralized systems—storing data on decentralized servers is more expensive as it requires multiple redundant copies of all data—and the difficulty updating and maintaining a decentralized system, most dApps still rely on centralized web servers to store their data or host their frontend (Cai et al., 2018). These centrally managed parts of a dApp can have security weaknesses or allow the party offering the dApp to unilaterally modify the dApp’s functionality even though the smart contract on the blockchain is not changed. For example, the company offering the dApp could route the transactions a user sends by using the front end of a different smart contract. For a user, such interference would be difficult to identify at first glance. However, once the transaction is sent to the blockchain, the user could check if the correct smart contract address is used as receiving address. On the other hand, if a centralized application provider changes some logic in their web application that is not noticeable at the front end, it is impossible for users to comprehend the changes. Therefore, already implementing parts of the dApp’s business logic as a smart contract offers greater transparency.

In summary, dApps have in common that they implement their core business logic as smart contracts on a blockchain, but can vary significantly in their degree of decentralization.

2.5 Specifics of the Ethereum blockchain

After Bitcoin, Ethereum is the second largest blockchain network with a market capitalization exceeding \$300 billion.[20] It was the first network to offer smart contracts (Buterin, 2014) and is used for diverse applications, unlike Bitcoin, which is mainly for payment. Although Ethereum is not the only platform providing smart contracts and the possibility to develop dApps, it offers and uses the highest number of dApps.[21] As the Ethereum blockchain is the empirical basis for this thesis, I explain its technical details that are particularly relevant for Chapter 5.

I chose Ethereum as a context for three reasons. First, as the most adopted platform for dApps, Ethereum therefore provides the best generalizability. Second, it has a well-defined technology stack and already established standards that enable a better comparison of applications offered on the blockchain. Finally, not only is it the oldest platform offering dApps, but also because it has attracted the biggest community of developers to improve the protocol and provide complements for the platform, it serves as a guidepost for the entire industry.

Ethereum has multiple layers of complexity, but since a technical review of the blockchain is beyond the scope of this thesis, I focus on the most important terms and characteristics that can explain how Ethereum works:

- **A global Turing machine**. Ethereum can track not just a coin’s ownership, but also the arbitrary data and code. Ethereum can also load data, run code, and store the results of data manipulation on the blockchain. In this way, it can change what is referred to as the state[22] of the data and a program. Like general-purpose computers, Ethereum is therefore ‘Turing complete’ and can run arbitrary programs. Unlike with a general-purpose computer, changes to the state are governed by the rules of a decentralized consensus and updated globally (Antonopoulos & Wood, 2019). Ethereum programs can be run anywhere

[20] https://coinmarketcap.com/de/currencies/ethereum/, accessed September 15, 2022.

[21] According to https://www.stateofthedapps.com/de/stats (retrieved January 22, 2022), Ethereum hosts 2,912 dApps based on 4,820 smart contracts with 9,301,000 daily users. The second most popular dApp platform (EOS) only hosts 331 dApps.

[22] A state is the recorded history of all preceding events or user interactions.

worldwide, yet result in a common state achieved through a decentralized consensus and stored on the blockchain. The creation of this distributed single-state world computer is Ethereum's innovation.

- **The Ethereum Virtual Machine (EVM)**. The EVM refers to all nodes[23] that participate in the Ethereum network and execute the blockchain's computations. The EVM is a virtual machine that executes smart contracts' compiled source code in the form of machine-readable bytecode. Whenever a smart contract is triggered, every node in the Ethereum network executes the contract and engages in finding a consensus on the correct execution of the contract according to the network's consensus rule. To compensate the nodes, users have to pay for this computational effort. To compute the fees, the EVM assigns costs to the operations a smart contract requires (e.g., costs for an if clause) and uses an accounting mechanism to track the consumption of computational effort a smart contract requires. Ethereum's measurement of computational effort is called *gas* (alluding to the virtual fuel that drives Ethereum). To increase the throughput and ensure transactions' computability, transactions and blocks have a *gas limit* that restricts the computational effort a smart contract execution may consume. The supply and demand for computational effort on Ethereum determine the price of a unit of gas (called *gas price*) paid in *Wei*. The gas price times the gas used by a transaction equals the transaction fee a sender has to send to pay for the execution of the transaction (for more details, see Chapter 5).
- **Ether**. Ether is Ethereum's native coin. It is tied to the network protocol and is used as general payment method on the network. Its name stems from the Greek letter 'Xi'. Ether can be subdivided into smaller units. The smallest is Wei and represents a quintillionth fraction ($1*10^{-18}$) of an Ether. Internally, all transactions on the Ethereum network are denominated in Wei. Ether also has various denominations in line with the International System of Units, but each denomination also has a colloquial name honoring some of the greatest minds in computing.
- **Accounts and wallets.** On Ethereum, every account is identified by an address (e.g., 0x02878FE24876747ab68528c50848CbA12A1Dd37d). This address is also referred to as the public key and is required for signing a transaction. A

[23] At the time of writing, 5,787 nodes were active in the network. Source: https://www.ethernodes.org/ accessed September 15, 2022.

wallet is a software application that helps to manage the coins and tokens owned by the account.[24] It stores the private key which is also required to authorize transactions but remains hidden from the public. Contrary to common intuition, the wallet does not store the value. It is only used to manage transactions and store the private key. The amount of Ether a wallet 'owns' is computed by tracing all transactions associated with the address. The number of tokens 'owned' by an address is stored in the token contract. On Ethereum, there are two types of accounts: externally owned accounts (EOAs) and smart contracts. Both can receive and send transactions to other EOAs or smart contracts, but only EOAs can initiate the execution of a smart contract. While a smart contract is being executed, it can also call other smart contracts, but cannot start transacting on its own. Another key difference is that only EOAs hold a private key (Bashir, 2020).

- **Smart contracts on Ethereum** are arbitrary computer programs running on top of the Ethereum blockchain. They are typically written in the human-readable language Solidity (similar to JavaScript) or Vyper (similar to Python). Both are high-level scripting languages that predefine a plethora of commands that the EVM can interpret. Appendix A-1 depicts and explains a simple example of a smart contract. It is OpenZeppelin's[25] standard implementation of an ERC20 token contract used for a variety of dApps.
- **Transactions on Ethereum.** Sending a transaction on Ethereum requires a public and a matching private key. A transaction sender can create a transaction by indicating its recipient (or rather its address) and the value of Ether they want to send. Along with the Ether, they can also send additional data to use as input for a smart contract. Before a transaction can be processed, it is added to the pool of pending transactions. There, miners typically pick transactions that maximize their profit by trying to pack a block as full as possible (they try to reach the gas limit) with transactions that indicate the highest willingness to pay for a unit of gas. On Ethereum, there are three types of transactions: normal Ether transfers from one EOA to another; contract calls from one EOA to a smart contract address; and contract creation calls where an EOA sends a trans-

[24] One of the most popular wallets, MetaMask is a web-based wallet that runs as a browser extension and is also optimized for using dApps.

[25] OpenZeppelin is a crypto cybersecurity service company (https://www.openzeppelin.com/about).

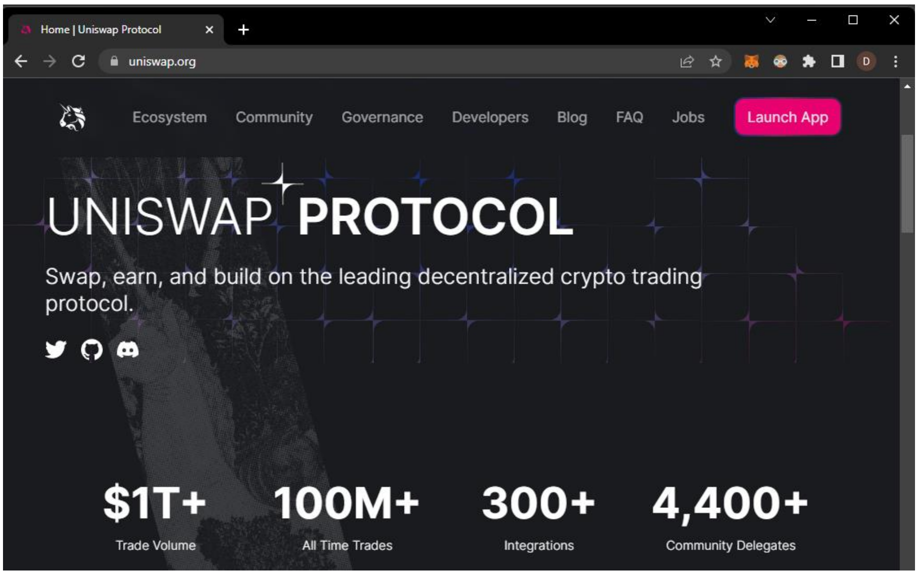
action to the 0x0 address. The EVM then interprets the data sent with the transaction as the smart contract code and assigns a new address to the newly created contract. Whenever a transaction is sent to a smart contract address, the EVM executes the bytecode stored at this address.

- **Tokens on Ethereum** are defined and managed by smart contracts. Ethereum uses several token standards to ensure tokens' compatibility. The most popular are the ERC20 standard for fungible tokens and the ERC721 standard for non-fungible tokens. These are de facto standards that define the minimum functionality and interface a smart contract can offer. An example is a function that allows the transfer of tokens. These standards are necessary since they enable wallet applications to interact with a token contract's interface and thus manage the funds at that address.

2.6 Interacting with a dApp on Ethereum

For readers who have never interacted with a dApp, I use the Uniswap dApp—one of the most popular DeFi (decentralized finance) dApps—as a visual example. I explain all the steps users must take to transact with a dApp. Most of the steps listed below are universal for all dApps. Only step 3 is specific to Uniswap but resembles other dApps.

- **Step 1: Accessing the website and launching the dApp.** Most dApps have a landing page (Uniswap's landing page, shown in Figure 3, is https://uniswap.org/). This website gives general information on the dApp and the company providing it. To launch the dApp, users can either click the 'launch App' button or access it directly through its dedicated URL (https://app.uniswap.org/#/swap). Both take the user to Uniswap's dApp interface (Figure 4).



UNISWAP PROTOCOL

Swap, earn, and build on the leading decentralized crypto trading protocol.

Figure 3: Uniswap's landing page

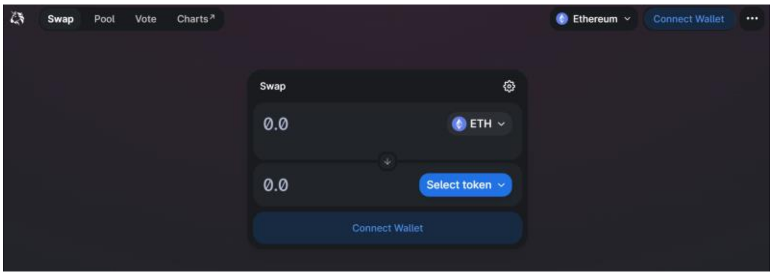
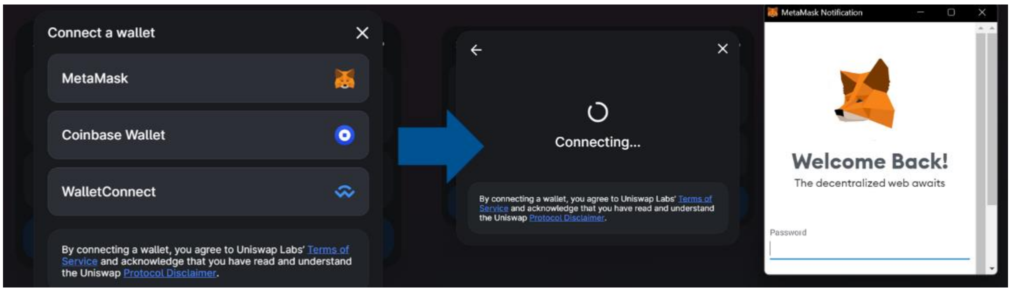


Figure 4: Uniswap's dApp interface

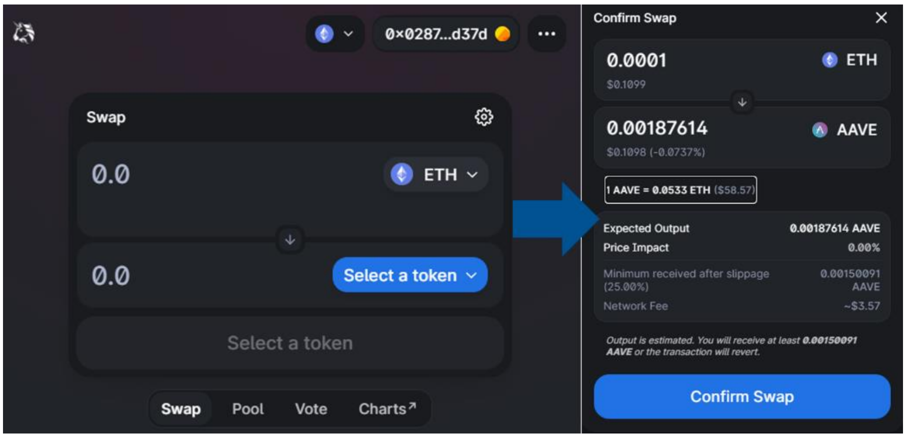
- **Step 2: Connecting and logging into the wallet.** Before users can interact with the dApp's interface and modify their transaction, they have to connect their wallet. Clicking on the 'connect wallet' button shown in Figure 4, prompts a pop-up window that allows users to select their wallet provider (left hand panel in Figure 5). MetaMask is the most popular wallet provider for dApps as it is compatible with the majority of dApps. It stores a user's private key safely and eases the process of signing transactions. After choosing the MetaMask wallet, users log into their MetaMask browser plug-in (right hand panel in Figure 5).



By connecting a wallet, you agree to Uniswap Labs' Terms of Service and acknowledge that you have read and understand the Uniswap Protocol Disclaimer.

Figure 5: Connecting and logging into a wallet

- **Step 3: Initiating a transaction (indicate swap amount).** Once users have connected their wallet, the dApp recognizes that a wallet has been connected (the top right corner of Figure 6 now shows the wallet address) and allows users to modify the transaction through the dApp's graphical interface. For this example, I chose to call the Uniswap dApp swap function and exchange 0.0001 ETH for 0.00187614 AAVE.[26] After clicking the 'confirm swap' button, the dApp assembles the transaction in the background and asks the user to sign it.



0.0 ETH

0.0

0.0001 ETH
$0.1099

0.00187614 AAVE
$0.1098 (-0.0737%)

1 AAVE = 0.0533 ETH ($58.57)

Expected Output 0.00187614 AAVE
Price Impact 0.00%
Minimum received after slippage (25.00%) 0.00150091 AAVE
Network Fee ~$3.57

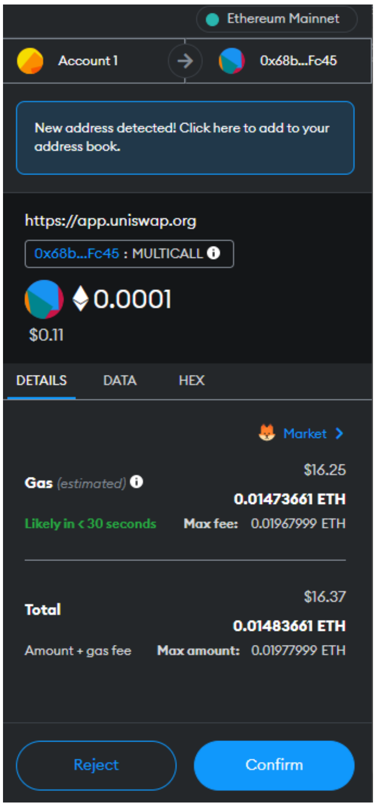
*Output is estimated. You will receive at least **0.00150091 AAVE** or the transaction will revert.*

Figure 6: Initiating a transaction

- **Step 5: Signing a transaction.** Once users have confirmed the transaction in the dApp, the dApp notifies the wallet plug-in. MetaMask then prompts users to another pop-up window (Figure 7). This window shows all the transaction information (the smart contract address and dApp website, the transaction value, and charged transaction fees) and allows users to modify general transaction data. For example, users can change the transaction fees by adjusting the gas price and thus influence how miners might prioritize the transaction and how fast it will be processed. By clicking the 'confirm' button, users sign the transaction with their private key and MetaMask sends the transaction to the

[26] AAVE is the token on another DeFi dApp (https://aave.com/).

pool of pending transactions where it waits to be picked up by a miner. To validate that the transaction is sent to the correct smart contract, users can copy the smart contract address and look it up on one of the blockchain explorers. Etherscan (etherscan.io/) is one of the most popular explorers for the Ethereum network. Etherscan also allows dApp developers to upload and verify their smart contract. If a dApp developer has verified the source code, then users can find it by looking up the smart contract’s address on Etherscan and thus see how the EVM will process the transaction. Figure 8 shows that Uniswap has verified the smart contract associated with the dApp’s swap function.



Ethereum Mainnet

Account 1 → 0x68b...Fc45

New address detected! Click here to add to your address book.

https://app.uniswap.org

0x68b...Fc45 : MULTICALL

0.0001

$0.11

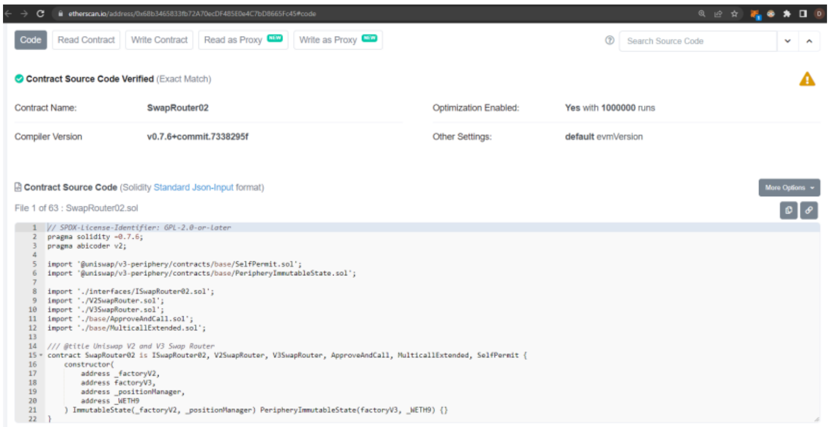
Gas (estimated) $16.25 0.01473661 ETH

Likely in < 30 seconds Max fee: 0.01967999 ETH

Total $16.37 0.01483661 ETH

Amount + gas fee Max amount: 0.01977999 ETH

Figure 7: Signing a transaction with MetaMask



Contract Source Code Verified (Exact Match)

Contract Name: **SwapRouter02**

Optimization Enabled: **Yes** with **1000000** runs

Compiler Version **v0.7.6+commit.7338295f**

Other Settings: **default** evmVersion

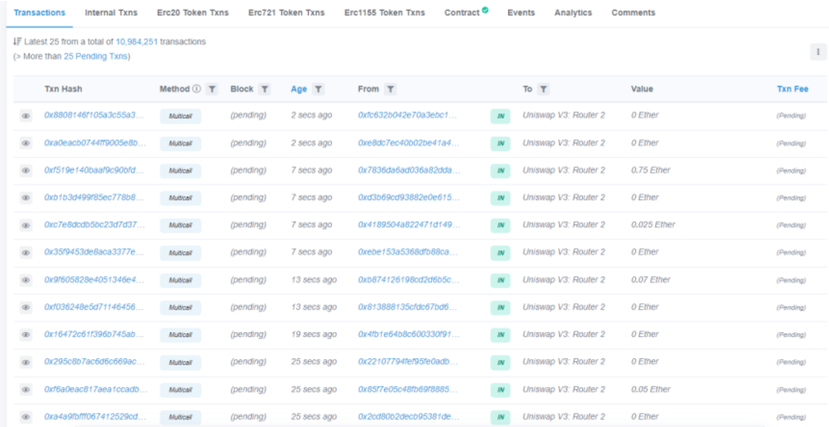
Contract Source Code (Solidity Standard Json-Input format)

File 1 of 63 : SwapRouter02.sol

```
// SPDX-License-Identifier: GPL-2.0-or-later
pragma solidity =0.7.6;
pragma abicoder v2;

import '@uniswap/v3-periphery/contracts/base/SelfPermit.sol';
import '@uniswap/v3-periphery/contracts/base/PeripheryImmutableState.sol';

import './interfaces/ISwapRouter02.sol';
import './V2SwapRouter.sol';
import './V3SwapRouter.sol';
import './base/ApproveAndCall.sol';
import './base/MulticallExtended.sol';

/// @title Uniswap V2 and V3 Swap Router
contract SwapRouter02 is ISwapRouter02, V2SwapRouter, V3SwapRouter, ApproveAndCall, MulticallExtended, SelfPermit {
    constructor(
        address _factoryV2,
        address factoryV3,
        address _positionManager,
        address _WETH9
    ) ImmutableState(_factoryV2, _positionManager) PeripheryImmutableState(factoryV3, _WETH9) {}
}
```

Figure 8: Uniswap’s verified smart contract on Etherscan.io

- **Step 6: Checking the status of the transaction and waiting for its execution**. After users have successfully signed their transaction and MetaMask has submitted it to the pool of pending transactions, they can check the status of the transaction on a blockchain explorer by looking up their transaction hash. Figure 9 is a sample of pending transactions to one of Uniswap’s smart contracts. Once the status of the transaction changes from pending to a block number, then the transaction has been successfully executed and added to the blockchain. This also completes a user’s transaction with a dApp. It is important to note that the verification of a transaction and thus the interaction with a dApp can take from several minutes up to days if the network is congested. Especially in times of congestion, the transaction might never even be processed if the user indicates a too low gas price.



Latest 25 from a total of 10,984,251 transactions
(> More than 25 Pending Txns)

Txn Hash	Method	Block	Age	From		To	Value	Txn Fee
0x8808146f105a3c55a3...	Multicall	(pending)	2 secs ago	0xfc632b042e70a3ebc1...	IN	Uniswap V3: Router 2	0 Ether	(Pending)
0xa0eacb0744ff9005e8b...	Multicall	(pending)	2 secs ago	0xe8dc7ec40b02be41a4...	IN	Uniswap V3: Router 2	0 Ether	(Pending)
0xf519e140baaf9c90bfd...	Multicall	(pending)	7 secs ago	0x7836da6ad036a82dda...	IN	Uniswap V3: Router 2	0.75 Ether	(Pending)
0xb1b3d499f85ec778b8...	Multicall	(pending)	7 secs ago	0xd3b69cd93882e0e615...	IN	Uniswap V3: Router 2	0 Ether	(Pending)
0xc7e8dcdb5bc23d7d37...	Multicall	(pending)	7 secs ago	0x4189504a822471d149...	IN	Uniswap V3: Router 2	0.025 Ether	(Pending)
0x35f9453de8aca3377e...	Multicall	(pending)	7 secs ago	0xebe153a5368dfb88ca...	IN	Uniswap V3: Router 2	0 Ether	(Pending)
0x9f605828e4051346e4...	Multicall	(pending)	13 secs ago	0xb874126198cd2d6b5c...	IN	Uniswap V3: Router 2	0.07 Ether	(Pending)
0xf036248e5d71146456...	Multicall	(pending)	13 secs ago	0x813888135cfdc67bd6...	IN	Uniswap V3: Router 2	0 Ether	(Pending)
0x16472c61f396b745ab...	Multicall	(pending)	19 secs ago	0x4fb1e64b8c600330f91...	IN	Uniswap V3: Router 2	0 Ether	(Pending)
0x295c8b7ac6d6c669ac...	Multicall	(pending)	25 secs ago	0x22107794fef95fe0adb...	IN	Uniswap V3: Router 2	0 Ether	(Pending)
0xf6a0eac817aea1ccadb...	Multicall	(pending)	25 secs ago	0x85f7e05c48fb69f8885...	IN	Uniswap V3: Router 2	0.05 Ether	(Pending)
0xa4a9fbfff067412529cd...	Multicall	(pending)	25 secs ago	0x2cd80b2decb95381de...	IN	Uniswap V3: Router 2	0 Ether	(Pending)

Figure 9: Pending transactions on Etherscan.io

3 Smart contracts on a Blockchain: Transaction Governance with the Potential of Deductive Certainty

3.1 Introduction

IS researchers have extensively examined governance mechanisms, particularly the role of trust in client-vendor relationships (e.g., Gefen et al., 2003; Guo, Straub, Zhang, & Cai, 2021; McKnight et al., 2002a). In general, governance mechanisms allow economic actors to establish and maintain exchange relationships (Poppo & Zenger, 2002; Williamson, 1985). They do so by allowing clients to form beliefs about the other party's future behavior. For established governance mechanisms, actors arguably form these beliefs primarily based on induction. Relational governance relies on unwritten, informal rules self-enforced by trust and norms (e.g., Gulati & Nickerson, 2008; Guo et al., 2021; Li, Poppo, & Zhou, 2010), and thus agents form beliefs about each other's behavior based on cues and prior experience. Formal or contractual governance specifies parties' obligations in a contract enforced by the legal system (e.g., Huber, Fischer, Dibbern, & Hirschheim, 2013; Schepker, Oh, Martynov, & Poppo, 2014). A legal contract offers some deductive element by specifying consequences of certain actions, but induction is required since contracts are typically incomplete, language is ambiguous, parties can violate contracts, and the legal system is not fully predictable. This inductive nature of established governance mechanisms implies that the other party's future behavior can never be predicted with certainty and reemphasizes the importance of trust to facilitate exchange (Uzzi, 1997). IS scholars have focused on the formation of trust in online vendors because trust is commonly understood as the antidote to doubts and fears arising from the faceless and intangible nature of online transactions and is thus a key facilitating factor for the adoption and continuing use of online services (e.g., Fang, Qureshi, Sun, McCole, Ramsey, & Lim, 2014; Gefen et al., 2003; McKnight et al., 2002b).

Lumineau et al. (2020) proposed blockchains—decentralized cryptographic systems that share an ongoing list of transactions among networked parties—as an alternative governance mechanism that might fundamentally change how online vendors govern their relationships and conduct transactions in the future. Instead of relying on trust, norms, or a third party, the decentralized blockchain system ensures automatic enforcement of obligations through a carefully designed consensus algorithm.

While we know that the blockchain generally enforces obligations arising from transactions, we do not yet understand how parties form beliefs about the outcome of a specific blockchain-based transaction. One option might be to use available inductive cues based on earlier experiences with the same transaction partner. If there is no prior experience, parties could alternatively look for cues that allow inferring the other party's integrity, benevolence, and competence (McKnight et al., 2002b), or rely on third-party certificates that suggest the other party is trustworthy (Pavlou & Gefen, 2004). But what if such cues are absent? Every transaction on a blockchain, except simple money transfers, is mediated by a smart contract, a pre-programmable and automatically executed computer script that can act upon data stored on the blockchain (Murray et al., 2019). Smart contracts specify a clear triggering condition and a unique outcome for every action they can execute (Halaburda et al., 2019; Tapscott & Tapscott, 2018), and in this sense are implementations of complete contracts. They are highly heterogeneous, and by implication so are the transactions they mediate. The automatic enforcement of a transaction, ensured by the blockchain, does not imply that both parties' expectations for a transaction are fulfilled—rather, this depends on what is specified in the smart contract. We therefore have to shift the focus from the blockchain infrastructure to the smart contract as studied artifact in order to understand how clients form beliefs about the outcome of a blockchain-mediated transaction with their vendors, and consequently how blockchain governance works on the transaction level.

Importantly, a vendor offering a transaction based on a smart contract has the option to disclose that smart contract's source code and have a third party certify that the compiled code on the blockchain is indeed derived from the disclosed source code. This option allows prospective transaction partners to predict the outcome of a proposed transaction based on pure logic. While induction is still required to a slight extent regarding the reliability of the environment (the blockchain infrastructure and the Internet), the proposed bilateral relationship can be assessed by studying the smart contract's source code, using deduction only. This deductive process enables an ex-ante certain prediction of the other party's behavior. We refer to this certainty as *deductive certainty* and argue that a blockchain transaction mediated by a smart contract with a disclosed and verified source code offers a new way to govern transactions with certainty. It differs fundamentally from established forms of governance that rely on induction and beliefs that only have some probability of becoming true.

As promising as the vision of deductive certainty might seem, achieving it requires a full deductive process—i.e., reading and understanding the entire source code; and the literature on information processing suggests that transacting parties will trade off between the effort required to read the smart contract and the risk incurred by not reading it (Fiske & Taylor, 1991; Petty & Cacioppo, 1986). This raises several questions: given the tradeoff, will smart contracts achieve their potential in practice? In other words, does the possibility to achieve deductive certainty implied by the availability of a smart contract's source code actually matter for the governance of client-vendor relationships on a blockchain? If so, under what conditions? The answers to these research questions are important for three reasons.

First, this allows us to explain the empirical observation that some vendors offering blockchain-based applications only rely on a disclosed source code, while others still invest in forming trust. Consequently, we can better understand the role of trust formation in the context of this novel IT system. Second, it helps us understand under what conditions smart contracts can fully substitute other forms of transaction governance and enable a 'trust-free' system as claimed by several proponents of this new technology (Beck et al., 2016; Greiner & Wang, 2015; Notheisen, Cholewa, & Shanmugam, 2017), and how we can integrate the possibility of attaining deductive certainty in our existing knowledge on governance mechanisms. Finally, we can gain insight into whether smart contracts on a blockchain will change how vendors govern their transactions in the future.

To answer these questions, we apply elements of the theory of reasoned action (TRA) (Fishbein & Ajzen, 1975) paired with arguments from the widely cited elaboration likelihood model (ELM; Petty & Cacioppo, 1986). We argue that, if deductive certainty is too costly to attain, smart contracts alternatively allow agents to form what we call *deduction-related trusting beliefs*. These beliefs stem from the mere possibility of achieving deductive certainty, based on either reading and verifying parts of the source code, relying on others having checked it, or simply knowing that the source code is amenable to inspection. Deduction-related trusting beliefs generally exist alongside induction-related ones.

Drawing on the TRA and prior work on trust formation (e.g., McKnight et al., 1998), we hypothesize that both types of trusting beliefs should be associated with a higher intention to engage in the exchange relationship; for vendors offering their services based on a smart contract, this translates into a larger number of exchange relationships (i.e., clients). To understand the interplay of induction-related and deduction-related trusting beliefs, we again rely on ELM arguments and hypothesize that their usefulness is moderated by the risk associated with the transaction, as well as by the effort required to form deduction-related trusting beliefs.

We empirically test our hypotheses on a novel sample of 536 decentralized applications (dApps). These applications differ in whether their source code is disclosed and verified (in which case they allow users to attain deductive certainty or form deduction-related beliefs), and to what extent their vendor provides signals allowing clients to form induction-related trusting beliefs. Using this variation, we find empirical evidence that dApps which allow both deductive and induction-related beliefs are more successful in establishing new exchange relationships. Moreover, we find evidence of complementarity between the two types of beliefs, with the potential to achieve deductive certainty through a smart contract reinforcing relational governance. To corroborate our theoretical arguments and empirical findings, we also conduct a supplementary user survey.

Our work makes several contributions to theory and practices, which we briefly list here and elaborate in the Discussion section. First, it extends prior research on governance mechanisms. It shows that induction and deduction provide a useful perspective on how parties form beliefs about a proposed transaction's reliability and help explain the effectiveness and interplay of diverse governance mechanisms. It also adds to the ongoing debate in IS and management research on whether governance mechanisms are complements or substitutes (e.g., Guo et al., 2021; Huber et al., 2013; Poppo & Zenger, 2002). Second, we add to the literature on online trust formation (e.g., Fang et al., 2014; Gefen et al., 2003; McKnight et al., 2002b; Pavlou & Gefen, 2004) by emphasizing that contrary to the claim that transactions mediated by a blockchain are supposedly 'trust-free' (Greiner & Wang, 2015), establishing trust is still an important endeavor for dApp vendors. Third, contribute to prior work on blockchain governance as we extend Lumineau et al.'s work (2020) by emphasizing the role of smart contracts for blockchain governance and introducing the concepts of deductive certainty and deduction-related trust and as we add to Hsieh and Vergne (2023), who empirically study the role of blockchain governance on the platform level, by empirically investigating the role of smart contracts in governing transactions between users and dApps. Fourth, we contribute by showcasing the use of a novel dataset that links transaction records stored on a blockchain with off-chain data on dApps, thus highlighting a new way to study this novel IT artifact.

Regarding practice, our work suggests that vendors offering their services through dApps should not only rely on a verified source code but also build trust by signaling their integrity, benevolence, and ability on their website. We highlight smart contracts as a promising new way for vendors burdened by a weak institutional environment (an untrustworthy legal system) to establish exchange relationships.

3.2 Blockchains and Smart contracts

The prevailing literature (e.g., Bartoletti & Pompianu, 2017; Egelund-Müller, Elsman, Henglein, & Ross, 2017; Lumineau et al., 2020) often considers blockchains and smart contracts as Siamese twins; yet they are two separate concepts (Halaburda, 2018), and it is important to treat them as such. First, while the blockchain infrastructure only facilitates simple money transfers, smart contracts allow the specification of arbitrary rules and thus enable a limitless diversity of transactions (Buterin, 2021). Hence, to understand blockchain governance on the transaction level, we have to examine smart contracts. Second, as smart contracts can exist without a blockchain (Szabo, 1994), understanding what governance mechanism elements they offer without a blockchain provides a more generalizable and useful notion applicable beyond the blockchain context. We therefore present the main characteristics of both concepts separately, then discuss how they work together to form a new governance mechanism.

3.2.1 Blockchains

Blockchains are an innovative combination of existing technologies, including cryptography and distributed databases (Narayanan & Clark, 2017). Their underlying idea dates back to Haber and Stornetta (1991) but has gained global renown through the introduction of Bitcoin by Nakamoto (2008). In essence, every blockchain is a ledger that is shared and maintained by a set of networked parties instead of one central authority. All participating parties share and keep an identical copy of the ledger. The ledger represents a linear event log of transactions bundled together and stored in blocks of a pre-defined size. Every block is timestamped and chained to its predecessor through a cryptographic hash function (Yermack, 2017). Thus, a blockchain is an append-only database in which every block is cryptographically linked to the first one, the ‘genesis block’ (Beck et al., 2016).

Two technical features distinguish a blockchain from a common database: a *decentralized consensus mechanism* that governs which transactions are valid and thus entered into the ledger, and the *immutability* of the past record ensured by cryptography.

The *decentralized consensus mechanism* means that all parties networked in the blockchain system must agree on the validity of a transaction. It contrasts with a centralized system (e.g., a bank) where one central authority that all parties must rely on and trust determines the validity of a transaction and distributes information. The decentralized consensus mechanism specifies how all parties verify, accept, or reject transactions together. While the consensus mechanism can vary from blockchain to blockchain (e.g., proof-of-work, proof-of-stake, Byzantine fault tolerance), all mechanisms ensure that it is prohibitively costly for a single party to gain control over the network and add invalid transactions.

3.2.2 Smart Contracts

Smart contracts, not to be confused with legal contracts, are the obligations translated into a computer script in the form of unambiguous if/then clauses, thus allowing machines to enforce the execution of these obligations automatically (Szabo, 1994). While smart contracts gained prominence thanks to being combined with the blockchain's technical features (Halaburda et al., 2019), the notion of smart contracts emerged before the advent of blockchain technology. An example of a simple smart contract not based on a blockchain is an automated recurring payment that an account holder sets up with their bank (Halaburda, 2018). We suggest that two of smart contracts' characteristics importantly affect their potential to create novel ways of governing relationships.

Contractual agreements in the form of algorithms. Unlike legal contracts where rights and obligations are written in human language which requires interpretation due to its ambiguity, obligations in a smart contract are written in the form of an algorithm in a computer language (Solidity in the case of the Ethereum blockchain). Since computers cannot deal with ambiguity, these algorithms have to be written as clear if/then instructions, which require the contract to be complete and its requirements codifiable (Lumineau et al., 2020). For each possible action, the programmer has to define a clear input, triggering condition, and output. Importantly, smart contracts are usually only stored in the form of byte code on a blockchain (Fröwis & Böhme, 2017). The party offering a smart contract has the option to disclose that contract's human-readable source code and have a third party verify that it corresponds with the executable code on the blockchain, a point we elaborate below. If a party elects to do so, then smart contracts—similarly but more formally than legal contracts—can create a shared understanding about contributions and payoffs which is central to a governance mechanism's cooperation function (Gulati et al. 2012).

Machine-based enforcement. A second important characteristic is that smart contracts run without the need for human interaction in machine-driven systems (Hsieh et al., 2018). As soon as a smart contract's triggering condition is met, the pre-specified outcomes are automatically enforced. Thus, smart contracts are not subject to the unpredictability of human actors, ensure accountability, and align expectations between the transacting parties, thus mitigating coordination issues (Gulati et al. 2012).

A blockchain supports these features of smart contracts in two ways: first, a blockchain provides a *secure execution environment*. When deployed on a blockchain, a smart contract can be triggered by transactions sent to the smart contract address. Once this happens, all networked parties execute the algorithms and compute the result of the smart contract. The smart contract's output is only stored on the blockchain if the networked parties agree on the validity of the smart contract execution according to the consensus mechanism. In this way, the smart contract is executed without the possibility of unilateral interference from any party. Second, due to the *immutable nature of the blockchain database*, the blockchain enables smart contracts to rely on a shared and agreed-upon version of the truth in the form of the data stored on the blockchain. This is important since smart contracts, once triggered, are automatically enforced without the possibility of renegotiation in the event of errors (Halaburda et al., 2019). Incorrect data would inevitably lead to incorrect results and, due to automation, this would happen over and over again. Moreover, since the smart contract is stored on the blockchain, it is not possible for any party to change the specified agreements, which again enhances accountability (Yermack, 2017).

In sum, smart contracts deployed on a blockchain enable parties to predict the outcome of a transaction with certainty. This insight has important implications for governance mechanisms.

3.2.3 The Role of Smart Contracts in Blockchain Governance

To conceptualize governance mechanisms, we look at the combination of an economics and management perspective proposed by Lumineau et al. (2020: 9), who see them as "the institutional arrangement[s] through which an agreement is enforced" when two parties engage in transactions. Governance mechanisms are necessary since transactions would otherwise be hindered by opportunism and bounded rationality (Simon, 1957; Williamson, 1985). An effective governance mechanism comprises two functions: *cooperation* and *coordination* (Gulati, Wohlgezogen, & Zhelyazkov, 2012; Malhotra & Lumineau, 2011). The cooperation function relates to aligning interests and goals (Gulati et al., 2012; Salvato, Reuer, & Battigalli, 2017). To this end, governance mechanisms need enforcement provisions that limit uncooperative behaviors by creating a shared understanding of contributions and payoffs (Srinivasan & Brush, 2006). The coordination function relates to aligning expectations and actions to achieve jointly determined goals (Gulati et al., 2012). This is done by providing accountability, predictability, and a common understanding between transacting parties (Okhuysen & Bechky, 2009).

So far, the debate about how to design an effective governance mechanism has centered on relational and contractual governance. However, Lumineau et al. (2020: 1) proposed that "blockchains offer a way to enforce agreements and achieve cooperation and coordination that is distinct from both traditional contractual and relational governance," thereby introducing blockchains as a new governance mechanism. They argue that a fundamental difference between relational and contractual governance is how blockchain governance enforces agreements. Relational governance rests on trust in the other party and shared norms, and is self-enforced through the desire to keep the relationship intact (Poppo, Zhou, & Ryu, 2008). With contractual governance, parties rely on formal, legally binding contracts that specify their rights and obligations (Poppo & Zenger, 2002; Zhou & Poppo, 2010). Such contracts can be enforced by a court or arbitrator granting legal remedies such as compensation or cancellation

(Williamson, 1985). In contrast to these established mechanisms, blockchain governance is an autonomous system in the form of code-based rules and algorithms that lead to automatic enforcement of rights and obligations by the network once certain triggering conditions are met (Lumineau et al., 2020).

When explaining how blockchain governance solves cooperation and coordination issues arising from transactions, Lumineau et al. (2020) essentially treat the smart contract and the blockchain infrastructure as a single concept. While this abstraction eases comprehensibility and captures the blockchain's enforcement function, we argue that distinguishing both —the blockchain infrastructure which is common to all transactions, and the smart contract which is specific to the transactions it governs—provides a more detailed understanding of how this new governance mechanism fulfills its cooperation and coordination function. Focusing on the smart contract reveals how blockchain governance allows potential transaction partners to form beliefs about the outcome of a proposed transaction, and how these beliefs differ from those based on established governance mechanisms. Understanding how smart contracts facilitate a new way to form beliefs is important as this is what constitutes their real power.

The smart contract details roles, responsibilities, and all possible outcomes of a specific transaction along with detailed enforcement provisions in the form of code-based rules. Uncooperative behavior is excluded by design as only a pre-defined set of inputs can trigger the desired outcome. Thus, the smart contract facilitates cooperation by limiting behavior to a pre-defined algorithm. Given this predictability, smart contracts also align expectations about the actions to jointly achieve determined goals, hence perform the coordination function. If the party offering a smart contract has disclosed its source code and had a third party verify its congruence with the byte code stored on the blockchain, then potential transaction partners can scrutinize the source code to check, using deductive reasoning, that the smart contract indeed performs the advertised function. As the following section explains, this possibility allows potential transaction partners to achieve *deductive certainty* and to form *deduction-related beliefs* about the outcome of the proposed transaction.

The above implies that even though relationships are governed by the same blockchain, variation in the specific smart contract can lead to varying effectiveness of the overall governance. The blockchain's role is to provide a secure execution environment for the enforcement specified in the smart contract and ensure the integrity of the data inputs. In this regard, the blockchain is analogous to the legal system, and the smart contract to a legal contract; and just as a fraudulent legal contract can exist even in an effective legal system, so can fraudulent smart contracts on an effective blockchain. In either case, a successful transaction requires both: a contract reflecting the parties' intentions and an effective system to enforce them. The main difference is that the legal system and contracts are written in human language, hence require interpretation, while the blockchain and smart contracts are specified in deterministic computer code.

3.3 Forming beliefs about transactions governed by a smart contract

We have shown how smart contracts on a blockchain fulfill governance mechanism requirements by solving coordination and cooperation problems in exchange relationships. In this section, we theorize that smart contracts on a blockchain differ from established governance mechanisms as they facilitate a different cognitive process to form beliefs about the other party's expected behavior and the transaction outcome. We argue that this difference is what creates the new technology's potential, enabling transactions that would not be possible under established governance mechanisms. To develop this argument, we borrow from epistemology and posit that relational and contractual governance primarily allow transacting parties to form expectations by induction, while expectations based on smart contracts can be derived by deduction alone.

3.3.1 Induction in Relational and Contractual Governance

Relational governance relies on trust and social processes to form expectations about other parties' future behavior and thus reduces the risk of opportunistic behavior and exchange hazards (Poppo & Zenger, 2002).

The relational governance mechanisms that allow parties to form expectations can be studied from an economic or sociological perspective, and both feature in the literature on trust formation in the online context (Beldad, Jong, & Steehouder, 2010). Economists rely on a game-theoretic and transaction cost perspective and emphasize the rational origins of relational governance based on the calculative nature of trust (Guo et al., 2021; Srinivasan & Brush, 2006; Williamson, 1993). In their view, beliefs in the other party's reliability are based on calculating the costs and benefits of opportunistic behavior (Lewicki & Bunker, 1996; Poppo, Zhou, & Li, 2016). Although this calculative perspective lends itself to a deductive process, the costs and benefits of opportunistic behavior are seldom based on explicit rules and are thus difficult to determine. Hence, economic actors still need to resort to induction when they infer the costs and benefits of opportunistic behavior from past observations. Sociologists emphasize knowledge about the other party and the relationship that has emerged from a prior exchange (e.g., Granovetter, 1985; Gulati, 1995; Uzzi, 1997). More recent empirical work shows that past knowledge of the other party facilitates forming beliefs about their future behavior by casting a “shadow of the future” and thereby enables future interaction (Poppo et al., 2008: 39). Particularly with online transactions, it is difficult to form trust due to their impersonal and one-time nature, however, scholars have shown that signals of a vendor’s integrity, benevolence, and ability also spur positive beliefs about the other party’s future behavior (e.g., Beldad et al., 2010; McKnight et al., 2002a, 2002b). Gefen and Straub (2004) found that even the mere presence of photographs of the vendor leads to positive considerations about the other party’s benevolence and thus increases their perceived trustworthiness. All these processes are inductive in nature as they rely purely on inferring an abstract feature of the other party (trustworthiness) from specific instances in the present or past. Reflecting on this inductive understanding, Simmel (1950) highlighted trust as weak inductive knowledge.

Contractual governance relies on formal rules to reduce the risk of opportunistic behavior and exchange hazards (Guo et al., 2021; Macneil, 1977; Williamson, 1985). Such formal rules allow parties to form expectations about future behaviors by limiting or incentivizing each party’s actions (Hoetker & Mellewigt, 2009). Contracts can offer deductive elements by specifying the consequences of certain actions. Typically, the more complex the contract, the more precisely the other party’s behavior can be deduced (Poppo & Zenger, 2002), at least in principle. This supposedly deductive process, however, is limited by three inherent characteristics of contracts. First, formal contracts are riddled with the ambiguity of natural language. Potentially different interpretations hinder deduction and may lead to varying expectations about the other party’s future behavior. Second, even if a contract is complete, the parties involved can still violate it. Finally, while a party can try to enforce a breached contract in court, predicting the outcome of court proceedings is limited by the legal system’s imponderability. In addition, the actual enforcement constitutes a deviation from what could have been deduced from the contract since this is costly and time-consuming. Past research has shown that managers actively consider this limitation and tailor their use of contracts to the strength of the legal system (Zhou & Poppo, 2010). Due to these limitations, contracting parties must still rely on prior instances of how the other party has complied with contracts, how contracts have been interpreted, and signals indicating the strength and predictability of the legal system. As such signals are typically cues based on past behavior rather than behavior deducible from explicit rules, transacting parties relying on contractual governance still need to resort to induction.

The fact that both relational and contractual forms of governance rely on induction and thus require the availability of cues, limits their applicability. This precludes transactions with no existing relationship providing information about the other party or signaling that both parties wish to continue the relationship, as well as transactions where the legal system fails to provide credible signs of reliability. A case in point is ad hoc cross-border exchange relationships with vendors from emerging markets that typically suffer from both a lack of shared past experience and an unreliable legal system. A further example is exchange relationships on a blockchain. Due to the pseudonymous nature of a blockchain (Catalini & Gans, 2020) and the current absence of legal regulations (Maume & Fromberger, 2019), existing governance mechanisms cannot explain the presence of numerous different exchange relationships on such a system.

3.3.2 The Possibility of Deductive Certainty

Smart contracts on a blockchain differ from established forms of governance as parties do not need to rely on inductive cues to form beliefs about their counterpart's future behavior. By limiting all possible actions to a set of pre-specified ones with deterministic triggering conditions, smart contracts with a disclosed and verified source code allow transacting parties to process every step of a proposed transaction with pure logic and deduce the outcome before engaging in the transaction. It is not possible to deviate from this outcome as the smart contract is collectively enforced by the blockchain network (Murray et al., 2019). The unpredictability of human behavior, as well as the possibility of renegotiation, are excluded by design (Halaburda et al., 2019). Therefore, transacting parties can predict the outcome of a smart contract transaction with certainty. As this certainty is achieved by pure deduction, we refer to it as *deductive certainty*.

However, just offering a smart contract to govern a transaction does not automatically lead to deductive certainty. Two conditions must be fulfilled to achieve deductive certainty. First, the party offering the smart contract must deliberately decide to disclose the smart contract's source code and have a third party verify that it coincides with the byte code running on the blockchain (smart contracts are usually only stored in the form of byte code on a blockchain, while the human-readable source code resides with the party offering the smart contract, see Fröwis & Böhme, 2017). An important consideration is that even though the verification is an automated process whereby the published source code is compiled again and compared to the bytecode stored on the blockchain, it requires trust in the verifying third party. This trust, however, is not specific to an individual transaction nor transacting parties and is hence excluded from our theoretical considerations. The second condition is that a prospective transaction partner needs to take advantage of the option to process all of a transaction's conditions and actions by deductive reasoning. As with any contract, it is necessary to read and understand the smart contract as completeness does not guarantee that all obligations are specified as desired by both parties. Accordingly, having a disclosed and verified source code implies what we term the *possibility of deductive certainty*. Anecdotal evidence from our interviewees suggests that they typically look up a smart contract's source code and only interact after reading at least parts of it. This is confirmed by forum entries where users actively ask where they can find a smart contract's source code.[27] There are also reasons why it is not always beneficial to reveal and verify the source code. For instance, a verified source code opens the door for imitators who copy the source code and offer the same smart contract-based service. Furthermore, revealing the source code can allow malicious third parties to identify and exploit security weaknesses.

If we can show that the potential to achieve full deductive certainty matters to the transacting parties, this will provide stronger evidence that smart contracts on a blockchain can indeed lead to a new, more effective form of governance. Since the possibility of achieving deductive certainty arguably matters most when two unfamiliar parties decide to enter a relationship, and since prior research has shown that pre-adoption beliefs differ from post-adoption beliefs about information technology (Karahanna, Straub, & Chervany, 1999; Kim, Xu, & Koh, 2004), we limit our theorizing to situations where neither party can rely on experience from a history of prior transactions.

3.3.3 Cognitive Processes to Form Beliefs in Smart Contract Transactions

To understand how the possibility to achieve deductive certainty impacts the governance of exchange relationships, we applied a cognitive perspective on how smart contracts and blockchain governance make transacting parties feel safe enough to engage in exchange behavior with an unfamiliar party. We adopted a cognitive perspective since scholars have proven it useful for understanding governance mechanisms and particularly their interplay (Weber, 2017; Weber & Bauman, 2019). We argue that smart contracts on a blockchain facilitate two cognitive processes to form beliefs about their reliability as a governance mechanism: new *deduction-related* and established *induction-related belief* formation. According to the theory of reasoned action (TRA) (Fishbein & Ajzen, 1975), beliefs provide the basis for attitudes and intentions and are thus important antecedents of behavior—in our case engaging in an exchange relationship. Since prior research broadly supports the correlation between beliefs and the intention to perform certain behavior (Ajzen, 1988; Mayer, Davis, & Schoorman, 1995; McKnight et al., 2002b), more positive beliefs about the underlying governance mechanism should result in more exchange behavior.

[27] See: https://ethereum.stackexchange.com/questions/2251/how-to-get-source-code-of-an-already-deployed-contract for an example of a user asking how to get an already deployed smart contract's source code (accessed September 15, 2022).

In the context of the TRA, beliefs are probability judgments concerning an object and an attribute (Fishbein & Ajzen, 1975). Consequently, in our case, forming a belief is establishing a link between the governance mechanism, in the form of a smart contract on a blockchain (object), and its reliability regarding the enforcement of obligations in an exchange relationship (attribute). Since reliability can only be observed in hindsight, parties considering entering a relationship governed by a smart contract need to infer beliefs about it. We argue that a smart contract on a blockchain allows forming deduction-related and inductive beliefs that differ regarding the mode of inference.

Forming Deduction-related Beliefs in Smart Contract Transactions. Deduction-related beliefs are inferred by a deductive process to ascertain the reliability of a smart contract. In a deductive process, all premises are logically linked with conclusions. If all premises are true, then only one conclusion can be true, uncertainty is dissolved, and certainty is achieved (Sternberg & Mio, 2009). Deduction-related beliefs are formed by reading the smart contract source code—if disclosed and verified—and by understanding, step by step, based on deterministic computational logic, how predefined inputs are inevitably linked to predefined outputs. It is important to note that such pure logic-based beliefs are only possible due to the immutable, predetermining, and automatically executed nature of the smart contract that excludes the possibility of opportunistic behavior (Halaburda et al., 2019), and the deficient reliability of human behavior more generally. Deduction-related beliefs are deduced from pure reason, therefore resemble a priori knowledge that is independent of experience. They climax in deductive certainty, a state where the transacting parties can predict the outcome of a transaction ex-ante with certainty, and thus do not need to rely on any other beliefs (e.g., trusting beliefs in the other party).

