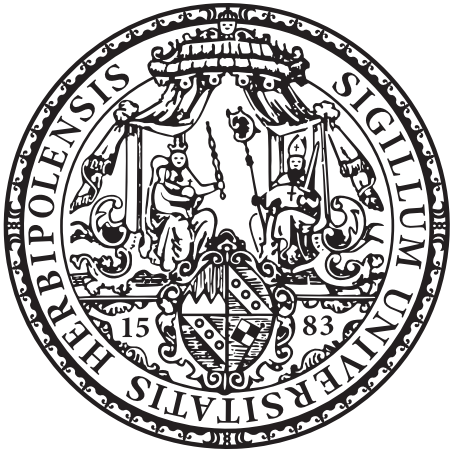
Julius-Maximilians-Universität Würzburg

Wirtschaftswissenschaftliche Fakultät



Inaugural Dissertation

To obtain the academic degree

Doctor rerum politicarum (Dr. rer. pol.)

Challenges and Solution Approaches for Blockchain Technology

Adrian Hofmann

May 06, 2022

Author:
Adrian Cornelius Sylvester Hofmann, M. Sc.

Am Sonnfeld 6
97076 Würzburg

1. Supervisor:
Prof. Dr. A. Winkelmann

2. Supervisor:
Prof. Dr. C. Flath

Acknowledgments

My first thanks go to my doctoral advisor, Prof. Dr. Axel Winkelmann. You gave me the freedom to pursue my own research but still challenged me to improve myself. You always provided me with opportunities to show my strength or confront my weaknesses. Thank you for all the support you provided me over the last four years.

Secondly, I want to thank Prof. Dr. Christoph Flath for serving as my second supervisor, for his critical view, and always inspiring talks about any topic, whether research-related or other nerd topics.

I want to thank all my co-authors, without whom this thesis would have been impossible. Here special thanks go to Dr. Marcus Fischer and Dr. Florian Imgrund, who involved me in their research from day one. I want to thank Prof. Dr. Christian Janiesch. We had different opinions on many things, which helped me grow as a researcher. I also want to thank my other co-authors, Fabian Schatz, Fabian Gwinner, Julian Kolb, Luc Becker, and Norman Pytel. We had great synergies, which resulted in great research results.

Next, I want to thank the rest of my colleagues at the Chair of Business Management and Business Information Systems. I am thankful to have been able to work with such a diverse team. Together we mastered many challenges and had much fun along the way. Here my special thank goes to Andrea Müller. I am sure that I caused you many headaches, and I am very thankful that you still always helped me with a smile.

I especially want to thank Jonas Wanner for supporting me as a colleague, friend, and business partner. Our journey together is far from over.

I also want to thank all the project partners in the research projects DiHP and PIMKoWe, as well as the Projektträger Karlsruhe (PTKA) and the Bundesministerium für Bildung und Forschung (BMBF) who supervised and funded these research projects.

Finally, I want to thank my colleague and dearest friend Chiara Freichel. Our joint research projects, long nights working or studying were always fruitful. This thesis would not have been half as fun without your emotional and organizational support.

Abstract

The digital transformation facilitates new forms of collaboration between companies along the supply chain and between companies and consumers. Besides sharing information on centralized platforms, blockchain technology is often regarded as a potential basis for this kind of collaboration. However, there is much hype surrounding the technology due to the rising popularity of cryptocurrencies, decentralized finance (DeFi), and non-fungible tokens (NFTs). This leads to potential issues being overlooked. Therefore, this thesis aims to investigate, highlight, and address the current weaknesses of blockchain technology: Inefficient consensus, privacy, smart contract security, and scalability.

First, to provide a foundation, the four key challenges are introduced, and the research objectives are defined, followed by a brief presentation of the preliminary work for this thesis. The following four parts highlight the four main problem areas of blockchain. Using big data analytics, we extracted and analyzed the blockchain data of six major blockchains to identify potential weaknesses in their *consensus* algorithm. To improve *smart contract security*, we classified smart contract functionalities to identify similarities in structure and design. The resulting taxonomy serves as a basis for future standardization efforts for security-relevant features, such as safe math functions and oracle services. To challenge *privacy* assumptions, we researched consortium blockchains from an adversary role. We chose four blockchains with misconfigured nodes and extracted as much information from those nodes as possible. Finally, we compared *scalability* solutions for blockchain applications and developed a decision process that serves as a guideline to improve the scalability of their applications.

Building on the scalability framework, we showcase three potential applications for blockchain technology. First, we develop a token-based approach for inter-company value stream mapping. By only relying on simple tokens instead of complex smart-contracts, the computational load on the network is expected to be much lower compared to other solutions. The following two solutions use offloading transactions and computations from the main blockchain. The first approach uses secure multiparty computation to offload the matching of supply and demand for manufacturing capacities to a trustless network. The transaction is written to the main blockchain only after the match is made. The second approach uses the concept of payment channel networks to enable high-frequency bidirectional micropayments for WiFi sharing. The host gets paid for every second of data usage through an off-chain channel. The full payment is only written to the blockchain after the connection to the client gets terminated.

Finally, the thesis concludes by briefly summarizing and discussing the results and providing avenues for further research.

Zusammenfassung

Die digitale Transformation ermöglicht neue Formen der Zusammenarbeit zwischen Unternehmen entlang der Wertschöpfungskette, sowie zwischen Unternehmen und Verbrauchern. Neben dem Austausch von Informationen auf zentralen Plattformen wird die Blockchain-Technologie häufig als mögliche Grundlage für diese Art der Zusammenarbeit angesehen. Allerdings gibt es auch durch die steigende Popluarität von Cryptowährungen, Decentralized Finance (DeFi) und non-fungible Tokens (NFTs) einen großen Hype um die Technologie. Dieser führt dazu, dass mögliche Probleme übersehen werden. Daher sollen in dieser Thesis die derzeitigen Schwachstellen der Blockchain-Technologie aufgezeigt und ausgewählte Maßnahmen zur Verbesserung aufgezeigt werden. Es werden dabei vier Schwachstellen betrachtet: Ineffizienter Konsens, Datenschutz, Sicherheit von Smart Contracts und Skalierbarkeit.

Als Grundlage werden zunächst die vier zentralen Herausforderungen vorgestellt und die Forschungsziele definiert. Es folgt eine kurze Darstellung der Vorarbeiten für diese Arbeit. In den folgenden vier Teilen wird für jedes der vier Hauptproblembereiche von Blockchain ein spezifisches Problem beleuchtet: Ineffizienter Konsens, Datenschutz, Smart-Contract-Sicherheit und Skalierbarkeit. Mit Hilfe von Big-Data-Analytics wurden die Blockchain-Daten von sechs großen Blockchains extrahiert und analysiert, um potenzielle Schwachstellen in deren *Konsens*-Algorithmus zu identifizieren. Um die Sicherheit von Smart Contracts zu verbessern, wurden die Funktionalitäten von Smart Contracts klassifiziert, um Ähnlichkeiten in Struktur und Design zu identifizieren. Die sich daraus ergebende Taxonomie dient als Grundlage für künftige Standardisierungsbemühungen für sicherheitsrelevante Funktionen, wie z. B. sichere mathematische Funktionen und Orakeldienste. Um die Annahmen von *Datenschutz* in Frage zu stellen, wurden Konsortialblockchains aus der Angreiferperspektive untersucht. Es wurden vier Blockchains mit falsch konfigurierten Knoten betrachtet mit dem Ziel so viele Informationen wie möglich aus diesen Knoten zu extrahieren. Schließlich wurden Lösungen für die Skalierbarkeit von Blockchain-Anwendungen verglichen und einen Entscheidungsprozess entwickelt, der als Leitfaden für die Verbesserung der Skalierbarkeit ihrer Anwendungen dient.