Although gaining deductive certainty is theoretically possible, in practice it is difficult. As Fröwis and Böhme (2017) have shown, even simple smart contracts can be complex, containing many possible dependencies and potential security vulnerabilities which all have to be understood. Nonetheless, we argue that the mere possibility to achieve deductive certainty can lead to deduction-related beliefs about the reliability of a smart contract because: transacting parties can take some deductive steps by reading parts of the source code; they can rely on other parties that have undertaken steps or the full deductive process (e.g., security audits that have proven a smart contract's reliability); or simply because they know that the source code is amenable to inspection. We conclude that offering the possibility of deductive certainty by disclosing a smart contract's verified source code allows interested parties to form deduction-related beliefs, thus increasing the probability of them entering the proposed relationship governed by the smart contract. Hence, we hypothesize:

> **Hypothesis 1.** *The possibility to achieve deductive certainty leads to more exchange relationships governed by the respective smart contract.*

Forming Induction-related Beliefs in Smart Contract Transactions. Induction-related beliefs about a smart contract's reliability to govern a relationship are inferred by an inductive process. Unlike a deductive process, an inductive process means extrapolating general, often unobservable rules from specific observations. Beliefs achieved by induction are true with some probability but are never certain (Sternberg & Mio, 2009). Since smart contracts are usually embedded in a broader setting, for instance in a web application, they provide not only cues for a deductive process, but also inductive cues. Based on prior research, we know that transacting parties can use such cues—signals of integrity, benevolence, and ability (McKnight et al., 2002b), links to other trustworthy parties (Stewart, 2003), published transaction metrics (Brengman & Karimov, 2012), photographs of the persons offering the service (Gefen & Straub, 2004), or even the application's usefulness and ease-of-use (Gefen et al., 2003)—to form positive beliefs, in the form of trust, and that these beliefs lead to more exchange behavior (e.g., Doney & Cannon, 1997; Gefen et al., 2003; Stewart, 2003).

Beliefs formed on the basis of inductive cues from the web application embedding a smart contract can increase beliefs in the smart contract governance of a relationship in two-fold ways: first, they allow inferences about the quality of the source code. For example, if the party offering the smart contract has crafted a high-quality web application, then the adopting party will form trusting beliefs in the other party's abilities (Gefen et al., 2003). In turn, these positive beliefs in the other party's ability serve as basis to extrapolate the soundness of the smart contract. Second, they allow reliance on relational governance. For example, if the vendor has signaled a high level of benevolence, the belief in their benevolence may create the perception that the vendor will always act in the best interest of their customers and do everything to fulfill their obligations, even if mistakes happen. Either way, it should be the case that the more positive inductive cues a smart contract-based application provides, the greater the chance a party forms positive beliefs and is more willing to rely on it to govern the relationship. In sum, inductive cues provided by the party offering a smart contract to govern an exchange relationship allow the other party to form inductive beliefs about the reliability of the exchange relationship and hence lead to more exchange relationships. Accordingly, we hypothesize:

> **Hypothesis 2.** *Inductive cues provided by the party offering a smart contract lead to more exchange relationships governed by the smart contract.*

3.3.4 The Moderating Role of Risk and Cost of Deductive Certainty

As the formation of beliefs involves gathering information (Fishbein & Ajzen, 1975), research on information processing is a good means of understanding our theory's boundary conditions. A great deal of information processing research has focused on the "dual-process" perspective (Fiske & Taylor, 1991: 138). A core tenet of this research is that motivation and personal capabilities are crucial for the amount of cognitive attention paid to a piece of information. Petty and Cacioppo (1986: 128) suggest with their widely-cited elaboration likelihood model of persuasion (ELM) that "when conditions foster people's motivation and ability to engage in issue-relevant thinking, the 'elaboration likelihood' is said to be high." (For others who share this view, see e.g., Elsbach & Elofson, 2000.) Thus, people are more likely to invest time and effort in accessing relevant information and forming beliefs that are well-grounded if external conditions require this. One such external motivation should be the risk associated with a transaction governed by a smart contract. The more money at stake, the more the transacting parties should be willing to process information thoroughly regarding reliability. Similarly, the literature on trust formation sees humans as 'cognitive misers' (Liu & Goodhue, 2012), perceiving that it is human nature to strike a balance between cognitive effort and acceptable risk (Baer, van der Werff, Colquitt, Rodell, Zipay, & Buckley, 2018). Consequently, the risk associated with a transaction is often considered an important factor influencing the formation of trust (McKnight et al., 2002b). Higher risk should therefore be linked to a stronger need for belief in a smart contract's reliability, and go hand in hand with increased salience of deduction-related and inductive cues. Accordingly, we hypothesize:

> **Hypothesis 3a.** *The level of risk associated with a transaction has a positive moderating effect on the relationship between the possibility of deductive certainty and the number of exchange relationships governed by the smart contract.*
>
> **Hypothesis 3b.** *The level of risk associated with a transaction has a positive moderating effect on the relationship between inductive cues and the number of exchange relationships governed by the smart contract.*

The ELM also suggests that anticipated effort influences the likelihood of elaboration. The higher the anticipated effort to process a certain cue, the more likely people will forgo processing this cue and instead seek information that is easier to comprehend (Petty & Cacioppo, 1986). Since deducing the outcome of a future transaction by reading and understanding a smart contract's source code requires more time and effort the more complex the smart contract, the ELM suggests that increasing complexity will make people forgo deductive belief formation and rely more on inductive trust cues. This reasoning suggests that the cost of achieving deductive certainty has a moderating effect on how it and inductive cues can impact the number of exchange relationships.

> **Hypothesis 4a.** *The cost to achieve deductive certainty has a negative moderating effect on the relationship between the possibility of deductive certainty and the number of exchange relationships governed by the smart contract.*

> **Hypothesis 4b.** *The cost to achieve deductive certainty has a positive moderating effect on the relationship between inductive cues and the number of exchange relationships governed by the smart contract.*

Since we can only measure the effort required to achieve deductive certainty in transactions that allow for the possibility of deductive certainty, we cannot test Hypothesis H4a. Overall, our hypotheses led to the research model depicted in Figure 10.

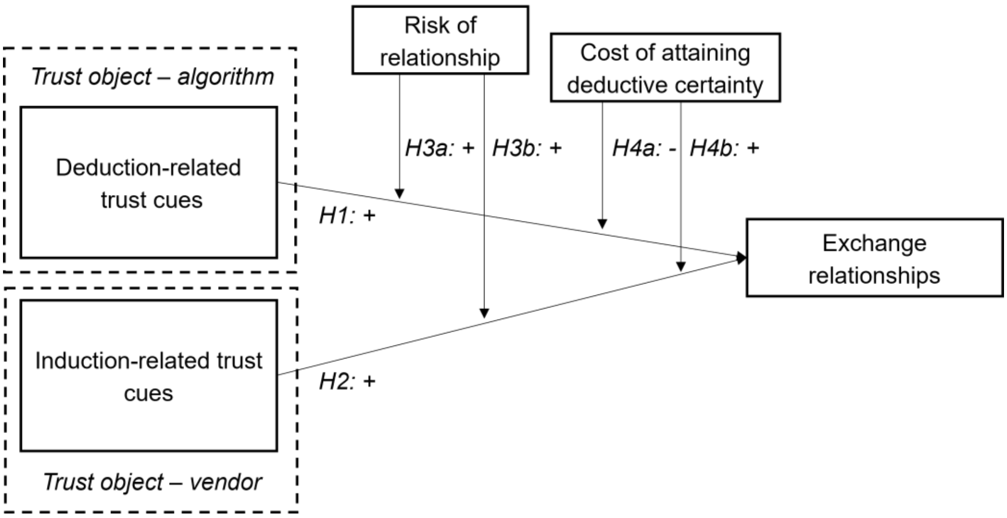


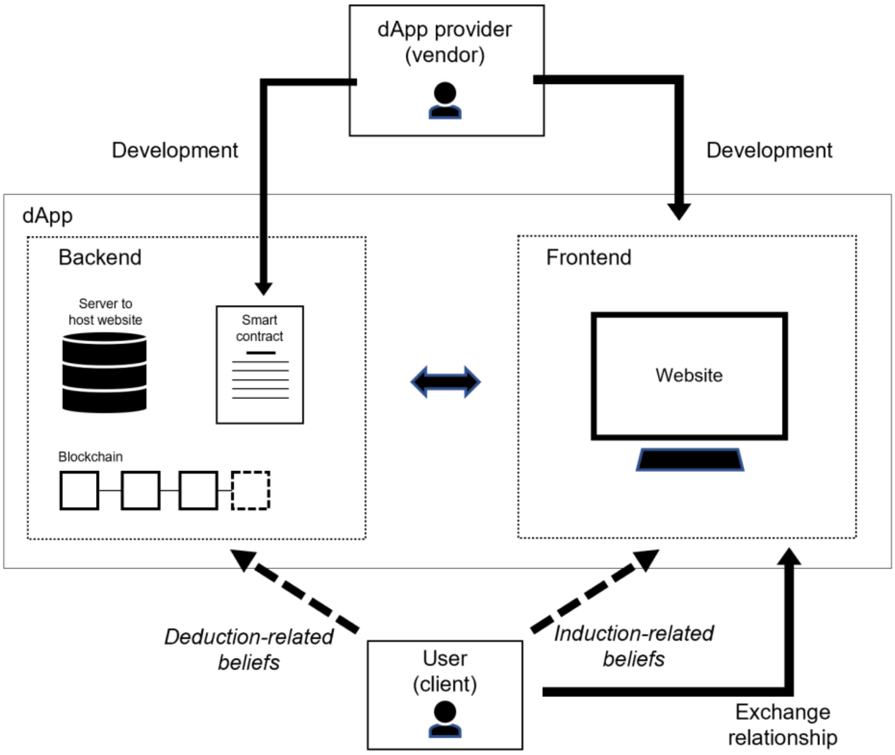
Figure 10: Research model

3.4 Method and data

3.4.1 Research Context

The empirical context for this study is vendors offering their services in the form of smart contract-based applications on the Ethereum blockchain. Ethereum was the first blockchain offering the possibility to run smart contracts. Besides simple money transfers, these smart contracts allow the implementation of arbitrarily complex transactions and thus enable the development of decentralized applications (dApps). Figure 11 illustrates the general dApp setup, the parties involved, and the objects of deduction- and induction-related trust formation.

Figure 11: dApp setup and objects of induction and deduction-related trust



With more than 540,000 daily active users,[28] a daily transaction value of over $20 billion, and a market cap of over $197 billion,[29] the Ethereum blockchain is no longer a niche phenomenon and has the potential to meaningfully change business relationships. Even though other blockchain platforms, such as EOS or Steem, also offer smart contracts, we chose dApps on the Ethereum blockchain as our context because it offers the largest number of active applications based on smart contracts covering diverse application areas (e.g., finance, social, entertainment, gambling). Furthermore, with its inbuilt programming language (Solidity) and standardized method to deploy and run smart contracts, it provides a comparable context and institutional surrounding for all units of observation in our sample. The Ethereum blockchain has been the study context for prior work (Fröwis & Böhme, 2017), however, our study examines a new aspect of smart contracts—their deductive character—not previously investigated in published research.

3.4.2 Data Collection and Sample

The study exploits a novel data set. The sample comprises cross-sectional data on 536 decentralized applications (dApps), based on one or more (in our sample precisely one) smart contracts. Every time a new user decides to send a transaction to a dApp's smart contract, they are entering a new exchange relationship with the party offering the dApp. This relationship

[28] https://ycharts.com/indicators/ethereum_daily_active_addresses, accessed September 15, 2022.
[29] https://coinmarketcap.com/, accessed September 15, 2022.

is governed by the smart contract. Since some dApps allow for deductive certainty by publishing and verifying their contract's source code while others do not, this setup is well suited to test our hypotheses.

We retrieved the data from multiple data sources and proceeded as follows: dApps were identified on *State of the dApps* (www.stateofthedapps.com), a not-for-profit curated directory of dApps running on various blockchains, with a focus on Ethereum. State of the dApps is widely acknowledged in the blockchain community and actively supported and used by Vitalik Buterin.[30] Anyone can enter data in the directory, but quality and integrity are controlled by the website's authors. At the time of data collection (29 April 2019), a total of 1,892 Ethereum-based dApps were listed on State of the dApps. To be included in the sample, a dApp had to be live at the time of data collection (i.e., running on the Ethereum main net, not on one of its test nets, which excluded 763 projects) and have exactly one smart contract (excluding another 593 projects). This allowed us to link the number of unique transactions (newly formed relationships) between a smart contract and a dApp. In our sample, dApps are distributed across five categories: Lotteries/Games (238, 44.4%), Finance (77, 14.4%), High risk (119, 22.1%), Social (73, 13.7%), and others (29, 5.4%). These categories are based on www.stateofthedapps.com (accessed April 11, 2019). We collected our data in two steps. First, we automatically retrieved online data from three different sources: State of the dApps, Etherscan.io, and Google Big Query. As outlined above, www.stateofthedapps.com allowed us to identify relevant projects and retrieve high-level data on them, such as the website link, general project information, and the smart contract address. Etherscan.io (www.etherscan.io), an analytics platform for Ethereum, provided more technical details about the smart contract, for example whether its source code has been published and verified, the date of deployment, and ultimately the source code.[31] As it is linked to the data stored on the blockchain, Etherscan.io is also acknowledged as a reliable data source by other scholars in the field (Fröwis & Böhme, 2017).

For data on the transaction level, we used the public Ethereum dataset available on Google Big Query.[32] It is a real-time image of the Ethereum blockchain, prepared so that the data can be queried with SQL commands in Google Cloud, which dramatically reduces processing time. To link the off-chain data on our dApps to the on-chain transaction records, we searched all records on the Ethereum blockchain for transactions sent to the smart contract

[30] www.stateofthedapps.com/about

[31] For an example of an Etheroll smart contract source code, see: https://etherscan.io/address/0xa52e014b3f5cc48287c2d483a3e026c32cc76e6d#code.

[32] For a detailed description of the dataset, see https://cloud.google.com/blog/products/data-analytics/ethereum-bigquery-public-dataset-smart-contract-analytics.

addresses of all the dApps in our sample and aggregated these transactions per dApp. Our second step was to manually collect data related to each dApp and rate available inductive cues. We used two independent raters for the rating process. To ensure a common understanding, we derived a framework of important inductive cues from the literature on relational governance and trust formation in particular (Gefen et al., 2003; Lim, Sia, Lee, & Benbasat, 2006; Mayer et al., 1995; McKnight et al., 1998; McKnight et al., 2002b). Subsequently, all raters assessed the first ten dApps, discussed differences, and developed word anchors for each level of all variables.[33] Finally, each rater evaluated all 536 projects. If there was disagreement, we used mean values. The overall interrater reliability of subjective trust cues was acceptable (lowest Cohen's Kappa = 0.799) and consistent with other studies using several raters in the field of management (Dahl et al., 1999; Corbett, 2007; Gregoire, Barr, & Shepherd, 2010; Mueller & Shepherd, 2016).

3.4.3 Dependent Variable

The number of unique exchange relationships is our dependent variable. It reflects the users' decision to start a new exchange relationship with the party offering the dApp. According to the TRA, such behavior is only possible with a sufficient level of positive belief in the reliability of the exchange relationship and thus in the governance mechanisms on which it relies. We calculated this number by leveraging the fact that all transactions, including the sender and recipient addresses, are time-stamped and stored on the Ethereum blockchain. To obtain the number of unique relationships, we counted the unique senders for each smart contract. Since this variable is highly right-skewed, we used its logarithm (Becker, Robertson, & Vandenberg, 2019). Retrieving the number of unique relationships for the Ethereum blockchain is not a trivial task, and since the number of those using a contract was not indicated on the company's website, herding effects can be excluded.

3.4.4 Independent Variables and Moderators

Possibility of Deductive Certainty: We captured this dApp characteristic with a binary variable that indicates whether the smart contract's source code is openly available and verified to match the bytecode running on the blockchain. This is in line with our argument that users can only achieve deductive certainty or form deduction-related beliefs if a smart contract's source code is publicly available and verified by a third party to coincide with the byte code on the blockchain and thus readable for users. We collected this variable based on a public database that indicates whether a smart contract's source code was verified to coincide with

[33] All items and word anchors associated with their corresponding levels are shown in Appendix B-2.

its byte code. This database comes from Etherscan.io, the most popular platform publishing a smart contract's source code, providing a de facto standardized way to technically verify that the published source code matches the bytecode deployed on the blockchain. All dApps in our sample with a verified source code verified it immediately after deploying the contract on the main net and verified it only on Etherscan.io. Thus, all users of a dApp with a verified source code could rely on the same information.

Induction-related Cues: Since trust is seen as the focal construct of relational governance (Poppo & Zenger, 2002; Weber & Bauman, 2019), we measured induction-related cues with items commonly used to capture trusting beliefs (Mayer et al., 1995; McKnight et al., 1998; McKnight & Chervany, 2001). These cues relate to the *perceived integrity*, *benevolence*, and *ability* of the company offering the smart contract and have been confirmed as important by prior empirical research (Mayer & Gavin, 2005). All constructs were rated on a five-point Likert scale with pre-defined word anchors for each level. Moreover, we rated *perceived usefulness* and *perceived ease of use* and the general *website appearance* on a five-point Likert scale, and availability of *third-party certificates* and *structural assurance* as dummy variables, since those constructs have also been conceptually linked to trust formation (Gefen et al., 2003; McKnight et al., 1998), and validated in the field of e-commerce and mobile banking (McKnight et al., 2002a; Zhou, 2012). Although multicollinearity was only moderately high—the highest Variance Inflation Factor (VIF) was 7.7—we performed an exploratory factor analysis to take into account the theoretical linkages among our constructs, their high correlations, and other scholars' concerns that results obtained by treating them as separate constructs might be driven by multicollinearity (Mayer & Gavin, 2005). This factor analysis supports a one-factor solution, where all of the above-mentioned variables load on the same factor. Cronbach's alpha of .92 indicates sufficient internal consistency. To construct the factor variable, we used the corresponding variables' factor scores. Below, we refer to this factor as inductive cues as it reflects the inductive way these cues are processed to create induction-related beliefs.

Risk Associated with the Smart Contract: We operationalized the risk associated with the relationship using the transaction value measured by the logged mean amount of Ether (Ethereum's internal cryptocurrency used for all smart contract transactions) sent to a smart contract for every user's first transaction. While risk is generally determined both by the amount at stake and the perceived probability of losing it, the latter is unobservable.

Cost of Deductive Certainty: To operationalize the cost of attaining deductive certainty, we used the log of the source code's length, measured by lines of code. We argue that a longer source code is ceteris paribus associated with a higher effort to read and verify the code, hence with a higher effort to achieve deductive certainty.

3.4.5 Control Variables

Besides our main explanatory variables, we controlled for the smart contract's age and category. Controlling for age is important as an older dApp has had more time to attract users. It should also control for the reputation that might form over time. However, given that most dApps in our sample are fairly new and not usually covered in the media, reputation should not be a concern. We controlled for category, since entertainment or gaming applications might require a different level of belief in the reliability of a governance mechanism than financial service applications. For a subset of dApps (n = 130), we could not identify a standalone website—these dApps were only presented at Stateofthedapp.com. Since those dApps were rated at 1 (the minimum value) for inductive trust cues, we introduced a dummy variable for having a website to account for potential systematic differences.

3.5 Results

Table 1 presents the means, standard deviations, and correlations. As noted above, no VIF statistics exceed the canonical cutoff of 10 (the highest VIF was 7.7) but due to the theoretical linkages between individual inductive cues, their high correlations, and scholars' multicollinearity concerns (Mayer & Gavin, 2005), we summarized them as one factor.[34]

We tested our hypotheses with a moderated OLS multiple regression analysis. Table 2 shows our regression models. Model 1 is a control model; Model 2 adds the direct effects of having a verified source code and the factor comprising all inductive cues; Model 3 adds the interactions with transaction value, our proxy for risk associated with the transaction; and Model 4 adds an interaction between the direct effects to assess whether the deduction-related beliefs and inductive cues are complements. Finally, in Models 5 and 6 we analyzed only dApps (n = 434) with a verified source code, since the length of code, our measurement for the cost of attaining deductive certainty, can be only observed for those dApps. Accordingly, in Model 5 we introduced the length of code as a control variable, and in Model 6 we added the interaction of the code length with inductive trust cues.[35]

[34] Using the original variables instead of other factors to calculate our models led to qualitatively the same result in terms of direct effects and interactions. The only significant constituent of inductive cues is perceived ability. As this might be due to high correlations with other variables of this construct, we computed factors to avoid falsely attributing the effect to only one variable.

[35] For all models, we assessed linearity, heteroskedasticity, autocorrelation, and normal distribution of error terms (Hair et al. 2014). To check regression assumptions, we made several residual plots, finding all residuals evenly spread across the entire scale and distributed normally. No value had extreme leverage on our model, nor were there indications of autocorrelation or quadratic relationships.

Table 1: Descriptive statistics

Variables	N	Mean	s.d.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1. Log of new relationships	536	1.65	1.12	1														
2. Verified code	536	0.81	0.39	0.25***	1													
3. Integrity rating	536	2.01	1.22	0.29***	0.04	1												
4. Benevolence rating	536	2.14	1.24	0.23***	-0.002	0.75***	1											
5. Ability rating	536	1.98	1.29	0.30***	0.05	0.89***	0.74***	1										
6. Perceived usefulness	536	1.85	1.28	0.22***	-0.04	0.76***	0.82***	0.79***	1									
7. Perceived ease of use	536	2.22	1.24	0.27***	0.06	0.82***	0.75***	0.83***	0.73***	1								
8. Site appearance	536	0.00	0.98	0.29***	0.06	0.80***	0.76***	0.84***	0.76***	0.83***	1							
9. Third-party certificates	536	0.08	0.28	0.30***	0.09**	0.41***	0.38***	0.44***	0.44***	0.38***	0.41***	1						
10. Structural assurance	536	0.05	0.22	0.12***	0.04	0.32***	0.28***	0.36***	0.35***	0.29***	0.31***	0.48***	1					
11. Inductive cues (Factor)	536	2.27	1.36	0.29***	0.05	0.91***	0.87***	0.93***	0.86***	0.91***	0.92***	0.44***	0.35***	1				
12. Site available	536	0.76	0.43	0.14***	0.06	0.51***	0.56***	0.48***	0.45***	0.56***	0.55***	0.17***	0.13***	0.61***	1			
13. Age	536	16.35	5.14	0.19***	0.04	0.17***	0.28***	0.18***	0.28***	0.16***	0.15***	0.18***	0.10***	0.21***	0.10**	1		
14. Log of transaction value	536	0.05	0.14	0.13***	0.11***	-0.15***	-0.22***	-0.15***	-0.18***	-0.12***	-0.10**	-0.12***	-0.22***	-0.17***	-0.11***	-0.24***	1	
15. Length of code	434	2.59	0.38	0.18***	n/a	0.15***	0.02	0.15***	0.09**	0.10**	0.12***	0.13***	0.01	0.11**	-0.03	-0.14***	0.15	1

Table 2: Regression results

	Dependent variable: Number of unique relationships (log10)					
	Model 1	Model 2	Model 3	Model 4	Model 5 (only verified)	Model 6 (only verified)
Deduction-related beliefs						
Verified code		0.54*** (0.11)	0.49*** (0.11)	0.50*** (0.11)		
Induction-related beliefs						
Inductive cues		0.49*** (0.05)	0.51*** (0.05)	0.20** (0.10)	0.54*** (0.06)	0.45 (0.33)
Interactions						
Verified code x transaction value			1.16** (0.59)	1.08* (0.58)		
Inductive trust cues x transaction value			-0.51** (0.22)	-0.45** (0.22)	-0.49 (0.30)	-0.50* (0.30)
Inductive trust cues x length of code						0.03 (0.12)
Inductive trust cues x Verified code				0.39*** (0.10)		
Controls						
Transaction value	0.84*** (0.32)	0.68** (0.29)	0.18 (0.43)	0.19 (0.43)	1.16*** (0.40)	1.16*** (0.40)
Website available	0.34*** (0.11)	-0.18* (0.11)	-0.17 (0.11)	-0.15 (0.11)	-0.12 (0.11)	-0.12 (0.11)
Category – Games	-0.48*** (0.14)	-0.22* (0.13)	-0.22* (0.13)	-0.21 (0.13)	-0.27** (0.14)	-0.27* (0.14)
Category – High risk	-0.10 (0.16)	0.19 (0.15)	0.19 (0.15)	0.20 (0.15)	0.17 (0.15)	0.18 (0.15)
Category – Other	-0.37 (0.23)	-0.30 (0.20)	-0.30 (0.20)	-0.31 (0.20)	-0.04 (0.24)	-0.04 (0.24)
Category – Social	-0.51*** (0.17)	-0.35** (0.15)	-0.35** (0.15)	-0.37** (0.15)	-0.50*** (0.17)	-0.50*** (0.17)
Age	0.07*** (0.01)	0.05*** (0.01)	0.05*** (0.01)	0.05*** (0.01)	0.07*** (0.01)	0.07*** (0.01)
Length of code					0.31*** (0.12)	0.31*** (0.12)
Constant	0.60*** (0.23)	0.60*** (0.22)	0.61*** (0.22)	0.59*** (0.21)	0.05 (0.40)	0.04 (0.40)
Observations	536	536	536	536	434	434
R^2	0.16	0.32	0.33	0.35	0.39	0.39
Adjusted R^2	0.15	0.31	0.32	0.33	0.38	0.38
Residual std. error	1.03 (df = 528)	0.93 (df = 526)	0.92 (df = 524)	0.91 (df = 523)	0.87 (df = 423)	0.88 (df = 422)
F statistic	14.60*** (df = 7; 528)	27.56*** (df = 9; 526)	23.46*** (df = 11; 524)	23.25*** (df = 12; 523)	27.59*** (df = 10; 423)	25.03*** (df = 11; 422)

Note: *p<0.1; **p<0.05; ***p<0.01

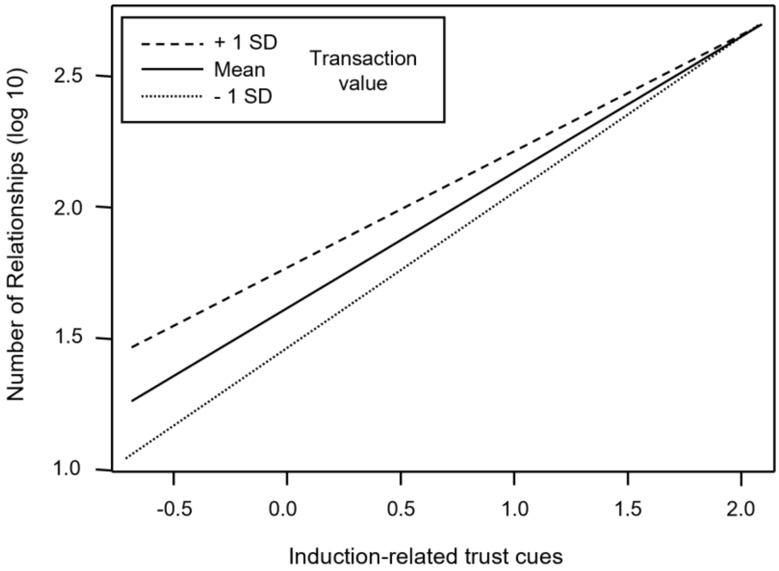
Our results indicate that the possibility of deductive certainty, offered by revealing and verifying a smart contract's source code, is positively associated with a higher number of exchange relationships in support of Hypothesis 1. This relationship is significant in Models 2-4 in Table 2 (all at p < 0.001). Based on the coefficient in Model 2, offering users a verified source code and thus the possibility to achieve deductive certainty, is associated with a 71.1 percent increase in the number of exchange relationships.

Hypothesis 2 predicts that inductive cues have a positive effect on induction-related beliefs and should therefore be associated with a higher number of exchange relationships. Given that the coefficients of inductive cues are positive and highly significant (Models 2, 3, 5 = $p < 0.001$, Model 4 = $p < 0.05$), this hypothesis is also supported. Based on the coefficient in Model 2, increasing the factor score by one unit leads to a 63.2 percent increase in the number of exchange relationships.

Hypothesis 3a predicts a positive moderating effect of transaction value (our measurement of risk associated with a transaction) on the association between having a verified source code and the number of new exchange relationships. In Models 3 and 4, we found weak support for this hypothesis ($p = 0.049$ and $p = 0.065$, respectively).

Hypothesis 3b predicts a positive moderating effect of transaction value on the association between inductive cues and the number of new exchange relationships. While we found significant results in Models 3, 4, and 6, the interaction coefficient is negative in all models, suggesting a negative, not a positive moderation. To illustrate this (Model 3), we show in Figure 12 that the association between inductive cues and number of exchange relationships is weaker for dApps with higher average transaction values.

Figure 12: Interaction Induction-related trust cues x transaction value

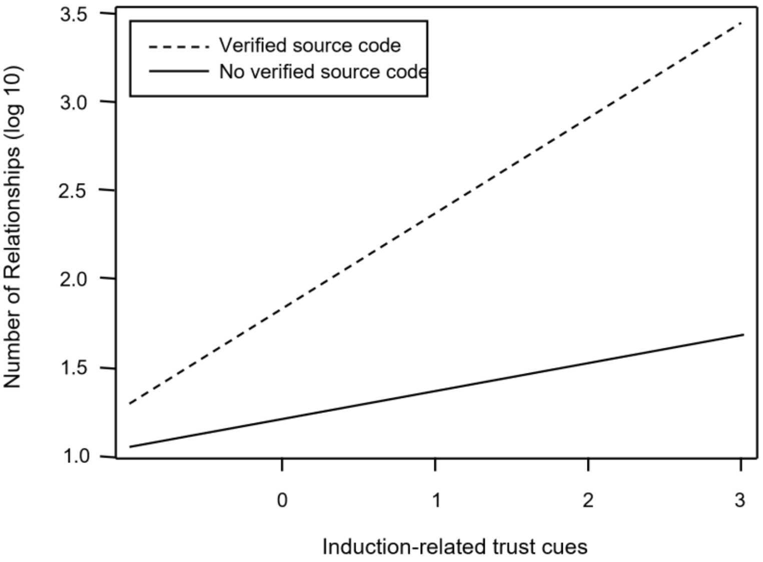


Hypothesis 4a cannot be tested with our data because having a verified source code is currently the only way to operationalize deduction-related trust. Other deduction-related trust cues like the availability of code audit certifications (already discussed in the Ethereum community),[36] would be a promising area for future research.

Hypothesis 4b predicts a positive moderation of the length of code (cost of deductive certainty) on the association between inductive cues and the number of new exchange relationships. The regression coefficient is insignificant, so the result does not allow us to reject the null hypothesis.

While we did not explicitly hypothesize complementarity between inductive cues and the possibility of deductive certainty, a positive and significant interaction ($p < 0.001$, Model 4) suggests that they complement each other. Conditional to having a verified source code, the association of inductive trust with the number of newly formed relationships seems to be stronger. This finding is depicted in Figure 13.

Figure 13: Interaction induction-related trust cues x verified source code



3.6 Additional Analyses

3.6.1 Supplementary User Survey

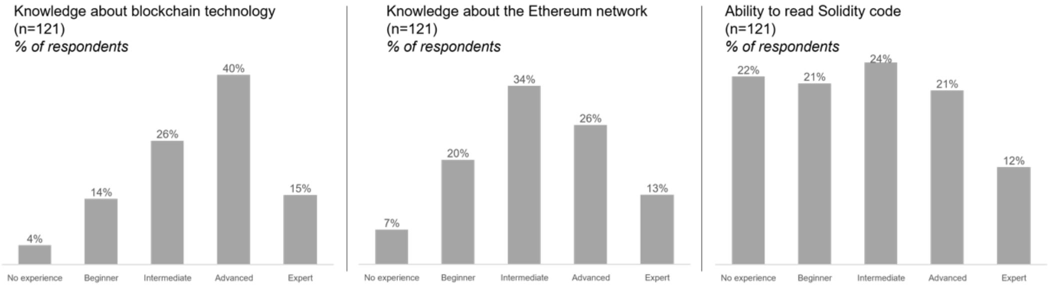
Our cross-sectional data does not allow causal claims that users really care about a verified source code because they want to use it to form deduction-related trust. For instance, there could be an omitted variable bias: possibly, more thorough vendors disclose and verify their source code as this is good practice, and at the same time they are more successful at attracting users. Thus, further analyses are needed to support our theoretical claim. We chose to conduct a user survey because this can help gain a representative understanding of users’ behavior and motivation. The aim of this survey was to discover if and why users care about a verified

[36] https://mitsoftware.com/en/token-smart-contract-audit-certification-service/, accessed September 15, 2022.

source code. In addition to the questions in our supplementary analysis, the survey questionnaire included a comprehensive set of other questions regarding users' trust formation process. These questions form the basis for the study described in Chapter 4, where the survey procedure is explained in detail.

To conduct the survey, we used a novel survey dApp[37] specifically developed to send questionnaires to dApp users on Ethereum (Weiss & Obermeier, 2021). We asked users about their demographics, their experience, and knowledge in the field of blockchain, whether they care about a verified source code, and finally to what extent they usually read the source code when considering interacting with a new dApp. We ran the survey from October 2021 to March 2022 and received a total of 121 responses. Our respondents ranged in age from 17 to 61 years old, with an average of 31. Most of our participants have a background in computer science (39 percent) or engineering (30 percent) and at least a bachelor's degree (86%). As Figure 14 shows, our sample comprises users who vary significantly in their general knowledge of blockchain technology, their knowledge about the Ethereum network, and their ability to read smart contract source codes. While 22 percent indicated having no experience in reading Solidity code, 33 percent rated their ability as advanced or even expert. This finding provides evidence that considerable numbers of dApps users can achieve deductive certainty, and thus constitute a basis for others to form deduction-related trust. At the same time, probably due to limited Solidity skills, many users will complement any trust they have formed, directly or indirectly, based on deduction, with induction-related trust.

Figure 14: Survey respondents' blockchain knowledge

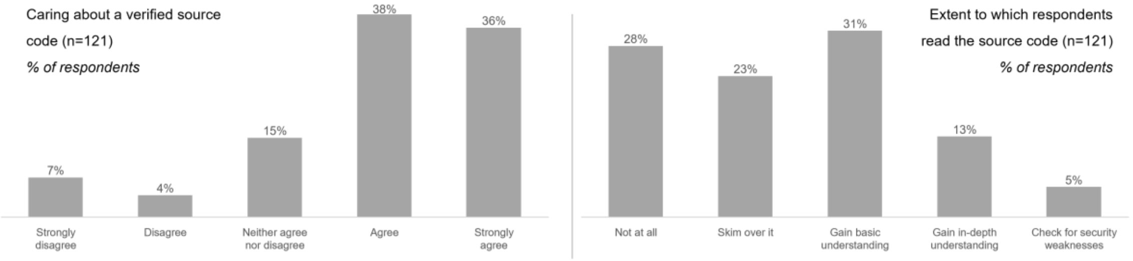


Furthermore, 74 percent of the respondents indicated they care about a verified source code (Figure 6), with 57 percent of those ticking "*because I want to read it*" as a reason. These results are in line with our theoretical predictions, providing evidence that the correlation between having a verified source code and the number of users is probably not spurious but is because users appreciate the possibility to form deduction-related trust. Regarding the actual reading of a smart contract source code, we noted that 5 percent of our respondents conduct a thorough security screening and another 13 percent gain an in-depth understanding

[37] https://blockchain-surveys.herokuapp.com/home.

of the code (Figure 15). On the one hand, this supports our argument that deductive certainty and the dispensability of trust are merely theoretical possibilities for most users. On the other hand, there is clearly potential for forming deduction-related trust: 72 percent of our respondents at least skim through the code, thus engaging to some extent in a deductive process, and all can form deduction-related trust based on the knowledge that some users have checked the code in detail.

Figure 15: Caring about and reading a verified smart contract



3.6.2 Robustness Tests

We conducted several robustness tests to control for peculiarities in our dataset.

Only dApps with Websites: Not all dApps listed on Stateofthedapp.com have a separate website allowing users to read about the dApp and form induction-related beliefs. dApps with no website have on average fewer users ($\text{mean}_{\text{log10 site available}}$=1.76, $\text{mean}_{\text{site available}}$=3359.35; $\text{mean}_{\text{log10 no site}}$=1.34, $\text{mean}_{\text{no site}}$=237.92; two-sided t-test: $p < 0.01$). To account for this potential bias, we ran the same analysis presented above with a restricted sample of only dApps with a website (n=406). This restricted sample supports Hypotheses 1 and 2; also the interaction term for both types of trust remains significant in Model 3. All other interaction terms retaining their sign but are no longer significant.

Restricting Outliers with High Average Transaction Value: Seven dApps exhibit average logarithmic transaction values three standard deviations (SD = 0.14) above the mean (0.05). Although their leverage is below a Cook's distance of 0.5 in our initial model, a visual analysis suggested winsorizing them. After limiting those outliers to the 95th percentile, again, the main effects and the complementarity term remain robust, while the interaction terms lose significance.

Coarsened Exact Matching: With this robustness test, we aimed to reduce model dependence and bias in two steps. First, we calculated simple t-tests for all independent variables. We found a significant difference between dApps with and dApps without a verified source code only regarding third-party certificates. Second, we applied coarsened exact matching to our dataset. We matched our sample based on all variables except the number of exchange relationships (the dependent variable), having a verified source code (the treatment variable), and the dApp category (since matching on category would exclude too many observations; instead, we accounted for categories through dummy variables). We manually selected cutoffs for the coarsening and pruned areas with no matches. In total, we matched 64 dApps that had a verified source code with 64 dApps without one. According to the regression analysis in Table 3, depending on the variables used for matching, having a verified source code is still positively associated with the number of users ($\beta = .52$; $p<0.01$). Although this procedure does not allow controlling for unobservable confounders (Kennedy, 2011; Wooldridge, 2010), it should mitigate a potential bias due to observable heterogeneity in our data set.

Table 3: Coarsened Exact Matching Regression Results

	Dependent variable: Number of relationships (log 10)
Verified source code	0.52*** (0.16)
Category – Games	-0.11 (0.26)
Category – High risk	0.07 (0.29)
Category – Other	-0.39 (0.47)
Category – Social	-0.37 (0.34)
Constant	1.18*** (0.25)
Observations matched	128
Observations unmatched with verified source code	36
Observations unmatched without verified source code	372

Note: *p<0.1***p<0.05**p<0.01*

3.7 Discussion and conclusion

Lumineau et al. (2020) describe blockchain technology as a new governance mechanism that differs fundamentally from established relational and contractual governance. We complement their study by pointing out the different roles of the blockchain and smart contracts. The blockchain ensures automatic, machine-based enforcement of agreements and thus corresponds to the legal system regarding contractual governance; the smart contract parallels the legal contract in that it specifies the transaction and allows parties to form beliefs about their counterpart's future behavior. Importantly, it does so in a way that differs qualitatively from how other governance mechanisms function: because of *the possibility to achieve deductive certainty*, smart contracts on a blockchain allow transacting parties to prove the soundness of a specific transaction and attain certainty about its outcome ex-ante. Thus, they do not need to rely on trust in the other party (relational governance) or the imponderability of a legal system (contractual governance) to govern the exchange relationship.

To understand the relevance and limitations of this potential, we took an epistemological perspective on how governance mechanisms allow transacting parties to form beliefs about the reliability of an exchange relationship. We argue that smart contracts on a blockchain facilitate a purely deductive process and can thus lead to full deductive certainty, while established governance mechanisms have to resort to an inductive process that leads to predictions with only some probability. Using TRA arguments (Fishbein & Ajzen, 1975) and the information processing literature (e.g., Petty & Cacioppo, 1986), we theorized that even if full deductive certainty is not reached, the mere possibility to attain certainty can lead to positive beliefs about the reliability of a transaction and thus result in more exchange behavior. We also theorize that this deductive process will be supplemented by classic inductive cues contingent on the risk associated with the transaction and the effort required to attain deductive certainty.

To the best of our knowledge, our empirical analysis is the first attempt to study the role of smart contracts in governing transaction. It shows that providing the possibility to achieve deductive certainty by revealing a smart contract's source code and having it verified, is associated with more exchange relationships (Hypothesis 1). We also found evidence that classic inductive cues as we know them from relational governance and the online trust formation literature still matter (Hypothesis 2). This finding suggests that the possibility of deductive certainty and deduction-related beliefs complement rather than replace classic inductive trusting beliefs. The novel mechanisms differ qualitatively as they do not refer to subjective beliefs about human traits but to provable properties of transaction mechanisms in the form of written code and algorithms. This finding is further emphasized as we discovered evidence of complementarity between the possibility of deductive certainty and inductive cues. Furthermore, we found that transaction risk (proxied by the transaction's value) positively moderates the relationship between the possibility of deductive certainty and the number of exchange relationships, suggesting that deduction-related beliefs become more important if more money is at stake (Hypothesis 3). In contrast, we found that risk negatively moderates the relationship between inductive cues and the number of exchange relationships, contradicting Hypothesis 4. A possible explanation is that high-risk users rely on deduction-related rather than induction-related beliefs, despite the greater effort required, since it allows them to attain a higher absolute level of belief in the reliability of a transaction. If the risk associated with the transaction is low, however, users can reach a satisfactory level of such beliefs through easier-to-process inductive cues. This interpretation would confirm Luhmann's (1979) perspective on trust as a mechanism that reduces complexity and effort. Our results regarding interaction terms should, however, be considered with caution since they did not pass our more conservative robustness tests. Therefore, this interpretation needs to be backed by further studies focusing on decision-making at the individual user level.

Our work offers four main contributions. First, we contribute to the literature by introducing an epistemological perspective on how different governance mechanisms allow belief formation about transaction reliability. We show that the cognitive processes of induction and deduction provide a useful perspective to understand the effectiveness, limitations, and interplay of various governance mechanisms. Particularly since IT systems often rely on logic-based algorithms instead of human behavior, this new perspective allows a better assessment of when IT-based transaction governance might be superior to other governance forms. In doing so, we join other scholars investigating the cognitive processes around governance mechanisms (Guo et al., 2021; Weber, 2017; Weber & Bauman, 2019). We also add to the ongoing debate whether different governance mechanisms complement or substitute each other (e.g., Hoetker, 2005; Hoetker & Mellewigt, 2009; Huber et al., 2013; Poppo & Cheng, 2018; Poppo & Zenger, 2002), as we found complementarity between smart contract-based and relational governance. We argue that the degree of certainty about the other party's future behavior and how it is obtained are further explanatory factors. In situations where certainty is achieved through a purely deductive process, trust and relational governance are no longer necessary.

Second, we contribute to the online trust formation literature by showing that trust is still relevant for adopting dApps that run on supposedly "trust-free" blockchain networks. This contribution is important as it calls for trust scholars to focus on this new field. Future scholars could study the relative importance of diverse trust cues or investigate the role of institutional safeguards such as regulations. It would also be interesting to study the role of deduction-related trust in an ongoing relationship, especially with cases of flawed relational trust and trust repair.

Third, we extend the work of Lumineau et al. (2020) by emphasizing the role of smart contracts for blockchain governance and introducing the concept of deductive certainty. Through theorizing about the possibility of deductive certainty, we pinned down the novel mechanism whereby blockchain governance allows transacting parties to form beliefs about the reliability of transactions. Investigating this mechanism allowed us to understand that blockchain governance is not limited by the codifiability and tacitness of a transaction (i.e., whether it is possible to represent a transaction in computer code and run it on a blockchain, Lumineau et al., 2020) but is also based on the smart contract features (e.g., verified source code or complexity of code) governing a specific transaction. Only the latter allowed us to explain various smart contracts' different performance on the same blockchain.

Finally, we contribute by showcasing the use of a novel data set and how it can help us understand the adoption and use of dApps on decentralized platforms. Particularly because all transactions with such dApps are meticulously recorded and publicly available on the blockchain, this new data has the potential to open up fruitful avenues for further research.

Our findings also have practical implications. Organizations offering smart contracts on blockchains should carefully consider how to leverage a mixture of blockchain governance and relational governance contingent on the complexity of the transaction and smart contract. According to our data, both a verified source code and presenting inductive cues as we know them from relational governance are positively associated with forming new exchange relationships. Therefore, contrary to the notion of being a “trust-free” system (Notheisen et al., 2017), organizations offering transactions based on smart contracts on a blockchain should not ignore relational governance and trust formation. Our study also shows that smart contracts on a blockchain are not only an automation tool but provide a novel way to ensure the reliability of transactions where previous governance mechanisms alone would fail. Thus, online vendors may consider this new technology as a governance tool that can enable transactions in environments where contractual governance, due to a weak legal system, and relational governance, due to a lack of prior history with the other party, do not suffice (e.g., transactions with vendors from emerging markets).

The study of smart contracts and blockchain is relatively new, and there are many promising avenues for future research. Our study illustrates blockchain’s potential as a promising data source for future studies: we were able to construct a novel dataset comprising not only every single transaction sent to a smart contract, but also (if disclosed) the contract’s source code, information published online by the organization offering the contract, and hand-collected data on all units in our sample. The fact that every transaction is timestamped and stored together with the sender address and amount of cryptocurrency sent in the transaction allowed us to analyze the characteristics of each user’s very first transaction and thus study the decision to engage in a new exchange relationship in the form of real transactions. Moreover, we captured all the subjective facets of relational governance evaluations through a multi-round rating.

Our study has limitations. The possibility of deductive certainty (a disclosed and verified smart contract) could also be interpreted as a factor causing more positive induction-related beliefs (perceived integrity, ability, benevolence). To account for that fact, we explicitly ignored the possibility of deductive certainty while coding established inductive dimensions, and conducted user interviews to discover if users actually read the source code. This interrelationship may have distorted our findings but could be resolved with further research on such attributional processes. Herding effects might have biased our results. However, since retrieving the number of unique users of a dApp from the Ethereum blockchain is not a trivial task, and this number is not indicated on the respective company’s website, we assume this bias is of minor importance. Finally, endogeneity concerns remain. For instance, we cannot fully control for companies’ advertising efforts. We tried to account for such effects during manual coding (of third-party certificates and structural assurance) and by adding as many controls as possible. Reassuringly, since most dApps are small at an early development stage, we can assume that most user acquisition is done via the dApp’s website, where all users are exposed to the same cues. We refrain from causal claims but see our results as initial findings on a topic that merits further study. An experimental study could help mitigate such concerns and improve internal validity. In terms of external validity, our results might be biased by a high share of early users and comparatively simple smart contracts. On the other hand, firms engaging with more complex smart contracts in the future will be willing to employ specialists and invest more effort in checking a smart contract's source code. Our study is limited by an observability bias as we can only observe conducted transactions, not potential transactions that did not materialize due to a lack of positive beliefs in reliability. The possibility of deductive certainty could be explored beyond blockchain and smart contracts: other algorithm-backed contracts as opposed to the blockchain-supported contracts studied here. Moreover, our study results could extend theoretical considerations about transaction costs—a field closely related to trust research and already linked to smart contracts and blockchain (Halaburda et al., 2019).

4 How do I trust in a trust-free system? Exploring trust formation in dApps on blockchains.

This Chapter continues studying the supposedly "trust-free" properties of dApps and complements Chapter 3 by changing the perspective from the dApp provider to the dApp user. It investigates how a user's personal disposition and perception of the transaction environment influence the formation of four distinct trusting beliefs and how these shape the user's trusting behavior. As both chapters are related, parts of the argumentation already featured in Chapter 3. But to allow readers to read chapter 4 independently, I repeat the major points.

4.1 Introduction

"You don't have to trust the counterparty. You don't have to rely on third-party agents. Transactions can be done trustlessly and safely with the help of blockchain-enabled smart contracts."[38]

With the inception of the internet and electronic commerce, scholars have emphasized the difficulty and importance of online trust formation (Gefen et al., 2003; McKnight et al., 2002b; Stewart, 2003). Mainly because the web environment does not allow parties to physically inspect products or directly observe the e-vendor's characteristics, the literature on online trust formation has put great effort into offering vendors strategies that promote the trust required to convince consumers to transact in the impersonal environment of the internet (Gefen, Benbasat, & Pavlou, 2008).

Recent advances in blockchain technology have led its proponents to question the core tenet of this literature. Instead of suggesting to alleviate the increased uncertainty of the internet through trust-building measures, such as transferring trust from the physical to the online realm (Stewart, 2003) or relying on trusted third-party institutions (Pavlou & Gefen, 2004), they argue that blockchain technology enables "trust-free systems" (Beck et al., 2016: 1), and thus potentially removes the need for trust altogether. This potential has not only been picked up by the media (Economist, 2015, 2017) and academics (e.g., Glaser, 2017) but also by blockchain app providers who claim, like in the introductory quote, that their application does not require trust in anyone.

The notion of a "trust-free" app rests on the idea that all a transaction's conditions and actions with the app are predefined in deterministic computer programs called "smart contracts." These are immutably stored on a blockchain and automatically enforced by all networked parties in the blockchain network once the smart contract receives transactions (see

[38] https://www.stateofthedapps.com/de/dapps/trustless-escrow, last updated June 14, 2022.

Section 2.2 for technical details). Since all of a transaction's potential outcomes have to be predefined and the transaction is only included in the blockchain ledger once all networked parties have reached consensus on the correct execution of the smart contract, opportunistic behavior and the possibility of renegotiation are excluded by design (Halaburda et al., 2019). The exclusion of opportunistic behavior supposedly renders trust dispensable and has led many blockchain-based app providers to promote their apps as "trust-free" or "trustless" and hence omit all trust-building efforts advocated by the online trust formation literature.

Despite the enthusiasm for trust-free transactions, there are also considerable doubts that smart contracts on a blockchain can deliver their promise to remove the need for trust in transactions. For example, Hawlitschek et al. (2018) established the notion of a trust frontier that separates human action from the blockchain system, arguing that whenever the blockchain system requires real-world information based on human action, a trusted interface is needed. Moreover, Ahangama and Poo (2016) argue that blockchain technology merely replaces trust in humans with trust in algorithms. Our discussion in Chapter 3 illustrates that even though it is theoretically possible to obtain certainty about the outcome of a transaction, due to the high costs involved, parties might not invest in the effort required to understand all possible outcomes of a transaction and instead rely on trust as a complexity reduction mechanism (Luhmann, 1979). The sample in Chapter 3 shows that some app providers still invest considerable efforts into signaling their trustworthiness by emphasizing their integrity, benevolence, and competence on their websites. This effort would not be necessary if their trustworthiness was irrelevant.

To resolve these conflicting perspectives, we explored users' trust formation process in this new field. We developed and tested a trust-building model in an attempt to explain how users form trust in blockchain-based applications that can supposedly run without trust. This addresses the questions: *Is trust in blockchain-based applications still necessary for users' decision to transact with them? And if so, how do users form that trust?*

The model we propose to study these questions is based on the Theory of Reasoned Action (Fishbein & Ajzen, 1975) and its trust formation-specific modifications by McKnight et al. (1998), who investigated how individuals form trusting beliefs (about the other party's trustworthiness), how trusting beliefs lead to trusting intentions (individuals' intentions to make themselves vulnerable to the trustee's actions), and finally how trusting intentions lead to behavior that exposes the trustor to be susceptible to the trustee's actions. Based on this logic, we theorize four ways that users (trustors) form trusting beliefs in decentralized applications (trustees) and how these beliefs then relate to engaging in transactions with decentralized applications (trusting behavior). With this model, we argue that blockchain-based applications are not trust-free but offer users a new way to form trust. This is distinct from prior ways of forming trust as it relies on understanding the transaction rules specified in the smart contract and the *possibility of deductive certainty* (introduced in Chapter 3) instead of assessing the other party's trustworthiness.

The context of this study is decentralized applications (dApps) on the Ethereum blockchain. These are blockchain-based apps (comparable to apps in the AppStore or Google Play Store). In contrast to iOS or Android apps, which run their code on centralized servers, dApps use a smart contract to encode the logic of a transaction with the dApp and a blockchain to store and execute transactions (Leiponen et al., 2021). According to stateofthedapps.com, there are currently over 2000 dApps on Ethereum offering a multitude of different services such as cryptocurrency exchange (e.g., Uniswap), lotteries (e.g., Etheroll), collectible games (e.g., CryptoKitties), insurance (e.g., Etherisk), or media sharing (e.g., Upfiring). These dApps can be used by anyone with an Ethereum-compatible wallet and a browser.

To test the trust formation model, we created a survey and used a novel approach to distribute it among dApp users. This approach relies on a new survey decentralized application which we developed specifically for our study.[39] This survey dApp allowed us to target actual users of dApps, only sending the survey to wallet addresses that have already transacted with dApps on Ethereum and thus had formed some trust in these applications. In addition, since respondents had to send their response as a transaction to our survey dApp’s smart contract, we could pseudonymously match their response to their past transaction record publicly documented on the Ethereum blockchain. Based on 121 survey responses, we found that users rely on a new way of forming trust based on the possibility of reading and understanding the smart contract. But we also found that users still complement this new way of forming trust with conventional ways established in the online trust formation literature (e.g., Beldad et al., 2010; Gefen et al., 2003; Gefen et al., 2008; McKnight et al., 2002b). Although a sampling bias induced by our survey tool[40] might limit the generalizability of our results, it provides the first empirical evidence of users’ trust formation process, allowing us to untangle the relative importance of different trusting beliefs in those who decide to interact with a dApp on the Ethereum blockchain.

[39] Daniel Obermeier developed the survey tool jointly with Johannes Weiss and published the development process at ICIS 2021, see Weiss and Obermeier (2021).

[40] To understand how users formed trusting beliefs in dApps and how this influenced their trusting behavior, the survey targeted users with prior transaction experience. The resulting sample might therefore suffer from a dependent variable bias.

This study has important contributions for research and practice. For research, it improves the understanding of how users form trust in an environment that is supposed to be trust-free. It extends existing online trust formation models by McKnight et al. (1998) ((2002b), making them applicable to the blockchain context and blockchain-based applications. It thus adds a new way of forming trust to these models and confirms the findings in Chapter 3, that in practice, dApps are not trust-free but allow the formation of deduction-related trust that complements classical trust formation approaches. Moreover, our novel trust formation model allowed us to investigate which dispositions and behaviors are associated with this new form of trust. This extension is essential as it will enable us to better understand the contingencies of users' trust-formation processes and why different users rely on different trust-building strategies.

This study also contributes to broader research by presenting an innovative approach to collecting data that allows for studying actual user behavior. In the past, most trust formation studies had difficulty observing trusting behavior and thus had to resort to measuring intentions instead of actual behavior (e.g., Gefen et al., 2003; McKnight et al., 2002a, 2002b). Our survey dApp allowed us to link survey responses pseudonymously to past trusting behavior and thus study more closely the concept of interest (behavior instead of mere intentions). Regarding practice, this study contributes by demonstrating to dApp providers that they should still care about the formation of trust and by providing four manageable strategies to enhance users' trust in their application.

The remainder of this chapter begins with our review of the literature on trust formation to provide a sound theoretical basis and outline important aspects of trust formation in decentralized applications. Subsequently, we describe the new trust formation model we developed comprising four distinct ways for users to form trust in decentralized applications. Then we describe the survey process, explain how we tested the proposed trust formation model, and present the results of our analysis. Finally, we discuss this study's contributions, implications, and limitations.

4.2 Theoretical foundations

Among management and organizational researchers, trust is widely recognized as a critical enabler of successful relationships on different levels (McKnight et al., 1998), and social exchange (e.g., Poppo et al., 2016; Zaheer & Venkatraman, 1995). For individuals, trust allows for cooperative behavior (Colquitt, LePine, Zapata, & Wild, 2011), fosters innovation and knowledge transfer (Dirks, 1999; Tsai & Ghoshal, 1998), and enhances individuals' performance when joining organizations as newcomers (Baer et al., 2018; Schaubroeck, Peng,

& Hannah, 2013). For organizations, trust is essential as it enables exchange relationships with other organizations (Lado, Dant, & Tekleab, 2008), for instance, in alliances (Faems, Janssens, Madhok, & van Looy, 2008; Lioukas & Reuer, 2015) or joint ventures (Inkpen & Currall, 2004; Polidoro, Ahuja, & Mitchell, 2011), facilitates efficient transactions (Nooteboom, 1996), and is thus connected to the organizational-level outcome of performance (Molina-Morales & Martínez-Fernández, 2009; Zaheer, McEvily, & Perrone, 1998). For these reasons, trust has also been characterized as a source of competitive advantage for organizations (Barney & Hansen, 1994).

Building on the seminal works of Mayer et al. (1995), McKnight et al. (1998), and Rousseau, Sitkin, Burt, and Camerer (1998), information systems research has confirmed that trust is also important when it comes to adopting new technology. According to this literature, trust is crucial as it helps users to overcome perceptions of uncertainty and risk rooted in the unfamiliarity with a new technology and engage in transactions that make them vulnerable to another party’s actions (McKnight et al., 2002a). Particularly since the introduction of the internet and electronic commerce, scholars have investigated new ways of forming trust. This step was necessary as the internet has shifted the trust object from a person or an organization towards the technology and the organization deploying the technology (Beldad et al., 2010). This has enabled users to transact with drastically more unfamiliar parties but simultaneously removed trust cues that users had long relied on. For instance, on the internet, it is no longer possible to physically inspect the product (Grazioli & Jarvenpaa, 2000) or directly observe attributes and the other party's behavior while looking them in the eye (Ba, Whinston, & Zhang, 1999).

Consequently, research has invested much effort in studying the antecedents of trust in online transactions (Beldad et al., 2010). For example, McKnight et al. (2002b) investigated what information on a homepage leads to more positive trust perceptions. Grabner-Kräuter and Kaluscha (2003) showed that besides trust in the seller, trust in the functionality and reliability of the e-commerce system matters, too. Gefen et al. (2003) found that a web page's perceived ease of use can enhance trust perceptions and lead to intentions to use a website. And Stewart (2003) provided evidence that trust can also be transferred from the physical realm to the website by using pictures of the physical store and hyperlinks. More recent studies have extended this knowledge beyond e-commerce to related fields such as online banking (e.g., Zhou, 2011) or mobile apps (Sarkar, Chauhan, & Khare, 2020).

Although all these studies have significantly improved our understanding of how trust is formed, the implicit and subjective nature of trust has led to a multifaceted discourse about the construct (McEvily, 2011; Rousseau et al., 1998). Therefore, it is important to specify what we mean by “trust” before we move on to investigate how trust is formed in the context of decentralized applications. The following sections define trust and briefly summarize the theoretical foundations of the trust formation process in order to understand the development of our model. We also describe a transaction with a dApp as the setting for this study.

4.2.1 Definition of trust

To define trust, we applied the definition by Mayer et al. (1995), who see trust as a willingness to be vulnerable to another party’s actions. This willingness is rooted in perceptions about the other party’s attributes that allow inferences about their trustworthiness and future behavior. We adopted this definition as it is most often used in settings involving the formation of new relationships (e.g., Baer et al., 2018; McKnight et al., 1998).

Whereas earlier accounts treated trust in a unidimensional way (Cook & Wall, 1980; Roberts & O'Reilly, 1974), more recent studies agree on the multidimensional nature of trust both regarding its roots and the stages of its formation (e.g., Lewis & Weigert, 1985; Mayer & Gavin, 2005; McKnight et al., 1998; McKnight et al., 2002a; Stewart, 2003). In line with McKnight et al. (1998), who build on Fishbein and Ajzen’s (1975) Theory of Reasoned Action, we broke down multidimensional trust into two concepts that lead to *trusting behavior*: *trusting beliefs* reflect a person’s beliefs about the other party’s integrity, ability, and benevolence (Mayer et al., 1995), whereas *trusting intentions* reflect a person’s willingness to be vulnerable to another party’s actions (Mayer et al., 1995; McKnight et al., 1998). Such a distinction is important for two reasons. First, trustors may form trusting beliefs in the other party but are still unwilling to make themselves vulnerable to their actions. For instance, they may have positive beliefs about the other party’s trustworthiness but second thoughts about the contextual factors beyond both parties’ control (Stewart, 2003). Despite the presence of trusting beliefs, these second thoughts would diminish trusting intentions and thus ultimately prevent trusting behavior. Second, the distinction allows us to separate trust formation (formation of trusting beliefs and intentions) from trust outcomes (trusting behavior) which increases conceptual clarity. Hence, we see trust formation as the process of forming trusting beliefs and intentions that then lead to trusting behavior.

Another facet of trust that can cause conceptual ambiguity is that trust is dynamic (Rousseau et al., 1998) and has to be treated differently at different stages of a relationship (McKnight et al., 1998). Whereas trust in existing relationships can be based on a history of shared experience and is therefore mainly relational in nature (Bigley & Pearce, 1998; McKnight et al., 2002b), such experience and relation-based trust are not available to strangers on first encounters (McKnight et al., 1998). This study focuses on initial trust, because forming trust when neither party is familiar with each other is very difficult but decisive for all future transactions. Furthermore, the benefit of removing the need for trust by being a "trust-free" application should have its most significant impact at the beginning of new relationships. Hence, we exclude considerations about relational trust.

4.2.2 Theoretical foundations of trust formation

Regarding the antecedents of trust, diverse streams of the trust formation literature have identified different sources of trust that can be clustered according to type: personality-based trust, cognition-based trust, institution-based trust, knowledge-based trust, and calculative-based trust (Gefen et al., 2003). The first three are more relevant for forming initial trust. The others are more critical where there is an existing relationship (McKnight et al., 1998). As our study investigates how users form trust in a dApp they have no prior experience with, we focus on the first three antecedents to build our trust formation model and exclude relation-based trust considerations pertaining to knowledge-based and calculative-based trust.[41] For the sake of completeness, here is a brief explanation of all these antecedents.

Personality-based trust reflects a person's general propensity to trust others and thus the disposition to believe that others are typically reliable (Gefen et al., 2003; Mayer et al., 1995). This disposition to trust is especially important in the initial stages of a relationship when social cues based on past behavior and experience with the other party are not yet available (McKnight et al., 1998). Later, as both parties interact more regularly, such disposition becomes less important because both parties are more influenced by knowledge based on past interactions. Importantly, this type of trust mainly depends on the trustor's general trusting stance and less on a specific trustee's characteristics.

Cognition-based trust views trusting beliefs as the outcome of cognitive processes to assess available cues allowing us to gauge the other person's trustworthiness (Gefen et al., 2003). Two examples are categorization and evaluation processes.[42] In a categorization process, individuals rely on unit grouping and stereotypes to compare how similar the other party is to them or a supposedly trustworthy party (Morgan & Hunt, 1994). A higher perception of similarity leads to a more positive assessment of trustworthiness. An evaluation process refers to the human tendency to try to regain some sense of personal control in an uncertain situation

[41] Given the conceptual diversity of trust, many other mechanisms could influence its formation. For instance, visual appearance (Beldad et al. , 2010), the size of the organization (Jarvenpaa, Tractinsky, and Vitale 2000), the website's perceived ease of use and usefulness (Gefen et al. 2003), or trust transfer from an offline presence (Stewart 2003). To provide a parsimonious trust formation model, we deem these mechanisms beyond the scope of this research.

[42] McKnight et al. (1998) build on Langer (1975) and refer to these processes as "illusion of contract." In our opinion, the term "illusion" is confusing as it suggests that trust cues do not relate to the other party's trustworthiness though at least some cues are plausibly helpful signals that indeed correlate with the other party's trustworthiness. To avoid this confusion, we do not use "illusion of control" but simply describe it as the process of evaluating trust cues.

(i.e., where the other party's future behavior cannot be observed) by paying attention to cues that allow inferences about the other party's trustworthiness (Langer, 1975; McKnight et al., 1998). Accordingly, both processes rely on available information that allows inferring the other party's trustworthiness to compensate for the lack of first-hand transaction experience and shared history. Therefore, similar to personality-based trust, cognition-based trust is crucial in the early stages of a relationship when experiential accounts are not yet available. One important difference compared to personality-based trust is that cognition-based trust is specific to a trusting relationship and thus has to be formed for every new relationship.

Institution-based trust stems from one's sense of security from guarantees, safety nets, and other safeguarding structures provided by the transaction environment (Zucker, 1986). As discussed earlier, the literature has identified multiple sources of institution-based trust, especially regarding online transactions (e.g., Beldad et al., 2010; Gefen et al., 2008; Pavlou, 2002; Pavlou & Gefen, 2004). Besides legal recourse and regulatory safeguards, scholars have identified technical protection mechanisms such as encryption and the Transport Layer Security (TLS) protocol to secure internet connections as sources of trust (Ratnasingam & Pavlou, 2002). These types of trust, similar to personality-based trust, are not specific to a trusting relationship and apply to all trusting relationships and trustors within the same institutional context.

Knowledge-based trust is created when parties have gathered enough knowledge about the other party to predict its future intentions and behavior (Gefen et al., 2003). Therefore, this type of trust is often considered a prediction process (Doney, Cannon, & Mullen, 1998). Especially past transaction experience allows parties to better understand what is happening in the present and what will likely happen in the future (Luhmann, 1979). Furthermore, it allows the transacting parties to understand how the other party usually conducts business (Kumar, Scheer, & Steenkamp, 1995). However, as this type of trust requires an existing relationship between the trustor and trustee, and grows with the knowledge gathered during the relationship, it is less relevant at the start of a transaction. Consequently, as our study focuses on users with no prior experience of a dApp, we excluded this form from our trust formation model.

Calculative-based trust is based on economic and utilitarian principles and involves calculative processes in the form of rational assessments of the costs and benefits of opportunistic behavior (Hosmer, 1995; Lewicki & Bunker, 1996). People build trust by recognizing that the cost of being caught outweighs the benefits of cheating. In other words, calculative-based trust does not require perceiving the other party as trustworthy as long as it can be assumed that the other party acts rationally and will not harm itself (Gefen et al., 2003). This form of trust is therefore also called deterrence-based trust (Shapiro, Sheppard, & Cheraskin, 1992). However, scholars have raised concerns that this should not be considered a form of trust as it is based on the absence of harmful intentions, not the belief in the other party's positive intentions, which is required by the generally accepted definition of trust (Sitkin & Roth, 1993). Thus, calculative-based trust may be closer to low levels of distrust than to what is commonly denoted as trust (Rousseau et al., 1998). It also requires an ongoing relationship as this is the only way parties can analyzes the costs and benefits of losing this relationship. If no relationship exists or the relationship is at an early stage, there is typically less to lose. Hence, we argue that calculative-based trust considerations are less important for initial trust formation and are thus excluded from our study.

4.2.3 Trust formation in the context of dApps

We describe the typical setting of a dApp transaction and outline common features of dApps that might be relevant for building trust. We briefly recap how users interact with a dApp and what information is typically available to them that could influence their cognition-based, institution-based, and personality-based trust considerations. For a more thorough review of this interaction, see Section 2.6 and the Uniswap dApp example.

Like other web applications, dApps can be accessed by any web browser and navigated through a graphical user interface (frontend). Consequently, dApps can provide all the typical trust cues. As research shows, trust cues appear abundantly on web application websites and may pertain to a high quality of the information provided (Kim, Song, Braynov, & Rao, 2005), a well-organized design (Grabner-Kräuter & Kaluscha, 2003), easy navigation (Chau, Hu, Lee, & Au, 2007), the app provider's social presence (e.g., through photographs, (e.g., Gefen & Straub, 2004; Riegelsberger, Sasse, & McCarthy, 2005), privacy statements (Lauer & Deng, 2007; Palmer, Bailey, & Faraj, 2000), third-party signals in the form of affiliations with trusted companies (Stewart, 2003) or features like a "contact us" button (Gefen et al., 2003).

Unique to dApps is how their backend works, as they run their transaction logic (backend code) in the form of a smart contract on a blockchain instead of a centralized server. To transact with a dApp, users have to connect their third-party wallet to the dApp and send transactions to its smart contract. The frontend typically eases this process by providing an interface connected to the smart contract's functions. If users click a button, the front end drafts a transaction and prompts them to their third-party wallet app where they can review all transaction information before confirming by signing with their private key. Once successfully signed, the transaction is broadcast to the pool of pending transactions, where it waits to be automatically executed and verified by all miners. The transaction only becomes effective after one of the miners has picked it up, executed the smart contract, and agreed with all other miners that the smart contract has been executed correctly. Only if this has happened, is the result of the transaction stored in a block and transmitted back to the front end. If the miners' consensus does not verify the transaction, then the whole interaction is reverted.

This unique difference has important implications for how cognition-based trust can be formed in dApps. As shown by prior research and outlined above, ordinary web applications commonly display information about app providers and even reveal details of their location and identity in the form of personal photographs (Gefen & Straub, 2004; Riegelsberger et al., 2005). With dApps, there is often a lack of such information, which impedes the formation of trust in the provider; dApps try to compensate for this lack of relational trust by relying on blockchain technology and smart contracts to provide more transparency about their underlying processes. Instead of relying on trust in the dApp provider to reduce the complexity of the interaction (Luhmann, 1979), dApps reduce this complexity by breaking down and pre-specifying all actions and possible outcomes in the form of logic-based and immutable computer code. In contrast to an ordinary app, where the transaction logic runs as a back-end code on a web server typically hidden from the user, dApps run their transaction logic as a smart contract on a blockchain. The immutable nature of a blockchain implies that dApp providers need to pre-specify all the conditions, actions, and outcomes of a transaction, that these specifications can no longer be changed, or that changes can be easily tracked (requires uploading a new smart contract). The pre-specification, however, does not automatically allow users to inspect the smart contract. On blockchains like Ethereum, the smart contract is only stored as machine-readable byte code (Fröwis & Böhme, 2017). The human-readable source code resides with the dApp providers. Only if the providers have revealed the source code and verified that it coincides with the byte code running on the blockchain, can users then inspect the smart contract. Etherscan.io (https://etherscan.io/verifyContract) is a third-party service provider offering a technical procedure that allows dApp providers to publish and verify their smart contract's source code. As discussed in Chapter 3, a verified source code allows users to inspect the smart contract, independently verify that the smart contract—hence also the dApp—actually does what it is supposed to do, and achieve certainty about the outcome of the transaction even before it takes place. To emphasize that this certainty is the outcome of full transparency, immutability, and automated execution, which allows deducing every step of the transaction from its previous steps, we introduced the concept of *deductive certainty* in Chapter 3. We further theorized that the potential of deductive certainty allows forming a type of trust fundamentally different from established forms as it depends on the transaction logic, not on the inferences from characteristics of the party offering the dApp. This fundamentally different way of forming trust is also the reason why established models might have led to confusion about trust formation and to the claim that transactions based on smart contracts on a blockchain run without the need for trust, when in reality they merely changed how trust is formed.

To highlight the difference between these types of trust, we refer to cognition-based trust based on the possibility of deductive certainty as *deduction-related trust* and the other as *induction-related trust*. “Induction-related” means this form of trust is built on gathering different trust cues that allow inference of the other party’s trustworthiness and good intentions (trusting beliefs) but never achieve complete certainty about the outcome of a transaction. To build the trust formation model, we applied both concepts and explain them with their antecedents in the next section. We also applied the same technical setup as described in Chapter 3 (for a reminder, see Figure 16).

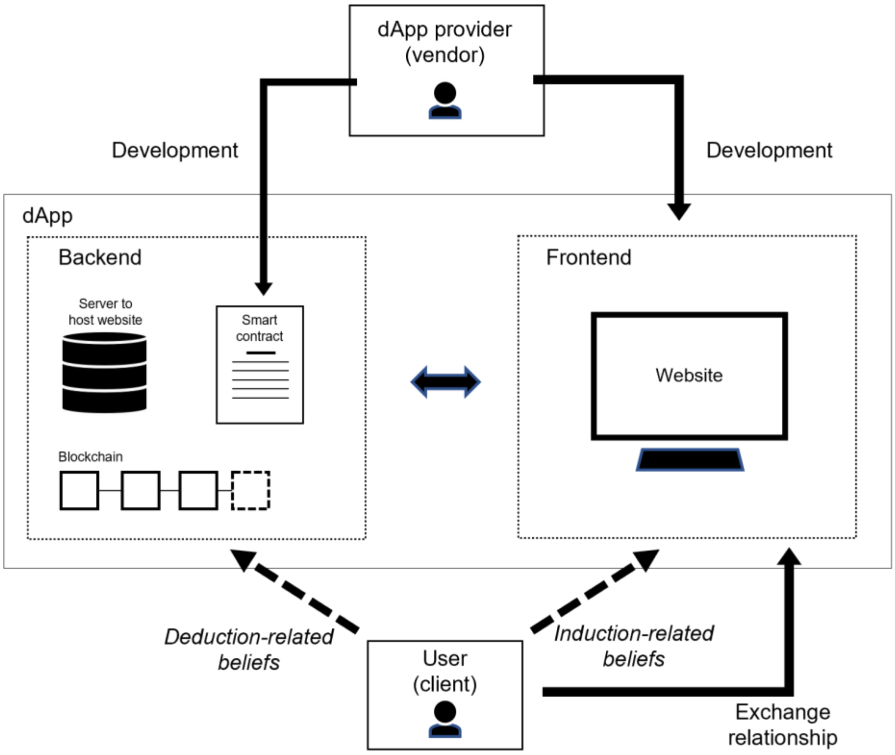


Figure 16: Objects of trust for a dApp transaction (Source: Section 3.3.3, p. 47)

To summarize, Figure 16 depicts the objects of direct cognition-based trust in the context of dApps. Although a dApp provider can still be subject to relationship-based trust considerations and thus a source of trust for the dApp, the pseudonymous nature and more substantial reliance on the technical properties of smart contracts and blockchain technology mean that the salience of the dApp provider as an object of trust is lower than for ordinary apps.

Regarding institution-based trust, a transaction with a dApp takes place on the blockchain infrastructure running on top of the internet. Therefore, the blockchain infrastructure is an additional layer to the technical and legal structures of the internet supporting the likelihood of transaction success. The blockchain infrastructure is defined by a protocol that ensures encryption, pseudonymity, and decentralized verification of transactions according to a predefined set of rules (Beck et al., 2017).[43] Since the protocol clearly defines these rules and forces everyone to adhere to them, the blockchain infrastructure is a source of institution-based trust. In contrast to direct trust in the object of trust (the dApp and its provider), this source of trust relates to the periphery of a dApp transaction and is thus universal for all transactions on the blockchain. In addition to providing a sense of security, institutional surroundings can fuel doubts about the security of a transaction and create the perception that it is risky to transact in this environment, especially if it is new or unfamiliar (McKnight et al., 2002b; Pavlou & Gefen, 2004). Whereas nowadays web browsing and transacting with standard web applications feel safe for many people, transacting with dApps exposes users to new threats. As recent research has shown, users can fall victim to scams and malicious behavior like pyramid schemes (Kell, Yousaf, Allen, Meiklejohn, & Juels, 2021), fraudulent ICOs (Zetzsche, Buckley, Arner, & Foehr, 2017), frontrunning (Daian et al., 2020), bugs in smart contracts (Zhang, Xiao, & Luo, 2020), the dApp provider knowingly or unknowingly lying about the smart contract's functionality, or simply erroneous user input (Froehlich, Hulm, & Alt). Even though these threats stem from different sources and involve various malicious parties (e.g., dApp providers, miners, users, third parties), they lead to a loss in the transaction value and the perception that transacting with dApps is risky. This is particularly troubling since the lack of regulation and legal recourse on a blockchain system implies that lost money is almost impossible to recover. Therefore, it is also important to consider the new risks arising from a transaction with a dApp when studying how users form trust in dApps.

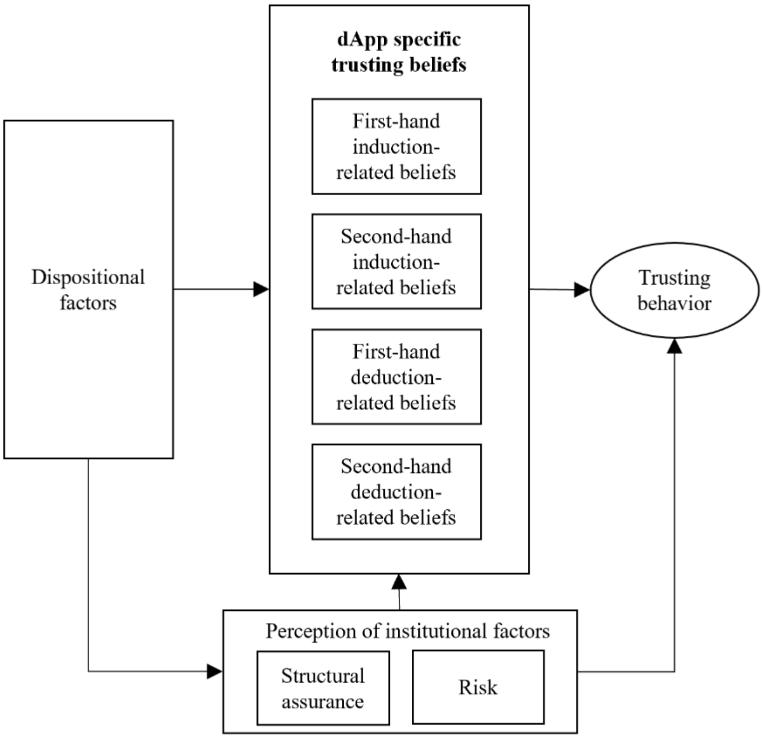
Finally, regarding disposition-based trust, presumably personal disposition correlates with the propensity to use a new dApp. As prior research has shown that dispositional factors may influence trust formation (McKnight et al., 2002a), they must also be considered in the context of dApps.

[43] Notably, most blockchain platforms also publish the protocol as open-source code, thus theoretically allowing users to independently verify that all rules have been implemented as agreed. Since this verification is only required the first time users decide to enter the platform, not every time they consider transacting with a new dApp, we did not consider this trust formation process.

4.3 A new trust formation model for dApps

With our trust formation model for dApps (see Figure 17), we posit that three sets of mechanisms—dispositional factors, perception of institutional factors, and dApp-specific trusting beliefs—influence users' trust in a dApp and thus their trusting behavior. Whereas dispositional and institutional factors are universal for all dApp transactions, trusting beliefs that are the consequence of processing dApp-specific information by induction or deduction have to be acquired for every dApp. This implies that disposition-based and institution-based types of trust are not active parts of every trust formation process. However, they still need to be considered as they may color the formation of dApp-specific trusting beliefs.

Our model details four distinct types of trusting beliefs. These account for the fact that dApps offer a new way to form trust that differs fundamentally from ordinary web applications. They also highlight why an extension of existing trust formation models is needed for dApps. In the following section we discuss our derived model's concepts, outline their linkages, and develop the hypotheses.


dApp specific trusting beliefs
First-hand induction-related beliefs
Second-hand induction-related beliefs
First-hand deduction-related beliefs
Second-hand deduction-related beliefs
Dispositional factors
Trusting behavior
Perception of institutional factors
Structural assurance
Risk

Note: Boxes that contain other boxes are not measurable constructs but are categories of constructs or "second-order" constructs.

Figure 17: A trust formation model for dApps

4.3.1 Trusting behavior

The final variable of interest to a dApp provider is users' trusting behavior, specifically if users are willing to interact with an unfamiliar dApp by sending transactions to its smart contract. dApps and their providers need to convince potential users to send a transaction because that is the only way a smart contract will be executed, and the dApp functionality used. To send a transaction, users must send money. The value of a transaction on Ethereum comprises different elements: an arbitrary amount of Ether (Ethereum's own cryptocurrency) and tokens (commonly ERC20 or ERC721), which are arbitrary digital assets defined by a smart contract and often an alternative payment method specific to the dApp (Antonopoulos & Wood, 2019). All transactions must involve fees to pay miners to verify the transaction (Bashir, 2020). The first two values are optional, but the transaction fee is mandatory for every transaction with a dApp. Consequently, if a transaction does not achieve the desired outcome, the transaction fee is lost. Even though transaction fees may seem marginal compared to the value of Ether or tokens, the fact that every transaction carries a fee always poses some level of risk to the user. Currently, due to the high gas fees on Ethereum, transaction fees can range from a few dollars for a simple money transfer to a few hundred dollars for interacting with a complex smart contract. For instance, in April 2022, the average gas fee equaled $42. We argue that the potential loss of these fees exposes users to considerable risk. Besides dropping in value, a failed transaction means the user does not receive the promised service or product. Resolving a vendor nonperformance issue on a blockchain platform is especially difficult as the party offering a service is often unknown, and transactions are typically irreversible. Furthermore, the lack of legal regulations on a blockchain platform impedes legal recourse.

To overcome the perceived risks of losing money and not receiving a promised service or product, trust is necessary to convince users to send transactions. Accordingly, we see sending a transaction to an unfamiliar dApp as an important consequence of trust, and thus an instance of trusting behavior.

4.3.2 Trusting beliefs

According to McKnight et al. (2002b: 303), "[t]rusting beliefs are perceptions of the trustworthiness of the object of trust." As established earlier, in a transaction with a dApp, the object of trust is twofold: the dApp and the dApp provider. Both offer different trust cues regarding the epistemological conjectures they allow for and thus should lead to different types of trusting beliefs. As the dApp offers a new way of forming trust, we differentiate between trusting beliefs in the dApp provider and in the actual dApp.

Induction-related trusting beliefs. Regarding the dApp provider as object of trust, the idea of trusting beliefs is rooted in early studies that considered the essence of trust to be perceptions about the other party's character traits. Scholars highlighted perceptions about the ethical character (Ring & van de Ven, 1994), the ability (Gabarro, 1978), predictability

(Rempel, Holmes, & Zanna, 1985), or a combination of diverse attributes as important sources for trusting beliefs. Later research integrated these different beliefs into three main dimensions (Mayer et al., 1995; McKnight et al., 1998): integrity, benevolence, and ability. Beliefs in the other party's integrity relate to what extent this party is perceived as honest and will keep its promises. Beliefs about benevolence relate to the perception that the other party will act in your best interest. Beliefs in ability relate to the perception whether the other party has the skills required to deliver the agreed outcomes. Although these core beliefs have slightly different names in some studies (e.g., integrity, benevolence, competence in McKnight et al., 2002b), conceptually, they refer to the same constructs. Since this set of beliefs has been used extensively in empirical research on buyer-seller relationships in general (Crosby, Evans, & Cowles, 1990; Doney & Cannon, 1997) and particularly for online relationships (Jarvenpaa et al., 2000; McKnight et al., 2002b; Ridings, Gefen, & Arinze, 2002), we also use them in the context of dApp transactions. Accordingly, if a trustor perceives that the trustee possesses the above mentioned traits, they are more likely to engage in an exchange relationship as they think the trustee will fulfill the promised outcome, act in the trustor's best interests if something goes wrong, and has all the requisite skills to deliver the agreed outcome. For example, an honest dApp provider will not lie when promising a specific service. A benevolent dApp provider will reimburse users if there are technical problems (e.g., bugs in the smart contract) and will be accommodating in the case of erroneous user input. And a competent dApp provider is less likely to accidentally program bugs in the smart contract. Consequently, having higher trusting beliefs in the dApp provider should relate positively to trusting behavior.

Trusting beliefs in the dApp provider can be further differentiated by their original source. Users can form beliefs about the other party's integrity, benevolence, and ability by evaluating information that the dApp provider offers (McKnight et al., 2002b). For instance, if a dApp provider publishes a code of conduct or the team members' track record on its website, potential users can infer beliefs about the dApp provider's integrity and ability. Users can also rely on second-hand information from another party to evaluate a dApp provider's integrity, benevolence, and ability. In the trust formation literature, these reputation-based beliefs have long been seen as crucial trust builders, particularly in commercial relationships (Doney & Cannon, 1997). Although both sources lead to beliefs about the same characteristics, it is important to distinguish them as they represent independent paths where one can be relevant while the other is not. For instance, if a dApp does not provide any information on its website, other parties' experiential accounts would still allow a new user to form beliefs about the trustworthiness of the dApp provider. Vice versa, if a dApp does not yet have a reputation, trusting beliefs can be formed based on the information offered by the dApp provider. Importantly, reputation also requires trust in the second-hand source. To create more conceptual clarity and understand the individual impact of each source of these beliefs, we decided to keep them as separate constructs.

What both types of beliefs share is their epistemological origin. Both are the outcome of an inductive process. Induction means that users must gradually gather first-hand or second-hand cues that allow them to abstract from a single observation a general prediction of the other party's trustworthiness. This prediction, however, only allows for probabilistic conclusions about the provider's future behavior and never predicts its actions with certainty. For instance, the fact that a dApp provider publishes a code of conduct on its website does not necessarily mean it will stick to this code. But confronted with uncertainty, Langer (1975) explains, people will use such cues to form tentative beliefs then look for more cues to confirm their beliefs until they have formed enough trust to feel confident about interacting with the other party. Due to this process, Simmel (1950) described trust as "weak inductive knowledge." Based on Simmel's notion, we refer to beliefs about the dApp provider's latent trust-related characteristics (integrity, benevolence, and ability) as *induction-related trusting beliefs*. Following on from the discussion above, we distinguish first-hand and second-hand induction-related beliefs.

We argue that first-hand and second-hand induction-related trusting beliefs relate positively to trusting behavior as they will recommend the dApp provider as a desirable exchange partner. Regarding first-hand induction-related trusting beliefs, if users form positive beliefs about the trustworthiness of the dApp provider from its website and believe it is honest, benevolent, and competent, then they will feel confident that the dApp provider will abide by the same traits in the exchange relationship. For example, if they perceive the provider as honest, they expect it to fulfill the agreements as promised. If they perceive it as benevolent, they expect it not to harm them intentionally. And if they perceive it as competent, they expect it to have built an unproblematic dApp. The same logic should apply with second-hand induction-related trusting beliefs, if users rely on others to tell them about their experience with the dApp provider and their assessment of a dApp provider's traits. Accordingly, we offer the following hypotheses:

Hypothesis 1a. *First-hand induction-related trusting beliefs are positively related to trusting behavior.*

Hypothesis 1b. *Second-hand induction-related trusting beliefs are positively related to trusting behavior.*

Deduction-related trusting beliefs. Regarding the dApp as object of trust, trusting beliefs are the trustor’s perception that the dApp is set up in the right way to deliver the promised transaction outcome. Therefore, rather than relating to latent and difficult-to-assess human traits (integrity, benevolence, and ability), these beliefs refer to manifest technical features that can be audited. This implies that trusting beliefs in the dApp result from a different epistemological reasoning. Instead of inducing the other party’s trustworthiness and potential future behavior, the dApp as the object of trust allows us to deduce the outcome of a transaction by relying on logical conclusions. Because deploying a smart contract requires pre-specifying all a transaction’s conditions in deterministic computer code and ensures that these conditions cannot be changed by any party (Halaburda et al., 2019), potential users can go through all these conditions and link them logically with their outcome. While legal contracts or ordinary web applications allow us to deduce the outcome of a transaction to some extent, dApps enable a full deductive process, thus achieving deductive certainty about the outcome of a transaction (see Chapter 3). Legal contracts cannot provide full deductive certainty as a party’s refusal or inability to fulfill obligations may delay the transaction's execution or jeopardize it altogether. Moreover, legal contracts also require interpretation and enforcement by a court whose decisions are usually not fully predictable. Ordinary web applications also fail to provide full deductive certainty as their transaction logic usually runs in the backend of a server controlled by the app provider and thus typically remains hidden from the user. This allows the app provider to change the transaction logic anytime without the user noticing. With a dApp, the entire transaction logic runs as a smart contract on the blockchain. Due to the immutable nature of the blockchain, changing the transaction logic would require deploying a new contract and could be easily tracked by the user. However, just offering a dApp on a blockchain does not automatically enable users to obtain deductive certainty as only the bytecode (the machine-readable version of the smart contract) is by default publicly visible (Fröwis & Böhme, 2017). To allow users to read the smart contract, the dApp provider has to reveal and verify the human-readable source code (see Section 2.2). Users can only process all the conditions and actions of a dApp transaction by deduction and achieve deductive certainty if the smart contract’s source code is openly available and verified. Then trust becomes dispensable as the transaction is no longer associated with risk. Obtaining deductive certainty, however, requires considerable skill and effort as users have to access, read, and understand the smart contract’s source code. Even skilled users might avoid this effort since, as we know from the information processing literature, humans typically strike a balance between effort and acceptable risk (Elsbach & Elofson, 2000; Petty & Cacioppo, 1986). Therefore, even if deductive certainty is achievable, and trust can become dispensable, it is not likely many people will obtain it. Yet, as we argue in Chapter 3, parts of the deductive process such as reading portions of the source code and even the mere possibility to do so can lead to forming trusting beliefs because of the association with a feeling of control. The conscious choice or mere possibility to read parts of the source code evoke this feeling of control. Since these beliefs stem from the possibility of a fully deductive process or some deductive steps, we refer to them as deduction-related trusting beliefs.

It is reasonable to assume that with induction-related as well as deduction-related trusting beliefs, users rely on both first-hand and second-hand information. Second-hand deduction-related trusting beliefs are based on information (e.g., security reports or audit certificates) from third parties (auditors) who have processed the respective dApp's deduction-related cues. As these beliefs also mitigate users' perceived risks and allow users to feel confident that transacting with a dApp is secure, these beliefs should also lead to more trusting behavior. Accordingly, we hypothesize:

Hypothesis 1c. *First-hand deduction-related trusting beliefs are positively related to trusting behavior.*

Hypothesis 1d. *Second-hand deduction-related trusting beliefs are positively related to trusting behavior.*

4.3.3 Institutional factors

There is ample research on how institutional surroundings impact the formation of trust (e.g., Pavlou, 2002; Pavlou & Gefen, 2004; Zucker, 1986). The idea of institution-based trust is rooted in prior sociological work findings that forming trust in others is facilitated by institutional factors (i.e., legal or regulatory structures) which make an environment feel secure and safe to transact in (Zucker, 1986). Especially at the beginning of a relationship, when the lack of familiarity hampers understanding and predicting the other party's behavior, institution-based trust is crucial for fostering exchange (McKnight & Chervany, 2001). Structural assurance is the most popular institution-based trust construct mentioned in prior research (Gefen et al., 2003; e.g., McKnight et al., 1998; McKnight et al., 2002b).[44] In the context of a transaction with a new dApp on a blockchain platform, structural assurance concerns all the blockchain mechanisms in place to ensure the correct execution of a transaction. These mechanisms (i.e., no double-spending, private-key cryptography, decentralized decision-making through a consensus algorithm, and rules for the automated execution of smart contracts) are defined

[44] Situational normality is another often cited institution-based trust mechanism (McKnight et al. 1998). It reflects to what extent users perceive a situation as normal and posits that the feeling of normality creates a sense of security and hence trust. We excluded situational normality as we argue that the blockchain environment is still in its infancy and developing rapidly, so it is difficult for users to judge what is "normal."

by the blockchain protocol and have to be fulfilled by all parties wanting to transact on the platform. If users conduct a transaction on a blockchain platform, they can assume it will be processed according to these rules as they are enforced by a combination of economic incentives and cryptography (Catalini & Tucker, 2018). Therefore, a blockchain's structural assurance is the protocol and its rules.

McKnight and Chervany (2001) argue that structural assurance influences trusting beliefs because people are more likely to develop them in a secure and safe environment. In other words, positive perceptions of the environment are likely to color the other party's perception. This argument was later confirmed by empirical evidence from other studies (McKnight et al., 2002b; Pavlou & Gefen, 2004). Accordingly, we also hypothesize a positive association between the structural assurance provided by the blockchain protocol and trusting beliefs. However, we only hypothesize this relationship for structural assurance and deduction-related trusting beliefs since the protocol only supports the contract's correct enforcement but does not relate to the trustworthiness of the party offering the dApp. Rather than incentivizing and promoting honest behavior, the blockchain protocol assumes malicious parties and rules out opportunistic behavior by design. It even obfuscates the transacting parties' identity.

Consequently, we hypothesize the following:

Hypothesis 2a. *The perceived structural assurance provided by blockchain technology is positively related to first-hand deduction-related trusting beliefs.*

Hypothesis 2b. *The perceived structural assurance provided by blockchain technology is positively related to second-hand deduction-related trusting beliefs.*

The role of trust in initiating an exchange relationship is to help people overcome their perceptions of risk. As discussed, even if the blockchain protocol tries to rule out opportunistic behavior in the execution of a transaction, people can still lie about the transaction rules encoded in the smart contract or make mistakes when writing the contract. Furthermore, the prevalence of scams, pyramid schemes, security issues, and hacking in the public media has cast blockchain platforms in a bad light. For example, the notion of Ethereum as a "Dark Forest"[45] suggests threats are lurking at every corner. Given the relevance of these concerns, we explicitly included the perceived risk of transacting on a blockchain platform in our model. This is in line with McKnight et al.'s (2002b) notion of perceived web risk. We translated this to the dApps context, referring to it as "perceived blockchain risk," the extent to which a user is convinced that sending transactions on a blockchain platform is unsafe.

[45] https://www.paradigm.xyz/2020/08/ethereum-is-a-dark-forest, accessed September 15, 2022.

McKnight et al. (2002b) have shown that perceived web risks negatively relate to trusting intentions. Therefore, we argue that perceived blockchain risks should have a negative association with trusting behavior (i.e., the consequence of trusting intentions) and hypothesize the following:

Hypothesis 2c. *The perceived risk of transacting on the blockchain network is negatively related to trusting behavior.*

4.3.4 Dispositional factors

People differ in their consistent tendency to be willing to depend on and trust others (McKnight et al., 1998). Therefore, a person's disposition to trust is an important construct that influences the development of specific trusting beliefs (McKnight et al., 2002a). Despite scholars agreeing on the theoretical importance of dispositional factors, researchers encountered difficulties proving the hypothesized relationship between dispositional factors and trust formation (Holmes, 1991). For example, Johnson-George and Swap (1982) found that constructs like disposition do not predict an individual’s trust. In contrast, Mayer et al. (1995) presented a review of organizational research supporting the importance of dispositional factors for forming trust. Based on this research, they proposed that a person’s disposition to trust is particularly important for forming trust if there is no prior information available about the trustee. McKnight et al. (1998: 477) build on this proposition and try to explain the mixed empirical findings by arguing that “the time frame of the relationship is important in predicting the effects of disposition to trust.” In their view, dispositional factors are salient at the beginning of a relationship but will be overlaid by other factors in an ongoing relationship.

In the context of e-commerce, McKnight et al. (2002a) presented empirical evidence on the importance of dispositional factors by showing that *faith in humanity* and a person’s *trusting stance*—two subconstructs of disposition to trust—are positively related to trusting beliefs and trusting intentions; also that faith in humanity reflects a person’s tendency to assume others are generally honest, well-meaning, and reliable. They separated this construct into three subconstructs for faith generally in others’ integrity, benevolence, and competence and found significant associations between faith in humanity and the formation of trusting beliefs. Applying this idea to our context, the general others are the dApp providers. Hence, *general faith in dApp providers* reflects a person’s tendency to assume that the dApp provider is honest, benevolent, and able to deliver the promised outcome of a transaction. *Trusting stance* is an economic choice variable. It is not about perceptions of general others’ attributes, but a personal approach to dealing with others (McKnight et al., 2002a). It reflects a person’s tendency to assume, regardless of the other’s characteristics, that dealing with other people will have a beneficial outcome (Riker, 2017). Applying this idea to our context, where the trusting beliefs refer to beliefs in the dApp provider's characteristics (i.e., induction-based beliefs) and beliefs enabled by underlying technology (i.e., deduction-based beliefs), we extend the work of McKnight et al. (2002a) and posit two separate subconstructs for trusting stance: *trusting stance towards people* and *trusting stance towards technology*. Trusting stance towards people is discussed in the literature and refers to a user's strategy to trust dApp providers until they prove them wrong. Trusting stance towards technology refers to a user's strategy to assume that a technology works as intended until they encounter serious flaws. Differentiating both stances is important in the context of dApps as users might not care about the party offering the dApp as they assume everything is pre-specified in the smart contract. However, to form trusting beliefs, they still need to make assumptions about the functioning of the blockchain platform and the correct execution of the smart contract.

The formation of trusting beliefs may differ for users with high versus low dispositional factors. For example, while users with a high faith in dApp providers might value as positive a dApp provider's trust-building signals, such as talking about their honesty and benevolence, users with low faith in dApp providers might perceive such trust-building attempts as suspicious and consequently be more reluctant to form trusting beliefs. As these beliefs pertain to the dApp provider's characteristics, they are induction related. The formation of deduction-related trusting beliefs should not be affected since these do not depend on the perceptions of the dApp provider but solely on the perceptions of the smart contract's transaction logic. Similarly, the trusting stance towards other people should also influence the interpretation of the relationship with the dApp provider and thus lead to induction-related trusting beliefs. The trusting stance towards technology only relates to the perception that technology in general is reliable and thus should only influence the formation of deduction-related trusting beliefs. Accordingly, we hypothesize:

Hypothesis 3a/b. *Faith in dApp providers is positively related to first-hand/second-hand induction-related trusting beliefs.*

Hypothesis 3c/d. *Trusting stance toward other people is positively related to first-hand/second-hand induction-related beliefs.*

Hypothesis 3e/f. *Trusting stance toward technology is positively related to first-hand/second-hand deduction-related trusting beliefs.*

Moreover, users' knowledge of blockchain technology should also matter for forming trust in a dApp. On the one hand, it should influence to what extent users can read the smart contract and understand that its encoded rules are connected to the outcome of a potential transaction with a smart contract. Therefore, it should influence to what extent users rely on first-hand deduction-related trusting beliefs. On the other hand, a better understanding of the technology should also lead to a better understanding of all the safeguards that blockchain technology provides for the secure execution of a transaction and thus should also influence the perceived structural assurance. This leads to the following hypothesis:

Hypothesis 3g/h. *Knowledge about blockchain technology is positively related to first-hand deduction-related trusting beliefs and the perceived structural assurance provided by blockchain technology.*

4.4 Method

Besides contributing to the theory of trust formation by developing a new trust formation model that accounts for the possibility of deductive certainty, this study also offers a new method for conducting surveys on a blockchain. This approach is especially useful as it allows us to pseudonymously link survey responses to participants' transaction history. Specifically, it allows us to link trust formation issues with observing the person's actual transaction behavior stored on the blockchain. Studying actual behavior was often called for in prior research (McKnight et al., 2002a) but was difficult to implement because most trust studies relied on surveys and self-report questions regarding the behavioral outcome of trust. Our survey method aims to provide initial insights into how current dApp users typically form trust in this new type of application, if their trust formation process differs from prior trust formation models, and if this influences their transaction behavior. Since we rely on observing past behavior, this study represents a snapshot. The results of future studies may change because of the dynamic developments with dApps and the ongoing diffusion of blockchain technology bringing an influx of new users with different characteristics. However, even if the relative importance of deduction-based and induction-based trusting beliefs changes for future users, our findings, that both are important and must be comprised in a trust model for dApps, should remain unchanged.

Given that this new approach has distinct advantages but also important caveats compared to ordinary online surveys, we first describe the novel survey tool, before discussing the survey design, scale development, data collection and sampling, and finally, how we conducted the survey.

4.4.1 A dApp-based survey tool

We used a survey dApp specifically developed for this study.[46] Certain features of the dApp are important for understanding our survey process and might impact the outcome. For a detailed technical description of all our dApp's features, see Weiss and Obermeier (2021).

The survey dApp has three main components: a client or website (frontend) that provides the interface to participate in the survey, a webserver (backend part 1) that stores the data (survey and survey responses), and a smart contract that communicates with the blockchain and manages the survey process (backend part 2). Figure 18 depicts all the main components, interacting parties, and actions associated with the survey process.

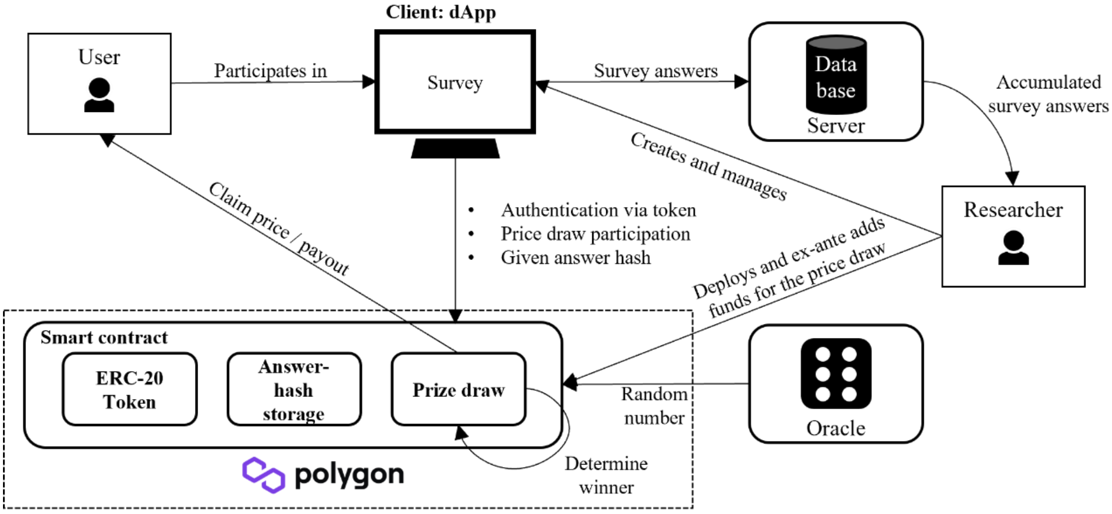


Figure 18: Our survey dApp setup (Source: Weiss & Obermeier, 2021: 5)

While the client and server are similar to other commonly used survey tools (e.g., Google forms, Unipark, SurveyMonkey, Qualtrics), the smart contract distinguishes our dApp. It pre-specifies the survey process and automatically manages every step automatically without the researcher being able to interfere. This setup, aiming to enhance trust and transparency in the overall survey process (Weiss & Obermeier, 2021), is divided into three phases: (1) survey development and deployment, (2) survey participation, and (3) survey completion.

In the first phase, the researcher compiles the questionnaire and specifies all the survey's surrounding conditions (survey period, participation requirements, and prize draw) in the dApp's survey creation area. Once all conditions are predefined, the dApp automatically creates a smart contract and drafts a contract creation transaction. After the survey creator has

[46] The survey tool was jointly developed by Daniel Obermeier and Johannes Weiss. The development process is documented in the ICIS 2021 proceedings (Weiss and Obermeier 2022). The idea, concept, and first version of the smart contract were developed by Daniel Obermeier, the final dApp by Johannes Weiss and David Stuebing. For the survey dApp, see: https://www.blockchain-surveys.com/home .

confirmed and signed this transaction with their wallet and private key, the smart contract is deployed on the blockchain,[47] and the survey is made accessible to participants.