Aufbauend auf dem Skalierbarkeitsframework werden potenzielle Anwendungen für die Blockchain-Technologie vorgestellt. Zunächst wurde einen Token-basierter Ansatz für die Abbildung eines unternehmensübergreifenden Wertstroms entwickelt. Da sich hier nur auf einfache Token anstelle von komplexen Smart-Contracts gestützt wurde, sollte die Rechenlast im Netzwerk im Vergleich zu anderen Lösungen deutlich geringer sein. Die beiden anderen Lösungen nutzen die Auslagerung von Transaktionen und Berechnungen aus der Haupt-

blockchain heraus. Der erste Ansatz nutzt Secure Multiparty Computation, um das Matching von Angebot und Nachfrage für Produktionskapazitäten in ein vertrauensfreies Netzwerk auszulagern. Erst wenn das Matching erfolgt ist, wird die Transaktion in die Haupt-Blockchain geschrieben. Der zweite Ansatz nutzt das Konzept von Payment Channel Networks, um hochfrequente bidirektionale Micropayments für WiFi-Sharing zu ermöglichen. Der Host wird für jede Sekunde der Datennutzung über einen Off-Chain-Chanel bezahlt. Die vollständige Zahlung wird erst dann in die Blockchain geschrieben, wenn die Verbindung zum Client beendet wird.

Zum Abschluss der Arbeit werden die Ergebnisse zusammengefasst, diskutiert und Möglichkeiten für weitere Forschungsarbeiten aufgezeigt.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The blockchain technology recently became a rapidly evolving area for research and industry. Especially the financial industry adopted the technology. Fintech companies build up a trillion-dollar ecosystem on top of it (CoinMarketCap, 2021). However, this adoption is accompanied by a great amount of publicity surrounding blockchain technology. This results in a steady market growth besides major setbacks. Within the past year, the overall market capitalization of cryptocurrencies more than doubled, even though there were numerous large scale hacks (e.g. PolyNetwork (Jagati, 2021), BadgerDAO (Newar, 2021)), scams (e.g. SquidGame (Gola, 2021), AfriCrypt (Jenkinson, 2021)), and rising popularity of virtually worthless so-called "meme-coins" (Finneseth, 2021). Because of its overwhelming fame, blockchain technology's problems and inherent limitations are often overlooked. Therefore, it can be difficult to distinguish between marketing claims and technological foundations to understand the true nature of blockchain systems. Hence, this thesis aims to investigate, highlight, and address some of the current main weaknesses of blockchain technology. Additionally, we highlight three use-cases for blockchain technology that can benefit from the technology's strength and use different measures to counteract the weaknesses.

To understand the advantages and disadvantages of the technology, it is necessary to outline the basic mechanisms of blockchains. Blockchain's main feature is that it enables electronic transactions between mutually distrusting parties without the involvement of a trusted third party. Instead, all information is shared between all parties in a decentralized manner on a mutual ledger. Having a publicly available ledger has some profound privacy implications. Transactions are generally not encrypted so that they can easily be verified. For that purpose, they have to be cryptographically signed by the sender to ensure authenticity. Valid transactions are bundled in blocks and appended to the ledger. After a new block was added, previous blocks and transactions cannot be altered. Therefore, the ledger is often referred to as being "immutable". However, it would be more precise to refer to it as

"append-only". Not being able to easily revert faulty or fraudulent transactions significantly influences security considerations when using a blockchain. A consensus protocol determines the ordering in which the transactions are appended to the ledger. The only assumption for the consensus protocol is that a majority of the participants behave honestly. However, the decentralized consensus is the main performance bottleneck of most current blockchain systems. Additionally, the immutability and public availability of past transactions pose limitations on the usability of the technology in specific settings. The following chapter introduces the resulting key challenges of blockchain technology and provides a foundation for the solution approaches developed in this thesis.

1.1 Blockchain Key Challenges

The challenges imposed by the distributed nature of the technology can be summarized in four categories: *Inefficient Consensus*, *Privacy*, *Smart Contract Security*, and *Scalability* (Kolb et al., 2020a). In the following sections, these challenges are summarized shortly to provide a foundation for this thesis.