In the second phase, participants can answer the survey questionnaire and submit their responses to the smart contract. Participants first need authentication by connecting their wallets to the dApp and verifying that they possess a survey token granting them the right to take part in the survey. This token is defined and managed by the survey and can be used to invite participants. Alternatively, the contract can be set up so that participants withdraw tokens from a public faucet. While sending the token to a specific group of prospective participants enables limiting the survey to a specific group, their wallet addresses have to be known ex-ante. After the participants are authenticated with the smart contract, they can complete the questionnaire. Our dApp tool's answering process resembles other online survey tools except that at the end, the answers are not only submitted to a webserver but compiled in an answer hash sent to the smart contract. Thus, once they have completed the survey, participants have to submit their answers by signing a transaction with their wallet.

The final phase is automatically initiated once the predefined end of the survey period is reached. In this phase, the prize draw is held and it is no longer possible to take part in the survey. A random number is generated by calling the smart contract's "prepare random number" function. This function is public and can be called by anyone, but only once for each survey. The function calls an external oracle (the Chainlink random number oracle in our case) and requests a random number. Based on this random number, a predefined number of winners is drawn from the pool of participants. These winners can then withdraw the promised amount from the smart contract.[48]

Our dApp-based survey tool offers four unique advantages. First, as it requires user authentication to prove possession of a survey token, we can ensure that no user participates twice, which would distort the survey results. Incidentally, this authentication method does not rely on tracking personal information, thus avoids multiple participation and ensures data privacy. Second, since all users have to submit their answers by sending a transaction to our smart contract, we know their wallet addresses. As all the transactions a wallet has ever sent are publicly stored on the blockchain, we can use this information to observe the user's past transaction behavior and pseudonymously link it to their survey responses. Third, since all answers are hashed and stored on the blockchain, we can prove our data's integrity without

[47] The smart contract runs on the Ethereum-based Polygon blockchain, a layer-2 blockchain in the Ethereum main net. We chose the Polygon network due to Ethereum's high transaction fees, and as Polygon allows us to use the same wallet and wallet address as the Ethereum main net, we can track past transactions.

[48] The prize is not directly transferred to the winners since this would expose our smart contract to security threats. Instead, we used the common practice of implementing the Solidity Withdrawal Pattern.

revealing our participants' personal information. For example, by looking up our dApp's smart contract, we can easily verify the number of participants by counting the transactions that called the function "add answer hash."[49]

Finally, our smart contract's immutable and predefined nature allows us to offer monetary incentives for participating in our survey in the form of a tamper-resistant and automated prize draw. Unlike other online surveys where participants need to trust the researcher to conduct the prize draw and pay out the reward as stipulated, our solution allows participants to read the smart contract source code and form deduction-related beliefs in the reliability of the prize draw. Furthermore, the fact that our dApp only allows participants to start the survey after the rewards are escrowed into the smart contract and that the random number comes from an external source, prevents by design the researcher manipulating the prize draw.

Notwithstanding these advantages, our dApp has potential drawbacks. Especially its additional complexity of connecting with a wallet, authenticating and sending a transaction to the smart contract might reduce the overall response rate and bias our sample as it deters participants who are less technologically savvy. According to Dillman (1978), one important lever for increasing the response rate and mitigating such a bias is to reduce the cost of taking part in the survey. Thus, the dApp was developed to simplify using the application and automate as many processes as possible. Moreover, to ensure that participants with little prior experience in the blockchain and dApp space could use the survey dApp, a user study was conducted during the dApp development process (see Weiss & Obermeier, 2021). Another caveat is that due to high costs on the Ethereum main net, we decided to run the dApp on the Polygon net (i.e., an Ethereum-based layer-2 blockchain). Although Polygon is fully compatible with Ethereum and set up so that users can use the same wallet and wallet address to conduct transactions, survey participants might use a different wallet address for transactions on the Polygon network.

4.4.2 Survey design

We developed our survey based on a thorough review of the trust formation literature (see Section 4.2) and multiple rounds of feedback. Our final survey, with a total of 62 questions, was divided into three main sections: (1) demographics, background, and personal disposition; (2) smart contract-related trust formation; and (3) trust in the party offering the dApp. Each main section comprised multiple subsections to give participants a guidance structure.

[49] See Appendix C-2 for a screenshot of Polygonscan.io showing our smart contracts' publicly stored transaction records.

The survey first defined the acronym dApp to ensure all participants had a common understanding. We then outlined the motivation for our research project on an abstract level to not bias the survey results and presented key facts about our survey tool since it was new to all the participants.[50] Every main section also had a short introduction providing further instructions. For example, having explained that the context of the study was the Ethereum blockchain, we asked them to bear this platform in mind when answering the questions. Further, since we were interested in how users form trust in new dApps, we also asked them to think of a typical new (i.e., new to them) dApp that they might consider using and answer all questions with this hypothetical dApp in mind. We asked users about a hypothetical new dApp, not how they formed trust in the most recent new dApp they had already interacted with to ensure their answer pertained to an average and more generalizable new dApp rather than a specific dApp.

4.4.3 Scale development

The items used to operationalize our trust formation model are based on McKnight et al. (2002a) and adjusted based on a few other sources.[51] We mostly relied on reflective operationalizations of our constructs and only resorted to formative measures when there was a clear causal relationship, and reflective operationalizations would be inappropriate (Diamantopoulos, Riefler, & Roth, 2008; Podsakoff, MacKenzie, Podsakoff, & Lee, 2003). We used negatively worded items for some dispositional factors but not for trusting beliefs because negatively worded trust items are related to distrust (Wrightsman, 1991), which differs conceptually from trust (McKnight & Chervany, 2001). Finally, to keep the questionnaire at an acceptable length, we selected one item from each established and validated scale (Cronbach's alpha > 0.85) and only included these individual items.

To measure trusting behavior, we relied on the number of dApps a user has transacted with and the total number of transactions. Especially the number of dApps adopted is an important measurement, as every adoption of a new dApp requires forming initial trust in the dApp and the dApp provider. The total number of transactions allowed us to assess whether the trust formed had led to an active exchange relationship or only a one-time interaction.

We operationalized first-hand and second-hand induction-related trusting beliefs as formative constructs created by the subconstructs measuring perceptions about integrity, benevolence, and ability. We adapted these subconstructs from McKnight et al. (2002a), who based their scales on ones in the social psychology literature (e.g., Johnson-George & Swap,

[50] See Appendix C-3 for the introduction to our survey.

[51] Appendix C-5 shows the operationalization of our constructs and a correlation table.

1982; Rempel et al., 1985; Wrightsman, 1991). We selected one item per subconstruct since these constructs exhibited sufficiently high Cronbach's alphas (the lowest Cronbach's alpha was 0.91). We argue that the subconstructs of perceived integrity, benevolence, and integrity form trusting beliefs since each can be a causally independent source of trustworthiness perceptions. For instance, if a trustor believes a trustee is honest and benevolent but incompetent, the trustor could still have sufficient trust as they are convinced the trustee will not lie about mistakes and do everything possible to deliver the promised result.

First-hand and second-hand deduction-related trusting beliefs were also operationalized as formative constructs. Since these constructs are new to the pertinent literature, we based their development on the theoretical arguments in Section 4.3 and multiple feedback loops with users who were familiar with the subject but not survey participants. Consequently, to measure first-hand deduction-related trusting beliefs, we used four items that reflected different mechanisms in the deductive process: trusting beliefs thanks to the mere possibility of reading the source code without actually reading it, trusting beliefs after skimming through the source code to understand its basic idea and functionality, trusting beliefs as a consequence of reading the source code to understand that it is doing what it is supposed to do, and trusting beliefs as a result of a thorough check and verification that the smart contract is error-free. To measure second-hand deduction-related trusting beliefs, we used two sources of second-hand deduction-related cues: hearing from peers that the source code is error-free and security audits by third parties. Again, we operationalized these as formative constructs because they represent causally independent sources that can lead to trusting beliefs.

For institutional factors, we drew from McKnight et al. (2002b). They use two subconstructs to account for the impact of institutional surroundings on forming trusting beliefs and the decision to engage in trusting behavior in an online context. The first of these, *perceived web risk*, accounts for marketing researchers' finding that the perception of risk associated with an environment affects purchasing behavior (Peter & Tarpey, 1975). To account for how perceived risks arising from the blockchain transaction environment impact trusting behavior, we introduced the related construct *perceived blockchain risk*. To measure this construct, we used the three web risk items perceived by McKnight et al. (2002b) that refer to trusting intentions regarding paying on the internet and replaced "the web" with Ethereum as the institutional transaction context. For example, instead of "I hesitate to enter my credit card information on the web," we used "I hesitate to send money on Ethereum." The second construct McKnight et al. (2002b) used was *structural assurance of the web*. It accounts for the fact that the perception of available security guarantees and safety nets might color the formation of trusting beliefs. As discussed, blockchain technology also provides a distinct set of security features that should facilitate the formation of trust. To account for the impact of these mechanisms, we introduced the construct *structural assurance of the blockchain*. To measure this construct, we relied on a blockchain's core technical features that ensure the correct execution of a transaction (automated execution of smart contract through a decentralized consensus mechanism, immutability, and transparency, (Beck et al., 2017; Lumineau et al., 2020) and applied three items regarding the perception of these Ethereum blockchain features.

To account for dispositional factors, we relied on four subconstructs: (1) faith in dApp providers; (2) trusting stance towards people; (3) trusting stance towards technology; (4) experience with blockchain technology. As with all induction-related trusting beliefs, we dimensionalized faith in dApp providers into integrity, benevolence, and ability and adopted one item per dimension from an existing scale (McKnight et al., 2002a). Since our object of trust is a human component (the dApp provider) and a technical object (the dApp), we used two constructs that measure the prospective user's stance towards trusting the person and trusting the technology. To measure the stance towards people, we used the scale developed by McKnight et al. (2002a). To measure the stance towards technology, we adapted the same scale, replacing "people" as the object of trust with "technology." We further changed the wording of one item for each construct to a negative wording. Finally, we used three items relating to respondents' general knowledge about blockchain technology, their specific knowledge about the Ethereum protocol, and their ability to read the smart contracts' source code (Solidity).

Besides items that operationalized our constructs, we added questions regarding the survey participants' background and demographics, the reason why they use Ethereum, their first type of transaction, and transaction frequency.

4.4.4 Data source and sampling

As our study explores how users form trust in new dApps on Ethereum, it is therefore important to have a broad sample representing the current user base. However, targeting a representative sample of dApp users is complicated for two reasons. First, due to the infancy of the field and still-growing adoption, it is difficult to evaluate who qualifies as representative user. Particularly since early adopters' characteristics differ significantly from the later majority of users (Rogers, 1995), presumably their personal disposition towards new dApps, and hence their trust formation process will vary. The sample selection process must therefore ensure the inclusion of not only early adopters who are enthusiastic about all dApps since they might have a significantly different disposition to trust, possibly influencing how they form trusting beliefs. Second, the pseudonymous nature of the Ethereum blockchain complicates identifying and inviting specific users to the survey. While it is relatively simple to identify the wallet addresses interacting with dApps on Ethereum, finding wallet users' contact details is nearly impossible. It is not feasible to find their email addresses, the preferred tool for inviting participants to online surveys.

Given these difficulties, we applied a two-fold strategy to invite survey participants. First, we created a list of all users with experience of at least one of the dApps in the Chapter 3 sample. To identify users, we examined the sender addresses of all transactions the dApps in our sample received. From this list, we excluded all smart contract addresses and obvious non-human actors (i.e., trading bots).[52] We then created an ERC20 token and assigned our survey tool URL as its name, and sent this token to all users in our database. Such ERC20 token airdrops served as marketing campaigns for newly introduced tokens in the past.[53] We hoped that if users saw the token in their wallet, they would check out our website and participate in the survey. Although we distributed over one million tokens, none of our survey participants indicated this method was an acquisition channel.[54] The second part of our strategy was identifying dApp users via social media. We used chat rooms and forums that are popular among dApp users to share a link to our survey. We also searched for Ethereum-related groups on LinkedIn and posted an invitation to our survey. In this way, we tried to focus on forums and groups likely to be visited by general users of Ethereum dApps and avoid niche tech-focus forums. We also sent private LinkedIn messages to users who exhibited some connection with Ethereum or dApps in their profile. Approximately 25,000 people could have seen our posts through all LinkedIn groups.[55] The fact that it is impossible to account for double counting (the same person might be in different LinkedIn groups) or know how many people actually saw the invitation, explains the relatively small sample size of 121.

Due to this approach, our resulting sample might suffer from bias. As most participants were acquired from online forums, they might be more engaged in the field and thus have a different disposition to trust than less frequent users and, therefore, a different transaction

[52] To identify bots, we used a set of heuristics. For example, we excluded transaction senders that sent over 200 transactions on one day or reoccurring transactions on a regular basis.

[53] For more on ERC20 airdrops, see https://hackernoon.com/3-methods-to-conduct-an-airdrop-intro-to-batch-transfers-the-unsung-heroes-of-blockchain-c3555b4875fe.

[54] A plausible explanation is that just before our survey launch, MetaMask enabled the default visibility of unsolicited tokens airdropped to a wallet in order to mitigate spamming attacks.

[55] The Ethereum LinkedIn group alone had around 22,000 members at the time.

behavior. Taking part in the survey requires adopting a new dApp. Users who are rather skeptical about adopting new dApps might be reluctant to do our survey and are thus underrepresented in our sample. This is how two skeptical users replied to the invitation:

> “Sorry, I will not participate. I would never click on a random link that a stranger sends me over the internet.”

> “No thanks. I am sure this is a scam.”

To mitigate this censoring problem, we tried to provide all the established trust-building signals and simplify the use of our dApp as much as possible (see next section). Another source of potential bias is that we could not ex-ante select participants based on their characteristics and had to rely on participants’ self-selection. To better understand which users participated in our survey, we included additional questions about the respondent's background, skill level, why they entered the Ethereum platform, and through which channel they became aware of the survey.

4.4.5 Data collection

To achieve a satisfactory number of participants, we followed Dillman‘s (1978) suggestions to increase the survey’s response rate.

To illustrate the importance and relevance of the survey, we added a small section to the invitation explaining how the community of dApp users will benefit from our research and promised to provide all participants with an aggregated summary of our results. For direct messages on LinkedIn, we personalized all invitations and sent reminders after one week. We also tried to enhance trust in our research and survey dApp by answering all questions as soon as possible and providing further information regarding our research on request.

As incentive for participation, we implemented a fully automated and tamper-resistant prize draw in our survey dApp. This rewarded eight winners with 280 MATIC tokens worth around 500 euros at the time of the survey. To give curious users another reason to participate, we promoted the automated prize draw as an exciting new feature of our dApp.

To establish trust in our survey and survey tool, we provided all the trust cues discussed in our theory section. Regarding induction-related trust cues, we provided not only our names and contact details on the survey's website but also crafted an extensive explanation of why we developed the survey tool. Furthermore, we signaled benevolence by explaining that we want to create knowledge for the community of dApp users and developers and tried to demonstrate our ability by creating a user-friendly experience. Regarding third-party signals, we relied on the Technical University of Munich’s reputation, since Fox, Crask, and Kim (1988) have shown that university sponsorship tends to increase the response rate. Regarding deduction-related trust, we verified our smart contracts’ source code on polygonscan.com and relied on the ERC20 implementation suggested by OpenZeppelin, a well-known security audit firm that provides certified templates for developing smart contracts. Further, we assured the pseudonymity of all answers and also refrained from tracking any metadata (e.g., IP addresses) on our survey dApp website.

In order to minimize the cost of responding, we designed the survey to take no longer than 15 minutes and tried to simplify the participation process. In particular, we designed the authentication and response submission process in such a way that it mostly happened in the background and users only had to confirm changes if absolutely necessary (e.g., switching from the Ethereum main net to the Polygon network, withdrawing tokens from the faucet, confirming the final transaction). We provided detailed descriptions for each action a user had to take. We also reimbursed all transaction fees in advance so that users did not incur any costs for submitting their answers as a transaction to our smart contract.

We conducted our survey in two waves. The first wave opened the survey from August to October 2021. In the second wave, the survey was open from November 2021 to May 2022, and most participants were invited and responded early in 2022.

Both survey waves had the same setup. Our invitation provided a link to the survey landing page[56, 57] and a link to the verified smart contracts managing the survey.[58, 59] The first wave’s smart contract received 92 transactions that added answer hashes. The second wave’s smart contracts received 29 transactions, leading to a total of 121 survey responses. Since we could not observe how many people saw the invitation in forums or on LinkedIn, it is impossible to calculate a response rate.

We conducted a second survey wave not only to increase our sample size but also to allow us to rule out the impact of abnormal events that might have biased general perceptions of blockchain technology and of the Ethereum platform in particular (e.g., media reports about blockchain scams or poor security of dApps). To verify ex-post whether this was prob-

[56] Link to the wave one landing page: https://www.blockchain-surveys.com/surveys/610fa0683fd3fb00174e5491.

[57] Link to the wave two landing page: http://www.blockchain-surveys.com/surveys/618f9abd39f1bd63e7ebc6d1.

[58] Link to the wave one verified contract: https://polygonscan.com/address/0x6da6ee8d5d56b578994c4ce111d0ff73746dfbe0.

[59] Link to the wave two verified contract: https://polygonscan.com/address/0x87477d97bf068a0c3cd103043430726120f2118d.

lematic, we performed means-difference tests to compare to what extent first wave respondents differed from those in the second wave regarding their perception of Ethereum as a secure transaction environment. Overall, there were no major differences.[60]

In addition to sample bias, surveys can suffer from other biases, especially the common method bias (Podsakoff, MacKenzie, Lee, & Podsakoff, 2003). As our dependent variable (trusting behavior) was not measured but observed by linking respondents' wallet address to their transaction history on Ethereum, common method bias should not be a problem. Another bias associated with self-report measurements is social desirability (Nederhof, 1985). Since we did not ask about ethical behavior, social desirability should be less of a concern. Moreover, our survey's pseudonymity should further reduce these concerns (Joinson, 1999).

4.5 Research model testing and results

4.5.1 Descriptive statistics

Before presenting our data analysis method and measurement model, we provide an overview of the respondents' demographics and background This overview aims to illustrate the type of users in our sample and help us understand to what population of early users we might generalize our findings.

Most of our survey participants are between 21 and 30 years old (50 percent, Figure 19). Their ages range from 17 to 62, with a mean of 31. This distribution aligns with the observation that the crypto and blockchain space generally, though not exclusively, appeal to younger people.

[60] The p-values for all items related to perceptions of blockchain technology or the blockchain environment ranged from 0.72 to 0.15. Only second wave participants perceived dApp providers' ability as significantly high (mean of 3.39 vs. 3.77) with a p-value of 0.015.

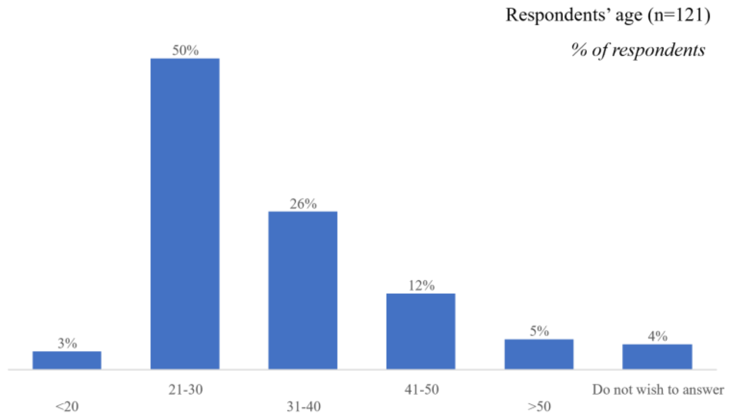


Figure 19: Respondents' age

The survey responses on education (see Figure 20) show that most participants have a background in either engineering (30 percent) or informatics (39 percent) and at least a Bachelor's degree (85 percent). These numbers indicate that, on average, our respondents are well-educated. One possible explanation for this high number is that being active in the blockchain space requires some technical knowledge. This was especially the case in the early days of Ethereum when there was a lack of support services and tutorials to ease dApps interaction and some technical expertise was required even to send transactions (e.g., how to install and use a crypto wallet). Furthermore, engineers and computer scientists were exposed earlier to this new technology, whose diffusion started among cryptography enthusiasts.

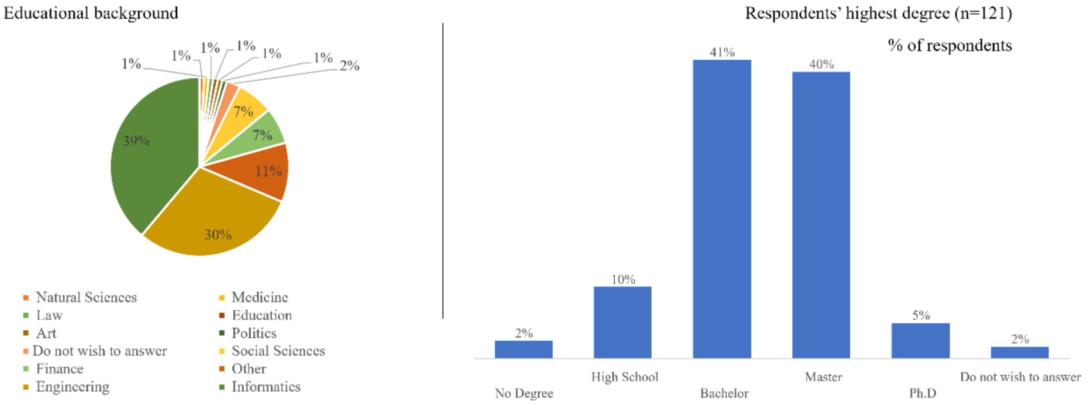


Figure 20: Respondents' educational background

According to Figure 21, most participants joined the platform to earn money by trading other cryptocurrencies (30 percent) or investing in Initial Coin Offerings (ICOs) (27 percent). This confirms the observation that most participants sent their first transaction in 2017 during the ICO hype. However, their responses also show that earning money is not the only reason why people initially joined the platform. A considerable number joined to develop new dApps and create value for others. Few people just joined for gaming and social activities. This indicates that Ethereum attracts users beyond investors and speculators.

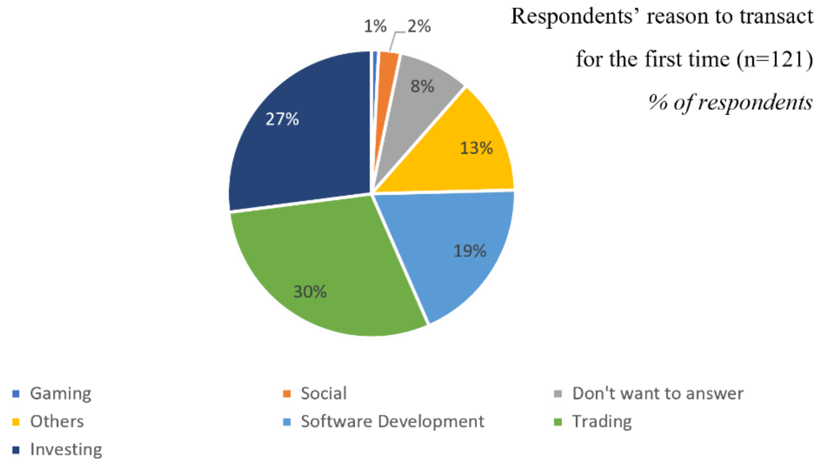


Figure 21: Respondents' reasons for joining Ethereum

Regarding the frequency of interaction, we found that most respondents only occasionally transact on Ethereum. Only 14 percent of our respondents indicate more than ten monthly transactions (Figure 22). Clearly, most users in our sample do not use dApps daily and are tentative about sending transactions. This observation, however, could be distorted by the high gas fees users had to pay during the survey period. These fees could have deterred users from sending transactions on the main net and resort to sidechains to explore new dApps.

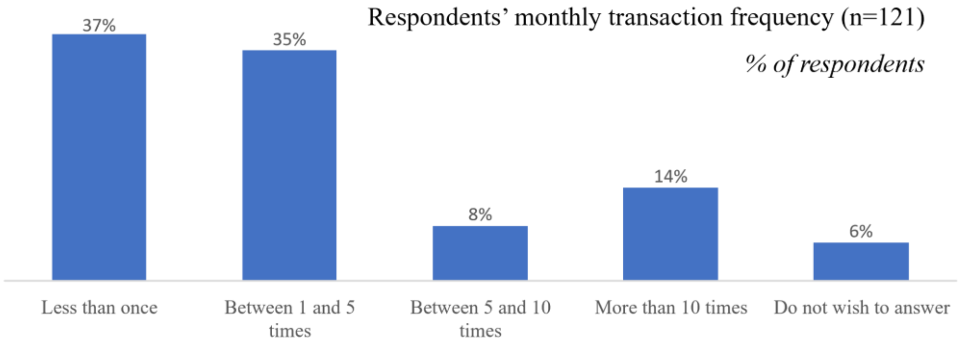


Figure 22: Respondents' monthly transaction frequency (self-reported)

The total number of transactions the participants in our sample have sent ranges between 1 and 781, with a mean of 110.6. While we observed fewer than or equal to 50 transactions for 50 percent of our sample, 42 percent had more than 50 transactions. Accordingly, our sample covers infrequent and frequent users. The respondents transacted with between one and 50 different dApps, with a mean of 11.4. Again, Figure 23 shows that our respondents are well distributed across the entire range, with an average of six to ten different dApps.

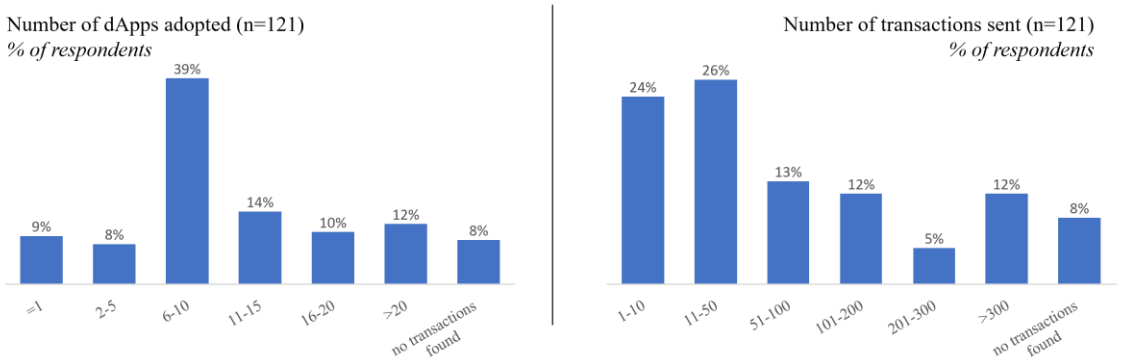


Figure 23: dApp adoption and usage

To understand to what extent our respondents could form deduction-related trust, we asked them about their technological knowledge. As Figure 24 shows, almost all respondents have some basic knowledge of blockchain technology, whereas slightly fewer respondents are experienced with the specifics of Ethereum. Nevertheless, 78 percent of our respondents have at least some experience in Solidity, which is needed to read smart contracts and thus can form first-hand deduction-related trust.

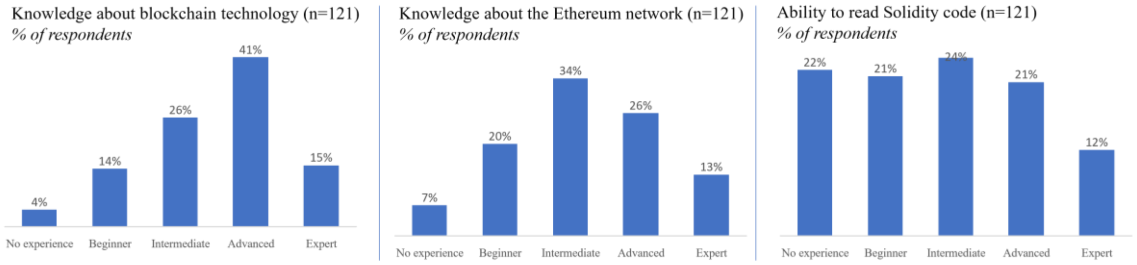


Figure 24: Blockchain, Ethereum, and Solidity knowledge

As the possibility of deductive certainty and forming deduction-related trust arguably changes how trust in dApps is formed, we present descriptive evidence that respondents indeed care about the availability of a verified source code. Figure 25 shows that 74 percent of our respondents care about a verified source code.

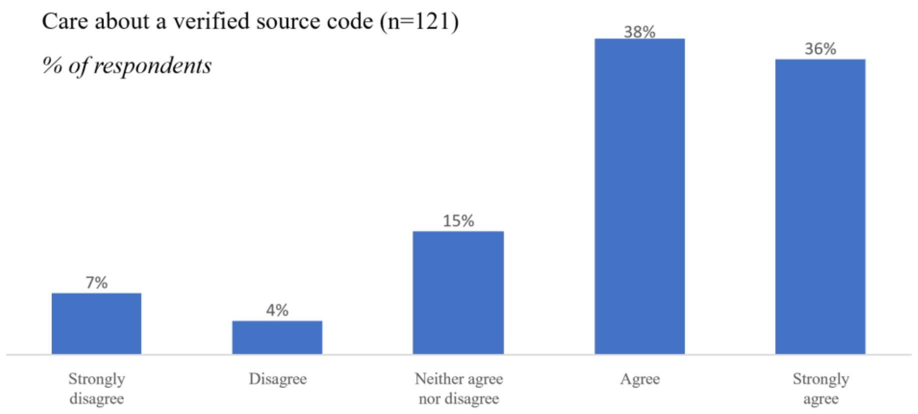


Figure 25: Caring about a verified source code

We also asked participants about their reliance on various trust-building sources. These sources had to be linked to one of the four trusting beliefs. Figure 26 shows to what extent our respondents rely on the four trust-building sources: first-hand inductive information from reading the dApp provider's website (upper left panel), second-hand information about the dApp provider (upper right panel), first-hand inductive information from reading the dApp source code (lower left panel), or others who have read the source code. Two observations are noteworthy.

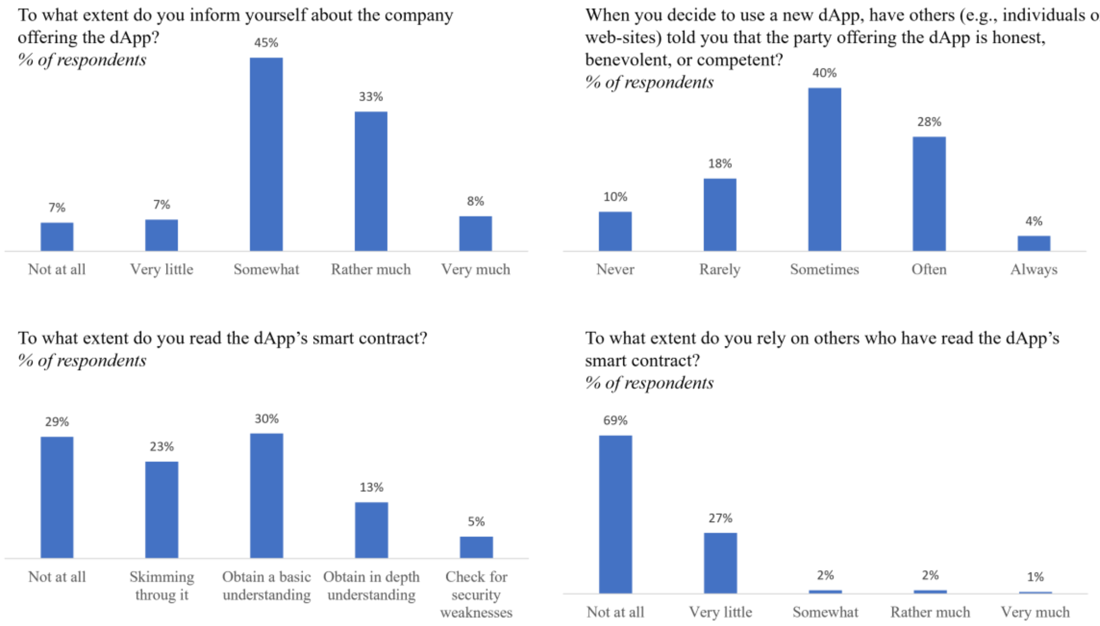


Figure 26: Reliance on Four sources of trusting beliefs

First, 72 percent of our participants (see left lower panel in Figure 26) state that they at least skim through the smart contract before deciding to adopt a new dApp. We see this as a first indicator that our respondents indeed rely on first-hand deduction-related trust formation. On the other hand, the fact that only 5 percent of the respondents typically check for security weaknesses before sending a transaction to a new dApp suggests that not many users achieve deductive certainty, and the majority still need to rely on some form of trust. However, it also shows that some users study the source code carefully, and thus serve as source for others' second-hand deduction-related trusting beliefs. Second, as the lower right panel in Figure 26 shows, few users rely on second-hand information about the smart contract's validity. One possible explanation is that security audits and certificates are in an early stage of development, and hence almost no dApp is certified.[61]

The more important a certain type of trusting belief is for a user, the more we would expect this user to invest in gathering related information. To check this point, we correlated the respondents' engagement in the four information gathering processes with the importance of the various trusting beliefs. Table 4 confirms our expectations, showing that each type of information gathering process is most strongly correlated with the associated type of trusting belief. This finding suggests our measurements are valid, and indicates that the four trusting beliefs come from four distinct information sources.

[61] Only 11 dApps in our sample refer to security audits on their website.

Table 4: Correlation between seeking trust cues and trusting beliefs

	FH-IND-TB	SH-IND-TB	FH-DED-TB	SH-DED-TB
To what extent do you inform yourself about the company offering the dApp (e.g., by consulting its website)?	0.35	0.05	0.25	-0.09
When you decide to use a new dApp, have others (e.g., individuals or websites) told you that the party offing the dApp is honest, benevolent, or competent?	0.08	0.46	0.18	0.12
To what extent do you read the dApp source code?	0.15	0.03	0.77	-0.19
To what extent do you rely on others who have read the dApp source code?	0.15	0.15	0.18	0.37

In addition to the above descriptive results, Appendix C-4 provides summary statistics and correlations for the variables in our model.

4.5.2 Data analysis method

We used the partial least squares (PLS) technique developed by Wold (1985) to analyze the measurement model and establish the reliability and validity of our constructs. PLS is a powerful modeling technique that simultaneously estimates measurement and structural components and is commonly used to investigate causal paths in structural equation models (SEM) (Fornell & Bookstein, 1982). It is also frequently used in information systems research (Marcoulides & Saunders, 2006). PLS is an alternative to linear structural relations (LISREL) models, which other trust formation researchers have used in similar settings (e.g., Gefen et al., 2003; McKnight et al., 2002a, 2002b). We chose PLS over LISREL for two reasons. First, the PLS approach enabled us to deal with reflective and formative constructs in the same model, thus proving useful for explorative analyses of structural equation models, and provided significant support for theory development (Götz, Liehr-Gobbers, & Krafft, 2010). Second, although PLS does not provide a silver bullet that works with every sample size (Marcoulides & Saunders, 2006), it can produce valid outcomes with a sample size below N=150 (Chin & Newman, 2000). To implement our trust formation model, we used SmartPLS 3 software (Ringle, Wende, & Becker, 2015) and followed best practice suggestions by Hair, Hult, Ringle, Sarstedt, Richter, and Hauff (2017). For our analysis, we applied Gerbing and Anderson's (1988) two-stage approach, where “the measurement model first is developed and evaluated separately from the full structural equation model” (p. 191). We now discuss the validity of our measurement model and explain why we excluded several items. Then we evaluate our full structural model and present the results regarding our hypotheses.

4.5.3 Measurement model

Our measurement model consists of reflective and formative constructs and both types require a different set of tests for checking their validity (Hair et al., 2017).

Evaluation of reflective constructs. To evaluate the reflective constructs' reliability and validity, we assessed their composite, convergent, and discriminant validity (Esposito Vinzi, Chin, Henseler, & Wang, 2010).

Table 5 shows all the items associated with our reflective constructs, their factor loadings, and composite reliability. Based on our assessment of the factor loadings and composite reliability, we left out the time TSP2 as it had a loading below 0.6, and its exclusion resulted in a higher composite reliability (0.94 instead of 0.82) and a higher Cronbach’s alpha (0.87 instead of 0.66). Similarly, we dropped item TST2 as it indicated a loading below 0.6 (TST2 = 0.28).

Table 5: Measuring internal consistency (reflective constructs)

Construct	Item	Factor loading	Composite reliability
Faith in dApp providers (FID)			0.85
	FID1	0.82	
	FID2	0.84	
	FID3	0.77	
Trusting stance towards people (TSP)			0.94
	TSP1	0.95	
	TSP2	0.93	
Trusting stance towards technology (TST)			0.73
	TST1	0.62	
	TST2	0.89	
Technological knowledge (TEK)			0.91
	TEK1	0.83	
	TEK2	0.89	
	TEK3	0.93	
Perceived blockchain risk (PBR)			0.85
	PBR1	0.87	
	PBR2	0.84	
Structural assurance blockchain technology (SABC)			0.80
	SABC1	0.76	
	SABC2	0.84	
	SABC3	0.65	
Trusting behavior (TB)			0.90
	NOT	0.90	
	NOD	0.92	

To evaluate convergent validity, which measures how well individual items reflect the same theoretical constructs, we applied Fornell and Larcker’s method (1981) to assess the average variance extracted (AVE). Table 6 shows that the AVE of all constructs exceeds the 0.5 threshold for a sufficiently high convergent validity.

Table 6: Average extracted variance (reflective constructs)

Construct	AVE
Faith in dApp providers (FID)	0.65
Trusting stance towards people (TSP)	0.89
Trusting stance towards technology (TST)	0.58
Technological knowledge (TEK)	0.78
Structural assurance blockchain technology (SABC)	0.57
Perceived blockchain risk (PBR)	0.74
Trusting behavior (TB)	0.83

To check for discriminant validity, which measures whether two constructs are empirically distinct, we applied the Heterotrait-Monotrait ratio (HTMT) (Henseler, Ringle, & Sarstedt, 2015). Henseler et al. (2015) suggest the HTMT ratio rather than the Fornell-Lacker criterion or examining cross-loadings as they do not detect a lack of discriminant validity reliably. Figure 7 shows that all our constructs are well below the conservative threshold of 0.85, and none of the 95 percent confidence intervals contains the value 1. This indicates that all our reflective constructs are independent and suggests that discriminant validity is achieved.

Table 7: HTMT ratio (reflective constructs)

Construct	1	2	3	4	5	6
1. Faith in dApp providers (FID)						
2. Trusting stance towards people (TSP)	0.31					
3. Trusting stance towards technology (TST)	0.46	0.41				
4. Technological knowledge (TEK)	0.24	0.15	0.14			
5. Structural assurance blockchain technology (SABC)	0.27	0.26	0.22	0.18		
6. Perceived blockchain risk (PBR)	0.27	0.08	0.07	0.45	0.19	
7. Trusting behavior (TB)	0.10	0.05	0.25	0.15	0.17	0.25

Overall, our composite reliability, convergent validity, and discriminant validity tests indicate that our reflective constructs are sufficiently reliable and valid.

Evaluation of formative constructs. There is an ongoing debate about the usefulness of formative constructs. Some scholars suggest that whenever possible, reflective rather than formative constructs are preferable as their indicators' weights depend on the outcome variable used to estimate them and can thus change substantially from study to study (Bagozzi, 2007; Wilcox, Howell, & Breivik, 2008). Moreover, they argue against using formative constructs because dropping indicators from formative constructs alters the conceptual meaning and can lead to fatal flaws in theory testing (Howell, Breivik, & Wilcox, 2007). Other scholars endorse formative measurements to account for the possibility of causal indicators (Diamantopoulos et al., 2008; Jarvis, MacKenzie, & Podsakoff, 2003; Podsakoff et al., 2003).

Taking this debate into account, we considered the advantages and disadvantages of formative constructs and only used them where indicators jointly determined the conceptual and empirical meaning of the construct but were not causally related to each other (Jarvis et al., 2003).

As discussed in Section 4.4.3, we operationalized the four trusting belief constructs as formative constructs. Specifically, we measured *first-hand induction-related trusting beliefs* (FH-IND-TB) with the three dimensions of belief in the other party's integrity, benevolence, and ability. These dimensions are distinct positive beliefs as they are causally independent, but as shown by prior literature, all lead to a more favorable assessment of the other party's overall trustworthiness (McKnight et al., 2002a). Based on a similar logic, we measured *second-hand induction-based trusting beliefs* (SH-IND-TB) as a formative construct comprising the same dimensions. We also measured *first-hand* (FH-DED-TB) and *second-hand deduction-related trusting beliefs* (SH-DED-TB) as formative constructs. Again, our items for these constructs represent distinct dimensions that can independently lead to these trusting beliefs. Regarding FH-DED-TB, users can either form trusting beliefs because they could obtain deductive certainty if they wanted to, or because they have read parts of the source code. Although both options lead to deduction-related trusting beliefs, these can be independent of each other. There are various sources of SH-DED-TB. Users can rely either on third parties' certificates or what they have heard from their peers about the soundness of the source code. Both are independent sources of second-hand deduction-related trusting beliefs and thus should have a formative relationship with this construct.

To ensure the validity of our reflective constructs, we took three steps suggested by Hair et al. (2017). First, to guide the development of our constructs, we consulted the trust formation literature (see Sections 4.2 and 4.3 for the theoretical foundations). Second, to ensure the dimensions to measure our constructs are mutually exclusive, we assessed their collinearity using the variance inflation factor (VIF) and followed recommendations by Diamantopoulos and Winklhofer (2001) to eliminate indicators that exceed the cutoff. Since the item battery FH-DED1-4 exhibited a VIF above the threshold value of 5, we left out FH-DED2 and FH-DED4. We kept FH-DED1 ("*I feel confident to interact with the dApp because I could read the source code if I wanted to*") and FH-DED3 ("*I feel confident to interact with the dApp because I read the source code (or parts of it) and understand that it does what it is supposed to do*") as both capture two unique dimensions of first-hand deduction-based trusting beliefs. Whereas FH-DED1 does not require reading of the source code and thus leads to trusting beliefs based only on the possibility of deductive certainty, FH-DED3 requires reading parts of the source code and taking deductive steps. Including only these two items reduces the VIF below the threshold of 5 (4.73 for both; see Table 8).

Table 8 indicates the variance inflation factor (VIF) for all items. Since no item exceeds a VIF of 5, every item presumably contributes unique explanatory power to the construct.

Table 8: VIF, loadings, and weights of formative constructs

Construct	VIF	Loadings	Weights (p-value)
First-hand induction-related trusting beliefs (FH-IND-TB)			
FH-INT	2.58	0.90	0.35 (0.10)
FH-ABI	2.58	0.98	0.70 (0.00)
First-hand deduction-related trusting beliefs (FH-DED-TB)			
FH-DED1	4.73	0.98	0.55 (0.09)
FH-DED3	4.73	0.97	0.47 (0.08)
Second-hand deduction-related trusting beliefs (SH-DED-TB)			
SH-DED1	1.07	0.81	0.66 (0.10)
SH-DED2	1.07	0.77	0.60 (0.10)

Note: To obtain p-values, we applied a bias-corrected, accelerated bootstrapping procedure with 5000 subsamples

Finally, to assess our constructs' external (nomological) validity, we evaluated the items' relative and absolute relevance to ensure our theoretical reasoning aligned with our measured constructs (Götz et al., 2010). Table 8 depicts factor loadings and the factor weights' p-value. Based on this analysis, we excluded SH-ABI (loading=0.42; p=0.59) and SH-BEN (loading=0.31; p=0.38) since neither indicated a factor loading over 0.5 (absolute importance) nor a significant p-value for the factor weight (relative importance) (Hair et al., 2017).

After these adjustments to our reflective and formative constructs, our measurement model fulfilled all the common reliability and validity requirements discussed by Hair et al. (2017).

4.5.4 Structural model

The next step was to evaluate the resulting structural model for testing our hypotheses (Gerbing & Anderson, 1988). For this we used four measurements: (1) the explained variance of our constructs; (2) our constructs' inner VIF; (3) the significance of the path coefficients; and (4) Q^2 and q^2 effect sizes (Hair et al., 2017).

First, in line with Fornell and Larcker (1981), we used the R^2 values to evaluate our proposed model's predictive power. As shown in Table 9, our model explains 20 percent of the variance in trusting behavior, 40 percent of first-hand induction-related beliefs, and 31 percent of first-hand deduction-related beliefs. Regarding second-hand trusting beliefs and the perception of structural assurance, our model is less predictive. With this exception, the observed R^2 values are within the range of previous trust formation studies (e.g., McKnight et al., 2002b).

Table 9: Coefficients of determination

Construct	R Square
Structural assurance blockchain technology (SABC)	0.03
First-hand induction-related trusting beliefs (FH-IND-TB)	0.40
Second-hand induction-related trusting beliefs (SH-IND-TB)	0.06
First-hand deduction-related trusting beliefs (FH-DED-TB)	0.31
Second-hand deduction-related trusting beliefs (SH-DED-TB)	0.09
Trusting behavior (TB)	0.20

Another way to assess predictive power is to evaluate the Q^2 and q^2 values. We obtained these values through a blindfolding procedure (Hair et al., 2017). Table 10 shows that Q^2 values range between 0.002 for perceived blockchain risk and 0.32 for first-hand induction-related trusting beliefs and thus align with our R^2 analysis. It confirms a moderate predictive power for first-hand induction-related and first-hand deduction-related trusting beliefs and only a low predictive power for both second-hand trusting beliefs. The q^2 effect sizes illustrate that all constructs have a relatively small but relevant effect size (Hair et al., 2017).

Table 10: Effects on the endogenous variable

Constructs	Q^2	q^2
First-hand induction-related trusting beliefs	0.32	0.03
Second-hand induction-related trusting beliefs	0.01	0.03
First-hand deduction-related trusting beliefs	0.28	0.04
Second-hand deduction-related trusting beliefs	0.01	0.01
Perceived blockchain risk	0.002	0.02

To evaluate if our constructs add unique explanatory power or should be aggregated into higher-order constructs, we assessed the inner VIF. No construct exceeded the threshold 5 (the highest VIF of 2.05 was for FH-DED-TB), suggesting that all constructs contribute unique explanatory power and do not need to be summarized (Hair et al., 2017).

Table 11 shows the standardized path coefficients we used to test our hypotheses. As expected, we found support for the positive relationship between first-hand induction-related and first-hand deduction-related trusting beliefs and trusting behavior (H1a and H1c). Regarding H1b and H1d, contrary to our expectation of a positive relationship between second-hand induction and deduction-related trusting beliefs with trusting behavior, we observed a significant but negative relationship. Our findings for institutional factors (H2a, H2b, and H2c) aligned with our hypotheses. Our hypotheses regarding dispositional factors (H3a-H3h) were only partially supported. We did find confirmation of the positive association between faith in the dApp provider and first-hand induction-related trusting beliefs (H3a), the relationship between trusting stance towards people and second-hand induction-related trusting beliefs (H3d), knowledge about blockchain technology associated with first-hand deduction-related trusting beliefs (H3g), and the perceived structural assurance provided by blockchain technology (H3h).

Table 11: Testing hypotheses

Hypotheses	Predicted direction	Standardized path coefficient
Trusting beliefs		
H1a. First-hand induction-related trusting beliefs– trusting behavior	+	0.21**
H1b. Second-hand induction-related trusting beliefs – trusting behavior	+	-0.24**
H1c. First-hand deduction-related trusting beliefs – trusting behavior	+	-0.16*
H1d. Second-hand deduction-related trusting beliefs – trusting behavior	+	0.24***
Institutional factors		
H2a. Structural assurance - first-hand deduction-related trusting beliefs	+	0.24***
H2b. Structural assurance - second-hand deduction-related trusting beliefs	+	0.25*
H2c. Perceived blockchain risk - trusting behavior	-	-0.18**
Dispositional factors		
H3a/b. Faith in dApp provider – first-hand/second-hand induction-related trusting beliefs	+/+	0.20*/0.14
H3c/d. Trusting stance towards people – first-hand/second-hand induction-related trusting beliefs	+/+	0.05/0.19*
H3e/f. Trusting stance towards technology – first-hand/second-hand deduction-related trusting beliefs	+/+	0.09/0.14
H3g/h. Knowledge about blockchain technology – first-hand deduction-related trusting beliefs / structural assurance	+/+	0.42***/0.16**

Note: Two-tailed test: ***p< 0.01 **p <0.05 *p<0.1; To obtain p-values, we applied a bias-corrected, accelerated bootstrapping procedure with 5000 subsamples

4.6 Discussion

In this section, we interpret the findings about our final trust formation model (Figure 27), discuss the limitations of our research, and provide practical and theoretical implications.

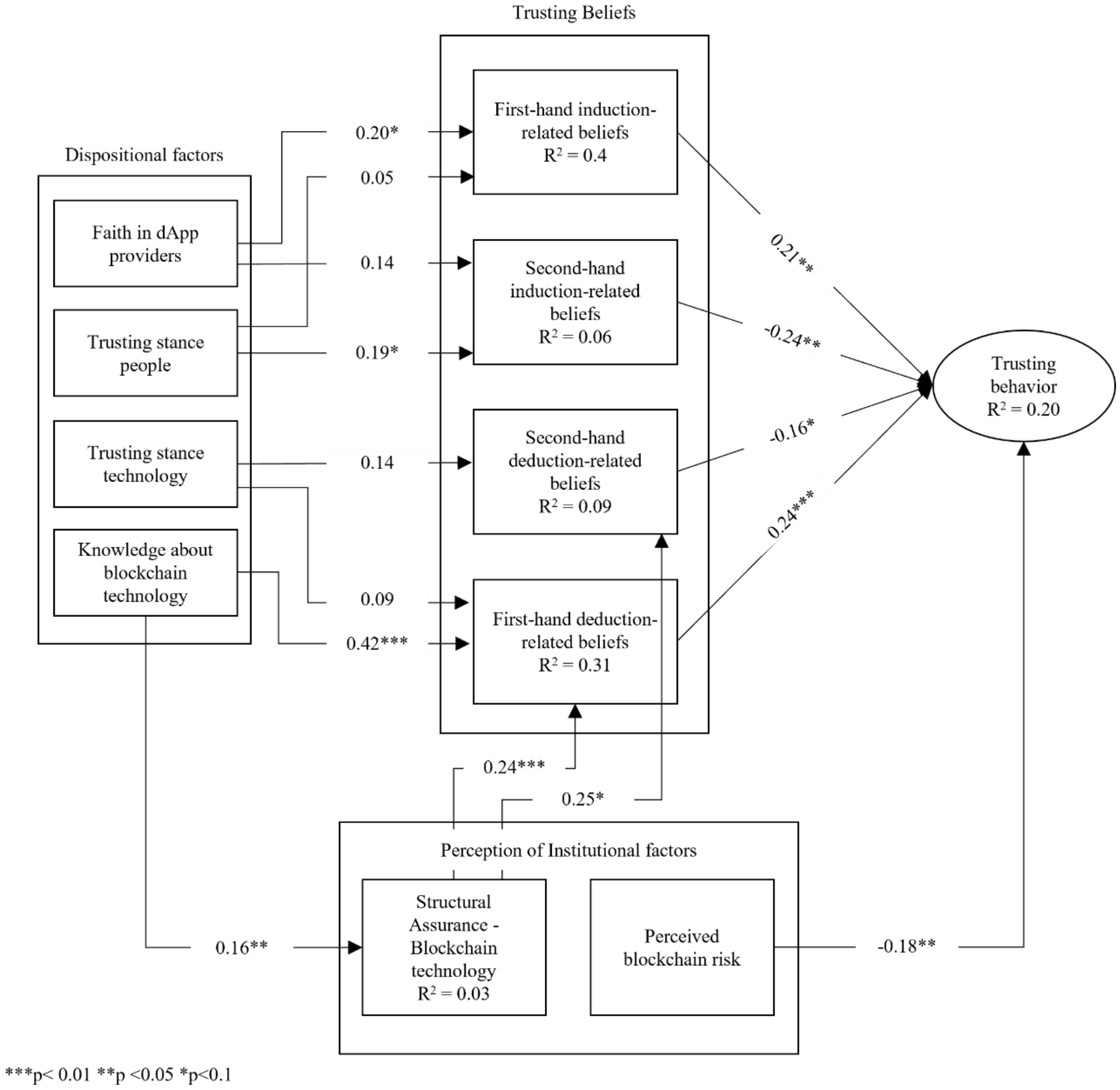

Trusting Beliefs
Dispositional factors
Faith in dApp providers
Trusting stance people
Trusting stance technology
Knowledge about blockchain technology
0.20*
0.05
0.14
0.19*
0.14
0.09
0.42***
First-hand induction-related beliefs R² = 0.4
Second-hand induction-related beliefs R² = 0.06
Second-hand deduction-related beliefs R² = 0.09
First-hand deduction-related beliefs R² = 0.31
0.21**
-0.24**
-0.16*
0.24***
Trusting behavior R² = 0.20
0.24***
0.25*
Perception of Institutional factors
Structural Assurance - Blockchain technology R² = 0.03
Perceived blockchain risk
0.16**
-0.18**
***p< 0.01 **p <0.05 *p<0.1

Figure 27: Trust formation model

4.6.1 Interpretation of our findings

Our empirical analysis largely supports our proposed trust formation model. Nevertheless, some results are contrary to our predictions and thus need further elaboration. On an abstract level, as we found that trusting beliefs are strongly related to the decision to transact with a new dApp, our model confirms that the pertinent trust formation literature remains valid in the context of dApps. This finding contradicts conventional wisdom that the blockchain is a “trust-free” system (Economist, 2015; 2017; Greiner & Wang, 2015) and concurs with other scholars who argue that as long as humans are involved, trust will always be relevant (Hawlitschek et al., 2018). However, our model also shows that building trust in dApps works differently than with traditional online applications. Our model enabled us to distinguish four distinct types of trusting beliefs. These differ regarding the source of information and basis of the cognitive process that formed them. The strongest predictors of trusting behavior are first-hand deduction-related trusting beliefs (β=0.24). This finding is highly interesting as it concerns a new type of trusting belief based on the possibility of attaining deductive certainty about the outcome of a transaction by reading and understanding a smart contract's source code instead of the perception of the other party's trustworthiness. It thus shows that trust formation in the context of dApps is less human-centric and more focused on the provable properties of the smart contract. Also supporting this interpretation is our observation that 71 percent of our survey respondents care about a verified source code, and also indicate that they at least skim through the smart contract. However, the fact that first-hand induction-related trusting beliefs are also positively associated with trusting behavior (β=0.21) provides evidence that trust in the party offering a dApp is still an important antecedent to trusting behavior. Contrary to the expected positive influence of second-hand trusting beliefs on trusting behavior, we observed a significant but negative relationship. There could be multiple reasons for these findings and all should be addressed in further research. One reason could be our model's relatively low predictive power for both constructs, which might therefore have failed to capture their real meaning. One solution might be to investigate additional antecedents of both these trusting beliefs and develop additional items that help to capture their variance. This finding could also be due to the current scarcity and ambiguity of second-hand trust cues; it would also explain why according to Figure 26 in Section 4.5.1, very few respondents rely on second-hand smart contract information and most even indicate not relying on it at all. Future research could revisit trust formation once more standard and credible third-party certificates are available to see whether the amount of people relying on second-hand information has changed and the path coefficients of second-hand trusting beliefs still hold. Another possible explanation is the observation presented in Figure 28. Respondents with below-median (<37) total transactions and a below median (<9) number of adopted dApps seem to rely more strongly on second-hand trusting beliefs than respondents with an above-median transaction and adopted dApp count. The resulting negative correlation is an alternative explanation for our observation that we cannot rule out or account for due to our limited sample. Interestingly, Figure 28 also shows that for above median respondents, first-hand deduction-related beliefs are the greatest source of confidence in the reliability of a dApp, whereas for below median respondents, second-hand induction-related trusting beliefs are the greatest source of confidence. An explanation for this pattern could be the different groups within our sample: plausibly, (a) tech-savvy crypto enthusiasts who can read the smart contract and verify its reliability; and (b) newcomers who depend on others for insights about the soundness of a dApp. The difference in these groups' adoption speed is yet another explanation for our findings. Again, our sample size did not allow us to perform within and between group analysis and we will therefore have to leave this investigation for future research. A further potential reason for these contradictory findings is that we had to drop SH-ABI and SH-BEN from our second-hand induction related construct. Doing so may have resulted in a different construct. Again, further research is required to understand the validity of our scale and come up with explanations for our findings.

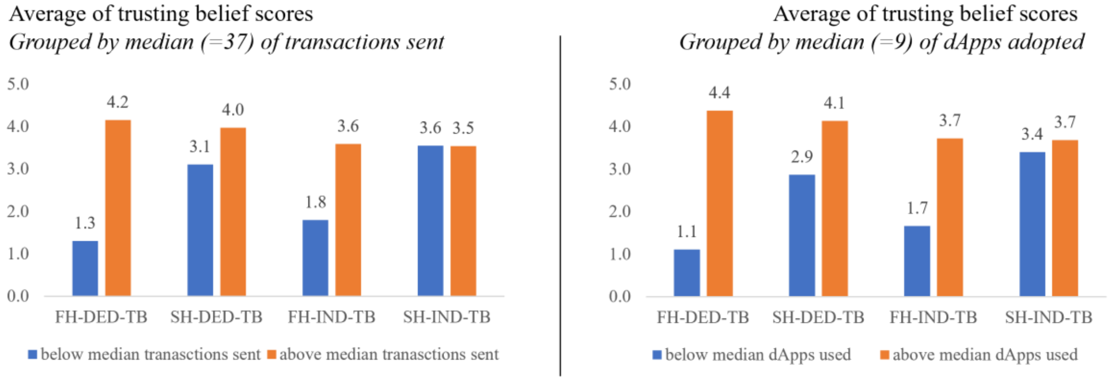


Figure 28: Average trusting belief scores (sample divided into median of transactions and adopted dApps)

Similar to research on how institutional surroundings impact the formation of trusting beliefs (e.g., Gefen et al., 2003; McKnight et al., 2002b; Pavlou & Gefen, 2004), we also found that the transaction environment matters for forming trusting beliefs. The feeling that Ethereum is a safe transaction environment thanks to the automated transaction execution, a decentralized consensus mechanism, transparency and immutability of records, fosters forming positive first-hand (β=0.16) and second-hand (β=0.24) deduction related to trusting beliefs. On the other hand, we also observed that the perceived risk associated with transacting on Ethereum negatively influences trusting behavior.

Regarding dispositional factors, we also found some confirmation that they matter for forming trust in new dApps. The general faith in dApp providers relates significantly to first-hand induction-related trusting beliefs (β=0.20). The trusting stance towards people, another dispositional factor claimed to be influential in forming trusting beliefs (e.g., McKnight et al., 1998), has a significant impact on second-hand but not first-hand induction-related trusting beliefs. We also found that knowledge about blockchain technology has a positive and significant direct impact on forming first-hand deduction-related trusting beliefs and a positive and significant indirect association with first-hand deduction-related trusting beliefs through a more positive perception of the Ethereum platform's structural assurances. Our findings show that while some dispositional factors matter, others are not relevant. One explanation is that dispositional factors become less important as familiarity with an interaction partner increases (Gefen et al., 2003; McKnight et al., 1998). Because Ethereum has been around since 2016, more and more users might be familiar with the transaction environment and dApps in general and are thus influenced by their first-hand experience rather than their general disposition.

4.6.2 Limitations and further research

Given that this study is a first attempt to measure trust formation in a novel and fast-developing field, it has noteworthy limitations. Some of these limitations offer a direct opportunity for further research to help overcome them. First, our study sample was not random. We invited prospective participants based on their affiliation to groups, their indicated interest on LinkedIn, or their participation in forums and chat rooms. Although we tried to focus on groups and forums that attract general users, we could not rule out that our respondents are users with a disproportionately high interest in and curiosity about Ethereum. A high level of curiosity about new dApps and better technical understanding could influence respondents' trust formation process or bias results. As the number of dApps being adopted is still rising, it is questionable whether our respondents represent average users in a later more stable situation. This means our study results are more generalizable to technology-savvy early adopters than to a later majority. To resolve this, future research could replicate our study once Ethereum has reached the mass market and investigate whether our initial findings—particularly the importance of first-hand deduction-related trusting beliefs—are still valid.

Another limitation is our relatively small sample size. Methodologically, a larger sample is beneficial as it would improve the precision of our PLS estimates (Marcoulides & Saunders, 2006). Moreover, a larger sample would allow covariance-based approaches such as LISREL (Chin & Newman, 2000), which other trust formation researchers have used to estimate their models (e.g., Gefen et al., 2003; McKnight et al., 2002a, 2002b). Regarding the generalizability of our results, our small sample raises another issue. Since respondents had to adopt a new dApp (our survey dApp) to take part in our survey, it is unclear whether this deterred skeptical users and only led curious and open-minded respondents to self-select into our study. To mitigate this censoring concern, we tried to make our dApp as user-friendly as we could and provide all common trust cues. Future research could use different survey techniques to scrutinize the generalizability of results. In addition, the small sample size limited the possible distinction between groups. Further analysis based on a larger sample could investigate why trust formation varies between users with different backgrounds and experiences, or the difference between users who entered Ethereum for various reasons (e.g., gaming vs. investing).

Although our observation of real behavior is a clear advantage over other trust formation studies that had to measure intentions, not actual behavior (e.g., Gefen et al., 2003; McKnight et al., 2002a, 2002b; Pavlou & Gefen, 2004), how we measured trusting behavior has limitations. We asked participants about forming trust in a typically new dApp they would consider using. We then related this knowledge about their typical trust formation process to their past transaction behavior. Obviously, not only their past transaction experience but also their idea of a typical new dApp might differ. Consequently, we also had to operationalize the trusting belief constructs so that they measured participants' beliefs when deciding to interact with a new dApp. This operationalization, however, did not allow us to assess the situation where users had not formed enough trust to transact with a new dApp. Thus, our study can explain the trusting beliefs that users typically rely on when adopting a new dApp but cannot explain how a lack of trusting beliefs influences the decision not to interact with a dApp. Future research could address both shortcomings with an experimental setup similar to McKnight et al. (2002b), where all respondents are confronted with the same new dApp.

As our second-hand induction-related trusting belief constructs had a low loading of ability and benevolence dimensions, we dropped them, despite being well-established dimensions of trusting beliefs (Mayer et al., 1995; McKnight et al., 1998). It would be interesting to investigate if this observation is due to our study setup, the adaption of existing items, or because regarding trust formation, the significance of these established trusting belief dimensions changed. We also dropped two items due to a high VIF related to why users read the source code (FH-DED2, FH-DED4) and three items related to participants' trusting stance towards people and technology. Since we developed these items, future research could replicate our study and modify our initial scales. Lastly, because our collected data was cross-sectional, we could not prove causality in our constructs.

Future research could extend our study's findings. For instance, it would be interesting to study how our model might change if we move beyond the formation of initial trust. We speculated that deduction-related trusting beliefs, institutional and dispositional factors would become less important as a relationship with the dApp provider develops and users gather first-hand experience with the dApp provider. Another interesting extension would be to consider previously studied adoption constructs, such as perceived usefulness and ease of use (Gefen & Straub, 2004). Scholars could then explore the relative importance and interaction with trust to gain a more holistic understanding of the adoption process.

4.6.3 Implications for practice and research

Our study has interesting implications for practice. The most important is recognizing that trust is still relevant in a supposedly "trust-free" system. This means that practitioners need to engage in trust-building strategies to foster the adoption of their dApps. Similar to prior studies on online trust formation, we confirm the multidimensionality of trust (Rousseau et al., 1998). This finding not only offers dApp providers different manageable levers to enhance their users' trust, but warrants that different trust-building strategies might be required depending on the targeted users. For instance, less technology-savvy users might not be able to read a dApp's source code to form deduction-related trust and thus prefer to rely on induction-related or second-hand deduction-related trusting beliefs. Consequently, dApp providers should not only advertise their verified source code but also talk about signals and engage in activities that allow a prospective user to gauge their integrity, benevolence, and thus evaluate their trustworthiness. They could focus on mechanisms such as endorsements by respected parties, a high-quality website, a customer hotline, or elaborating on their mission and vision statements. These new and manageable levers can also be particularly beneficial for dApp providers whose trust-building strategies are inhibited—for instance, in settings where forming trust is difficult because the dApp provider suffers from discrimination (e.g., due to race or origin) or if the legal system does not warrant enough institution-based trust for users to feel safe transacting with the dApp.

By showing that institutional safeguards play an important role in counteracting perceptions about the risk associated with a blockchain transaction, we suggest that dApp providers join forces and invest in educating their users about the safeguards provided by the blockchain infrastructure. Such efforts should be particularly promising as our analysis shows that more technical knowledge about blockchain technology is generally associated with more positive perceptions of the blockchain's institutional safeguards and that these perceptions, in turn, favorably foster the formation of trusting beliefs in a specific vendor. dApp providers must recognize that the current user base is more technology savvy than the general population. Therefore, they need to be aware of a potential shift in the relative importance of the various dimensions of trust and find better ways to implement cues that allow users to form second-hand trusting beliefs. It is important to emphasize second-hand trusting beliefs since our analysis shows that they currently do not work as intended.

Our study has three main implications for research. First, we validated that a diverse set of previously established, literature-based trust measurements are still relevant for dApps on a blockchain. Researchers can thus build on the same foundations when investigating this new space. Second, inspired by smart contracts' potential to allow users to obtain absolute certainty about the outcome of a transaction even before it takes place, we introduce a new way of forming trust and the concept of deduction-related trust. Deduction-related trust contrasts with all established induction-related trust cues as it focuses on a transaction's provable (i.e., with deductive logic derivable) elements. Researchers can use this novel differentiation to revisit established transaction governance mechanisms and scrutinize to what extent a governance mechanism allowing for deduction might influence trust formation. Arguably, even legal contracts allow some degree of deduction depending on how they are written. Investigating the impact on forming trust could provide a different perspective of established governance mechanisms and enhance our overall understanding of their effectiveness. Third, our study showcases using our own survey dApp to investigate how trust is formed in dApps running on top of a blockchain. For trust formation scholars, this is not only interesting as it opens up a new field where trust formation works differently compared to ordinary web apps but also as this novel approach allows us to pseudonymously link past transaction behavior with survey responses. Thus, it presents an opportunity to address the limitations of many previous studies on the formation of trust that cannot observe actual trusting behavior and only measure trust intentions (e.g., Gefen et al., 2003; McKnight et al., 2002a, 2002b). In addition, linking survey responses to actual transaction behavior might also open up fruitful avenues for related research areas, such as technology adoption, human-computer interaction research, or marketing research.

4.7 Conclusion

Blockchain platforms are hailed for disrupting our business world by removing the need for trust in transactions (Economist, 2017) and creating “trust-free” systems (Greiner & Wang, 2015). Whereas these claims are in stark contrast to previous research that deemed trust as the key to e-commerce (Keen, Balance, Chan, & Schrump, 2000), our study aims to investigate the role of trust formation in a supposedly “trust-free” environment. To delineate the role of trust in the context of decentralized applications on a blockchain, we developed a new model of trust formation that accounts for the possibility of obtaining certainty about the outcome of a transaction even before it takes place by processing all the transaction rules predefined in a smart contract. We argue that obtaining this deductive certainty—which makes trust dispensable—is not very likely due to the high costs involved. In our view, dApps offer users a new way to form trust based on the possibility of obtaining deductive certainty but not on its actual obtainment. We show that four distinct trusting beliefs are associated with trusting behavior (the adoption and use of a new dApp). We see this study as initial evidence that trust still matters when deciding to enter into an exchange relationship with a dApp, but also as evidence that the way this trust is established has changed.

5 Competition in a Market for Transactions: The Effect of Ethereum's Gas Price Mechanism on dApp Heterogeneity

5.1 Introduction

Blockchain technology disintermediates digital platforms by substituting a centralized authority with a market mechanism that ensures automated enforcement of transactions following pre-defined rules (Nakamoto, 2008). According to blockchain technology proponents, this disintermediation limits a platform provider's ability to modify the platform rules or exclude complementors unilaterally and allows architects to design platforms where the created value is distributed more evenly among all participating parties (Catalini & Tucker, 2018; Vergne, 2020). Gavin Wood, one of Ethereum's founding fathers, envisioned that blockchain technology would enable what he calls Web3.0, a new form of the World Wide Web that is fairer, more democratic, and free from powerful platform intermediaries that exploit their users' data (Wood, 2014a). With this vision, he spurred a whole new industry aiming to disrupt digital platforms across industries such as finance, gaming, insurance, and health.

However tempting this vision might seem, we need to bear in mind that disintermediation is no panacea without limitations. For example, blockchain platforms are known to incur greater coordination costs because protocol changes require community consensus and higher storage costs as the same data is replicated across different nodes (Pereira, Tavalaei, & Ozalp, 2019). For our study, we took a platform orchestration perspective and focused on another important limitation recently receiving burgeoning interest. Blockchain platforms truncate the platform provider's strategic tools to prioritize some transactions over others to orchestrate an appealing set of third-party applications (platform complements) and steer the direction of innovation when necessary (Leiponen et al., 2021).

One of platform providers' most powerful strategic tools is their ability to set prices and engage in price discrimination to enhance the quality of their services (e.g., Lin, 2020; Liu & Serfes, 2013; Wang & Wright, 2017). Blockchain platforms eliminate this tool, as no entity has the power to set prices for transacting on the platform unilaterally. The transaction price and how it is set are inherent parts of the reward mechanism required to incentivize nodes so that they make efforts to maintain the network. Although platform providers can initially design the overall reward and transaction fee mechanism, they cannot interfere unilaterally with setting the price of an individual transaction after the system is launched.

Currently, most blockchain platforms like Bitcoin and Ethereum rely on a market mechanism that sets the price for transacting on the platform (Buterin, 2014; Nakamoto, 2008). However, for this market mechanism to work, they also restrict the supply of transactions.[62, 63] Our goal is to investigate the consequences of this market mechanism from a platform orchestration perspective. We examined whether blockchain platforms—which remove the platform provider's ability to set prices and substitute them with a market mechanism—are a viable blueprint for platforms aspiring to host a variety of applications and become general-purpose.

We argue that such a market mechanism only prioritizes complements based on their users' transaction fee sensitivity. This is an efficient allocation for homogenous transactions (like those on the Bitcoin network). It can, however, lead to long-term inefficiencies with heterogenous complements as it favors some types over others depending on their current user's transaction fee sensitivity, not the value the complement might provide in the future. It does so by adding an externality in the form of congestion costs to the existing competition between complements in the same category: if one complement attracts more users and thus increases the demand for transactions, the transaction fees for all other complements—irrespective of the service they offer—rise as well, as they are all competing for the same supply of transactions. This is problematic because, as we show in section 5.4, several characteristics other than the quality of a complement determine its users' sensitivity toward transaction fees. Especially in times of congestion and high transaction fees, some complements will be used less, and as the platform provider has no tools to protect them, they may have to abandon the platform even though it would benefit the platform in the long term if the complement stayed.

Such an unregulated reduction of complement heterogeneity is not desirable, as we know from the platform competition literature that users value the diversity of platform complements. Thus, an unsolicited reduction of complements can hamper a platform's potential to leverage same-side and cross-side network effects (see Rietveld & Schilling, 2020). It also questions the neutrality of the blockchain and raises concerns about how this mechanism influences investment incentives for complementors and platform providers similar to the discussion around net neutrality (Choi & Kim, 2010). It also questions whether blockchain platforms that rely on a market mechanism to enforce the correct execution of transactions will be a viable option for Web 3.0, where all web applications have to run fully decentralized.

Despite the important implications and potentially detrimental effects of using a market mechanism to ensure the automated verification of transactions, there is scant research on

[62] Restricting supply is also necessary to maintain a sufficient level of decentralization. Unrestricted supply would favor nodes with greater power to compute and verify more transactions and thus exclude smaller nodes with less computational power.

[63] While changing supply will impact the price, it does not allow for price discrimination, and also requires a consensus by all miners on the Bitcoin network and an EIP (Ethereum Improvement Proposal) or community vote on Ethereum.

how such a mechanism and the lack of strategic tools to protect complements when necessary, impact the heterogeneity of complements offered on a platform. To fill this void, we posed the following research questions: how does a market mechanism for the decentralized verification of transactions affect the use of platform complements? What complements will blockchain platforms offer in the long run?

The context of the Ethereum blockchain provides a unique opportunity to study our research questions, for three reasons. First, Ethereum was the first platform to enable smart contracts, which are computer scripts that enable complementors to offer web applications (Buterin, 2014). These applications that run on top of a blockchain are also called decentralized applications or dApps (Wu et al., 2021). Accordingly, Ethereum qualifies as a multi-sided platform where complementors can offer arbitrary services to users. Ethereum is also the most popular platform for dApps, offering services in finance, gaming, social, insurance, and health. Second, Ethereum uses a market mechanism to allocate the limited supply of transactions among transaction senders (users of a dApp). This market mechanism resembles a first-price auction, where users must bid on how much they are willing to pay for the computational effort their transaction requires (Roughgarden, 2020). Third, Ethereum has served as blueprint for many other blockchain platforms now using a similar mechanism to allocate transactions and thus enhances the generalizability of our results.

For our empirical strategy, we used daily transaction data from a sample of 1,590 dApps on Ethereum and estimated different demand curves for different groups of dApps. To address the endogeneity issues arising from the simultaneous determination of transaction fees by demand and supply, we introduced Ethereum's difficulty bomb as a novel demand-side instrument that has led to exogenous variation in the supply of transactions.

Our analysis yielded several important findings. By finding a downward-sloping demand curve, we can confirm that the law of demand also applies to transactions on Ethereum. While this finding seems theoretically trivial, the ongoing debate on the prevalence of speculation activity, extreme volatility, and illicit transaction conduct, questions whether blockchain platforms are subject to standard supply and demand dynamics like in other financial markets (Foley, Karlsen, & Putniņš, 2019; Li, Shin, & Wang, 2018), and thus calls for empirical clarity before scholars can move on with investigations. We also found that groups of dApps vary significantly regarding their sensitivity towards transaction fees and that, in times of congestion, finance applications crowd out transactions to other applications by increasing the market price for transacting on the network. Our results suggest that building network effects and bundling transactions more efficiently are the only ways a dApp can influence sensitivity towards transaction fees.

Our research contributions are threefold. First, we contribute to the platform literature by exploring how competition induced by a market mechanism for allocating transactions affects the long-term heterogeneity of complements offered on such platforms. These insights are important, not only as they allow us to gauge such decentralized platforms' competitiveness compared to their centralized counterparts, but also because they help us to understand that it is unlikely that an existing blockchain platform will be able to cater for all types of dApps' needs and dominate all other platforms. Second, we contribute more specifically to the literature on platform orchestration by extending it to the realm of decentralized platforms and discussing which orchestration tools might still work when the platform provider's ability to steer transaction activity is limited. Finally, by providing a novel instrument that will help overcome endogeneity problems, we pave the way for future scholars wanting to leverage the rich data on a blockchain platform to investigate the economic dynamics.

The remainder of this chapter features: Section 5.2, explaining how we relate and contribute to the existing literature. Section 5.3 describes all the details needed to understand the process of transacting with an application on the Ethereum blockchain, and conceptualizes Ethereum as a market for transactions. Section 5.4 presents a conceptual framework that is the basis for our empirical analysis and Section 5.5 is a summary of our data. Section 5.6 discusses the empirical strategy to identify the demand curves for different types of applications. The results of our analysis are shown in Section 5.7. Finally, Sections 5.8 and 5.9 conclude with implications for platform providers, complementors, and policymakers and discuss avenues for further research.

5.2 Related Literature

For the theoretical foundation of our work, we draw on two streams of prior research.

5.2.1 Research into transaction fees on blockchain platforms

The first stream is the nascent literature that studies transaction fee mechanisms on blockchain platforms. Within this literature, scholars are already characterizing blockchains as marketplaces where miners offer their services to transaction senders, and they are studying the dynamics of these marketplaces from different theoretical perspectives. For instance, Basu et al. (2019) and Easley et al. (2019) build game theoretic models to analyze how Bitcoin's fee mechanism causes high variability in transaction fees and thus might deter miners (Basu et al., 2019) and users (Easley et al., 2019). Other scholars like Huberman, Leshno, and Moallemi (2017) and Donmez and Karaivanov (2021) use queuing theory to investigate the implications of transaction fee mechanisms on blockchains. Huberman et al. (2017) use this theoretical lens to study miners' entry and exit and find that Bitcoin's transaction fee mechanism protects users from monopoly pricing. Donmez and Karaivanov (2021) use queuing theory to investigate the determinants of transaction fees and reveal that changes in transaction demand and type of transactions are important factors associated with higher fees. A third stream of researchers builds on auction theory (e.g., Lavi, Sattath, & Zohar, 2017). Most notably, Ilk et al.'s (2021) perspective on Bitcoin's transaction fee mechanism shows that the basic forces of demand and supply determine the price of transactions on the Bitcoin platform. They also find that due to a relatively inelastic demand curve and a comparatively elastic supply curve, Bitcoin's current transaction fee mechanism can efficiently self-regulate transaction fees—higher fees stimulate mining to a much higher degree than that they dampen demand. There is ample general research on blockchain mining, some of which addresses transaction fees and their implications as a peripheral topic. For instance, Houy (2016) and Cong et al. (2021) provide a general analysis of Bitcoin's mining game and miners' behavior. Kroll, Davey, and Felten (2013) scrutinize the security of Bitcoin's mining mechanism and conclude that transaction fees only have limited importance. Arnosti and Weinberg (2018) develop a model that considers heterogenous cost structures among miners and explains how this heterogeneity fosters the concentration of mining power. Sapirshtein, Sompolinsky, and Zohar (2016) study the equilibrium between miners and conclude that a properly designed transaction fee mechanism only produces a reliable system in equilibrium if the miners are suitably small.

These accounts implicitly or explicitly focus on the implications of the mining process and develop suggestions how to improve the protocol, but only look at the implications of the transaction fees mechanism for miners or users (transaction senders). Despite their importance for the long-run success of second-generation platforms like Ethereum, enabling third parties to offer additional services in the form of dApps, the consequences of a transaction fee mechanism for these complementors are currently ignored. To fill this void, our research adds to the literature by being the first to investigate how the transaction fee mechanism impacts platform complements (i.e., dApps). Arguably, the transaction allocation mechanism can have severe implications for complementors if skyrocketing fees prevent users from sending transactions to the dApp.

On the empirical side, only a few accounts estimate how transaction fees impact the use of blockchain platforms, and most focus on the Bitcoin blockchain. For example, Ilk et al. (2021) provide empirical evidence that the basic economic theory (i.e., the law of demand) also holds for transactions on blockchains by finding a downwards-sloping demand and an upwards-sloping supply curve for transactions on the Bitcoin blockchain. For Ethereum, this evidence is still lacking. A few accounts investigate the relationship between network congestion and gas prices (Donmez & Karaivanov, 2021), gas prices and throughput (Azevedo Sousa et al., 2021; Spain, Foley, & Gramoli, 2020), or how high gas fees antagonize Ethereum's goal of inclusion and democratization by excluding users who cannot afford these rising fees (Cong, Tang, Wang, & Zhao, 2022). However, there is very little research analyzing the supply and demand dynamics on Ethereum and particularly how these impact the use of dApps. We argue that the possibility to offer dApps distinguishes Ethereum's potential and demarcates similar decentralized platforms' potential to compete with established centralized platforms like Apple's iOS or Google's Android. To know whether decentralized platforms can deplete the prevalence of established centralized platforms, one important step is to understand under what conditions platform complements must work on such decentralized platforms. By presenting empirical evidence for the impact of a decentralized transaction verification mechanism on the use of dApps, our work adds to the literature. As an aside, we also provide initial empirical evidence that basic economic theory applies to transactions on Ethereum and thus pave the way for further economic inquiries.

5.2.2 Research on platform competition and orchestration

The second stream of literature is on platform competition. This stream stems from early work on standard setting and standards battles (e.g., Church & Gandal, 1992; Cusumano, Mylonadis, & Rosenbloom, 1992; Shapiro & Varian, 2010), and seminal work by Katz and Shapiro (1985) and Farrell and Saloner (1986) on network effects. Prior work typically focuses on how platform providers can use strategic tools such as setting prices (e.g., Brynjolfsson & Kemerer, 1996; Gandal, 1994), investment in quality (e.g., Choi, 1994), or subsidizing complements (e.g., Riggins, Kriebel, & Mukhopadhyay, 1994) to their competitive advantage in settings with strong network effects. More recent literature is focusing on the relationship between platform providers and complementors and how platform providers can set rules and norms to orchestrate an appealing ecosystem of platform complements attracting as many users as possible.

For instance, Tudón (2022) investigates the platform providers' trade-off between fostering the entry of new complements and preventing platform congestion and finds that consumer welfare would drop significantly without prioritization on the supply side. Similarly, Panico and Cennamo (2020) investigate if too many complements affect the quality of the ecosystem depending on the nature of the complementors' increasing returns. They find that if complementors' network effects diminish on account of their network's size, a larger network of complementors will dilute the average quality of complements. Both studies thus question the often-oversimplified tenet of the literature that consumers prefer greater breadth and depth in the network. Other scholars echo this idea, suggesting that too many complements may cause coordination problems, increase coordination costs, and reduce consumers' value (e.g., Boudreau, 2012; Casadesus-Masanell & Hałaburda, 2014; Markovich & Moenius, 2009).

Regarding platform governance, O'Mahony and Karp (2020) investigate how the decentralization of decision rights on a platform influences participation. Based on an in-depth case study, they find that although the benefits depend on the platform's products, participants, and markets, for most people in their sample, participation increases with the platform's transition towards decentralized leadership. Related but in the context of blockchain-based platforms, Chen, Richter, and Patel (2021) find an inverted-u-shaped relationship between the decentralization of blockchain platforms and developer activity. Together, these studies emphasize that platform providers need a carefully planned strategy to determine how many and what type of complements they want on the platform.

We add to this literature by investigating blockchain platforms from a platform orchestration perspective. Blockchain platforms are an interesting novel phenomenon as they provide an alternative blueprint for established centralized platforms. They substitute a centralized authority with a market mechanism that ensures the correct and automated enforcement of transactions. By doing so, they truncate the platform provider's strategic tools to attract or exclude complements by setting prices, offering subsidies, or limiting entry. Hence, blockchain platforms limit the power of a strong "visible" hand by shifting the agency to the "invisible" hand of a decentralized market. Although platform providers can define the initial rules of this market, they cannot interfere with them afterwards. As a healthy ecosystem of complements is crucial for a platform's success, it is paramount to understand how the market mechanism used on blockchain platforms to verify transactions influences what type of complements such platforms will offer.

5.3 Background

Ethereum is the second-largest blockchain platform, with a market capitalization of $300 billion and over 1.2 million daily transactions.[64] It is the context of our study as it was the first blockchain platform to introduce smart contracts, which enable more complex transactions than simple money transfers and thus allow complementors to develop their own blockchain-based apps running on top of the blockchain (Buterin, 2014). As transactions differ regarding complexity and thus require differing computational effort to be executed by miners,

[64] https://etherscan.io/ (retrieved March 30, 2022).

Ethereum introduced a new market mechanism that incentivizes miners to compute more computationally expensive transactions. This market mechanism served as a blueprint for many other blockchain platforms that enable smart contracts and thus is seminal for the whole industry. In the following section, we briefly review the core features of Ethereum's market for transactions, focusing on the relevant economic aspects. For a more technical review, we refer to Antonopoulos and Wood (2019) and Wood (2014b).

5.3.1 Smart contracts and dApps

Smart contracts are immutable and automatically enforced computer programs running on top of a blockchain (Fröwis & Böhme, 2017). They allow developers to specify arbitrary agreements between two parties in the form of pre-defined obligations and rules written in computer code. If triggered by receiving a transaction, a smart contract is automatically enforced by the decentralized network according to pre-defined rules, making it impossible for parties to unilaterally alter or renegotiate the transaction's outcome with a smart contract (Halaburda et al., 2019).

As smart contracts enable arbitrary programs, they can be used to develop decentralized applications or dApps (Wu et al., 2021). dApps are blockchain-based apps that resemble normal web applications regarding their user interface but differ from normal web applications as they run their business logic as a smart contract on a decentralized blockchain platform. Due to the immutability and automated enforcement of the underlying smart contract, dApp users do not have to trust the dApp provider or rely on third-party institutions to fulfill their obligations but can read the smart contract and ascertain that the promised outcome will be delivered.[65] Therefore, dApps promise to solve problems of centralized control, limited access, downtime, censorship resistance, and trust issues arising from weak institutions (Leiponen et al., 2021).

DApps are the complements of interest for our study as they extend the functionality of the Ethereum network. Without dApps, Ethereum users could only use the network to send Ether (Ethereum's native cryptocurrency) to each other. With dApps, complementors can offer any arbitrary service. According to DappRadar, Ethereum currently hosts more than 3,600 dApps across finance, games, gambling, insurance, social media, property, and digital identity. It is Ethereum's vision to further grow the number and diversity of dApps offered

[65] For a detailed discussion on where smart contracts remove the need for trust in transactions, see Chapter 3.

on the platform and ultimately pave the way for Web3, a more inclusive and democratic version of the internet, where apps are available to everyone, with no downtime, censorship, entry restrictions, or central control of the data.[66]

5.3.2 Ethereum's market for transactions

To verify and enforce transactions users send to dApps, Ethereum uses a decentralized transaction verification and enforcement mechanism that relies on cryptography, a decentralized consensus mechanism, and economic incentives to substitute a centralized intermediary. Prior scholars have already characterized Bitcoin mining, which uses a similar mechanism, as a two-sided market (e.g., Basu, Easley 2019) and a market for data space more specifically (Ilk 2020). We also characterize Ethereum's transaction verification and execution process as a market but highlight important differences thanks to Ethereum's ability to run smart contracts and offer dApps.

Like on the Bitcoin network, transactions on Ethereum are not instantly effective but have to be verified by specific users called miners. At regular intervals, these miners select transactions from the pool of pending transactions, verify their validity according to Ethereum's protocol, bundle the transactions together, and participate in a computationally demanding puzzle known as "proof-of-work" (PoW). This puzzle requires miners to brute-force numerous hashes until they find a hash that satisfies the protocol's conditions. Only the miners that solve this puzzle first get to write their block onto the blockchain and receive the block reward in addition to all the transaction fees paid by senders. The mining of transactions comprises two tasks. First, the miner has to compute the transaction and check it against a list of rules—only if the transaction fulfills these rules can the miner add it to block. If just one transaction in a block does not fulfill the requirements, the entire block will be rejected by all the other miners. Second, the miner has to solve the proof of stake (PoS) puzzle by computing numerous hashes until they find a block hash that fulfills the requirements for a new block. Both tasks require computational effort. The update from PoW to Proof-of-Stake (PoS, an alternative consensus mechanism that does not require solving a computationally expensive puzzle, but randomly assigns miners the privilege to write a new block based on their stake tokens) will dramatically reduce the computational efforts required to find a new block. However, the update will not impact the amount of effort miners have to invest in verifying every individual transaction. In essence, the update to PoS will even increase the relative importance of the effort required to verify a transaction.

[66] https://ethereum.org/en/upgrades/vision/, accessed September 15, 2022.

In contrast to Bitcoin and to facilitate dApps and arbitrary transactions, Ethereum does not charge a fee per transaction but a fee for the computational effort a transaction requires. A transaction's computational effort is measured in *units of gas* according to a list of fixed gas requirements for every atomic computation. To maintain decentralization by ensuring that miners with less powerful machines can also participate in mining transactions, a block has a maximum gas limit (*block gas limit*). The Ethereum protocol also tries to keep the average time for finding a new block (*average block time*) within a 12 to 14 seconds interval (Wood, 2014b). These two limitations imply that the total amount of gas available has an upper limit. To allocate this limited gas supply, Ethereum uses a market mechanism that we conceptualize as a market for transactions or, more specifically, a market for the verification and enforcement of transactions.

The commodity sold on this market is the gas required to verify a transaction.[67] Accordingly, users (transaction initiators) are the buyers, whereas miners are the sellers of this commodity. On the supply side, the daily gas supply is fixed due to the block gas limit and the limited average block time. Although miners can decide to what extent they use this limit, they cannot change it individually. Changing this limit requires successful voting by all miners and a protocol update. Also, suppose more miners join the network and participate in the race to solve the mining puzzle. In that case, the network will increase the mining difficulty (the average number of hashes it takes to find a new block) to keep the average block time within the target window of 12 to 14 seconds and the gas supply fixed. To ensure a stable average block time, Ethereum adjusts the mining difficulty for every new block according to the following function:

$$block\ time_b = \frac{mining\ difficulty_b}{network\ hash\ rate_{b-1}}$$

Where $mining\ difficulty_b$ is the average number of hashes required to find a new block and $network\ hash\ rate_{b-1}$ is the number of hashes computed per second by all miners while searching for the previous block.

To incentivize miners to provide their computation service, they are awarded a mining reward for every block they find. This consists of a static block reward (at the time of writing, 2 Ether) for finding a new block plus the sum of all gas fees (usually measured in *GWei*; 1 Ether = 10^9 GWei) paid by all transactions *t* which a miner includes in this block. Hence, the mining reward for every block *b* is:

[67] Importantly, the transaction initiator only pays gas fees for the transaction computation not the computational effort the miner has to invest in solving the PoW puzzle required to find a new block.

$$mining\ reward_b = 2 + \sum_{\forall t \epsilon b} \frac{gas\ price_t \times gas\ used_t}{10^9}$$

On the demand side, users cast transactions to other externally owned accounts (simple Ether transfers to other users or wallets controlled by computers) or smart contracts. To initiate a transaction, users must indicate a *transaction gas limit* (the maximum amount of gas a miner is allowed to use to compute the transaction) and a *gas price* (the price the user is willing to pay for each unit of gas). If the gas limit is reached before the transaction is fully computed, the transaction will be aborted and not included in the block. Users only pay for the gas used if the computation is completed before reaching the limit. Also, only the actual amount of gas used is considered for the block gas limit. Accordingly, the fees a user has to pay for a transaction *t* are computed as follows:

$$transaction\ fees_t = \frac{gas\ price_t\ x\ gas\ used_t}{10^9}$$

As the gas supply is limited, transaction senders compete with other senders by choosing a gas price that is high enough that miners pick their transactions from the pool of pending transactions. Typically, miners engage in profit maximization (Basu et al., 2019). Hence, they sort transactions by the indicated gas price and requirement and fill up the block until its gas limit is reached. Especially in times of congestion, offering too low a gas price means that a transaction will not be picked up by any miner and ultimately be deleted from the pool of pending transactions. Although, in theory, transaction initiators can observe other initiators' gas price bids and adjust their bids accordingly, in line with Roughgarden (2020), we see this price mechanism as a first-price, sealed-bid auction. Our reasons for this type of auction are threefold. First, even though the pool of pending transactions is openly available, the peer-to-peer nature of the pool implies that not every participant sees every transaction simultaneously. Thus, it is difficult for initiators to determine what transactions were available to the miner when assembling the block. Second, although a block is found on average every 12-14 seconds, the exact timing of a block's discovery cannot be predicted. Therefore, initiators do not know when they need to be among the highest bidders. Third, some wallets already offer gas price suggestions that help to gauge a price that will highly likely lead to the transaction being included in one of the next blocks. However, these tools are only backward-looking. They suggest a gas price by extrapolating the gas prices that have led to including the transaction on one of the last blocks. If initiators want to ensure that their transaction is processed with certainty, they still need to exceed this suggestion and account for the possibility that other initiators will do so, too. This gas price mechanism has led to considerable fluctuations in the amount of gas used, and the price users have paid for a unit of gas. To illustrate this, Figure 29 shows the daily gas usage on the left and the daily average gas price on the right.

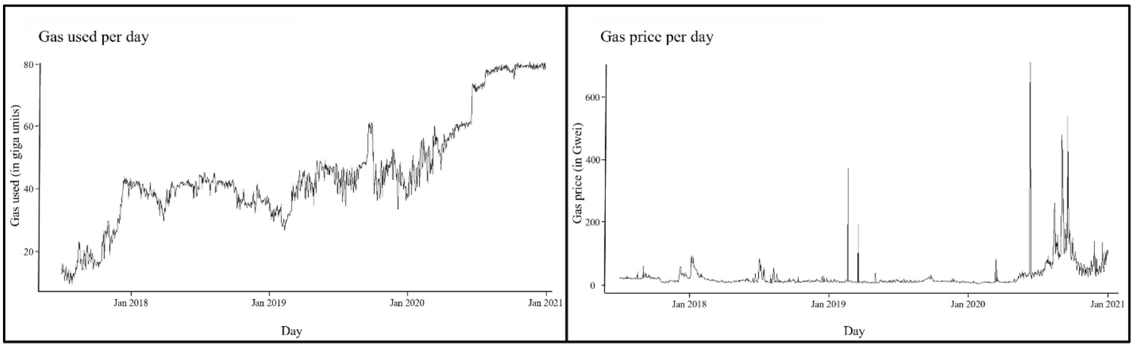


Figure 29: Daily gas used and gas price

In the next section, we develop a conceptual framework that explains the intuition underlying our empirical analysis. As our study focuses on the implications of Ethereum's market for transactions for the heterogeneity of complements offered on the network, the framework focuses on how gas fees impact the use of dApps. For an analysis of how gas fees impact users (transaction senders) and miners in the network, we refer to Cong et al. (2022) and Basu et al. (2019).

5.4 Conceptual framework

Here we discuss the intuition underlying our empirical analysis. Importantly, although our empirical analysis is—due to our selected instrumental variable—limited to a period when Ethereum relied on PoW as a consensus mechanism, our ensuing theoretical arguments also apply to the period when Ethereum updated to PoS.[68] The update to PoS only removed the computationally expensive puzzle of finding a new block but did not change the fact that users still need to compensate miners for verifying and enforcing their transactions by paying fees for the gas used.

The driving force behind our framework is that the use of a dApp—hence its success—on Ethereum depends on the use of the platform, which in turn depends on the use of other dApps. However, due to two countervailing forces, it is unclear if increasing the user base and dApp base benefits all dApp providers. On the one hand, entering dApps attract new users to the platform, foster the platform's adoption, and increase the number of potential dApp users. On the other hand, the limited supply of transactions combined with the first-price auction allocating this limited supply aggravate the direct competition among dApps by introducing a negative externality: new dApps and users increase demand and intensify the

[68] Our arguments should also apply to the period after EIP1559 (Ethereum Improvement Proposal). Although EIP1559 introduced a more flexible block gas limit and an upper limit to the fees users can pay miners to incentivize them to process their transaction quickly, it neither changed the fact that the gas supply is still fixed nor that users can outbid others by paying higher fees.

competition for the limited supply of gas. The increasing demand and competition raise congestion costs and gas prices. Because transaction initiators need to pay transaction fees to interact with every dApp, increasing gas prices reduce the overall utility and thus the use of dApps. Accordingly, the relative magnitude of these countervailing effects will determine how Ethereum's market for transactions impacts the success of the platform complements.

Although the net impact of increasing gas prices in response to more platform use is theoretically undetermined—due to the countervailing forces described above—we can analyze which characteristics of a dApp make it more vulnerable to changes in the gas price. Understanding this is not only useful for the complementors' decision to enter such a market but also for the platform provider, as it might have important implications for the heterogeneity of complements offered on the platforms. We hypothesize that depending on four characteristics, dApps are more or less sensitive to changes in the gas price and, therefore, better or worse equipped to compete in a transaction market.

First, the type of service a dApp offers should influence its sensitivity towards changes in the gas price. This intuition becomes clear when considering that some dApps provide social and entertainment services while others provide financial or security-related services. Although finance dApps do not necessarily provide more benefit to users than leisure-related dApps, it is easier to compute the expected benefit of a finance transaction. Therefore, it should be easier for users to evaluate if they still want to send a transaction, whereas the uncertainty and cognitive effort to gauge the expected benefit might deter others. Furthermore, finance-related transactions are often more time-sensitive, and as Donmez and Karaivanov (2021) show, users on Ethereum are more willing to pay higher gas fees for timely transactions. Some services might differ regarding their gas price elasticity due to the frequency of required interactions. For instance, property and identity-related dApps typically require only infrequent interaction, whereas gaming or finance dApps require regular interactions. Through frequent interactions, gas fees can quickly accumulate and deter usage.

Second, even within the same type of service, dApps can differ substantially in their transaction requirements, for example regarding the complexity of the underlying transaction and hence the gas required for its computation. On the one hand, gas requirement correlates with the complexity of the underlying functionality. On the other hand, it is also driven by the code's actual efficiency. Within the same type of service, where the functionality and complexity of transactions with dApps are similar, the code's efficiency should be the main determining factor for gas requirement. Especially in times of high gas prices, we expect users to be more sensitive to such differences and use dApps that require less gas for the same functionality. Another factor determining gas price sensitivity is the value transferred in transacting with a dApp. For example, finance dApps carry value to transfer money to other accounts or invest it (e.g., in a liquidity pool). Other dApps require users to pay for their services (e.g., getting data from an oracle) or purchase goods (e.g., NFTs). Considering that some NFTs are sold for well above $100,000,[69] evidently even gas fees of a few dollars are negligible. Depending on the average transaction value, we expect that dApps are sensitive to changes in the gas price.

Third, dApps' services also differ in overall quality or usability and hence the value they create for their users. Accordingly, some dApps are more appealing to users than others. These not only perform better at baseline but are also more likely to benefit from other dApps entering. For example, if many new dApps enter Ethereum, this should attract additional users since they appreciate product variety. But once the users join, they will disproportionately choose the dApp offering more benefit. This effect is intensified if the dApp benefits from network effects, which should be the case with currency exchanges, marketplaces, or social messengers. For such dApps, the increasing benefit thanks to the larger network could counterbalance the additional fees as a result of the intensified competition for gas among dApp users.

Fourth, a dApp's current performance should influence users' willingness to pay fees for transacting with it. Again, especially for a dApp that relies on network effects, the number of other users should increase the benefit of transacting with this dApp.

5.4.1 Implications for the platform

The heterogeneity of complements is a decisive factor in a platform's success (Rietveld & Schilling, 2020). Therefore, platform providers need to strategize how many and what types of complements they want to attract for orchestrating an ecosystem of complements that creates the most value for users. Because blockchains by design limit platform providers' strategic toolset to devise a healthy ecosystem of dApps, understanding what dApp features a market mechanism for the decentralized verification of transactions caters to is vital. As elaborated above, we expect some dApps to be more sensitive to changes in gas prices. When gas prices are high, it will be more difficult to attract users. If decline in use continues for a long time, the dApp might have to terminate its business and leave the platform. Consequently, characteristics associated with higher sensitivity to gas price should also be associated with a higher likelihood of an exit and a lower likelihood of entry, particularly when gas prices are high.

[69] CryptoPunk are sold for as much as 8,000 Ether. https://opensea.io/collection/cryptopunks

An unsuccessful dApp's exit might be desirable for the platform provider and users if this is due to poor quality (e.g., if the dApp relies on inefficient smart contracts that require more gas than the competition's smart contract). However, the exit might be less desirable if it is because the transaction verification mechanism discriminates against other dApp characteristics (e.g., type of service offered, or value of a transaction). To ascertain whether there is undesirable discrimination in transactions, we empirically investigated the drivers of a dApp's sensitivity hypothesized above and discuss their implications.

5.5 Data and sample construction

We combined block and transaction-level data stored publicly on the Ethereum blockchain with three different sources of supplementary off-chain data, such as the dApp category or exchange rate for one Ether or other tokens. Below we explain the data sources, the resulting sample, and the variables in our data set.

5.5.1 Data collection procedure and sample

We obtained our data from four different sources. First, we used the Ethereum ETL[70] to download all block-level and transaction-level data publicly stored on the Ethereum blockchain during our study period (July 1, 2017 to December 31, 2020).[71] The block-level data includes a unique identifier (block hash), a timestamp, the *difficulty of the block*, the *gas limit* indicating the maximum amount of gas that miners are allowed to use in this block, and the *gas used,* which is the sum of computational effort required to verify all transactions in this block. The transaction-level data contains the block hash, a sender and recipient address, the *gas used* by this transaction, and the *gas price* the sender has paid for one unit of gas in GWei (1 GWei = 10^{-9} Ether). Second, we consulted two websites that provide a curated list of dApps (stateofthedapps.com and defillama.com) to identify dApps running on Ethereum, their associated smart contract addresses, and the application category. This step, enabling us to trace the pseudonymous smart contract addresses on the blockchain to their respective dApp, is necessary because a dApp can have multiple smart contracts. Overall, we identified 1,590 dApps with 4,680 associated smart contracts active in our study period. As neither stateofthedapp.com nor defilama.com provides an exhaustive list of all smart contracts associated with a dApp, we further collected a list of all verified smart contracts from the

[70] https://ethereum-etl.readthedocs.io/en/latest/, accessed September 15, 2022.

[71] This study period allowed us to observe three periods when the additional difficulty induced by the difficulty bomb caused a shortage in gas supply (see Figure 2 and Ethereum Improvement Proposal 649, 1234, and 2384).

Etherscan API[72] and manually matched 1,316 additional smart contracts to the dApps in our sample. Through the smart contract addresses, we could link transactions with their associated dApps. We also used the Etherscan API to collect further daily network-level data, such as the *network utilization*, which measures to what extent the block gas limit has been used. Finally, we retrieved the daily prices for one Ether and other tokens associated with the dApps in our sample from the CoinGecko API.[73] To ensure all variables are on the same level and mitigate high-frequency variation in the data, we first merged the block-level and transaction-level data using the block hash reported for every transaction, then aggregated the resulting data at the daily level. Our consolidated dataset covers 1,279 daily observations. Table 12 shows the number of dApps per group of categories.[74]

Table 12: Groups of dApps in our sample

Groups	dApp categories	Examples	# dApps
Group 1	finance, exchanges, wallets, insurance, security	Sushi Swap, OmiseGo, Status, Nexus Mutual, Chainlink	507
Group 2	identity, property	ENS Manager, Decentraland	45
Group 3	games, marketplaces	Axie Infinity, Cryptokitties	464
Group 4	gambling, social, health	FunFair, Minds, BEAT	397
Group 5	energy, governance, media, storage	Dovu, Aaragon, CryptoTunes, XCloud	177

5.5.2 Data sets, variables, and measurement

Besides the daily aggregation, we employed two further aggregations resulting in two data sets that we used for our analysis. The first data set aggregates all transactions on the platform level. This allowed us to estimate network-level demand curves, compare the demand curves between Ether transfers between two externally owned accounts and dApp transactions, and estimate a separate demand curve for every group of dApps. This ensures comparability with other studies that conduct their analysis only on the network level (e.g., Donmez & Karaivanov, 2021; Ilk et al., 2021). The second data set aggregates transactions at dApp level and thus allowed us to include individual and time-fixed effects. Unless noted otherwise, both data sets comprise the following variables.