1.1.1 Inefficient Consensus

The consensus algorithm for a blockchain protocol is the mechanism that determines the ordering of the blocks, and hence the ordering of the transactions. A strict ordering of transactions is useful to prevent participants from double-spending a token in two simultaneous transactions. Therefore, each participant must agree on the same ordering of blocks at any time. Ensuring that each participant of a distributed system has the correct information of the system's state at any time is referred to as *byzantine fault tolerance* (Mingxiao et al., 2017). The main idea to ensure a byzantine fault tolerant, strict ordering is to let all users of the network vote for the next block that is appended to the blockchain. This would guarantee, that the majority of participants always agrees on the ordering. However, a simple voting mechanism is not viable in permissionless, public blockchains where participants are anonymous and can join or leave the network at any time. Since users are mostly anonymous, fake identities can be created easily to flood the network with fake votes. This is commonly known as a *Sybil-attack* (Douceur, 2002). To counteract this issue, multiple different consensus algorithms were developed in the past years. For a better understanding of the operating principles and limitations we introduce four of the most common algorithms.

The first blockchain consensus algorithm for permissionless, public blockchains was the Proof-of-Work (PoW) consensus featured in the Bitcoin and Ethereum networks. While this approach provides much freedom, it also has significant disadvantages. To avoid spamming new blocks into the network, participants must perform computationally intensive calculations that serve no other useful purpose and incur high energy costs. Recent studies have concluded that the energy consumption of Bitcoin's mining network exceeds that of entire

countries (de Vries, 2020). Additionally, because PoW consensus is non-deterministic, two participants can find a valid block at roughly the same time and propagate it through the network. The blockchain can temporarily split (fork). Until one of the chains is not discarded by the majority of the network, users can not be confident that their transaction cannot be reversed (Gervais et al., 2016). In addition to assumptions that apply to permissioned blockchains, these drawbacks have led to many other blockchain consensus protocols.

Proof-of-Stake (PoS) was conceptualized as a virtual PoW. Here, the resources are not denoted by computational power but monetary resources in the form of coins on the blockchain. For each new block, a validator is randomly selected. The random distribution is weighted by each node's number of coins staked. PoS has different assumptions on security compared to PoW. Instead of relying on a simple minority of honest nodes, PoS relies on two-third of the nodes being honest. Additionally, most PoS implementations suffer from a theoretical flaw: when two blocks are broadcasted simultaneously, nodes do not have to choose which one to keep. They can use their stake to produce new blocks for each version to maximize their reward instead of choosing which blockchain version they spend their resources on. This problem, referred to as the *nothing-at-stake problem* could result in a constantly split blockchain, where no transaction can be considered finalized (Nicolas, 2014).

The concept of Proof-of-Elapsed-Time (PoET) was introduced by Intel® and is based on hardware features of their processors (Dhillon et al., 2017). Therefore, it can only be run on Intel® processors that support the secure SGX enclave. This enclave enables trusted computations. Each node runs an instance of PoET in their SGX enclave. After a new block is produced, the protocol assigns a random timeout duration. Once a node's assigned timeout has expired, it can generate a cryptographic proof that it waited for the entire duration of its assigned timeout before transmitting the new block along with the proof to the network. While the nodes do not have to trust each other, they must trust that the Intel SGX implementation is secure. Therefore, the trust is shifted to the hardware manufacturer (Chen et al., 2017).

The computational load can be significantly reduced based on identity-based authentication schemes in a completely controlled environment, where each participant is unique and known among the network. An example of these is practical byzantine fault tolerance (PBFT) (Li et al., 2015; Bellare et al., 2009). Here, the next valid block is chosen by the votes of each participant. The system cannot be flooded with votes from fake identities since participants have unique identities. There is no risk of having forked chains since participants can only vote for one block in each round. While the computational power needed is limited, there is much more overhead for the communication between the nodes. Therefore, the number of nodes is limited to around 20 nodes (Sukhwani et al., 2017).