Our main variable of interest is the quantity of *gas used*. It refers to the daily amount of computational verification effort required by all transactions (on the network level in the

[72] https://etherscan.io/apis .

[73] https://www.coingecko.com/en/api/.

[74] To mitigate multicollinearity issues due to similar transaction patterns across similar categories, we aggregated the 17 categories into 5 groups offering comparable services. We identified the groups by applying a cluster analysis to variables like daily transaction count and transaction value.

first data set and the dApp level in the second data set) measured in Giga gas units. It is the goods supplied by the miners and requested by the transaction senders.

The *gas price* (in GWei) is what transaction senders have to pay for each gas unit. As this price varies according to the outcome of a first-price auction, we define it as the *market gas price* a sender would have to pay for their transaction just to make it into one of the blocks on a given day. We proxy this market price with the daily average of the bottom fifth percentile gas price recorded on each block that day in GWei. We used this proxy because there are blocks where for verification, miners circumvent the first-price auction mechanism by adding their own transactions with a gas price close to zero or even zero. Accordingly, the marginal gas price (the lowest gas price on a day when a transaction has just been added to a block) would not correctly reflect the market mechanism. We also ran several robustness checks with alternative gas price variables (different percentiles of the gas price in USD).

We define the variable *difficulty bomb* as the average additional difficulty induced by Ethereum's difficulty bomb on a given day. Next to the automated adjustment of the mining difficulty, the difficulty bomb is the second mechanism encoded in Ethereum's protocol that influences the total network difficulty (the average number of hashes it takes to find a block). The difficulty bomb is meant to force miners to switch from PoW to PoS once the PoS update is available. To this end, the difficulty bomb exponentially increases the mining difficulty until it is almost impossible to find new blocks by solving the PoW puzzle. As the plan right from the start was to switch to PoS at some point, the difficulty bomb was always part of the protocol. However, because the update to PoS was delayed several times, the difficulty bomb increased the difficulty too quickly, resulting in a disproportionate increase that was not reflected by the network hash rate and the discovery of significantly fewer blocks per day. Because the resulting gas shortage was not intentional (the plan was that PoS-blocks would grow at the same rate as they declined), the Ethereum community issued a protocol update that reverted the additional difficulty. Over our study period, this pattern occurred three times and is reflected in three protocol updates (EIP649, EIP1234, and EIP2384). As the difficulty induced by the difficulty bomb is not reported in any database, we leveraged the fact that Ethereum's protocol continually tried to keep the block time within the target window of 12-14 seconds and created the following variable: the difficulty induced by the difficulty bomb on a day *d* is the difference between the total observed difficulty and the theoretical difficulty required to achieve the target block time, given the current hash rate in the network. Accordingly, the difficulty bomb on a day *d* is:

$$
\begin{aligned}
&difficulty\ bomb_d \\
&\qquad = (network\ hash\ rate_d \times target\ blocktime) - difficulty_{\ observed,d}
\end{aligned}
$$

The unit of this variable is the average number of Tera hashes required to find a new block. Due to the exponential growth and fluctuation of the network difficulty within the target window, especially at the beginning of difficulty bomb activity, the added difficulty is not always distinguishable from zero. To account for this, although the difficulty bomb is always active, we only assigned a positive value to the difficulty bomb if the block time was noticeably above the target window (> 14s). Using this conservative approach, we only observed a difficulty bomb above zero on 16 percent (182) of all the days in our sample. To establish robustness, we also used different cutoffs and approaches to measure difficulty bomb activity. We discuss our instrument's relevance and exogeneity later in the empirical strategy and results section.

To ascertain to what extent miners fill the blocks on a given day, we measured *network utilization* as the fraction of total available gas (sum of all blocks' gas limit) used by all transactions on one day in percentages. It captures the platform's usage level and researchers previously used this to measure congestion (Donmez & Karaivanov, 2021).

In addition to these variables, we computed several measurements that allowed us to study each dApp's transaction requirements or usage patterns. These variables are only available on the dApp level and are thus only part of the second data set. To reflect the complexity of an interaction with a dApp, we measured the *average gas requirement* for transacting with a dApp. To reflect this requirement, we measured the *average value of Ether or tokens* a dApp receives as a proxy for the usual value of transactions with a dApp. In addition, we measured the following performance indicators for every dApp: *average daily transaction activity*, *average number of unique externally owned accounts (EOA) that transactions with a dApp (our proxy for users)*,[75] *average gas price users pay* for a transaction with a dApp, *average number of transactions per externally owned account* on a given day, and *surplus gas price* that transaction senders pay beyond the market gas price on a given day.

We also controlled for the following variables: *Ether price* measures the price of one Ether in USD on the day the transaction was executed; *Ether volatility* measures the daily change in the exchange rate of one Ether; *Gas limit* measures the sum of all block gas limits on one day and accounts for the fact that over our sample period, the total units of gas that can be used in a block have increased several times; and finally, *day of the week* and *year* dummy variables, and a *trend*.

[75] It is technically possible to differentiate smart contract addresses from wallet addresses, but not if a wallet address is controlled by a bot. Consequently, we do not call wallet addresses "users" but "externally owned accounts" to emphasize they do not necessarily correspond to human users. This variable is therefore only a proxy.

Table 13 and Table 14 provide descriptive statistics and correlation scores for all variables in our data sets.

Table 13: Descriptive statistics and correlations (network level data)

Variables	N	Mean	S.D.	1	2	3	4	5	6	7	8	9	10	11	12
1. Gas used	1,280	45.42	17.15	1											
2. Gas used group 1	1,280	18.96	18.65	0.88	1										
3. Gas used group 2	1,280	0.39	0.66	-0.50	-0.28	1									
4. Gas used group 3	1,280	2.43	1.77	-0.04	-0.25	-0.23	1								
5. Gas used group 4	1,280	0.86	0.61	-0.09	-0.27	-0.12	0.46	1							
6. Gas used group 5	1,280	0.56	0.53	-0.21	-0.20	0.09	-0.14	-0.42	1						
7. Market gas price	1,280	6.75	12.29	0.73	0.86	-0.16	-0.33	-0.33	-0.15	1					
8. Difficulty bomb	1,280	1.08	2.92	-0.48	-0.23	0.25	-0.25	-0.06	-0.05	-0.12	1				
9. Network utilization	1,280	0.83	0.13	0.73	0.53	-0.60	0.01	-0.20	0.03	0.45	-0.18	1			
10. Ether price	1,280	327.48	218.96	0.10	0.11	-0.04	-0.19	-0.62	0.64	0.13	-0.16	0.27	1		
11. Ether volatility	1,280	0.36	23.46	0.03	0.05	-0.01	0.04	-0.01	0.04	0.05	0.01	0.03	0.07	1	
12. Gas limit	1,280	0.01	0.002	0.93	0.90	-0.41	-0.08	-0.02	-0.29	0.75	-0.31	0.53	0.001	0.03	1

Table 14: Summary of statistics and correlations (dApp level data)

c	N	Mean	SD	Min	Max	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)
(1) Gas used	370,748	180,178	1,288,645	21	85,346,148	1																		
(2) Transaction activity	370,748	893	9,213	1	518,357	0.89	1																	
(3) EOA	370,748	288	3,143	1	168,900	0.82	0.97	1																
(4) Average transaction gas price	370,748	28	44	0.00	6,250	0.05	0.05	0.05	1															
(5) Market gas price	370,748	8	14	1	54	0.06	0.06	0.06	0.71	1														
(6) Difficulty bomb	370,748	65	210	0.00	1,610	-0.01	-0.01	-0.01	-0.09	-0.13	1													
(7) Network utilization	370,748	301	195	84	1,385	0.02	0.02	0.02	0.25	0.23	-0.20	1												
(8) Network utilization^2	370,748	0.24	20	-228	153	0.01	0.01	0.01	0.01	0.06	0.001	0.06	1											
(9) log(Ether price)	370,748	0.85	0.10	0.30	0.98	0.03	0.04	0.04	0.37	0.54	-0.12	0.34	0.04	1										
(10) log(Ether volatility)	370,748	0.73	0.17	0.09	0.97	0.04	0.04	0.04	0.40	0.57	-0.12	0.36	0.04	1.00	1									
(11) Gas limit	370,748	9,278	1,739	6,704	12,485	0.06	0.06	0.06	0.49	0.77	-0.21	0.12	0.06	0.51	0.54	1								
(12) Age	370,748	415	322	1	1,280	0.05	0.08	0.08	0.23	0.33	-0.11	-0.15	0.03	0.24	0.24	0.49	1							
(13)Average gas requirement	370,748	322	478	21	9,900	0.04	-0.02	-0.02	-0.05	0.01	-0.02	-0.08	-0.001	0.01	0.01	0.05	-0.10	1						
(14) Average value sent USD	370,748	366	3,656	0.00	99,002	0.01	-0.001	-0.0002	0.01	0.01	-0.001	0.02	0.001	0.01	0.01	0.01	0.01	0.01	1					
(15) Average token value sent USD	370,748	2,781	13,909	0.00	185,968	0.05	0.04	0.03	0.02	0.01	0.01	0.02	0.003	-0.002	0.0004	0.001	0.07	-0.03	0.01	1				
(16) Average daily transactions	370,748	893	5,695	1.00	71,089	0.53	0.62	0.61	0.02	0.01	-0.004	0.002	-0.0004	0.01	0.01	0.01	0.05	-0.04	-0.002	0.06	1			
(17) Average daily EOA	370,748	288	1,954	1.00	24,975	0.50	0.61	0.62	0.02	0.01	-0.01	0.002	-0.001	0.01	0.01	0.01	0.05	-0.04	-0.0002	0.05	0.99	1		
(18) Average transactions per EOA	370,748	6	20	1.00	354	0.03	0.01	-0.01	0.01	0.04	-0.005	-0.02	0.01	0.02	0.02	0.05	-0.04	0.09	-0.01	-0.02	0.01	-0.02	1	
(19) Transactions per EOA	370,748	6	44	1.00	4,488	0.07	0.02	-0.01	0.002	0.01	0.01	-0.01	0.001	-0.01	-0.01	0.003	-0.04	0.04	-0.01	-0.01	0.01	-0.01	0.46	1
(20) Surplus gas price paid	370,748	19	30	-129	6,249	0.04	0.05	0.04	0.93	0.44	-0.06	0.23	-0.01	0.25	0.27	0.29	0.15	-0.08	0.01	0.03	0.02	0.02	0.002	-0.001

5.6 Estimation strategy

Here we discuss the network, dApp level specifications, and the instrumental variable (IV) used in both specifications to address the endogeneity of the gas price.

5.6.1 Baseline network-level specification

The specification for our network-level analysis:

$$\log(Gas\ used_t) = \alpha_0 + \alpha_1 \log(Market\ gas\ price_t) + \alpha_2 Network\ utilization_t + \alpha_3 Network\ utilization_t^2 + \alpha_4 \log(Ether\ price_t) + \alpha_5 \log(Ether\ volatility_t) + \alpha_6 \log(Gas\ limit_t) + \mu_{dayofweek} + \mu_{year} + trend + u_t,$$

where gas used is the equilibrium gas demand aggregated over all executed transactions on the network or per group of dApps in the period t (day), with $\mu_{\text{dayofweek}}$ denoting the day of week fixed effects, μ_{year} the year fixed effects, and u_{t} the error term. We chose a log-log specification for gas used and market gas price in order to interpret α_1 as the price elasticity of demand. Due to the skewed distributions of Ether price, Ether volatility, and the gas limit, we used log-transformed versions of these variables in our specification. We also controlled for the level of network utilization. This allowed us to control to what extent miners use the available block gas limit on a given day, as used by other scholars to measure network congestion (Donmez & Karaivanov, 2021). We added a quadratic term to account for the nonlinear relationship between gas price and network utilization.[76]

In this model, log(*Gas used$_t$*) and log(*Market gas price$_t$*) are the endogenous variables, as both are jointly determined in equilibrium. To address this simultaneity issue, we used the *difficulty bomb* as an instrumental variable in a two-stage least squares approach (2SLS). In the first stage, we used the difficulty and all other control variables listed above to predict the log(*Market gas price$_t$*). In the second stage, we estimated the above specification by replacing the log(*Market gas price$_t$*) with its predicted value. The economic intuition underlying our approach is that we leverage the difficulty bomb as an exogenous supply shifter. Due to the consistent adjustment of the network difficulty and the resulting constant block time, the gas supply curve resembles a fixed vertical line. When the difficulty bomb is active, the added difficulty increases the block time and thus reduces the number of blocks on a given day. As the maximum gas a block can contain is limited, fewer blocks lead to a reduced gas supply

[76] We also computed the same model with a threshold specification, only adding the linear term and dummy variable that take on the value one if the utilization level exceeds 90 percent. The results were qualitatively the same for the magnitude and significance of our coefficients.

and hence a horizontal shift of the supply curve to the left. We exploited this supply shift to identify the demand curve.

We argue that the difficulty bomb is exogenous and only affects gas demand through the increased gas price for three reasons. First, it is programmed into the Ethereum protocol. Changing it requires a successful protocol update (called Ethereum Improvement Proposal or EIP), which is only possible after a majority vote, and is therefore an unlikely response to a short term market situation. Changes to the difficulty bomb can thus be seen as exogenous policy interventions. Second, as the difficulty level is not reported in wallet applications or by an API and has to be manually calculated (see above), ordinary Ethereum users were presumably not aware of the difficulty bomb's existence. Third, even if users were aware of the difficulty bomb, it is difficult for them to comprehend its exponential growth and differentiate its impact—at least in the initial phase—from normal fluctuations due to miners' entry and exit. It would also be difficult for users to predict every single miner's power and evaluate the cost structure if they cannot keep up with the difficulty level.

5.6.2 Baseline dApp-level specification

The specification for our dApp-level analysis is similar to the network-level specification above:

$$\log(Gas\ used_{dt}) = \alpha_0 + \alpha_1 \log(Market\ gas\ price_t) + \alpha_2 Network\ utilization_t + \alpha_3 Network\ utilization_t^2 + \alpha_4 \log(Ether\ price_t) + \alpha_5 \log(Ether\ volatility_t) + \alpha_6 \log(Gas\ limit_t) + age_{dt} + \mu_d + \mu_{dayofweek} + \mu_{year} + u_t,$$

with the difference that the index *d* denotes the dApp and the panel specification allowed us to conduct a within transformation, add μ_d as individual fixed effects, and control for the intrinsic growth of the dApp by adding age_{dt} as the number of days since the dApp entered the platform.

To address the simultaneity of gas demand and gas price, we leveraged the same procedure discussed above. To analyze the impact of the group and other time-variant or time-invariant characteristics of a dApp, such as the average gas requirement for a transaction or the typical value of transactions with a dApp, we interacted these variables with the log(Market gas price) variable. We discuss our results in the next section.

5.7 Results

This section presents two sets of results. The first reports the network-level analysis, including the estimate of a general demand curve for transactions on Ethereum and for each group of dApps. The second set is the results of our dApp-level analysis, investigating further characteristics of a dApp that determine its sensitivity towards changes in the gas price.

5.7.1 Baseline network-level results

Table 15 shows our 2SLS demand curve estimate. Column 1 presents the first stage results, predicting the gas price (*log(Market gas price)*) with our IV (*difficulty bomb*). Column 2 presents the second stage results, where we used the predicted gas price to estimate the price elasticity of gas demand (*log(Gas used)*). Finally, column 3 provides an OLS model for comparison.

Table 15: 2SLS model 1st and 2nd stages with an OLS benchmark (network level)

	(1)	(2)	(3)
	2SLS *1st stage*	2SLS *2nd stage*	OLS
	log(Gas price)	log(Gas used)	log(Gas used)
Difficulty bomb	0.10^{***} (0.02)		
log(Market gas price)		-0.69^{***} (0.16)	-0.04^{**} (0.02)
Network utilization	-3.03^{***} (0.35)	-1.58^{***} (0.43)	0.20 (0.19)
Network utilization^2	17.51^{***} (1.85)	10.38^{***} (2.60)	-0.33 (0.87)
log(Ether price)	0.09 (0.13)	0.06 (0.08)	0.12^{**} (0.05)
log(Ether volatility	-0.02 (0.02)	-0.01 (0.01)	0.001 (0.003)
log(Gas limit)	3.08^{***} (1.11)	3.02^{***} (0.99)	0.53^{*} (0.28)
$D^{Thursday}$	-0.04 (0.03)	-0.03 (0.02)	-0.001 (0.002)
D^{Friday}	0.01 (0.03)	0.005 (0.02)	-0.001 (0.003)
$D^{Wednesday}$	-0.02 (0.02)	-0.01 (0.02)	0.0002 (0.002)
D^{Monday}	-0.05 (0.03)	-0.03 (0.02)	-0.01^{*} (0.004)
$D^{Saturday}$	-0.02 (0.04)	-0.01 (0.02)	-0.01 (0.01)
D^{Sunday}	-0.03 (0.04)	-0.02 (0.02)	-0.01 (0.01)
D^{2018}	-1.21^{***} (0.20)	-0.85^{***} (0.26)	0.13 (0.19)
D^{2019}	-1.61^{***} (0.29)	-1.11^{***} (0.30)	-0.005 (0.24)
D^{2020}	-1.30^{**} (0.62)	-0.90^{**} (0.40)	-0.03 (0.27)
Trend	0.001 (0.001)	0.001^{*} (0.0005)	0.001^{***} (0.0003)
Constant	-13.30 (18.66)	-2.97 (12.00)	-7.81 (6.25)
Observations	1,279	1,279	1,279
R^2	0.79		0.94
F Statistic (df = 16; 1262)	305.20^{***}		$1,220.08^{***}$
C-D Wald F Stat.		85.06	
Stock-Yogo Critical Value		16.38	
Kleibergen-Paap LM Stat.		4.18**	

Note: Heteroskedasticity and autocorrelation consistent (HAC) standard errors are shown in parentheses, where the optimal bandwidth (23) is calculated in line with Newey and West (1987). *p<0.1; **p<0.05; ***p<0.01

Confirming our theoretical prediction, Columns 2 and 3 suggest a downwards-sloping demand curve for gas on Ethereum. The first stage in Column 1 shows that an increase in additional difficulty due to the difficulty bomb is significantly associated with increased gas prices. This is in line with our explanation that the added difficulty reduces the gas supplied—by reducing the number of blocks explored per day—and thus intensifies price competition among transaction senders. The coefficient of the difficulty bomb is highly significant despite the fact that we controlled for network utilization (i.e., the degree to which miners use the available block space), network utilization squared,[77] the exchange rate of Ether to USD, the daily fluctuation of this exchange rate, the block gas limit, as well as day of the week, year dummies, and a common trend.

Regarding the validity of our instrument, by comparing the first-stage with and without the instrument, we obtained an incremental F (305.20), well beyond the suggested cut-off of 10 (Stock & Yogo, 2005), thus suggesting that our instrument strongly correlates with the endogenous gas price. To test the relevance of our instruments, we computed the Stock-Yogo (Stock & Yogo, 2005) test for weak instruments, which shows that the Cragg-Donald-Wald F Statistic (85.06) exceeds the predetermined critical value (16.38). We also computed the Kleibergen-Paap LM Statistic (4.18) for under-identification, which is highly significant. These tests suggest that our instrument is both strong and relevant. Regarding its exogeneity, we explained above that the difficulty bomb does not impact the gas demand except through an increase in gas price as the mining difficulty is simply a "production factor" for miners that the casual Ethereum user is unlikely to track.

To assess the validity of our results, we compared the 2SLS estimate of α_1 (-0.69) with the OLS estimate of α_1 (-0.04)—Columns 2 and 3. An unobserved negative supply shock would shift the vertical supply curve to the left, leading to an intersection with the demand curve at a high price. Hence, the error term in our specification should be negatively correlated with the gas price. Accordingly, not controlling for endogeneity should lead to a downward bias in the OLS estimate of α_1. As the gas price effect is significantly greater for the 2SLS estimator, our results align with this theoretical expectation.

To interpret the effect of gas price (log(*gas price*)) on gas log(Gas used) demand, the coefficient of -0.69 implies that a 1 percent increase in the market price of a unit of gas reduces the amount of gas demand by 0.69 percent. Considering the average transaction on Ethereum consumes 184,000 units of gas (corresponding to a normal smart contract interaction), this equals a reduction of roughly 1,703 smart contract transactions per day or 14,923 Ether transfers which require 21,000 units of gas. As the median dApp only receives eight transactions per day, the order of magnitude of this effect can have significant economic implications for individual dApps if some are more affected than others.

[77] The inclusion of the quadratic term is suggested by a scatterplot showing a highly nonlinear relationship between network utilization and gas price. When network utilization exceeds 90 percent, the gas price increases dramatically. We also performed a robustness check using a threshold effect at 90 percent network utilization in the form of a binary variable that equals 1 if the utilization is above 90 percent and 0 otherwise, which we then interacted with the linear term. This finding is similar to Donmez and Karaivanov (2021), who tested the impact of congestion on gas price for a shorter observation period.

To establish robustness, we ran a series of alternative models to those reported in Table 16. We used the transaction count instead of gas demand as an alternative dependent variable and computed our models with different operationalizations of the market gas price. Table 16 presents the models with the average of the 25th percentiles of the gas price paid in blocks on a given day and the average gas price on a given day. Moreover, we normalized the gas price by the total supply of Ether to account for inflation. We also used different specifications of our instrumental variable. Table 16 shows our estimate of the demand curve using the difference between the number of blocks we expect based on the target block time and the actual number of blocks observed per day. We also studied various subsamples. Column 7 is the demand curve after we winsorized gas demand to the 5th and 95th percentile. Column 8 shows the results of limiting our study period to the final difficulty bomb. The reduced sample size could explain the insignificant result for this model but our robustness check results resemble our main results regarding magnitude and significance.

Table 16: Robustness checks (network level)

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	Baseline	Alternative Dependent variable	Alternative market gas price (25th percentile)	Alternative market gas price (average gas price)	Alternative market gas price (normalize by ETH supply)	Alternative instrument (block difference)	Subsample (5th-95th percentile gas used)	Subsample (specific difficulty bomb period)
	log(Gas used)	log(Transaction count)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)
log(Market gas price)	-0.69*** (0.16)	-0.63*** (0.15)	-0.80*** (0.20)	-1.83** (0.61)	-0.57 (0.14)	-0.75** (0.24)	-0.69** (0.19)	-2.70 (2.95)
Observations	1,279	1,279	1,279	1,279	1,279	1,279	1,279	101

Note: Heteroskedasticity and autocorrelation consistent (HAC) standard errors are shown in parentheses. *p<0.1; **p<0.05; ***p<0.01

This analysis provides initial empirical evidence that the well-established "law of demand" (Gale, 1955) applies to the verification of transactions on Ethereum. It also affirms that Ethereum's gas price mechanism introduces a form of price competition among transaction senders that counteracts the main prediction in the two-sided market literature (Katz and Shapiro 1985), that, due to the same-side network effect, an increase in the demand-side draws even more consumers to the market and subsequently increases demand. On Ethereum, a greater number of transaction senders not only increases the utility of transacting on Ethereum but also price competition. However, as the demand for gas is negatively associated with its price, the market mechanism underlying Ethereum's transaction verification process dampens the effectiveness of same-side network effects.

5.7.2 Differing Demand Curves per Group

We also estimated a specific demand curve for every group of dApps along with their confidence intervals. Table 17 shows the second stage results of this estimate. Each model takes the aggregated daily gas used by all dApps within the respective group as the dependent variable. Columns 2-6 show that the coefficients of log(*Market gas price*) vary significantly between the groups of dApps, indicating that they differ regarding their sensitivity to changes in the gas price.

Table 17: 2SLS models by group (network level)

	(1)	(2)	(3)	(4)	(5)	(6)
	2SLS *2nd stage*	2SLS *2nd stage*	2SLS *2nd stage*	2SLS *2nd stage*	2SLS *2nd stage*	2SLS *2nd stage*
	log(Gas used by all dApps)	log(Gas used by group 1)	log(Gas used by group 2)	log(Gas used by group 3)	log(Gas used by group 4)	log(Gas used by group 5)
log(Market gas price)	-0.45*** (0.14)	-0.29* (0.16)	0.09 (0.19)	-2.09*** (0.63)	-0.59*** (0.13)	-0.48*** (0.17)
Network utilization	-1.04*** (0.36)	-0.27 (0.41)	-0.84 (0.61)	-2.37 (1.67)	-0.91* (0.48)	-1.05** (0.51)
Network utilization^{2}	6.61*** (2.25)	2.51 (2.58)	2.89 (3.60)	17.04* (10.24)	5.44* (2.81)	7.20** (3.04)
log(Ether price)	0.20** (0.08)	0.39*** (0.08)	0.03 (0.09)	-0.02 (0.23)	-0.93*** (0.09)	0.37*** (0.10)
log(Ether volatility)	-0.0000 (0.01)	0.01 (0.01)	-0.02 (0.02)	-0.005 (0.03)	0.02 (0.02)	-0.02 (0.01)
log(Gas limit)	2.49*** (0.92)	1.56 (1.05)	-0.75 (1.07)	7.61*** (2.28)	1.88** (0.86)	2.68*** (0.91)
$D^{Thursday}$	-0.03 (0.02)	-0.02 (0.02)	0.02 (0.04)	-0.12 (0.08)	-0.05* (0.03)	-0.09** (0.04)
D^{Friday}	0.01 (0.02)	0.01 (0.02)	-0.04 (0.04)	0.03 (0.07)	-0.02 (0.03)	-0.13*** (0.04)
$D^{Wednesday}$	-0.002 (0.02)	0.004 (0.01)	-0.02 (0.03)	-0.06 (0.05)	-0.03 (0.02)	-0.06* (0.04)
D^{Monday}	-0.02 (0.02)	-0.01 (0.02)	-0.03 (0.04)	-0.10 (0.07)	-0.06** (0.03)	-0.12*** (0.03)
$D^{Saturday}$	-0.04 (0.03)	-0.07*** (0.03)	-0.09** (0.04)	0.13* (0.07)	-0.06* (0.03)	-0.13*** (0.05)
D^{Sunday}	-0.04 (0.02)	-0.08*** (0.02)	-0.08* (0.05)	0.14* (0.07)	-0.07** (0.03)	-0.13*** (0.05)
D^{2018}	-1.25*** (0.28)	-1.36*** (0.35)	-0.26 (0.31)	-1.29 (1.15)	-0.66** (0.28)	-0.23 (0.30)
D^{2019}	-1.53*** (0.32)	-1.80*** (0.40)	-0.23 (0.38)	-1.69 (1.43)	-0.41 (0.35)	0.22 (0.38)
D^{2020}	-1.35*** (0.38)	-1.61*** (0.42)	-0.29 (0.44)	-1.90 (1.35)	-0.34 (0.40)	1.37*** (0.42)
Trend	0.002*** (0.0004)	0.003*** (0.0005)	-0.001** (0.001)	0.002 (0.001)	0.0004 (0.001)	-0.003*** (0.001)
Constant	-0.03 (10.36)	-18.54 (12.14)	35.66** (14.67)	16.61 (30.89)	24.97* (13.23)	83.40*** (12.13)
Observations	1,279	1,279	1,279	1,279	1,279	1,279
C-D Wald F Stat.	85.06					
Stock-Yogo Critical Value	16.38					
Kleibergen-Paap LM Stat.	4.19**					

Note: Heteroskedasticity and autocorrelation consistent (HAC) standard errors are shown in parentheses, where optimal bandwidth (23) is calculated in line with Newey and West (1987). All models use the first stage regression reported in Table 4.

*p<0.1; **p<0.05; ***p< 0.01

To compare gas price elasticities, we computed their 95 percent confidence intervals. Figure 30 presents these intervals, showing that not all elasticities can be distinguished with sufficient confidence, but significant differences are noticeable. Especially games and marketplaces (group 3) seem to be far more sensitive to changes in gas prices than the dApps in groups 1 and 2. Considering that group 3 comprises collectible games such as crypto kitties, where the timing of the transaction does not matter as much as for finance or cryptocurrency exchange dApps, whose cryptocurrency prices change rapidly, this result seems plausible. The one-time nature and relatively high transaction values in group 2 (identity and property dApps) can explain why users are relatively insensitive to changes in the gas price.

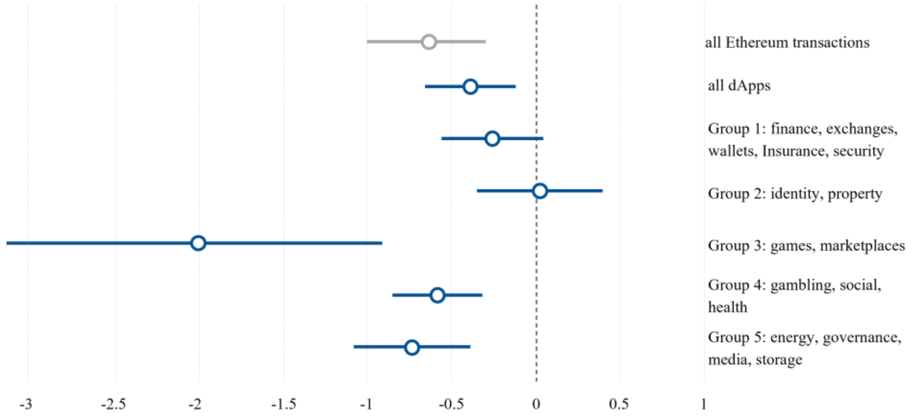


Figure 30: Price elasticities of demand per group of dApps

These findings show that dApps differ regarding sensitivity towards gas fees, as their users differ in their willingness to pay for transactions with different groups of dApps. Given that all dApps—irrespective of what service they offer—compete for the limited supply of gas, Ethereum's gas price mechanism harms cross-side network effects. That is, if additional transaction senders join the platform in response to more complements entering it, the difference in gas price sensitivity can mean some dApps will be used less and thus not benefit but actually lose users despite otherwise positive network effects.

5.7.3 Baseline dApp-level results

Similar to the baseline analysis of our network-level data, we used a 2SLS estimator and Ethereum's difficulty bomb to estimate the demand curve for our panel specification. In addition to all the other control variables in the preceding model, the panel specification allowed us to add individual-level fixed effects for each dApp. Table 18 shows the results of our estimates.

Table 18: Results of demand curve estimates: baseline model (dApp level)

	(1)	(2)	(3)
	log(Market gas price)	log(Gas used)	log(Gas used)
Difficulty bomb	0.20*** (0.0000)		
log(Market gas price)		-0.64*** (0.21)	0.27*** (0.05)
log(Ether price)	-0.0004 (0.01)	0.15*** (0.04)	0.18*** (0.04)
log(Ether volatility)	-0.01*** (0.0004)	0.01** (0.004)	0.02*** (0.003)
Network utilization	-2.36*** (0.06)	-1.20** (0.47)	0.30*** (0.11)
Network utilization^{2}	16.30*** (0.37)	8.59*** (3.29)	-1.89*** (0.68)
log(Gas limit)	2.40*** (0.03)	1.89*** (0.53)	0.13 (0.20)
Age	0.001*** (0.0000)	-0.002*** (0.0003)	-0.002*** (0.0002)
Year^{2018}	-0.82*** (0.02)	-0.68*** (0.22)	-0.09 (0.15)
Year^{2019}	-1.09*** (0.02)	-0.66*** (0.25)	0.07 (0.15)
Year^{2020}	-0.95*** (0.02)	-0.28 (0.24)	0.36** (0.16)
weekday^{Thursday}	-0.02*** (0.001)	-0.03*** (0.01)	-0.01* (0.01)
weekdays^{Friday}	0.02*** (0.001)	-0.02** (0.01)	-0.03*** (0.01)
weekdays^{Wednesday}	-0.005*** (0.001)	-0.001 (0.01)	0.002 (0.01)
weekdays^{Monday}	-0.02*** (0.001)	-0.03*** (0.01)	-0.02** (0.01)
weekdays^{Saturday}	0.01*** (0.002)	-0.07*** (0.01)	-0.08*** (0.01)
weekdays^{Sunday}	0.01*** (0.002)	-0.08*** (0.01)	-0.09*** (0.01)
log(Market gas price) × group 2			-0.43*** (0.15)
log(Market gas price) × group 3			-0.64*** (0.12)
log(Market gas price) × group 4			-0.49*** (0.10)
log(Market gas price) × group 5			-0.28*** (0.09)
Observations	370,392	370,392	370,392
R^{2}	0.78		0.11
Incremental F	121.39		
C-D Wald F Stat.		2542.47	118.07
Stock-Yogo Critical Value		16.38	26.87
Kleibergen-Paap LM Stat.		70.04***	25.16***

Note: Heteroskedasticity and autocorrelation consistent (HAC) standard errors are shown in parentheses. Interacted and squared variables are centered beforehand.

*p<0.1; **p<0.05; ***p<0.01

Regarding magnitude and significance, the baseline results from the first and second stages of our 2SLS estimate (see Columns 2 and 3) resemble our results at the aggregated network level. However, when we added the group of dApps as an interaction to the gas price (log(*Market gas price*)), we obtained different results for the reference category’s demand curve (group 1: finance, exchanges, wallets, insurance, security). With a positive and significant coefficient (0.27), our results suggest that the demand curve for this group of dApps is upwards-sloping. One explanation is that the entry of additional finance-related dApps caused an influx of high willingness-to-pay customers and that the network effects these finance-related dApps could achieve compensated for the higher transaction fees these transaction senders had to pay. This explanation is in line with prior research that describes networked goods (e.g., financial services) as having irregularities such as an upward-sloping demand curve for low quantity levels (Economides & Himmelberg, 1995). Particularly if a service relies on strong network effects, no one will pay for the product if no one else is using it. Although high willingness-to-pay users are typically beneficial for a platform, the fact that we observed downwards-sloping demand curves in the form of negative moderations of all other groups (see Table 7, Column 3) poses a risk, particularly in times of high transaction fees, that dApps from other groups are no longer used and finally have to leave the platform. This reduction in complement heterogeneity can ultimately harm the long-term attractiveness of Ethereum, especially as a general-purpose platform.

5.7.4 Heterogeneous effect of Ethereum gas price mechanism

We used our rich data to explore further the characteristics of dApps that impact their sensitivity towards the gas price. The first set of characteristics pertains to the formal requirements for transacting with a dApp. These characteristics are the amount of gas a transaction with a dApp requires and the typical value of Ether and tokens. To analyze these characteristics, we computed the total average for all these variables over every transaction a dApp received. Because this average is time-invariant, we interacted these variables with the gas price and groups in different models. In Table 19, Columns 1 and 4 show the two-way and three-way interaction models for the average gas requirement; Columns 2 and 5 show the interaction models with the average Ether value sent; and Columns 3 and 6 the models with the average token value sent.

Table 19: Interactions with transaction requirements (dApp level)

	(1)	(2)	(3)	(4)	(5)	(6)
	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)
log(Market gas price)	-0.66*** (0.21)	-0.64*** (0.21)	-0.73*** (0.21)	-0.59** (0.26)	-0.62** (0.27)	-0.82*** (0.30)
log(Market gas price) × log(Average gas requirement)	-0.06 (0.04)			0.02 (0.05)		
log(Market gas price) × log(Average value sent USD)		0.14*** (0.04)			0.15** (0.06)	
log(Market gas price) × log(Average token value sent USD)			0.31*** (0.04)			0.40*** (0.09)
log(Market gas price) × group 2				-0.17 (0.17)	-0.08 (0.18)	0.10 (0.20)
log(Market gas price) × group 3				-0.28* (0.15)	-0.24 (0.15)	0.03 (0.16)
log(Market gas price) × group 4				-0.17 (0.14)	-0.15 (0.14)	0.09 (0.16)
log(Market gas price) × group 5				0.04 (0.14)	0.09 (0.14)	0.23 (0.16)
log(Market gas price) × log(Average gas requirement) × group 2				-0.58*** (0.16)		
log(Market gas price) × log(Average gas requirement) × group 3				-0.24** (0.11)		
log(Market gas price) × log(Average gas requirement) × group 4				-0.19* (0.10)		
log(Market gas price) × log(Average gas requirement) × group 5				-0.003 (0.08)		
log(Market gas price) × log(Average value sent USD) × group 2					-0.25 (0.17)	
log(Market gas price) × log(Average value sent USD) × group 3					0.18 (0.16)	
log(Market gas price) × log(Average value sent USD) × group 4					-0.02 (0.08)	
log(Market gas price) × log(Average value sent USD) × group 5					-0.10 (0.11)	
log(Market gas price) × log(Average token value sent USD) × group 2						-0.19 (0.15)
log(Market gas price) × log(Average token value sent USD) × group 3						-0.05 (0.13)
log(Market gas price) × log(Average token value sent USD) × group 4						-0.21 (0.14)
log(Market gas price) × log(Average token value sent USD) × group 5						-0.24** (0.11)
Controls	YES	YES	YES	YES	YES	YES
log(Ether volatility)	0.01* (0.004)	0.01** (0.004)	0.01* (0.004)	0.01* (0.004)	0.01* (0.004)	0.01* (0.004)
Network utilization	-1.24*** (0.47)	-1.18** (0.47)	-1.31*** (0.48)	-1.27*** (0.48)	-1.25*** (0.48)	-1.34*** (0.49)
Network utilization^{2}	8.87*** (3.30)	8.48** (3.30)	9.37*** (3.36)	9.06*** (3.32)	8.96*** (3.36)	9.54*** (3.40)
log(Gas limit)	1.94*** (0.53)	1.88*** (0.53)	1.95*** (0.54)	1.95*** (0.54)	1.94*** (0.54)	1.99*** (0.55)
Age	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)
Year dummies	YES	YES	YES	YES	YES	YES
Weekday dummies	YES	YES	YES	YES	YES	YES

Note: Heteroskedastic and autocorrelation consistent (HAC) standard errors are shown in parentheses. Interacted and squared variables are centered beforehand. *p<0.1; **p<0.05; ***p<0.01

Regarding the gas requirement for transacting with a dApp, we did not find a significant two-way interaction effect between gas price and average gas requirement (Column 1), but significant three-way interactions between gas price, gas requirement, and groups 2, 3, and 4

(Column 4). These interactions indicate that for some groups of dApps, the two-way interaction differs significantly from the reference category (group 1). For instance, for gambling dApps, the negative coefficient of the three-way interaction (-0.24) implies that the negative impact of the gas price on gas demand is even stronger if the gambling dApp demands a high amount of gas for a transaction. On the other hand, for dApps in group 2, the coefficient of the three-way interaction is positive (0.58). This implies that, compared to the identity and property dApps in group 1, a high gas requirement somewhat counteracts the downward slope of the demand curve, reducing the sensitivity towards changes in the gas price. One possible explanation is the required frequency of interaction with a dApp. Unlike gambling and finance applications, where users obtain utility from regular interaction with dApps, identity and property dApps only require sporadic transactions. If a property dApp bundles more functionality into one transaction, not only the gas requirement but also the utility of the transaction increase. Accordingly, the user might be willing to accept high gas prices for this transaction as the additional gas fees become less relevant in relation to the one-time transaction effort. For users of gambling and finance applications, who benefit through more frequent interactions, greater functionality in a single transaction might increase the utility but, in the long run, also pile up more transaction fees as the additional function will be computed over and over again, not just a few times like with property dApps. Thus, users might be less inclined to comply with higher gas requirements as they prefer less complex, but dedicated functions realized through singular transactions. Another explanation is that as for example gambling dApps require frequent interaction, there is more pressure on such dApps to make their smart contracts more efficient in terms of gas requirement.

Regarding the average value (in Ether or other tokens) sent with a transaction to a dApp, we found a positive moderation of the negative demand curve (Columns 2 and 3). The positive interaction coefficients between the gas price and average Ether value (0.14) and token value (0.31), combined with the negative linear coefficient of the gas price (-0.64 and -0.74) are indicators that the gas price elasticity of dApps decreases with a higher average transaction value. This finding is in line with prior studies that showed users' fee sensitivity declined with the transaction value (e.g., Wang & Wright, 2017).

Regarding the three-way interactions (*log(Market gas price) x Average value or token value X group*), we found that only one out of eight coefficients is significant. This indicates that, apart from group 5, the positive and significant interaction of the transaction value with gas price does not differ across the groups of dApps and suggests that dApps receiving a higher average transaction value have a less elastic demand curve.

Table 20: Interactions with average performance indicators (dApp level)

c	(1)	(2)	(3)	(4)	(5)	(6)
	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)
log(Market gas price)	-0.67*** (0.21)	-0.68*** (0.21)	-0.64*** (0.21)	-0.81*** (0.29)	-0.81*** (0.29)	-0.59** (0.26)
log(Market gas price) × log(Average daily transactions)	0.16*** (0.06)			0.39*** (0.08)		
log(Market gas price) × log(Average daily EOA)		0.21*** (0.06)			0.39*** (0.07)	
log(Market gas price) × log(Average transactions per EOA)			-0.03 (0.04)			0.02 (0.06)
log(Market gas price) × group 2				0.08 (0.19)	0.06 (0.19)	-0.02 (0.15)
log(Market gas price) × group 3				-0.12 (0.15)	-0.13 (0.15)	-0.33** (0.15)
log(Market gas price) × group 4				0.01 (0.16)	0.02 (0.16)	-0.16 (0.14)
log(Market gas price) × group 5				0.22 (0.15)	0.22 (0.15)	0.06 (0.14)
log(Market gas price) × log(Average daily transactions) × group 2				-0.51*** (0.17)		
log(Market gas price) × log(Average daily transactions) × group 3				-0.64*** (0.14)		
log(Market gas price) × log(Average daily transactions) × group 4				-0.47*** (0.13)		
log(Market gas price) × log(Average daily transactions) × group 5				-0.45*** (0.11)		
log(Market gas price) × log(Average daily EOA) × group 2					-0.28* (0.16)	
log(Market gas price) × log(Average daily EOA) × group 3					-0.55*** (0.13)	
log(Market gas price) × log(Average daily EOA) × group 4					-0.38** (0.15)	
log(Market gas price) × log(Average daily EOA) × group 5					-0.46*** (0.11)	
log(Market gas price) × log(Average transactions per EOA) × group 2						-0.46*** (0.10)
log(Market gas price) × log(Average transactions per EOA) × group 3						-0.28** (0.12)
log(Market gas price) X log(Average transactions per EOA) × group 4						-0.12 (0.08)
log(Market gas price) × log(Average transactions per EOA) × group 5						0.03 (0.10)
log(Ether price)	0.15*** (0.04)	0.15*** (0.04)	0.15*** (0.04)	0.14*** (0.04)	0.15*** (0.04)	0.15*** (0.04)

log(Ether volatility)	0.01** (0.004)	0.01* (0.004)	0.01** (0.004)	0.01 (0.004)	0.01 (0.004)	0.01* (0.004)
Network utilization	-1.22** (0.48)	-1.25*** (0.48)	-1.21** (0.47)	-1.40*** (0.50)	-1.42*** (0.50)	-1.28*** (0.48)
Network utilization^{2}	8.73*** (3.34)	8.92*** (3.36)	8.66*** (3.29)	10.02*** (3.48)	10.16*** (3.51)	9.12*** (3.33)
log(Gas limit)	1.88*** (0.53)	1.89*** (0.53)	1.90*** (0.53)	2.08*** (0.55)	2.10*** (0.56)	1.95*** (0.54)
Age	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)	-0.002*** (0.0003)
Year dummies	YES	YES	YES	YES	YES	YES
Weekday dummies	YES	YES	YES	YES	YES	YES

Note: Heteroskedasticity and autocorrelation consistent (HAC) standard errors are shown in parentheses. Interacted and squared variables are centered beforehand.

*p<0.1; **p<0.05; ***p<0.01

Along with the requirements for transacting with a dApp, we also computed average performance indicators for each dApp. Table 20 reports the interaction result regarding the average daily number of transactions, the average daily number of EOA, and the average daily transactions per EOA. For the average daily transactions and average daily EOA, we found a positive and significant two-way interaction with the gas price. This suggests that the demand for gas for transactions with dApps with a high average of daily transactions and users is less impacted by changes in the gas price. However, by adding the group dummies to these two-way interactions, we found that this interaction differs significantly between dApps in group 1 and all other groups. Whereas dApps in group 1 still seem to benefit from a higher level of average transactions and EOAs—as indicated by the positive and significant two-way interactions between gas price and average number of transactions (Column 4, 0.39) and the average number of daily EOA (Column 5, 0.39)—the three-way interactions with all other groups are highly significant and negative. This indicates that the effect of receiving, on average, more transactions or having more unique EOAs transacting with these dApps is less prevalent or even makes them more sensitive to changes in the gas price. Again, network effects are a plausible explanation. Particularly finance dApps and cryptocurrency exchange dApps should benefit greatly from network effects. A gas price increase caused by an influx of additional users could be compensated by the additional benefit the growing number of users provides to finance and exchange dApps. At the same time, because dApps from other groups (e.g., property, gambling, identity, or storage) benefit less from network effects, they cannot compensate for the additional gas fees their users would have to pay to transact with them. Especially for dApps that already have a high average number of users but do not benefit from network effects, this can increase the sensitivity towards the gas price and reduce the demand for transactions—especially at times when gas supply is lower and price competition is fierce. For the average number of transactions per EOA (Columns 3 and 6), we only obtained a few significant results that did not allow us to infer systematic patterns.

To further investigate network effects, we analyzed the impact of dynamic usage indicators that vary for each dApp over time. Table 21 shows the interaction results of the daily ratio of transactions per EOA and the average price users were willing to pay above the market gas price. Regarding the number of transactions per EOA, we found a positive interaction (0.08, Column 2) between the number of transactions per EOA and the gas price (*log(Market gas price)*). According to the three-way interactions, except for group 5, this moderation does not vary significantly between the different groups of dApps. Because the interaction is even stronger in group 5 than for all other dApps, attracting heavy users might be a valid strategy to survive the competition in a transaction market. Considering that group 5 comprises dApps such as storage or energy services and the typically strong lock-in effects of these services, our findings seem plausible.

Table 21: Interactions with usage indicators (dApp level)

	(1)	(2)	(3)	(4)	(5)	(6)
	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)
log(Market gas price)	-0.44** (0.17)	-0.43** (0.17)	-0.38* (0.22)	-0.66*** (0.21)	-0.71*** (0.22)	-0.71** (0.29)
log(Transactions per EOA)	1.27*** (0.03)	1.28*** (0.04)	1.17*** (0.06)			
log(Market gas price) × log(Transactions per EOA)		0.08*** (0.03)	0.08** (0.04)			
log(Market gas price) × group 2			-0.15 (0.17)			-0.002 (0.19)
log(Market gas price) × group 3			-0.28** (0.13)			-0.15 (0.16)
log(Market gas price) × group 4			-0.21* (0.12)			-0.02 (0.16)
log(Market gas price) × group 5			0.05 (0.12)			0.12 (0.15)
log(Transactions per EOA) × group 2			-0.03 (0.15)			
log(Transactions per EOA) × group 3			0.35*** (0.08)			
log(Transactions per EOA) × group 4			0.01 (0.09)			
log(Transactions per EOA) × group 5			0.17 (0.11)			
log(Market gas price) × log(Transactions per EOA) × group 2			-0.13 (0.15)			
log(Market gas price) × log(Transactions per EOA) × group 3			-0.001 (0.07)			
log(Market gas price) X log(Transactions per EOA) × group 4			-0.02 (0.05)			
log(Market gas price) × log(Transactions per EOA) × group 5			0.16*** (0.06)			
log(Surplus gas price paid)				0.08*** (0.03)	-0.14*** (0.04)	-0.07 (0.07)
log(Surplus gas price paid) × log(Market gas price)					0.16*** (0.02)	0.16*** (0.03)
log(Surplus gas price paid) × group 2						-0.39*** (0.11)
log(Surplus gas price paid) × group 3						-0.35*** (0.11)
log(Surplus gas price paid) × group 4						-0.18 (0.11)
log(Surplus gas price paid) × group 5						0.14 (0.11)
log(Market gas price) × log(Surplus gas price paid) × group 2						0.11** (0.05)

log(Market gas price) × log(Surplus gas price paid) × group 3						0.09* (0.05)
log(Market gas price) × log(Surplus gas price paid) × group 4						-0.05 (0.05)
log(Market gas price) × log(Surplus gas price paid) × group 5						-0.18*** (0.05)
log(Ether price)	0.18*** (0.03)	0.18*** (0.03)	0.18*** (0.03)	0.14*** (0.04)	0.14*** (0.04)	0.13*** (0.04)
log(Ether volatility)	0.005 (0.003)	0.005 (0.003)	0.004 (0.003)	0.004 (0.005)	0.0003 (0.01)	-0.001 (0.01)
Network utilization	-0.73* (0.38)	-0.72* (0.38)	-0.82** (0.39)	-1.15** (0.46)	-1.41*** (0.48)	-1.48*** (0.49)
Network utilization^2	5.45** (2.66)	5.38** (2.67)	6.07** (2.72)	8.23** (3.20)	10.20*** (3.38)	10.67*** (3.44)
log(Gas limit)	1.39*** (0.45)	1.39*** (0.45)	1.47*** (0.46)	1.71*** (0.48)	1.85*** (0.49)	1.88*** (0.50)
Age	- 0.001*** (0.0002)	- 0.001*** (0.0002)	- 0.001*** (0.0002)	- 0.002*** (0.0003)	- 0.002*** (0.0003)	- 0.002*** (0.0003)
Year dummies	YES	YES	YES	YES	YES	YES
Weekday dummies	YES	YES	YES	YES	YES	YES

Note: Heteroskedasticity and autocorrelation consistence (HAC) standard errors are shown in parentheses. Interacted and squared variables are centered. *p<0.1; **p<0.05; ***p<0.01

Regarding the average surplus gas price that transaction senders are willing to pay on a given day for transacting with a dApp, we also observed a positive interaction with gas price (0.16, Column 5). Again, except for group 5, this moderation retains roughly the same direction and magnitude across the different groups.

The three-way interaction is only negative in group 5, implying that, compared to the dApps in group 1, group 5 dApps are more sensitive to changes in gas price when users overpay the market price. These could be periods with high fluctuations that expose users to high uncertainty about the gas price, forcing them to overpay for a certain inclusion of their transaction. One explanation for the negative three-way interaction is that users in this group are more sensitive to uncertainty related to overpaying and thus react by becoming more price sensitive.

5.7.5 Additional robustness checks

To assess the robustness of our panel specifications, we tested them against alternative measurements and samples. For example, we used the transaction count instead of gas used, different winsorization levels to restrict the impact of potential outliers, different percentile and winsorization levels for the market gas price together with the average gas price, and another measurement of the difficulty bomb, subtracting the observed number of blocks from the target number . We only conducted our analysis for the periods when the difficulty bomb was active. Table 22 reports the coefficients we obtained from the robustness tests. The results were consistent with our baseline specification.

Table 22: Robustness checks (dApp level data)

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
	Baseline	Alternative Dependent variable	Alternative market gas price (25th percentile)	Alternative market gas price (average gas price)	Alternative instrument (block difference)	Outliers (5th-95th percentile gas used)	Subsample (specific difficulty bomb period)
	log(Gas used)	log(Transaction count)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)	log(Gas used)
log(Market gas price)	-0.64*** (0.21)	-0.42** (0.19)	-0.57*** (0.18)	-0.82*** (0.26)	-1.03** (0.45)	-0.58*** (0.20)	-1.48* (0.87)
Observations	370,392	370,392	370,392	370,392	370,392	370,392	35,756

Note: Heteroskedasticity and autocorrelation consistent (HAC) standard errors are shown in parentheses. *p<0.1; **p<0.05; ***p<0.01

5.8 Additional analysis (survival analysis)

To investigate the impact of Ethereum's transaction verification mechanism on platform complements' heterogeneity, we examined our explanatory variables' simultaneous effect on the overall hazard-rate function using the semi-parametric Cox proportional-hazards regression analysis (Cox, 1972). Scholars previously used Cox-proportional hazard models to study market exit or entry (e.g., Agarwal & Gort, 2002; Huang, Ceccagnoli, Forman, & Wu, 2013). For our benchmark specification, we estimated the hazard of dApp d leaving the market on day t as:

$$h_{dt} = h_o(t)\ \exp\{\beta'_x x_t\}$$

where h_0(t) is the baseline hazard, $x_t$is a vector of explanatory and control variables pertaining to time t. With this model, we were not interested in predicting the exit time but the effect of gas price as a time-dependent covariate. For our analysis, we stratified our observations by the group of dApps. This allowed us to account for their different baseline hazard rates. To measure market exit, we leveraged the fact that stateofthedapps.com reports the status of dApps and classifies discontinued dApps as "abandoned." For the exact timing of the market exit, we took the date of the last transaction a dApp received. Table 23 presents the results of our analysis. Column 1 shows our benchmark specification and column 2 the gas price interacted with the group of dApps.

Table 23: Survival models

	(1)	(2)
	all dApps	all dApps
	stratified by group	stratified by group
log(Market gas price)	0.02 (0.09)	-1.7* (0.11)
log(Market gas price) × group 2		0.49** (0.23)
log(Market gas price) × group 3		0.15 (0.10)
log(Market gas price) × group 4		0.21** (0.09)
log(Market gas price) × group 5		0.22* (0.12)
Network utilization	-6.68 (8.24)	-6.89 (8.18)
Network utilization^2	4.01 (5.32)	4.15 (5.28)
log(Ether price)	-0.04 (0.14)	-0.02 (0.14)
log(Ether volatility)	0.01 (0.04)	0.01 (0.04)
log(Gas limit)	1.07 (0.71)	1.11 (0.71)
Year of entry dummies	YES	YES
Observations	783,619	783,619
Market exit events	399	3991
Log-likelihood	-2,088.394	-2,083.793

Note: Hazard ratios can be calculated by exponentiating the coefficients reported for each variable.

*p<0.1; **p<0.05; ***p<0.01

Our benchmark specification shows gas price has no significant impact on a dApp's survival. However, after interacting the gas price with the group of dApps (Column 2), we found that a 10 percent increase in the Market price (~0.095 increase in log(Market price) is associated with a reduced hazard rate (β = -1.7; hazard rate = exp(0.095*-1.7) = 0.851) of around 16.9 percent for our base category (group 1, finance dApps). The positive and (except for group 3) significant interactions indicate that all other groups of dApps benefit less from a higher gas price and face a higher likelihood of market exit. For instance, the reduced hazard rate for group 2 equals 10.9% (exp((-1.7 + 0.49)*0.095) = 0.891).

Our hazard model results suggest that an increase in the market gas price reduces the likelihood of a market exit on a given day, but groups differ significantly regarding this effect. Given that the gas price fluctuates rapidly, sometimes doubling or even tripling within a month (e.g., in January 2018, June 2020 at the start of the Defi hype), these results can be economically significant. The results seem plausible as an increase in gas price is typically on account of the increased demand for gas caused by more transaction activity with dApps. Again we can see that dApps in group 1 benefit more from this effect than other dApps and thus have an overall higher likelihood of staying in this market. This differentiating effect is problematic as it corroborates our main argument by showing that a transaction market disproportionately favors a specific type of dApp, leading to a long-run reduction in the heterogeneity of dApps offered on the Ethereum platform.

5.9 Conclusion

Decentralized blockchain platforms like Ethereum have been hailed for challenging the current dominance of centralized digital platforms in the digital economy (Murray et al., 2019; Vergne, 2020). Yet, little is known about how the decentralized transaction verification mechanism, which distinguishes blockchain platforms from their centralized counterparts, impacts platform performance by determining its usage and complements. To investigate this issue, we studied Ethereum's transaction verification mechanism as a market for transactions and used a panel data set of 1,590 dApps together with a novel supply-side instrument to estimate various price elasticities in the demand for transactions with dApps. We found strong evidence that Ethereum's current gas price mechanism leads to negative network effects (an increase in transaction demand makes transacting more expensive) that counteract the usually positive network effects on multi-sided platforms. Furthermore, we found that the relative magnitude of these effects depends on the characteristics of a dApp that are mostly predetermined. Particularly the type and complexity of the service a dApp offers are decisive factors. Across the board, the demand for transactions with finance or exchange dApps seems to be less impacted by changes in the gas price than dApps that offer games, gambling, social, or media-related services. This is especially problematic as the transaction verification mechanism adds a new externality to the existing competition on such platforms: all dApps—no matter what service they offer—must compete for the limited gas supply. Hence, it favors some dApps over others and ultimately forces disadvantaged dApps to leave the platform leading to a decrease in the heterogeneity of dApps offered on Ethereum and a reduced value for platform users who joined because of the variety of complements offered on the platform.

The main contribution of this work is to unpack the consequences of using a market mechanism instead of a central authority to allocate transactions for the dApps offered on a blockchain platform. Our results have several important implications for platform providers, complementors, and policymakers. For platform providers: as we found that the type of service and its complexity determine a dApp's sensitivity towards gas prices and thus its likelihood of entry or exit, platform providers have to consider these discriminatory effects when designing the transaction verification mechanism. Especially because the decentralized nature of blockchain platforms restricts their strategic toolset for orchestrating complements, such as entry restrictions or other means of prioritization, a transaction verification mechanism has to be designed carefully and align with the platform strategy. Carelessly expanding the complementor side (e.g., by promoting complementors to join the platform) in the hope that it naturally benefits the platform's performance, might be detrimental to the platform's long-term goals. Our analysis provides a case in point, as it shows that the current version of

Ethereum’s gas price mechanism favors finance and exchange dApps over others, thus contradicting Ethereum’s vision of becoming a general-purpose platform that caters to all sorts of dApps. Furthermore, it questions whether platforms with similar transaction verification mechanisms are viable options for Web3.0.

Regarding the implications for complementors: in a market for transactions, platform complementors not only need to pay attention to their direct competition but also carefully analyze the current and future congestion of the network and consider their own sensitivity towards gas fees compared to all other dApps on the platform. As our analysis shows that the gas required for transaction with a dApp is another important determinant of gas price elasticity, dApp providers need to consider how to bundle or split interactions with the dApp into one or multiple transactions.

Finally, regarding policymakers: policymakers and regulators are frequently concerned about antitrust competition between platforms. From this perspective, the reduced heterogeneity of complements on one platform might be desirable: it gives rise to other platforms more closely tailored to the complements’ needs and thus reduces the likelihood of one platform dominating the entire industry. Although the general impact of the transaction verification mechanism through creating multiple other platforms is beyond the scope of this paper, our results need to be considered in the regulatory process. A transaction verification mechanism like Ethereum’s could be a self-regulation tool for mitigating the “winner-takes-it-all” associated with digital platforms that rely heavily on network effects.

This paper has limitations that open up opportunities for further research. One limitation is that we only observed one platform. Even though our analysis suggests that the gas price mechanism on Ethereum might cause complementors to leave the network and join other platforms, we do not address cross-platform competition and substitution patterns. A natural extension of our work would be to analyze other blockchain platforms offering dApps and study platform complements' switching and multi-homing behavior. Another limitation is our sample of dApps and their associated smart contracts. Although we tried to include as many dApps as possible in our analysis and even manually matched smart contracts to these dApps, more dApps are running on Ethereum than our sample reflects. Especially dApps that are only accessible through Chinese websites might have slipped our attention and are not represented in our sample. Although our sample accounts for as much as 85 percent of all Ethereum transactions in some periods, our results should be seen as initial empirical evidence and would benefit from replications incorporating a different set of dApps or a more fine-grained perspective on the rich data available. Potentially promising ideas are zooming in on single days and tracing individual users’ bidding behavior or studying a dApp’s usage pattern in light of changing gas prices. Finally, due to this field’s infancy and rapid development, our results should be treated as preliminary and could be reevaluated after major protocol updates. One such change is Ethereum’s long-announced update from PoW to PoS. We predict this update will only eliminate the computationally expensive puzzle of finding a hash that fulfills properties required by the protocol, not the transaction’s computation and verification. Consequently, the gas price mechanism could become even more important as major driver of the cost to verify transactions. It would therefore be interesting to see how validators prioritize transactions and influence the use of dApps after the PoS update.

6 Summary and Outlook

Blockchains are more than just secure data bases. Their underlying technology has the potential to induce a paradigm shift in our digital economy by offering a new blueprint for decentralized and distributed digital platforms that outperform their centralized counterparts in transparency, inclusion, and democracy. But to achieve this vision, we first need to understand all the pitfalls and boundary conditions surrounding the new technology. This thesis sheds light on two important promises made by blockchain technology proponents and seeks to provide a better understanding of how and to what extent these promises will help reshape our current platform economy. These promises are to create supposedly trust-free systems, and disintermediate platforms by substituting a central authority with a decentralized market mechanism that ensures the verification and automated execution of transactions.

As there is hardly any empirical research on this subject, I chose dApps on Ethereum as the context of this dissertation and investigated how the promises impact the use of dApps. Ethereum lends itself as preferred empirical context because it was the first blockchain platform to offer smart contracts and dApps. It enabled a more versatile use of blockchain platforms beyond mere cryptocurrency transfers and has thus served as role model for many smart contract-enabling platforms.

In the first study, I theorize that smart contracts on a blockchain only theoretically have the power to remove the need for trust in transactions. In practice, they are unlikely to do so because their users would have to fully read and understand the smart contract before transacting with it. I argue that smart contracts enable a new way to form trust based on the possibility of reading the source code and ascertaining that it delivers the promised outcome without users having to actually read it. This trust is new as it can be purely deductive, whereas trust cues established in the trust formation literature are mainly processed by induction. Beyond contributing new concepts of deductive certainty and deduction-related trust, this study shows, based on a sample of 536 dApps on Ethereum, that dApps which provide both deduction-related and induction-related trust cues attract more users than dApps that only provide one or the other. This finding emphasizes that despite not removing the need for trust in transactions, smart contracts on a blockchain enable a new way of forming trust that enforces traditional trust formation efforts.

The second study revisits the concepts of deductive certainty and deduction-related trust to investigate them from a user's perspective. While existing trust formation models account for the possibility of forming trust by deduction, I created a new model that accounts for the possibility of deductive certainty and deduction-related trust. Using this model, I show

how dispositional and institutional factors influence users' cognitive processes to form deduction and induction-related trust. To test the updated trust formation model, I compiled a survey and used a novel survey dApp specifically developed for this study to send the survey to users who have already interacted with dApps and ask how they form trust. Another novel feature of the survey dApp is that it allows survey responses to be pseudonymously linked with respondents’ transaction history, thus analyzing their past trusting behavior. The survey’s findings corroborate the previous chapter's results. They show that dApp users rely on a mixture of deduction and induction-related trust when deciding to interact with a new dApp. Also that users who rely more strongly on deduction-related trust interact with a larger variety of dApps. Again, this study emphasizes that dApps can use deduction-related trust cues (e.g., verification of their source code on Etherscan) to foster their perceived trustworthiness and attract users who would not necessarily interact with the dApp if there were only traditional induction-related trust cues. This study also highlights that a user’s tendency to rely on deduction or induction-related trust cues depends on dispositional factors, such as their general trust in people, in technology, and their technical knowledge, along with institutional factors such as the perceived structural assurance provided by the blockchain and the perceived risk of transacting on a blockchain.

The third study contributes by investigating the downside of using a market for transactions instead of a central authority to ensure their correct execution. The downside is that a market mechanism for verifying transactions adds yet another externality to the existing competition between dApps (i.e., increased use of a dApp raises the price of interacting with other dApps through increasing the gas price) while simultaneously limiting the platform provider’s strategic tools to protect dApps from this competition when necessary. Based on a sample of 1,560 dApps on Ethereum, I found that the current market mechanism favors finance dApps over all others and that finance transactions crowd out transactions with all other dApps. Although it might seem efficient and fair to allocate the limited supply of transactions to dApps whose users are willing to pay the most (finance dApps), discriminating other dApps can thwart a platform’s long-term strategy to foster innovation and a healthy ecosystem of complements. It is questionable whether Ethereum can become a general-purpose platform hosting the full spectrum of dApps and serve as backbone for Web3.

Besides these theoretical contributions, each study also presents innovative methods, showcasing a different way to leverage rich and openly available blockchain data for empirical research. To the best of my knowledge, the initial version of Study 1 was also the first to link smart contracts’ addresses with their associated dApps and compute dApp usage numbers based on the smart contract transaction history stored on the blockchain. The advantage of this approach is that researchers do not need to rely on numbers reported by dApp or third-party providers but can see directly every transaction the dApp has received, its initiator, and the value sent with the transaction. The benefit of linking dApps with their associated smart contracts and stored transaction history is reinforced as Studies 2 and 3 also rely on this link to compute their dependent variable (various measurements of dApp usage). The second study is innovative regarding empirical methods as it uses a specifically developed survey dApp that enables surveys to be sent to dApp users and pseudonymously link their responses to their past transaction history. This approach allows researchers to use metrics computed from a user's transaction history as dependent variables instead of the typically self-reported variables in survey research, without causing data privacy issues. Study 3 introduces a novel instrument, Ethereum's difficulty bomb, demonstrating that it allows researchers to address important endogeneity issues regarding Ethereum's gas price and estimate demand curves for transactions on Ethereum.

This dissertation's findings also point to additional practical implications for dApp and blockchain platform providers. The first two studies urge dApp providers not to rely just on a verified source code and the supposedly trust-free nature of the blockchain, but rather see the verification of their smart contract as an additional measure enhancing its trustworthiness. This new way to form trust can be used, particularly by dApp providers with no proven track record of successful transactions or who are burdened with an untrustworthy institutional environment (e.g., a country with a notoriously weak legal system) to attract users who would otherwise not transact with them. However, dApp providers still need to consider that users' dispositional factors such as technological savvy determine their reliance on induction or deduction-related trust cues, and tailor the trust cues they provide accordingly. Study 3 emphasizes that dApp providers about to join a blockchain platform need to consider that they will be competing directly not just with similar dApps but also indirectly with all other dApps for a limited supply of transactions. If a dApp provider fears not being able to attract enough users willing to pay the network's high transaction fees, it might be wiser to enter a smaller platform with fewer users and lower transaction fees. For platform providers, the first two studies show that the perception of the platform's security (e.g., the structural assurances it provides) is an important factor for users' trust formation processes, for their decision to use a dApp, and thus ultimately deciding to use the platform. Platform providers should therefore invest in providing structural assurances, for example resistance to malicious behavior such as 51% attacks, front-running, man-in-the-middle attacks, and educating users about these. The third study urges platform providers to pay particular attention to designing a market mechanism that ensures the verification and enforcement of transactions. Most smart contract-enabling platforms currently use auction-based mechanisms that prioritize transactions based on their users' willingness to pay for the transaction. Thus, they may be inappropriate for creating an ecosystem of diverse platform complements.

As discussed in detail in the individual chapters, this dissertation's studies have limitations. I will not repeat them all but focus on the general limitations that concern all three studies and discuss how these could be addressed in future research. Ethereum has been the most prominent smart contract-enabling platform in recent years, but multiple competing platforms have now entered the market (e.g., Binance Smart Chain, Solana, Polygon, Avalanche). Most are built around the same principles but differ in their technical design. As not all our findings might generalize to these platforms, future research could therefore revisit the claims scrutinized in this dissertation and investigate whether the same findings can be replicated. Conducting Study 3 on other platforms might yield interesting findings regarding how an alternatively designed market mechanism might favor different types of dApps and if other platforms are more capable of hosting a broad variety of dApps. The second general limitation is the pseudonymity of transaction records. All studies rely on proxying users by wallet addresses (externally owned accounts). Although this proxy is common in the field, it could overestimate the number of users because one user might have multiple wallets and thus control multiple externally owned accounts. This limitation is complicated as it is not easy to establish the true numbers. One promising avenue is advanced network analysis and machine learning techniques that allow the identification of circular transaction patterns in accounts owned by the same person. A third limitation concerns the infancy of the field. The studies for this dissertation were conducted at a time when blockchain platforms were still in their infancy and mainly used by enthusiasts and people with more technological know-how. Therefore, it remains to be seen if this study's findings will still apply once blockchain technology has entered the mainstream. The field's infancy also questions the internal validity of all studies as the observed results could also be due to co-occurring distorting events instead of the claimed mechanism. Although I tried my best to mitigate such concerns, for example by using two survey waves for Study 2 and computing robustness checks in different periods for Study 3, I cannot exclude the influence of such distorting events. Future research could conduct experiments to corroborate the internal validity of the results presented here. Particularly Study 2 would benefit from experiments allowing researchers to purposefully manipulate a dApp's trust cues and investigate how users react to these changes.

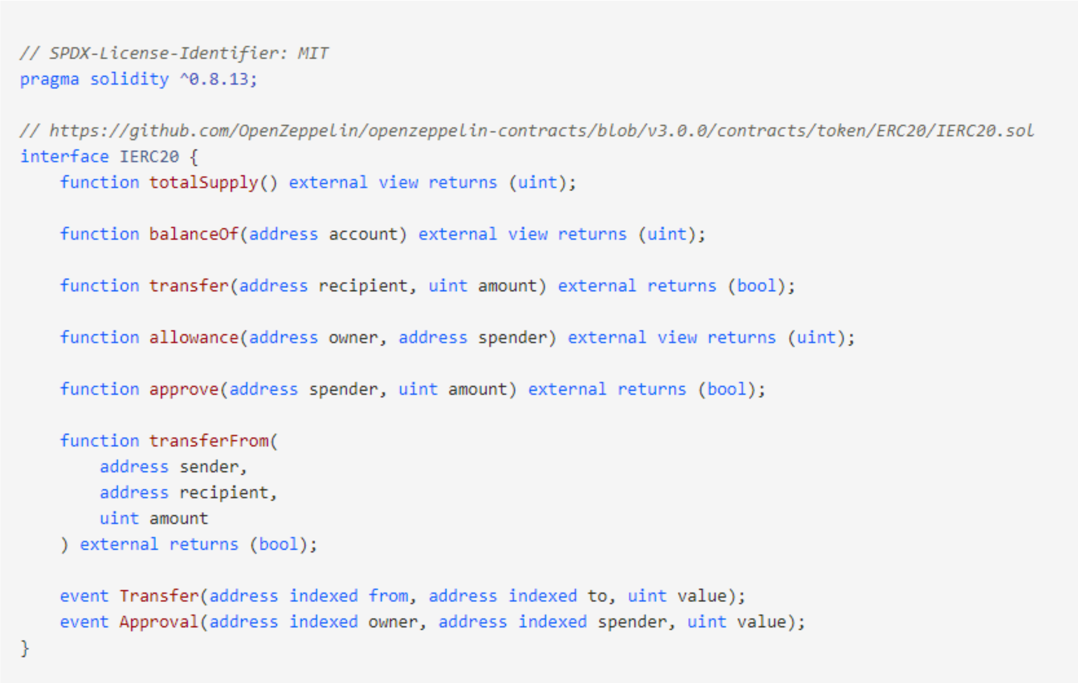
By scrutinizing two fundamental claims currently surrounding blockchain technology, I aim to contribute to a better understanding of what is hype and what is the real potential of this novel technology. However, given the rapid progress in the field and limited scope of my work, this is a humble contribution. Many economic, managerial, and organizational questions regarding this new technology remain unaddressed. I therefore hope my work not only contributes to specific literature but also inspires researchers in general to join this exciting field full of opportunities and help blockchain technology live up to its high promises and enable a future digital economy that is more transparent, inclusive, and democratic.

Appendix A

A-1 Example of an ERC20 token's standard functions

The smart contract below has two sections. The first specifies the following ERC20 interface functions: [78]

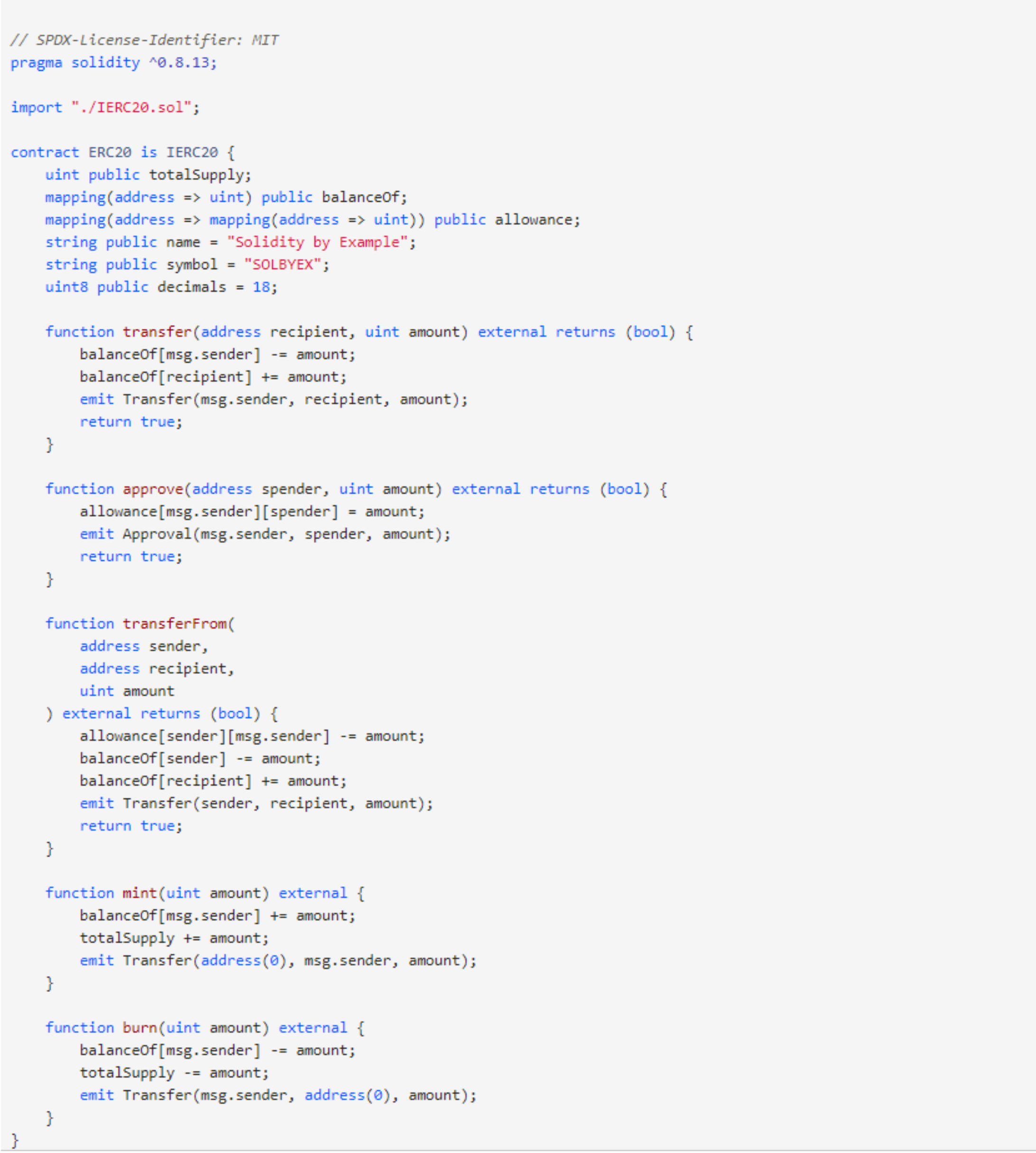
- TotalSupply: returns the total number of available tokens
- balanceOf: returns the number of tokens at the specified address
- transfer: transfers token from sender to a specified recipient
- allowance: allows a specified address to transfer tokens from the sender's account
- approve: used by the owner of the contract to authorize the given address to withdraw tokens from the owner's address.
- transferFrom: allows transferring tokens from one account to another



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

// https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.0.0/contracts/token/ERC20/IERC20.sol
interface IERC20 {
    function totalSupply() external view returns (uint);

    function balanceOf(address account) external view returns (uint);

    function transfer(address recipient, uint amount) external returns (bool);

    function allowance(address owner, address spender) external view returns (uint);

    function approve(address spender, uint amount) external returns (bool);

    function transferFrom(
        address sender,
        address recipient,
        uint amount
    ) external returns (bool);

    event Transfer(address indexed from, address indexed to, uint value);
    event Approval(address indexed owner, address indexed spender, uint value);
}
```

The second section imports the interface functions shown in the above contract, specifies these functions, then adds two functions that allow the minting (creating) and burning (deleting) of tokens.

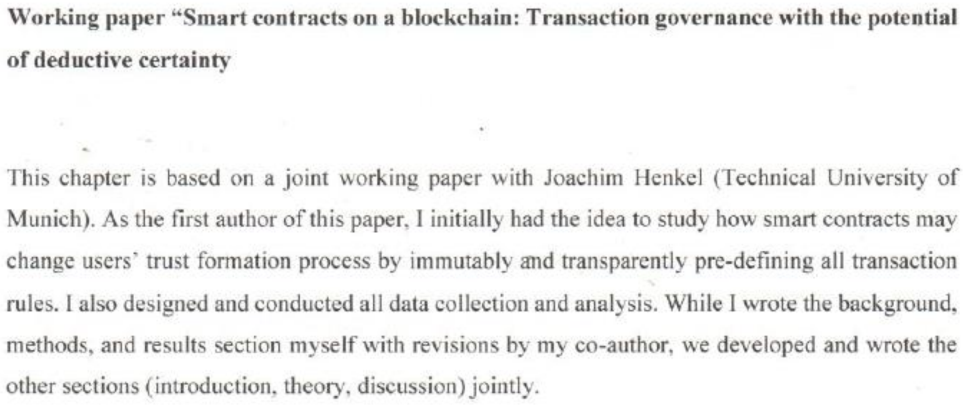
[78] https://solidity-by-example.org/app/erc20/, accessed September 15, 2022.



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

import "./IERC20.sol";

contract ERC20 is IERC20 {
    uint public totalSupply;
    mapping(address => uint) public balanceOf;
    mapping(address => mapping(address => uint)) public allowance;
    string public name = "Solidity by Example";
    string public symbol = "SOLBYEX";
    uint8 public decimals = 18;

    function transfer(address recipient, uint amount) external returns (bool) {
        balanceOf[msg.sender] -= amount;
        balanceOf[recipient] += amount;
        emit Transfer(msg.sender, recipient, amount);
        return true;
    }

    function approve(address spender, uint amount) external returns (bool) {
        allowance[msg.sender][spender] = amount;
        emit Approval(msg.sender, spender, amount);
        return true;
    }

    function transferFrom(
        address sender,
        address recipient,
        uint amount
    ) external returns (bool) {
        allowance[sender][msg.sender] -= amount;
        balanceOf[sender] -= amount;
        balanceOf[recipient] += amount;
        emit Transfer(sender, recipient, amount);
        return true;
    }

    function mint(uint amount) external {
        balanceOf[msg.sender] += amount;
        totalSupply += amount;
        emit Transfer(address(0), msg.sender, amount);
    }

    function burn(uint amount) external {
        balanceOf[msg.sender] -= amount;
        totalSupply -= amount;
        emit Transfer(msg.sender, address(0), amount);
    }
}
```

Appendix B

B-1 Contribution to Chapter 3



Working paper “Smart contracts on a blockchain: Transaction governance with the potential of deductive certainty

This chapter is based on a joint working paper with Joachim Henkel (Technical University of Munich). As the first author of this paper, I initially had the idea to study how smart contracts may change users' trust formation process by immutably and transparently pre-defining all transaction rules. I also designed and conducted all data collection and analysis. While I wrote the background, methods, and results section myself with revisions by my co-author, we developed and wrote the other sections (introduction, theory, discussion) jointly.

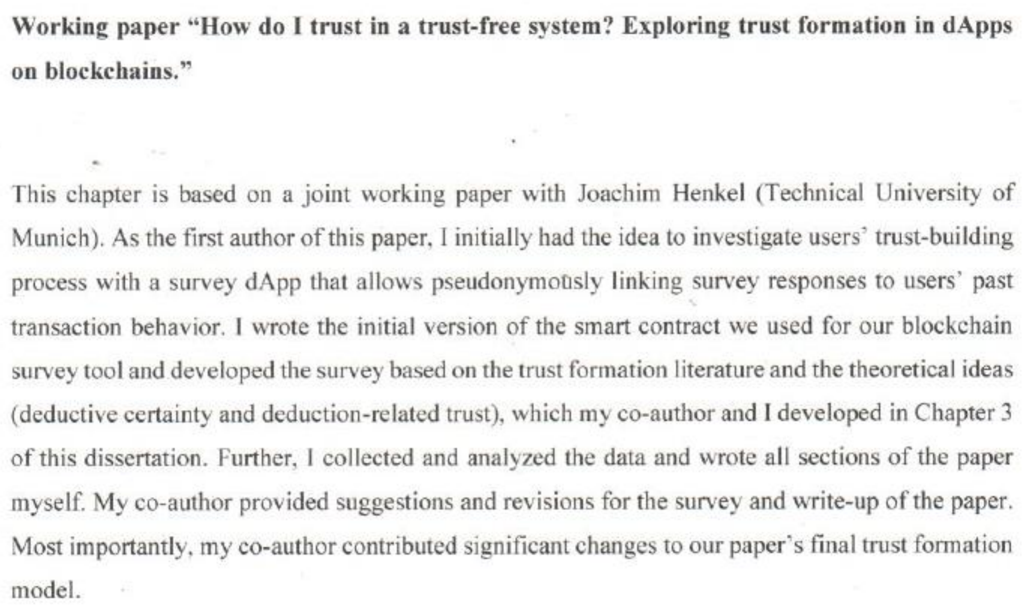
B-2 Induction-based trust items

Table 24: Inductive trust items – rating framework

Item	Level	Word anchor
Perceived Integrity	5	Clear explanation of conduct, explanation of transaction conditions, refer to clear rules, explain how they obey rules
	3	Clear explanation of conduct, explanation of transaction conditions,
	1	No explanation at all
Perceived Benevolence	5	Show they are acting in the interest of their users or society/community (e.g., no fees, 100% of the money goes to charity, 24/7 customer hotline to answer questions)
	3	Neutral, no explicit measures benefitting their users except service offered
	1	Harms some users (e.g., Ponzi schemes)
Perceived Ability	5	High-quality team, use and explain technical terms very well; well-elaborated technical appearance of the entire website and service
	3	Good quality team, some explanations of technical terms, clean technical appearance of website and service offered
	1	Poor technical appearance, no team information, no explanations
Perceived Usefulness	5	Clear problem statement and explanation of how this problem is solved, clear USP, service solves a significant problem with useful features
	3	Solves meaningful problem in a reasonable way
	1	Useless or random product not addressing a problem
Perceived Ease of Use	5	A step-by-step explanation of user journey (e.g., explainer videos); standalone web app allowing use of the service without any setup
	3	Meta Mask integration; transactions can be sent via user interface
	1	Unclear how to go about interacting with the smart contract; transactions have to be sent by typing in the smart contract ID (no user interface at all)

Appendix C

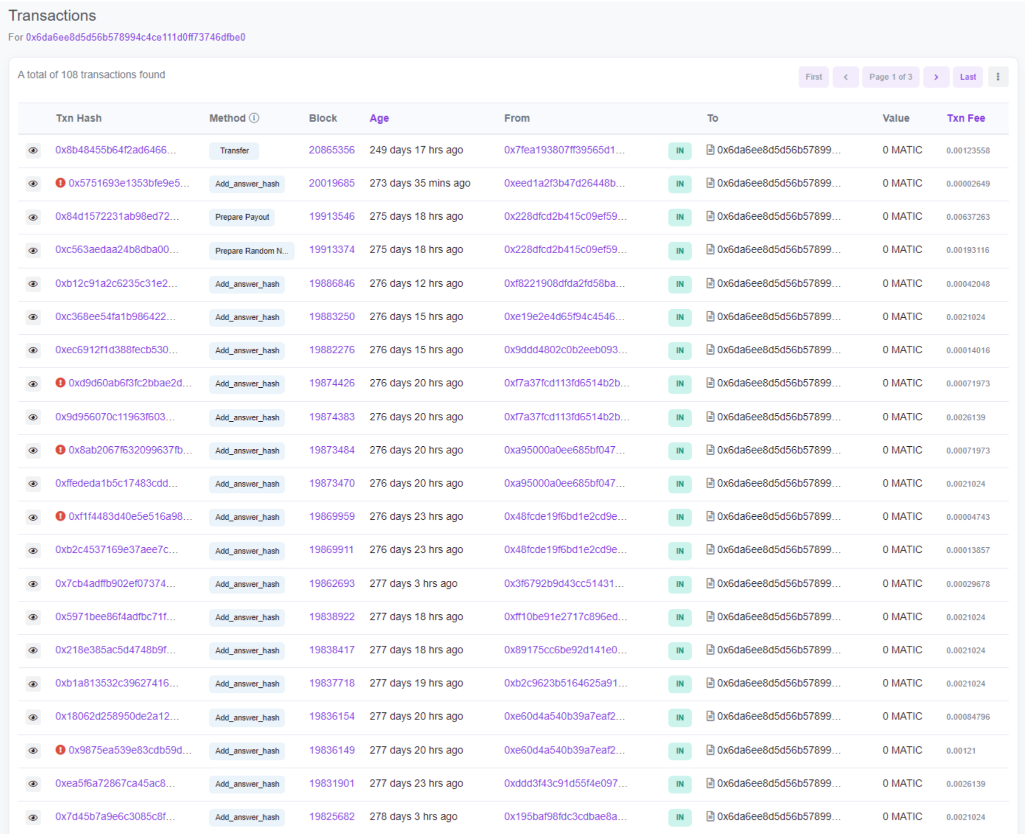
C-1 Contribution to Chapter 4



Working paper "How do I trust in a trust-free system? Exploring trust formation in dApps on blockchains."

This chapter is based on a joint working paper with Joachim Henkel (Technical University of Munich). As the first author of this paper, I initially had the idea to investigate users' trust-building process with a survey dApp that allows pseudonymously linking survey responses to users' past transaction behavior. I wrote the initial version of the smart contract we used for our blockchain survey tool and developed the survey based on the trust formation literature and the theoretical ideas (deductive certainty and deduction-related trust), which my co-author and I developed in Chapter 3 of this dissertation. Further, I collected and analyzed the data and wrote all sections of the paper myself. My co-author provided suggestions and revisions for the survey and write-up of the paper. Most importantly, my co-author contributed significant changes to our paper's final trust formation model.

C-2 List of transactions with the survey dApp



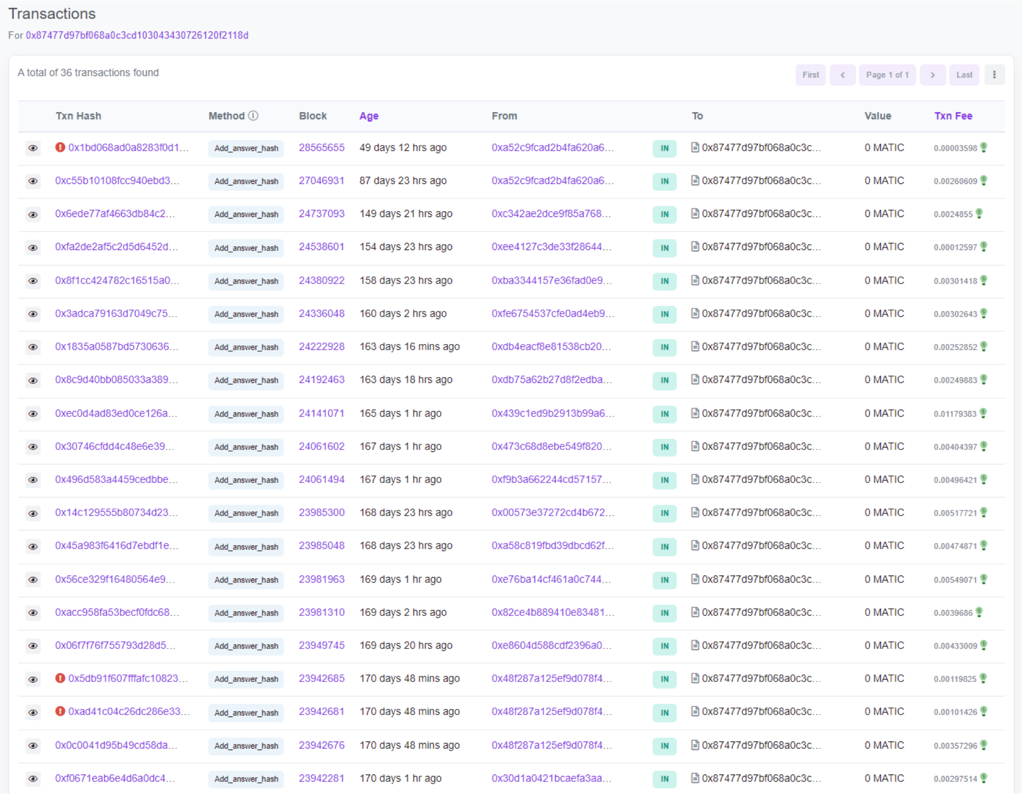
Transactions

For 0x6da6ee8d5d56b578994c4ce111d0ff73746dfbe0

A total of 108 transactions found

	Txn Hash	Method ⓘ	Block	Age	From		To	Value	Txn Fee
👁	0x8b48455b64f2ad6466…	Transfer	20865356	249 days 17 hrs ago	0x7fea193807ff39565d1…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00123558
👁	❗ 0x5751693e1353bfe9e5…	Add_answer_hash	20019685	273 days 35 mins ago	0xeed1a2f3b47d26448b…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00002649
👁	0x84d1572231ab98ed72…	Prepare Payout	19913546	275 days 18 hrs ago	0x228dfcd2b415c09ef59…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00637263
👁	0xc563aedaa24b8dba00…	Prepare Random N…	19913374	275 days 18 hrs ago	0x228dfcd2b415c09ef59…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00193116
👁	0xb12c91a2c6235c31e2…	Add_answer_hash	19886846	276 days 12 hrs ago	0xf8221908dfda2fd58ba…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00042048
👁	0xc368ee54fa1b986422…	Add_answer_hash	19883250	276 days 15 hrs ago	0xe19e2e4d65f94c4546…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.0021024
👁	0xec6912f1d388fecb530…	Add_answer_hash	19882276	276 days 15 hrs ago	0x9ddd4802c0b2eeb093…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00014016
👁	❗ 0xd9d60ab6f3fc2bbae2d…	Add_answer_hash	19874426	276 days 20 hrs ago	0xf7a37fcd113fd6514b2b…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00071973
👁	0x9d956070c11963f603…	Add_answer_hash	19874383	276 days 20 hrs ago	0xf7a37fcd113fd6514b2b…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.0026139
👁	❗ 0x8ab2067f632099637fb…	Add_answer_hash	19873484	276 days 20 hrs ago	0xa95000a0ee685bf047…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00071973
👁	0xffededa1b5c17483cdd…	Add_answer_hash	19873470	276 days 20 hrs ago	0xa95000a0ee685bf047…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.0021024
👁	❗ 0xf1f4483d40e5e516a98…	Add_answer_hash	19869959	276 days 23 hrs ago	0x48fcde19f6bd1e2cd9e…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00004743
👁	0xb2c4537169e37aee7c…	Add_answer_hash	19869911	276 days 23 hrs ago	0x48fcde19f6bd1e2cd9e…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00013857
👁	0x7cb4adffb902ef07374…	Add_answer_hash	19862693	277 days 3 hrs ago	0x3f6792b9d43cc51431…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00029678
👁	0x5971bee86f4adfbc71f…	Add_answer_hash	19838922	277 days 18 hrs ago	0xff10be91e2717c896ed…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.0021024
👁	0x218e385ac5d4748b9f…	Add_answer_hash	19838417	277 days 18 hrs ago	0x89175cc6be92d141e0…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.0021024
👁	0xb1a813532c39627416…	Add_answer_hash	19837718	277 days 19 hrs ago	0xb2c9623b5164625a91…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.0021024
👁	0x18062d258950de2a12…	Add_answer_hash	19836154	277 days 20 hrs ago	0xe60d4a540b39a7eaf2…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00084796
👁	❗ 0x9875ea539e83cdb59d…	Add_answer_hash	19836149	277 days 20 hrs ago	0xe60d4a540b39a7eaf2…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.00121
👁	0xea5f6a72867ca45ac8…	Add_answer_hash	19831901	277 days 23 hrs ago	0xddd3f43c91d55f4e097…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.0026139
👁	0x7d45b7a9e6c3085c8f…	Add_answer_hash	19825682	278 days 3 hrs ago	0x195baf98fdc3cdbae8a…	IN	0x6da6ee8d5d56b57899…	0 MATIC	0.0021024

Figure 31: Excerpt of transactions with a smart contract in our first survey wave



Transactions

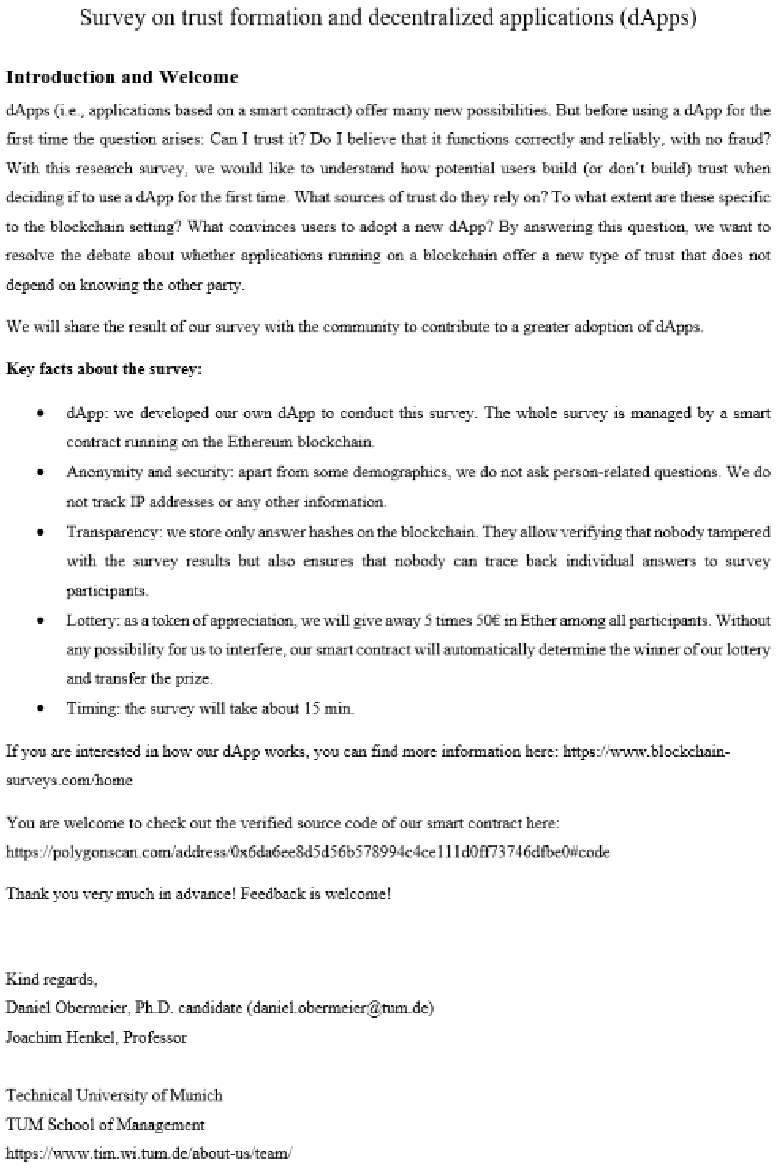
For 0x87477d97bf068a0c3cd103043430726120f2118d

A total of 36 transactions found

	Txn Hash	Method ⓘ	Block	Age	From		To	Value	Txn Fee
👁	❗ 0x1bd068ad0a8283f0d1…	Add_answer_hash	28565655	49 days 12 hrs ago	0xa52c9fcad2b4fa620a6…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00003598
👁	0xc55b10108fcc940ebd3…	Add_answer_hash	27046931	87 days 23 hrs ago	0xa52c9fcad2b4fa620a6…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00260609
👁	0x6ede77af4663db84c2…	Add_answer_hash	24737093	149 days 21 hrs ago	0xc342ae2dce9f85a768…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.0024855
👁	0xfa2de2af5c2d5d6452d…	Add_answer_hash	24538601	154 days 23 hrs ago	0xee4127c3de33f28644…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00012597
👁	0x8f1cc424782c16515a0…	Add_answer_hash	24380922	158 days 23 hrs ago	0xba3344157e36fad0e9…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00301418
👁	0x3adca79163d7049c75…	Add_answer_hash	24336048	160 days 2 hrs ago	0xfe6754537cfe0ad4eb9…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00302643
👁	0x1835a0587bd5730636…	Add_answer_hash	24222928	163 days 16 mins ago	0xdb4eacf8e81538cb20…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00252852
👁	0x8c9d40bb085033a389…	Add_answer_hash	24192463	163 days 18 hrs ago	0xdb75a62b27d8f2edba…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00249883
👁	0xec0d4ad83ed0ce126a…	Add_answer_hash	24141071	165 days 1 hr ago	0x439c1ed9b2913b99a6…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.01179383
👁	0x30746cfdd4c48e6e39…	Add_answer_hash	24061602	167 days 1 hr ago	0x473c68d8ebe549f820…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00404397
👁	0x496d583a4459cedbbe…	Add_answer_hash	24061494	167 days 1 hr ago	0xf9b3a662244cd57157…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00496421
👁	0x14c129555b80734d23…	Add_answer_hash	23985300	168 days 23 hrs ago	0x00573e37272cd4b672…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00517721
👁	0x45a983f6416d7ebdf1e…	Add_answer_hash	23985048	168 days 23 hrs ago	0xa58c819fbd39dbcd62f…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00474871
👁	0x56ce329f16480564e9…	Add_answer_hash	23981963	169 days 1 hr ago	0xe76ba14cf461a0c744…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00549071
👁	0xacc958fa53becf0fdc68…	Add_answer_hash	23981310	169 days 2 hrs ago	0x82ce4b889410e83481…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.0039686
👁	0x06f7f76f755793d28d5…	Add_answer_hash	23949745	169 days 20 hrs ago	0xe8604d588cdf2396a0…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00433009
👁	❗ 0x5db91f607fffafc10823…	Add_answer_hash	23942685	170 days 48 mins ago	0x48f287a125ef9d078f4…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00119825
👁	❗ 0xad41c04c26dc286e33…	Add_answer_hash	23942681	170 days 48 mins ago	0x48f287a125ef9d078f4…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00101426
👁	0x0c0041d95b49cd58da…	Add_answer_hash	23942676	170 days 48 mins ago	0x48f287a125ef9d078f4…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00357296
👁	0xf0671eab6e4d6a0dc4…	Add_answer_hash	23942281	170 days 1 hr ago	0x30d1a0421bcaefa3aa…	IN	0x87477d97bf068a0c3c…	0 MATIC	0.00297514

Figure 32: Excerpt of transactions with a smart contract in our second survey wave

C-3 Introduction to the survey



Survey on trust formation and decentralized applications (dApps)

Introduction and Welcome

dApps (i.e., applications based on a smart contract) offer many new possibilities. But before using a dApp for the first time the question arises: Can I trust it? Do I believe that it functions correctly and reliably, with no fraud? With this research survey, we would like to understand how potential users build (or don't build) trust when deciding if to use a dApp for the first time. What sources of trust do they rely on? To what extent are these specific to the blockchain setting? What convinces users to adopt a new dApp? By answering this question, we want to resolve the debate about whether applications running on a blockchain offer a new type of trust that does not depend on knowing the other party.

We will share the result of our survey with the community to contribute to a greater adoption of dApps.

Key facts about the survey:

- dApp: we developed our own dApp to conduct this survey. The whole survey is managed by a smart contract running on the Ethereum blockchain.
- Anonymity and security: apart from some demographics, we do not ask person-related questions. We do not track IP addresses or any other information.
- Transparency: we store only answer hashes on the blockchain. They allow verifying that nobody tampered with the survey results but also ensures that nobody can trace back individual answers to survey participants.
- Lottery: as a token of appreciation, we will give away 5 times 50€ in Ether among all participants. Without any possibility for us to interfere, our smart contract will automatically determine the winner of our lottery and transfer the prize.
- Timing: the survey will take about 15 min.

If you are interested in how our dApp works, you can find more information here: https://www.blockchain-surveys.com/home

You are welcome to check out the verified source code of our smart contract here:
https://polygonscan.com/address/0x6da6ee8d5d56b578994c4ce111d0ff73746dfbe0#code

Thank you very much in advance! Feedback is welcome!

Kind regards,
Daniel Obermeier, Ph.D. candidate (daniel.obermeier@tum.de)
Joachim Henkel, Professor

Technical University of Munich
TUM School of Management
https://www.tim.wi.tum.de/about-us/team/

Figure 33: Introduction to the survey

C-4 Summary statistics and correlations

Table 25 Summary statistics and correlations

Variables	N	Mean	s.d.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Transaction count	121	110.63	163.26	1																								
dApp count	121	11.42	9.71	0.84	1																							
FDP1	121	3.36	0.77	0.11	0.09	1																						
FDP2	121	3.33	0.86	0.10	0.14	0.59	1																					
FDP3	121	3.50	0.83	0.07	0.08	0.43	0.40	1																				
TSP1	121	3.66	1.02	0.06	0.09	0.20	0.29	-0.08	1																			
TSP3	121	3.59	1.04	0.05	0.05	0.21	0.26	-0.13	0.79	1																		
TST3	121	3.18	1.09	-0.14	-0.14	0.22	0.27	0.19	0.21	0.26	1																	
TEK1	121	3.50	1.02	0.10	0.11	0.02	0.19	-0.19	0.18	0.15	-0.07	1																
TEK2	121	3.18	1.09	0.05	0.05	0.04	0.20	-0.18	0.11	0.13	-0.02	0.85	1															
TEK3	121	2.79	1.33	0.12	0.12	0.19	0.24	0.04	-0.07	-0.06	0.01	0.58	0.67	1														
FH.INT	121	3.85	0.97	0.26	0.20	0.09	0.19	0.18	0.05	0.05	0.02	-0.03	0.02	-0.03	1													
FH.ABI	121	4.06	1.04	0.08	0.10	0.06	0.13	0.25	0.03	-0.04	-0.05	0.11	0.05	0.04	0.55	1												
SH.INT	121	3.49	1.05	-0.07	-0.19	0.20	0.10	0.10	0.15	0.20	0.02	-0.08	-0.01	-0.07	0.20	0.21	1											
SH.ABI	121	3.32	1.26	0.003	-0.04	0.08	0.11	0.21	0.09	0.07	0.22	-0.13	-0.11	0.04	0.27	0.32	0.47	1										
FH-DED1	121	3.25	1.63	0.16	0.17	0.16	0.12	0.21	-0.15	-0.12	-0.05	0.24	0.31	0.61	0.12	0.21	0.03	0.09	1									
FH-DED2	121	3.02	1.53	0.27	0.32	0.10	0.12	0.19	-0.15	-0.15	-0.07	0.22	0.23	0.59	0.11	0.28	-0.08	0.20	0.84	1								
SH-DED1	121	3.96	0.83	0.09	-0.02	0.15	0.18	0.18	0.04	0.10	0.16	-0.06	-0.06	-0.12	0.21	0.26	0.29	0.20	-0.07	-0.08	1							
SH-DED2	121	3.85	0.94	-0.001	-0.06	0.21	0.21	0.33	-0.07	-0.03	0.13	0.04	0.13	0.22	0.29	0.42	0.07	0.33	0.22	0.24	0.27	1						
SABC1	121	3.92	0.92	0.22	0.25	0.14	0.15	0.25	-0.20	-0.08	0.004	0.06	0.05	0.21	0.26	0.40	0.01	0.19	0.28	0.30	0.28	0.29	1					
SABC 2	121	4.19	0.79	0.03	0.06	0.10	0.08	0.14	-0.12	-0.03	0.09	0.10	0.11	0.21	0.14	0.16	-0.02	0.06	0.31	0.28	0.14	0.18	0.47	1				
SABC 3	121	3.98	0.93	0.04	0.08	-0.08	-0.04	0.18	-0.22	-0.17	-0.02	0.08	0.11	0.15	0.16	0.17	-0.19	0.04	0.27	0.20	0.09	0.23	0.31	0.50	1			
PBR1	121	2.25	1.00	-0.20	-0.07	-0.21	-0.20	-0.08	-0.16	-0.11	-0.08	-0.01	-0.01	0.02	-0.29	-0.08	-0.12	-0.05	0.08	0.08	-0.16	-0.10	-0.09	0.001	0.06	1		
PBR2	121	2.54	1.16	-0.27	-0.18	-0.13	-0.08	-0.20	-0.02	0.07	0.01	-0.05	-0.09	-0.04	-0.20	-0.12	0.05	-0.02	-0.09	-0.09	-0.002	-0.25	-0.08	-0.01	-0.17	0.49	1	
RISK	121	1.98	0.93	-0.07	0.03	-0.12	0.01	-0.03	-0.08	-0.09	-0.05	-0.01	-0.04	0.02	-0.07	0.02	-0.14	0.04	-0.05	0.06	-0.13	-0.04	0.06	0.003	0.09	0.16	0.09	1

C-5 Operationalization of survey measurements

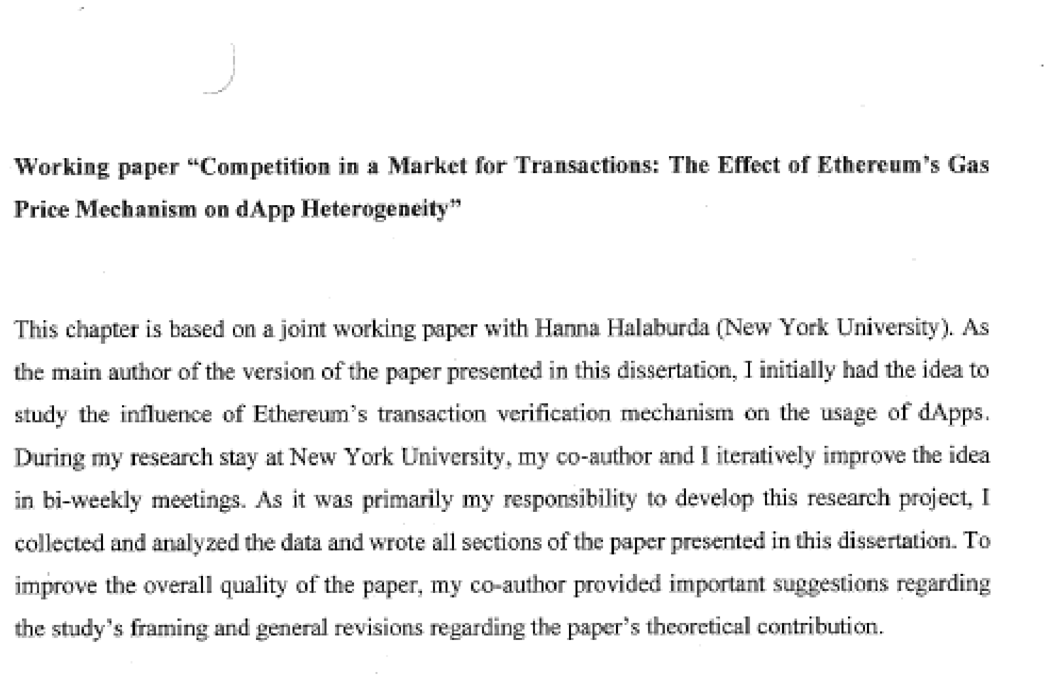
Table 26: Operationalization of measures

Construct	Abbreviation	Items
	Trusting beliefs	
First-hand induction-related trusting beliefs	FH-INT	1. I feel confident transacting with this company if I perceive it as honest.
	FH-BEN	2. I am confident transacting with this company if I have the feeling that the company would act in my best interests.
	FH-ABI	3. I feel confident transacting with this company if I perceive the company as competent and effective in providing its service.
Second-hand induction-related trusting beliefs	SH-INT	1. Do you feel confident interacting with a new dApp because others have told you about the other party being honest?
	SH-BEN	2. Do you feel confident interacting with a new dApp because others have told you that the other party would always act in your best interests?
	SH-ABI	3. Do you feel confident interacting with a new dApp because others have told you that the other party is competent and effective in providing its service?
First-hand deduction-related trusting beliefs	RSC-Y1	1. I feel confident transacting with a new dApp because I could read the source code if I wanted to.
	RSC-Y2	2. I feel confident transacting with a new dApp because I read the source code (or parts of it) and see what it does.
	RSC-Y3	3. I feel confident transacting with a new dApp because I read the source code (or parts of it) and understand that it does what it is supposed to do.
	RSC-Y4	4. I feel confident transacting with a new dApp because I read the source code and verify that it is free of errors
Second-hand deduction-related trusting beliefs	ORSC1	1. I feel confident transacting with a new dApp if I have heard from the blockchain community that the dApp's source code is secure.
	ORSC2	2. I feel confident transacting with a new dApp if I have seen the smart contract's security certificates on which a dApp is based.
	Trust search	
First-hand inductive search	FH-TS	1. When deciding to use a new dApp, to what extent do you typically inform yourself about the company offering it.
Second-hand inductive search	SH-INT-TS	1. When you decide to use a new dApp, have others (e.g., individuals or websites) told you that the party offing the dApp is honest?
	SH-BEN-TS	2. When you decide to use a new dApp, have others told you that the other party will help you if required?
	SH-ABI-TS	3. When you decide to use a new dApp, have others told you that the other party is competent?
Possibility for others to read the source code	SCV-SH	1. Why do you care about a verified source code? Because it allows others (e.g., the community, developers, and audit companies) to check for mistakes in it and inform the public about their findings.
	SCV-Signal	2. Why do you care about a verified source code? Because it signals that the party offering the dApp has nothing to hide.
Possibility to read the source code	SCV-FH	1. Do you care whether a dApp's source code is publicly disclosed and verified?
	SCV-Reading	2. Why do you care about a verified source code? Because I want to read it.
Reading the source code	RSC	1. To what extent do you read the dApp's source code?
	Institutional factors	
Structural assurance blockchain technology	SABC1	1. When considering whether to use a new dApp, I feel confident transacting with it because a consensus mechanism on the Ethereum blockchain ensures that the smart contract on which a dApp is based is executed automatically.
	SABC2	2. When considering whether to use a new dApp, I feel confident transacting with it because no-one on the Ethereum blockchain can change the result of transactions afterwards.
	SABC3	3. When considering whether to use a new dApp, I feel confident transacting with it because everyone on the Ethereum blockchain can see the data stored there.
Perceived blockchain risk	PBR1	1. In general, I think it is risky to send transactions on Ethereum.
	PBR2	2. I hesitate to spend money on Ethereum.
	PBR3	3. Generally, transactions on the Ethereum blockchain are executed correctly.

	Dispositional factors	
Faith in dApp provider	FDP1	1. In general, I think companies offering dApps on Ethereum are honest.
	FDP2	2. In general, I think companies offering dApps on Ethereum act in the best interest of their customers.
	FDP3	3. In general, I think companies offering dApps on Ethereum are competent and effective in providing services.
Trusting stance toward people	TSP1	1- I usually trust people until they give me a reason not to trust them.
	TSP2	2. I generally do not give people the benefit of the doubt when I first meet them.
	TSP3	3. My typical approach is to trust others until they prove I should not trust them.
Trusting stance toward technology	TST1	1. Generally, I believe that technology cannot be relied upon.
	TST2	2. I generally give new technology the benefit of the doubt.
	TST3	3. My typical approach is to think that new technology works as expected.
Knowledge about blockchain technology	TEK1	1. What is your level of knowledge about blockchain technology in general?
	TEK2	2. What is your level of knowledge about the Ethereum protocol and infrastructure?
	TEK3	3. What is your level of ability to read Solidity code?
	Others	
Attitude towards dApps	ATD1	I like to explore new dApps.
	ATD2	I am generally not interested in trying out new dApps.
	ATD3	Among my peers, I am usually the first to try out new dApps.

Appendix D

D-1 Contribution to Chapter 5



Working paper "Competition in a Market for Transactions: The Effect of Ethereum's Gas Price Mechanism on dApp Heterogeneity"

This chapter is based on a joint working paper with Hanna Halaburda (New York University). As the main author of the version of the paper presented in this dissertation, I initially had the idea to study the influence of Ethereum's transaction verification mechanism on the usage of dApps. During my research stay at New York University, my co-author and I iteratively improve the idea in bi-weekly meetings. As it was primarily my responsibility to develop this research project, I collected and analyzed the data and wrote all sections of the paper presented in this dissertation. To improve the overall quality of the paper, my co-author provided important suggestions regarding the study's framing and general revisions regarding the paper's theoretical contribution.

7 REFERENCES

Agarwal, R., & Gort, M. 2002. Firm and Product Life Cycles and Firm Survival. ***The American Economic Review***, 92(2): 184–190.

Ahangama, S., & Poo, D. C. C. 2016. Credibility of Algorithm Based Decentralized Computer Networks Governing Personal Finances: The Case of Cryptocurrency. In F. F.-H. Nah & C.-H. Tan (Eds.), ***HCI in Business, Government, and Organizations: eCommerce and Innovation***: 165–176. Cham: Springer International Publishing.

Ajzen, I. 1988. ***Attitudes, Personality, and Behavior*** (2nd edn.). Chicago: Dorsey Press.

Al-Breiki, H., Rehman, M., Salah, K., & Svetinovic, D. 2020. Trustworthy Blockchain Oracles: Review, Comparison, and Open Research Challenges. ***IEEE Access***, 8: 85675–85685.

Antonopoulos, A. 2018. ***Mastering bitcoin: Programming the open blockchain*** (2nd ed.). Beijing, Boston, Farnham, Sebastopol, Tokyo: O'Reilly.

Antonopoulos, A. 2020. May the fourth be with you, from: https://twitter.com/aantonop/status/ 12573.

Antonopoulos, A., & Wood, G. 2019. ***Mastering Ethereum: Building smart contracts and DApps*** (1st ed.). EBL-Schweitzer. Beijing, Boston, Farnham, Sebastopol, Tokyo: O'Reilly.

Arnosti, N., & Weinberg, M. 2018. Bitcoin: A Natural Oligopoly, forthcoming.

Azevedo Sousa, J., Oliveira, V., Valadares, J., Dias Gonçalves, G., Moraes Villela, S., Soares Bernardino, H., & Borges Vieira, A. 2021. An analysis of the fees and pending time correlation in Ethereum. ***International Journal of Network Management***, 31(3).

Ba, S., Whinston, A., & Zhang, H. 1999. Building Trust in the Electronic Market through an Economic Incentive Mechanism. ***Proceedings of the 20th International Conference on Information Systems***: 208–213. USA: Association for Information Systems.

Baer, M., van der Werff, L., Colquitt, J., Rodell, J., Zipay, K., & Buckley, F. 2018. Trusting the “look and feel”: Situational normality, situational aesthetics, and the perceived trustworthiness of organizations. ***Academy of Management Journal***, 61(5): 1718–1740.

Bagozzi, R. P. 2007. On the meaning of formative measurement and how it differs from reflective measurement: comment on Howell, Breivik, and Wilcox (2007). ***Psychological Methods***, 12(2): 229-37; discussion 238-45.

Barney, B., & Hansen, M. 1994. Trustworthiness as a source of competitive advantage. ***Strategic Management Journal***, 15: 175–190.

Bartoletti, M., & Pompianu, L. 2017. An empirical analysis of smart contracts: platforms, applications, and design patterns. In M. Brenner, K. Rohloff, J. Bonneau, A. Miller, P. Y. Ryan, V. Teague, A. Bracciali, M. Sala, F. Pintore & M. Jakobsson (Eds.), ***Financial Cryptography and Data Security***. Cham: Springer International Publishing.

Bashir, I. 2020. ***Mastering Blockchain*: *A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more*** (3rd ed.). Birmingham, UK: Packt Publishing Limited.

Basu, S., Easley, D., O'Hara, M., & Sirer, E. 2019. Towards a Functional Fee Market for Cryptocurrencies, forthcoming.

Beck, R., Avital, M., Rossi, M., & Thatcher, J. 2017. Blockchain technology in business and information systems research. ***Business & Information Systems Engineering***, 59: 381–384.

Beck, R., Czepluch, J., Lollike, N., & Malone. 2016. Blockchain - the gateway to trust free cryptographic transactions. ***Proceedings of the 24th European Conference on Information Systems (ECIS)***, forthcoming.

Becker, T., Robertson, M., & Vandenberg, R. 2019. Nonlinear Transformations in Organizational Research: Possible Problems and Potential Solutions. ***Organizational Research Methods***, 22(4): 831–866.

Beldad, A., Jong, M., & Steehouder, M. 2010. How shall I trust the faceless and the intangible? A literature review on the antecedents of online trust. ***Computers in Human Behavior***, 26(5): 857–869.

Bigley, G., & Pearce, J. 1998. Straining for shared meaning in organization science: Problems of trust and distrust. ***Academy of Management Review***, 23: 405–421.

Boudreau, K. 2012. Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation. ***Organization Science***, 23(5): 1409–1427.

Brengman, M., & Karimov, F. 2012. The effect of web communities on consumers' initial trust in B2C e-commerce websites. ***Management Research Review***, 35(9): 791–817.

Brynjolfsson, E., & Kemerer, C. 1996. Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market. ***Management Science***, 42(12): 1627–1647.

Buterin, V. 2014. Ethereum Whitepaper. Downloaded on September 13, 2022, from: https://ethereum.org/en/whitepaper/.

Buterin, V. 2021. A Next-Generation Smart Contract and Decentralized Application Platform. Downloaded on April 15, 2021, from: https://ethereum.org/en/whitepaper/.

Cai, W., Wang, Z., Ernst, J., Hong, Z., Feng, C., & Leung, V. C. 2018. Decentralized Applications: The Blockchain-Empowered Software System. ***IEEE Access***, 6: 53019–53033.

Caldarelli, G. 2020. Real-world blockchain applications under the lens of the oracle problem. A systematic literature review. ***2020 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)***. IEEE.

Casadesus-Masanell, R., & Hałaburda, H. 2014. When Does a Platform Create Value by Limiting Choice? ***Journal of Economics & Management Strategy***, 23(2): 259–293.

Catalini, C., & Gans, J. 2020. Some simple economics of the blockchain. ***Communications of the ACM***, 63(7): 80–90.

Catalini, C., & Tucker, C. 2018. Antitrust and Costless Verification: An Optimistic and a Pessimistic View of the Implications of Blockchain Technology. ***SSRN Electronic Journal***: 1–14.

Chatterjee, K., Goharshady, A., & Pourdamghani, A. 2019. Probabilistic Smart Contracts: Secure Randomness on the Blockchain. ***2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)***. IEEE.

Chau, P., Hu, P., Lee, B., & Au, A. 2007. Examining customers' trust in online vendors and their dropout decisions: An empirical study. ***Electronic Commerce Research and Applications***, 6(2): 171–182.

Chen, Y., Richter, J., & Patel, P. 2021. Decentralized Governance of Digital Platforms. ***Journal of Management***, 47(5): 1305–1337.

Chin, W., & Newman, P. 2000. Structural equation modeling analysis with small samples using partial least squares. In ***Statistical strategies for small sample research***: 307–341. Thousand Oaks, CA: Sage Publications, Inc.

Choi, J. 1994. Network Externality, Compatibility Choice, and Planned Obsolescence. ***The Journal of Industrial Economics***, 42(2): 167.

Choi, J., & Kim, B. 2010. Net Neutrality and Investment Incentives. ***RAND Journal of Economics***, 41(3): 446–471.

Christidis, K., & Devetsikiotis, M. 2016. Blockchains and Smart Contracts for the Internet of Things. ***IEEE Access***, 4: 2292–2303.

Church, J., & Gandal, N. 1992. Network Effects, Software Provision, and Standardization. ***The Journal of Industrial Economics***, 40(1): 85.

Colquitt, J., LePine, J., Zapata, C., & Wild, R. 2011. Trust in typical and high-reliability contexts: Building and reacting to trust among firefighters. ***Academy of Management Journal***, 54: 999–1015.

Cong, L., Tang, K., Wang, Y., & Zhao, X. 2022. Inclusion and Democratization Through Web3 and DeFi? Initial Evidence from the Ethereum Ecosystem. ***SSRN Electronic Journal***, forthcoming.

Cong, L. W., He, Z., & Li, J. 2021. Decentralized Mining in Centralized Pools. ***The Review of Financial Studies***, 34(3): 1191–1235.

Cook, J., & Wall, T. 1980. New work attitude measures of trust, organizational commitment and personal need non-fulfilment. ***Journal of Occupational Psychology***, 53: 39–52.

Cox, D. 1972. Regression Models and Life-Tables. ***Journal of the Royal Statistical Society Series B (Methodological)***, 34(2): 187–220.

Crosby, L., Evans, K., & Cowles, D. 1990. Relationship Quality in Services Selling: An Interpersonal Influence Perspective. ***Journal of Marketing***, 54(3): 68–81.

Cusumano, M., Mylonadis, Y., & Rosenbloom, R. 1992. Strategic Maneuvering and Mass-Market Dynamics: The Triumph of VHS over Beta. ***Business History Review***, 66(1): 51–94.

Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., & Juels, A. 2020. Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability. ***2020 IEEE Symposium on Security and Privacy (SP)***. IEEE.

Diamantopoulos, A., Riefler, P., & Roth, K. 2008. Advancing formative measurement models. ***Journal of Business Research***, 61(12): 1203–1218.

Diamantopoulos, A., & Winklhofer, H. 2001. Index Construction with Formative Indicators: An Alternative to Scale Development. ***Journal of Marketing Research***, 38(2): 269–277.

Dillman, D. 1978. ***Mail and telephone surveys: The total design method***. Wiley New York.

Dirks, K. T. 1999. The effects of interpersonal trust on work group performance. ***Journal of Applied Psychology***, 84: 445–455.

Doney, P., & Cannon, J. 1997. An Examination of the Nature of Trust in Buyer-Seller Relationships. ***Journal of Marketing***, (61): 35–51.

Doney, P., Cannon, J., & Mullen, M. 1998. Understanding the Influence of National Culture on the Development of Trust. ***Academy of Management Review***, 23(3): 601–620.

Donmez, A., & Karaivanov, A. 2021. Transaction fee economics in the Ethereum blockchain. ***Economic Inquiry***, 60: 265–292.

Dutra, A., Tumasjan, A., & Welpe, I. 2018. Blockchain is changing how media and entertainment companies compete. ***MIT Sloan Management Review***: 39–46.

Easley, D., O'Hara, M., & Basu, S. 2019. From mining to markets: The evolution of bitcoin transaction fees. ***Journal of Financial Economics***, 134(1): 91–109.

Economides, N., & Himmelberg, C. 1995. "Critical Mass and Network Size with Application to the US FAX Market. ***Mimeo, Stern School of Business at New York University)***, forthcoming.

Economist. 2015. The trust machine: The promise of the blockchain. ***The Economist***, forthcoming.

Economist. 2017. ***If blockchain ran the world - Disrupting the trust business***.

Egelund-Müller, B., Elsman, M., Henglein, F., & Ross, O. 2017. Automated Execution of Financial Contracts on Blockchains. ***Business & Information Systems Engineering***, 59(6): 457–467.

Elsbach, K., & Elofson, G. 2000. How the packaging of decision explanations affects perceptions of trustworthiness. ***Academy of Management Journal***, 43(1): 80–89.

Esposito Vinzi, V., Chin, W. W., Henseler, J., & Wang, H. 2010. ***Handbook of Partial Least Squares***. Berlin, Heidelberg: Springer Berlin Heidelberg.

Faems, D., Janssens, M., Madhok, A., & van Looy, B. 2008. Toward an integrative perspective on alliance governance: Connecting contract design, trust dynamics, and contract application. ***Academy of Management Journal***, 51: 1053–1078.

Fang, Y., Qureshi, I., Sun, H., McCole, P., Ramsey, E., & Lim, K. H. 2014. Trust, Satisfaction, and Online Repurchase Intention. ***MIS Quarterly***, 38(3).

Farrell, J., & Saloner, G. 1986. Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation. ***The American Economic Review***, 76(5): 940–955.

Fishbein, M., & Ajzen, I. 1975. ***Belief, attitude, intention and behavior: An introduction to theory and research***. Reading MA: Addison-Wesley.

Fiske, S., & Taylor, S. 1991. ***Social cognition.*** MacGraw-Hill series in social psychology. New York, NY: MacGraw-Hill.

Foley, S., Karlsen, J., & Putniņš, T. 2019. Sex, Drugs, and Bitcoin: How Much Illegal Activity Is Financed through Cryptocurrencies? ***The Review of Financial Studies***, 32(5): 1798–1853.

Fornell, C., & Bookstein, F. 1982. Two Structural Equation Models: LISREL and PLS Applied to Consumer Exit-Voice Theory. ***Journal of Marketing Research***, 18(4): 440–452.

Fornell, C., & Larcker, D. 1981. Evaluating Structural Equation Models with Unobservable Variables and Measurement Error. ***Journal of Marketing Research***, 18(1): 39–50.

Fox, R., Crask, M., & Kim, J. 1988. Mail Survey Response Rate: A Meta-Analysis of Selected Techniques for Inducing Response. ***Public Opinion Quarterly***, 52(4): 467.

Friedlmaier, M., Tumasjan, A., & Welpe, I. 2016. ***Disrupting Industries With Blockchain: The Industry, Venture Capital Funding, and Regional Distribution of Blockchain Ventures***.

Froehlich, M., Hulm, P., & Alt, F. Under Pressure. A User-Centered Threat Model for Cryptocurrency Owners. In ***2021 4th International Conference on Blockchain Technology and Applications***.

Fröwis, M., & Böhme, R. 2017. In code we trust? Measuring the control flow immutability of all smart contracts deployed on Ethereum. In J. Garcia-Alfaro, G. Navarro-Arribas, H. Hartenstein & J. Herrera-Joancomartí (Eds.), ***Data Privacy Management, Cryptocurrencies and Blockchain Technology***: 357–372. Cham: Springer International Publishing.

Gabarro, J. 1978. The development of trust, influence and expectations. ***Interpersonal behavior Communication and understanding in relationships***: 290–303.

Gale, D. 1955. The Law of Supply and Demand, Vol. 3. ***Mathematica Scandinavica***, 3(1): 155–169.

Gandal, N. 1994. Hedonic Price Indexes for Spreadsheets and an Empirical Test for Network Externalities. ***The RAND Journal of Economics***, 25(1): 160.

Gefen, D., Benbasat, I., & Pavlou, P. 2008. A research agenda for trust in online environments. ***Journal of Management Information Systems***, 24(4): 275–286.

Gefen, D., Karahanna, E., & Straub, D. W. 2003. Trust and TAM in online shopping: An integrated model. ***MIS Quarterly***, 27(1): 51–90.

Gefen, D., & Straub, D. 2004. Consumer trust in B2C e-Commerce and the importance of social presence: experiments in e-Products and e-Services. ***Omega***, 32(6): 407–424.

Gerbing, D., & Anderson, J. 1988. An Updated Paradigm for Scale Development Incorporating Unidimensionality and Its Assessment. ***Journal of Marketing Research***, 25(2): 186–192.

Glaser, F. 2017. Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain enabled System and Use Case Analysis. ***Proceedings of the 50th Hawaii International Conference on System Sciences***, forthcoming.

Götz, O., Liehr-Gobbers, K., & Krafft, M. 2010. Evaluation of Structural Equation Models Using the Partial Least Squares (PLS) Approach. In V. Esposito Vinzi, W. W. Chin, J. Henseler & H. Wang (Eds.), ***Handbook of Partial Least Squares***: 691–711. Berlin, Heidelberg: Springer Berlin Heidelberg.

Grabner-Kräuter, S., & Kaluscha, E. 2003. Empirical research in on-line trust: a review and critical assessment. ***International Journal of Human-Computer Studies***, 58(6): 783–812.

Granovetter, M. 1985. Economic Action and Social Structure: The Problem of Embeddedness. ***American Journal of Sociology***, 91(3): 481–510.

Greiner, M., & Wang, H. 2015. Trust-free systems - a new research and design direction to handle trust-issues in P2P systems: The case of bitcoin. ***Twenty-first Americas Conference on Information Systems***, forthcoming.

Gulati, R. 1995. Does familiarity breed trust? The implications of repeated ties for contractual choice in alliances. ***Academy of Management Journal***, 38(1): 85–112.

Gulati, R., & Nickerson, J. 2008. Interorganizational trust, governance choice, and exchange performance. ***Organization Science***, 19(5): 688–708.

Gulati, R., Wohlgezogen, F., & Zhelyazkov, P. 2012. The Two Facets of Collaboration: Cooperation and Coordination in Strategic Alliances. ***Academy of Management Annals***, 6(1): 531–583.

Guo, W., Straub, D., Zhang, P., & Cai, Z. 2021. How Trust Leads to Commitment on Microsourcing Platforms: Unraveling the Effects of Governance and Third-party Mechanisms on Triadic Microsourcing Relationships. ***MIS Quarterly***, 45(3): 1309–1348.

Haber, S., & Stornetta, S. 1991. How to time-stamp a digital document. ***Journal of Cryptology***, (3): 99–111.

Hair, J., Hult, T., Ringle, C., Sarstedt, M., Richter, N., & Hauff, S. 2017. ***Partial Least Squares Strukturgleichungsmodellierung*****:** ***Eine anwendungsorientierte Einführung*** (1. Auflage). Munich: Vahlen.

Halaburda, H. 2018. Blockchain revolution without the blockchain? ***Communications of the ACM***, 61(7): 27–29.

Halaburda, H., Levina, N., & Min, S. 2019. Understanding smart contracts as a new option in transaction cost economics. ***Proceedings of the 40th International Conference on Information Systems, Munich, Germany***, forthcoming.

Hawlitschek, F., Notheisen, B., & Teubner, T. 2018. The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy. ***Electronic Commerce Research and Applications***, 29: 50–63.

Henseler, J., Ringle, C., & Sarstedt, M. 2015. A new criterion for assessing discriminant validity in variance-based structural equation modeling. ***Journal of the Academy of Marketing Science***, 43(1): 115–135.

Hoetker, G. 2005. How much you know versus how well I know you: selecting a supplier for a technically innovative component. ***Strategic Management Journal***, 26(1): 75–96.

Hoetker, G., & Mellewigt, T. 2009. Choice and performance of governance mechanisms: matching alliance governance to asset type. ***Strategic Management Journal***, 30(10): 1025–1044.

Holmes, J. 1991. Trust and the appraisal process in close relationships. In ***Advances in personal relationships: A research annual, Vol. 2***: 57–104. Oxford, UK: Jessica Kingsley Publishers.

Hosmer, L. 1995. Trust: The Connecting Link between Organizational Theory and Philosophical Ethics. ***Academy of Management Review***, 20(2): 379–403.

Houy, N. 2016. The Bitcoin Mining Game. ***Ledger***, 1: 53–68.

Howell, R., Breivik, E., & Wilcox, J. 2007. Is formative measurement really measurement? Reply to Bollen (2007) and Bagozzi (2007). ***Psychological Methods***, 12(2): 238–245.

Hsieh, Y.-Y., & Vergne, J.-P. 2022. The future of the web? The coordination and early-stage growth of decentralized platforms. ***Strategic Management Journal***, 44(3): 829–857.

Hsieh, Y.-Y., Vergne, J.-P., Anderson, P., Lakhani, K., & Reitzig, M. 2018. Bitcoin and the rise of decentralized autonomous organizations. ***J. Org. Design***, 7(1): 709.

Huang, P., Ceccagnoli, M., Forman, C., & Wu, D. J. 2013. Appropriability Mechanisms and the Platform Partnership Decision: Evidence from Enterprise Software. ***Management Science***, 59(1): 102–121.

Huber, T., Fischer, T., Dibbern, J., & Hirschheim, R. 2013. A Process Model of Complementarity and Substitution of Contractual and Relational Governance in IS Outsourcing. ***Journal of Management Information Systems***, 30(3): 81–114.

Huberman, G., Leshno, J., & Moallemi, C. 2017. Monopoly Without a Monopolist: An Economic Analysis of the Bitcoin Payment System. ***SSRN Electronic Journal***, forthcoming.

Ilk, N., Shang, G., Fan, S., & Zhao, J. 2021. Stability of Transaction Fees in Bitcoin: A Supply and Demand Perspective. ***MIS Quarterly***, 45(2): 563–692.

Inkpen, A., & Currall, S. 2004. The coevolution of trust, control, and learning in joint ventures. ***Organization Science***, 15: 586–599.

Jarvenpaa, S., Tractinsky, N., & Vitale, M. 2000. Consumer trust in an Internet store. ***Information Technology and Management***: 45–71.

Jarvis, C., MacKenzie, S., & Podsakoff, P. 2003. A Critical Review of Construct Indicators and Measurement Model Misspecification in Marketing and Consumer Research. ***Journal of Consumer Research***, 30(2): 199–218.

Johnson-George, C., & Swap, W. 1982. Measurement of specific interpersonal trust: Construction and validation of a scale to assess trust in a specific other. ***Journal of Personality and Social Psychology***, 43: 1306–1317.

Joinson, A. 1999. Social desirability, anonymity, and Internet-based questionnaires. ***Behavior research methods, instruments, & computers a journal of the Psychonomic Society, Inc***, 31(3): 433–438.

Karahanna, E., Straub, D., & Chervany, N. 1999. Information Technology Adoption Across Time: A Cross-Sectional Comparison of Pre-Adoption and Post-Adoption Beliefs. ***MIS Quarterly***, 23(2): 183.

Katz, M., & Shapiro, C. 1985. Network Externalities, Competition, and Compatibility. ***The American Economic Review***, (75): 424–440.

Keen, P., Balance, C., Chan, S., & Schrump, S. 2000. ***Electronic commerce relationships: Trust by design***. Englewood Cliffs, NJ: Prentice-Hall.

Kell, T., Yousaf, H., Allen, S., Meiklejohn, S., & Juels, A. 2021. ***Forsage: Anatomy of a Smart-Contract Pyramid Scheme***. arXiv.

Kennedy, P. 2011. ***A guide to econometrics*** (6th ed.). Malden, Mass.: Blackwell.

Kim, D., Song, Y., Braynov, S., & Rao, H. 2005. A multidimensional trust formation model in B-to-C e-commerce: a conceptual framework and content analyses of academia/practitioner perspectives. ***Decision Support Systems***, 40(2): 143–165.

Kim, H.-W., Xu, Y., & Koh, J. 2004. A Comparison of Online Trust Building Factors between Potential Customers and Repeat Customers. ***Journal of the Association for Information Systems***, 5(10): 392–420.

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. 2016. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. ***IEEE Symposium on Security and Privacy, SP 2016***: 839–858.

Kroll, J., Davey, I., & Felten, E. 2013. The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries. ***12th Workshop on the Economics of Information Security (WEIS 2013) Washington, DC***: 1–21.

Kumar, N., Scheer, L., & Steenkamp, J.-B. 1995. The Effects of Supplier Fairness on Vulnerable Resellers. ***Journal of Marketing Research***, 32(1): 54–65.

Lado, A., Dant, R., & Tekleab, A. 2008. Trust-opportunism paradox, relationalism, and performance in interfirm relationships: Evidence from the retail industry. ***Strategic Management Journal***, 29: 401–423.

Langer, E. 1975. The illusion of control. ***Journal of Personality and Social Psychology***, 32(2): 311–328.

Lauer, T., & Deng, X. 2007. Building online trust through privacy practices. ***International Journal of Information Security***, 6(5): 323–331.

Lavi, R., Sattath, O., & Zohar, A. 2017. Redesigning Bitcoin's fee market, forthcoming.

Leiponen, A., Thomas, L., & Wang, Q. 2021. The dApp economy: a new platform for distributed innovation? ***Innovation***: 1–19.

Lewicki, R., & Bunker, B. 1996. Developing and maintaining trust in work relationships. In R. M. Kramer & T. R. Tyler (Eds.), ***Trust in organizations***. Thousand Oaks: CA: Sage.

Lewis, J., & Weigert, A. 1985. Trust as a social reality. ***Social Forces***, 63: 967–985.

Li, J., Poppo, L., & Zhou, K. Z. 2010. Relational mechanisms, formal contracts, and local knowledge acquisition by international subsidiaries. ***Strategic Management Journal***, 31(4): 349-370.

Li, T., Shin, D., & Wang, B. 2018. Cryptocurrency Pump-and-Dump Schemes. ***SSRN Electronic Journal***, forthcoming.

Lim, K., Sia, C., Lee, M., & Benbasat, I. 2006. Do I Trust You Online, and If So, Will I Buy? An Empirical Study of Two Trust-Building Strategies. ***Journal of Management Information Systems***, 23(2): 233–266.

Lin, S. 2020. Two-Sided Price Discrimination by Media Platforms. ***Marketing Science***, 39(2): 317–338.

Lioukas, C., & Reuer, J. 2015. Isolating trust outcomes from exchange relationships: Social exchange and learning benefits of prior ties in alliances. ***Academy of Management Journal***, 58: 1826–1847.

Liu, B., & Goodhue, D. 2012. Two Worlds of Trust for Potential E-Commerce Users: Humans as Cognitive Misers. ***Information Systems Research***, 23(4): 1246–1262.

Liu, Q., & Serfes, K. 2013. Price Discrimination in Two-Sided Markets. ***Journal of Economics & Management Strategy***, 22(4): 768–786.

Luhmann, N. 1979. ***Trust and power***. Chichester UK: John Wiley & Sons.

Lumineau, F., Wang, W., & Schilke, O. 2020. Blockchain governance-a new way of organizing collaborations? ***Organization Science***, 32(2): 500–521.

Macneil, I. 1977. Contracts: Adjustment of long-term economic relations under classical, neoclassical, and relational contract law. ***Nw. UL Rev.***, 72: 854.

Maese, V., Avery, A., Naftalis, B., Wink, S., & Valdez, Y. 2016. Cryptocurrency: A primer. ***Banking LJ***, 133: 468.

Malhotra, D., & Lumineau, F. 2011. Trust and Collaboration in the Aftermath of Conflict: The Effects of Contract Structure. ***Academy of Management Journal***, 54(5): 981–998.

Marcoulides, G., & Saunders, C. 2006. Editor's Comments: PLS: A Silver Bullet? ***MIS Quarterly***, 30(2): iii–ix.

Markovich, S., & Moenius, J. 2009. Winning while losing: Competition dynamics in the presence of indirect network effects. ***International Journal of Industrial Organization***, 27(3): 346–357.

Maume, P., & Fromberger, M. 2019. Regulation of Initial Coin Offerings: Reconciling U.S. and E.U. Securities Laws. ***Chicago Journal of International Law***, 19(19): 548–585.

Mayer, R., Davis, J., & Schoorman, D. 1995. An integrative model of organizational trust. ***Academy of Management Review***, 20(3): 709–734.

Mayer, R., & Gavin, M. 2005. Trust in management and performance: Who minds the shop while the employees watch the boss? ***Academy of Management Journal***, 48(5): 874–888.

McEvily, B. 2011. Reorganizing the boundaries of trust: From discrete alternatives to hybrid forms. ***Organization Science***, 22: 1266–1276.

McKnight, H., & Chervany, N. 2001. What trust means in e-commerce customer relationships: An interdisciplinary conceptual typology. ***International Journal of Electronic Commerce***, 6(2): 35–59.

McKnight, H., Choudhury, V., & Kacmar, C. 2002a. Developing and validating trust measures for e-commerce: An integrative typology. ***Information Systems Research***, 13(3): 334–359.

McKnight, H., Choudhury, V., & Kacmar, C. 2002b. The impact of initial consumer trust on intentions to transact with a web site: A trust building model. ***Journal of Strategic Information Systems***, 11: 297–323.

McKnight, H., Cummings, L., & Chervany, N. 1998. Initial trust formation in new organizational relationships. ***Academy of Management Review***, 23(3): 473–490.

Mehrwald, P., Treffers, T., Titze, M., & Welpe, I. 2019. Blockchain Technology Application in the Sharing Economy: A Proposed Model of Effects on Trust and Intermediation. ***Proceedings of the Annual Hawaii International Conference on System Sciences***. Hawaii International Conference on System Sciences.

Molina-Morales, X., & Martínez-Fernández, T. 2009. Too much love in the neighborhood can hurt: How an excess of intensity and trust in relationships may produce negative effects on firms. ***Strategic Management Journal***, 30: 1013–1023.

Morgan, R., & Hunt, S. 1994. The Commitment-Trust Theory of Relationship Marketing. ***Journal of Marketing***, 58(3): 20–38.

Murray, A., Kuban, S., Josefy, M., & Anderson, J. 2019. Contracting in the Smart Era: The Implications of Blockchain and Decentralized Autonomous Organizations for Contracting and Corporate Governance. ***Academy of Management Perspectives***, forthcoming.

Nakamoto, S. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. Downloaded on April 15, 2021, from: https://bitcoin.org/bitcoin.pdf.

Narayanan, A., & Clark, J. 2017. Bitcoin's academic pedigree. ***Communications of the ACM***, 60(12): 36–45.

Nederhof, A. 1985. Methods of coping with social desirability bias: A review. ***European Journal of Social Psychology***, 15(3): 263–280.

Nooteboom, B. 1996. Trust, opportunism and governance: A process and control model. ***Organization Studies***, 17: 985–1010.

Notheisen, B., Cholewa, J., & Shanmugam, A. 2017. Trading real-world assets on blockchain. ***Business & Information Systems Engineering***, 59(6): 425–440.

Okhuysen, G., & Bechky, B. 2009. Coordination in Organizations: An Integrative Perspective. ***Academy of Management Annals***, 3(1): 463–502.

O'Mahony, S., & Karp, R. 2020. From proprietary to collective governance: How do platform participation strategies evolve? ***Strategic Management Journal***, 43(3): 530–562.

Palmer, J., Bailey, J., & Faraj, S. 2000. The Role of Intermediaries in the Development of Trust on the WWW: The Use and Prominence of Trusted Third Parties and Privacy Statements. ***Journal of Computer-Mediated Communication***, 5(3): 0.

Panico, C., & Cennamo, C. 2020. User preferences and strategic interactions in platform ecosystems. ***Strategic Management Journal***, 43(3): 507–529.

Pavlou, P. 2002. Institution-based trust in interorganizational exchange relationships: The role of online B2B marketplaces on trust formation. ***Journal of Strategic Information Systems***, 11: 215–243.

Pavlou, P., & Gefen, D. 2004. Building effective online marketplaces with institution-based trust. ***Information Systems Research***, 15: 37–59.

Pereira, J., Tavalaei, M., & Ozalp, H. 2019. Blockchain-based platforms: Decentralized infrastructures and its boundary conditions. ***Technological Forecasting and Social Change***, 146: 94–102.

Peter, J. P., & Tarpey, S. L. X. 1975. A Comparative Analysis of Three Consumer Decision Strategies. ***Journal of Consumer Research***, 2(1): 29.

Petty, R., & Cacioppo, J. 1986. The Elaboration Likelihood Model of Persuasion. ***Advances in Experimental Social Psychology***, 19: 123–205.

Podsakoff, P., MacKenzie, S., Lee, J.-Y., & Podsakoff, N. 2003. Common method biases in behavioral research: a critical review of the literature and recommended remedies. ***The Journal of applied psychology***, 88(5): 879–903.

Podsakoff, P., MacKenzie, S., Podsakoff, N., & Lee, J. Y. 2003. The mismeasure of man(agement) and its implications for leadership research. ***The Leadership Quarterly***, 14(6): 615–656.

Polidoro, F., Ahuja, G., & Mitchell, W. 2011. When the social structure overshadows competitive incentives: The effects of network embeddedness on joint venture dissolution. ***Academy of Management Journal***, 54: 203–223.

Poppo, L., & Cheng, Z. 2018. Complements versus substitutes in business-to-business exchanges. ***The Routledge Companion to Trust***, forthcoming.

Poppo, L., & Zenger, T. 2002. Do formal contracts and relational governance function as substitutes or complements? ***Strategic Management Journal***, 23(8): 707–725.

Poppo, L., Zhou, K. Z., & Li, J. 2016. When can you trust “trust”? Calculative trust, relational trust, and supplier performance. ***Strategic Management Journal***, 37(4): 724–741.

Poppo, L., Zhou, K. Z., & Ryu, S. 2008. Alternative Origins to Interorganizational Trust: An Interdependence Perspective on the Shadow of the Past and the Shadow of the Future. ***Organization Science***, 19(1): 39–55.

Ratnasingam, P., & Pavlou, P. A. 2002. Technology Trust: The Next Value Creator in B2B Electronic Commerce. ***Proceedings of the 2002 IRMA International Conference, Seattle, WA***: 19–22.

Rempel, J., Holmes, J., & Zanna, M. 1985. Trust in close relationships. ***Journal of Personality and Social Psychology***, 49(1): 95–112.

Ridings, C. M., Gefen, D., & Arinze, B. 2002. Some antecedents and effects of trust in virtual communities. ***Journal of Strategic Information Systems***, 11: 271–295.

Riegelsberger, J., Sasse, A., & McCarthy, J. 2005. The mechanics of trust: A framework for research and design. ***International Journal of Human-Computer Studies***, 62(3): 381–422.

Rietveld, J., & Schilling, M. 2020. Platform Competition: A Systematic and Interdisciplinary Review of the Literature. ***Journal of Management***: (in press).

Riggins, F., Kriebel, C., & Mukhopadhyay, T. 1994. The Growth of Interorganizational Systems in the Presence of Network Externalities. ***Management Science***, 40(8): 984–998.

Riker, W. 2017. The Nature of Trust. In J. T. Tedeschi (Ed.), ***Social power and political influence***: 63–81. Abingdon, Oxon, New York, NY: Routledge, Taylor & Francis Group.

Ring, P., & van de Ven, A. 1994. Developmental process of copperative interorganizational relationships. ***Academy of Management Review***, (19): 90–118.

Ringle, C., Wende, S., & Becker, J. 2015. ***SmartPLS 3***. Boenningstedt: SmartPLS GmbH.

Risius, M., & Spohrer, K. 2017. A blockchain research framework: What we (don't) know, where we go from here, and how we will get there. ***Business & Information Systems Engineering***, 59: 385–409.

Roberts, K., & O'Reilly, C. 1974. Measuring organizational communication. ***Journal of Applied Psychology***, 59: 321–326.

Rogers, E. 1995. ***Diffusion of innovations*** (4th ed.). New York, London, Toronto: Free Press.

Roughgarden, T. 2020. Transaction Fee Mechanism Design for the Ethereum Blockchain: An Economic Analysis of EIP-1559, forthcoming.

Rousseau, D., Sitkin, S., Burt, R., & Camerer, C. 1998. Not so different after all - a cross discipline view of trust. ***Academy of Management Review***, 23: 393–404.

Rückeshäuser, N. 2017. Typology of distributed ledger based business models. ***Proceedings of the 25th European Conference on Information Systems (ECIS)***: 2202–2217.

Salvato, C., Reuer, J., & Battigalli, P. 2017. Cooperation across Disciplines: A Multilevel Perspective on Cooperative Behavior in Governing Interfirm Relations. ***Academy of Management Annals***, 11(2): 960–1004.

Sapirshtein, A., Sompolinsky, Y., & Zohar, A. 2016. Optimal Selfish Mining Strategies in Bitcoin, forthcoming.

Sarkar, S., Chauhan, S., & Khare, A. 2020. A meta-analysis of antecedents and consequences of trust in mobile commerce. ***International Journal of Information Management***, 50: 286–301.

Schaubroeck, J., Peng, A., & Hannah, S. 2013. Developing trust with peers and leaders: impacts on organizational identification and performance during entry. ***Academy of Management Journal***, 56: 1148–1168.

Schepker, D., Oh, W.-Y., Martynov, A., & Poppo, L. 2014. The many futures of contracts: Moving beyond structure and safeguarding to coordination and adaptation. ***Journal of Management***, 40(1): 193–225.

Seidel, M.-D. 2018. Questioning Centralized Organizations in a Time of Distributed Trust. ***Journal of Management Inquiry***, 27(1): 40–44.

Shapiro, C., & Varian, H. 2010. ***Information rules*: *A strategic guide to the network economy*** ([Nachdr.]). Boston, MA: Harvard Business School Press.

Shapiro, D., Sheppard, B., & Cheraskin, L. 1992. Business on a Handshake. ***Negotiation Journal***, (8): 365–377.

Sheldon, M. 2021. Auditing the Blockchain Oracle Problem. ***Journal of Information Systems***, 35(1): 121–133.

Simmel, G. 1950. ***The Sociology of Georg Simmel***. Translated, edited and with an introduction by Kurt H. Wolff. New York: Free Press.

Simon, H. 1957. ***Models of man; social and rational***. Oxford, UK: Wiley.

Sitkin, S., & Roth, N. 1993. Explaining the limited effectiveness of legalistic "remedies" for trust/distrust. ***Organization Science***, (4): 367–392.

Spain, M., Foley, S., & Gramoli, V. 2020. ***The Impact of Ethereum Throughput and Fees on Transaction Latency During ICOs***.

Srinivasan, R., & Brush, T. 2006. Supplier Performance in Vertical Alliances: The Effects of Self-Enforcing Agreements and Enforceable Contracts. ***Organization Science***, 17(4): 436–452.

Sternberg, R., & Mio, J. 2009. ***Cognitive psychology*** (International student ed., 5. ed.). Belmont, CA: Wadsworth.

Stewart, K. J. 2003. Trust Transfer on the World Wide Web. ***Organization Science***, 14(1): 5–17.

Stock, J., & Yogo, M. 2005. Testing for Weak Instruments in Linear IV Regression. In J. H. Stock & D. W. K. Andrews (Eds.), ***Identification and inference for econometric models: Essays in honor of Thomas Rothenberg*** (Paperback): 80–108. New York: Cambridge university press.

Szabo, N. 1994. Smart contracts. Downloaded on April 15, 2021, from: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html.

Tapscott, D., & Tapscott, A. 2018. ***Blockchain revolution*: *How the technology behind Bitcoin and cryptocurrency is changing the world*** (Updated edition). London: Portfolio Penguin.

Tsai, W., & Ghoshal, S. 1998. Social capital and value creation: The role of intrafirm networks. ***Academy of Management Journal***, 41: 464–476.

Tudón, J. 2022. Prioritization vs. congestion on platforms: evidence from Amazon's Twitch.tv. ***The RAND Journal of Economics***, 53(2): 328–355.

Uzzi, B. 1997. Social Structure and Competition in Interfirm Networks: The Paradox of Embeddedness. ***Administrative Science Quarterly***, 42(1): 35.

Vergne, J. P. 2020. Decentralized vs. Distributed Organization: Blockchain, Machine Learning and the Future of the Digital Platform. ***Organization Theory***, 1(4): 1-26.

Voshmgir, S. 2020. ***Token economy*: *How the Web3 reinvents the internet*** (2nd ed.). Berlin: Berlin; BlockchainHub.

Vukolić, M. 2016. The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In J. Camenisch & D. Kesdoğan (Eds.), ***Open Problems in Network Security***: 112–125. Cham: Springer International Publishing.

Wang, Z., & Wright, J. 2017. Ad valorem platform fees, indirect taxes, and efficient price discrimination. ***The RAND Journal of Economics***, 48(2): 467–484.

Weber, L. 2017. A Sociocognitive View of Repeated Interfirm Exchanges: How the Coevolution of Trust and Learning Impacts Subsequent Contracts. ***Organization Science***, 28(4): 744–759.

Weber, L., & Bauman, C. 2019. The cognitive and behavioral impact of promotion and prevention contracts on trust in repeated exchanges. ***Academy of Management Journal***, 62(2): 361–382.

Weiss, J., & Obermeier, D. 2021. How Blockchain Can Enhance Trust and Transparency of Online Surveys. ***ICIS 2021 Proceedings***, forthcoming.

Werbach, K. 2018. ***The blockchain and the new architecture of trust.*** Information policy series. Cambridge, MA, London, UK: MIT Press.

Wilcox, J., Howell, R., & Breivik, E. 2008. Questions about formative measurement. ***Journal of Business Research***, 61(12): 1219–1228.

Williamson, O. 1985. ***The Economic Institutions of Capitalism*** 36. New York: Free Press.

Williamson, O. 1993. Calculativeness, Trust, and Economic Organization. ***The Journal of Law and Economics***, 36(1, Part 2): 453–486.

Wold, H. 1985. Partial least squares. S. Kotz and NL Johnson (Eds.), Encyclopedia of statistical sciences (vol. 6). ***Wiley, New York***, forthcoming.

Wood, G. 2014a. ÐApps: What Web 3.0 Looks Like, from: http://gavwood.com/dappsweb3.html.

Wood, G. 2014b. ***Ethereum: A secure decentralised generalised transaction ledger***.

Wooldridge, J. 2010. ***Econometric analysis of cross section and panel data*** (2nd ed.). Cambridge, MA: MIT Press.

Wrightsman, L. 1991. Interpersonal Trust and Attitudes toward Human Nature. In ***Measures of Personality and Social Psychological Attitudes***: 373–412. Elsevier.

Wu, K., Ma, Y., Huang, G., & Liu, X. 2021. A first look at blockchain-based decentralized applications. ***Software: Practice and Experience***, 51(10): 2033–2050.

Yermack, D. 2017. Corporate Governance and Blockchains. ***Review of Finance***, 21(1): 7-31.

Zaheer, A., McEvily, B., & Perrone, V. 1998. Does trust matter? Exploring the effects of interorganizational and interpersonal trust on performance. ***Organization Science***, 9(2): 141–159.

Zaheer, A., & Venkatraman, N. 1995. Relational governance as an interorganizational strategy: An empirical test of the role of trust in economic exchange. ***Strategic Management Journal***, 16: 373–392.

Zetzsche, D., Buckley, R., Arner, D., & Foehr, L. 2017. The ICO Gold Rush: It's a Scam, It's a Bubble, It's a Super Challenge for Regulators. ***SSRN Electronic Journal***, forthcoming.

Zhang, P., Xiao, F., & Luo, X. 2020. A Framework and DataSet for Bugs in Ethereum Smart Contracts. ***2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)***. IEEE.

Zhou, K., & Poppo, L. 2010. Exchange hazards, relational reliability, and contracts in China: The contingent role of legal enforceability. ***Journal of International Business Studies***, 41(5): 861–881.

Zhou, T. 2011. An empirical examination of initial trust in mobile banking. ***Internet Research***, 21: 527–540.

Zhou, T. 2012. Understanding users' initial trust in mobile banking: An elaboration likelihood perspective. ***Computers in Human Behavior***, 28(4): 1518–1525.

Zucker, L. G. 1986. Production of trust: institutional sources of economic structure 1840-1920. ***Research in Organization Behavior***, 8(1): 53–111.