In summary, there is always a trade-off between decentralization, security and scalability. In theory, reaching a decentral consensus over mutually distrusting and potentially anonymous participants in a scalable manner is postulated to be impossible (Halpin, 2020). The *decentralization trilemma* states that an application can at most fulfil two of the three properties:

security, scalability and decentralization. Hence, a decentralized system is either insecure or not scalable (Halpin, 2020). All consensus protocols trade-off among these design points. In public blockchains, the PoW and PoS protocols impose a cost on participation by requiring participants to commit computational power or deposit tokens, respectively. The PoET protocol replaces the cost of participation with a trusted hardware environment, shifting the trust to the hardware manufacturer. In permissioned settings, participants are authenticated, which enables efficient voting mechanisms such as PBFT.

1.1.2 Smart Contract Security

Blockchain smart contracts can be viewed as computer programs executed by the blockchain network. As such, smart contracts can contain programming errors that allow users to exploit security flaws. However, unlike most programs, the smart contract code is written in an immutable state, so it cannot easily be updated to fix these flaws. Additionally, smart contracts are often used in high-stakes scenarios, such as financial applications. This combination of circumstances requires smart contracts to guarantee correctness and security (Huang et al., 2019).

The hacks listed at the beginning of this chapter highlight that there is still room for improvement in smart contract security. Despite an extensive expert review of smart contract source code, these applications were exploited. Therefore, a common criticism is that the programming languages for smart contracts are not suitable for these applications. A set of solutions has been proposed to improve upon this potential weakness. The first one is to keep the existing programming language that is easy to program, then translate it to an intermediary language that is easier to verify. The second solution is to use a different language entirely that enforces stricter paradigms, such as functional programming (Wang et al., 2020).

To further improve security and utilize the availability of different programming languages, a strategy called *N-of-N-version Programming* can be deployed. Instead of writing the smart contract once, N different programmers write N versions of the same smart contract in their language of choice. Users do not call these contracts directly but call a parent smart contract. This parent contract delegates the call to each of the N implementations. If they all return the same result, it is treated as correct. If only one contract returns a different result, the transaction can safely be reverted (Singh et al., 2020).

1.1.3 Privacy

Blockchain networks are often praised for their ability to perform anonymous transactions. The term “cryptocurrency” suggests that transactions are encrypted on the blockchain to maintain confidentiality. However, this is generally not the case except for a few specific implementations. Transactions must be transmitted in clear text so that nodes can verify the transaction before appending it to the ledger. Therefore, transaction input, output, sender,

and receiver are publicly visible on the ledger. For each address on the blockchain, the entire transaction history and the current balance of tokens are visible to everyone else on the network (Andola et al., 2021).

Although such a blockchain address can not directly be linked to the owner, there are techniques to deanonymize transactions and link addresses to personally identifiable information. These methods rely on available information for a subset of addresses, for example, from publicly available information such as donation addresses, information available from centralized exchange providers, and analytic methods on the transaction graph (Biryukov and Tikhomirov, 2019). The methods are currently used to identify criminals who use cryptocurrencies to obfuscate their payment traces from illegal activities (Paquet-Clouston et al., 2019).

While technologies exist that allow the verification of encrypted transactions, such as zero-knowledge proofs, they are currently rarely used in blockchain networks due to their high computational overhead for transaction verification (Hopwood et al., 2020). Additionally, implementing functionalities beyond simple payments such as smart contracts for these systems is more complex than for traditional systems. Finally, the encrypted data is still publicly available. The utilized encryption schemes might become insufficient in terms of security in the future, making all transaction details transparent again (Barker and Roginsky, 2010).

These privacy concerns are a significant obstacle to adopting public blockchains for businesses. Companies fear losing their competitive advantage if most of their transaction data become public. Therefore, permissioned blockchains have attracted interest from the business community (Kolb et al., 2020a). In a permissioned blockchain, the ledger is only shared between authorized actors. Here, permissions can be set individually so that some actors can only act as observers. Therefore, actors who mutually distrust each other but have aligned interests can collaborate on such a network. While such a setup mitigates some of the privacy concerns, a single actor could intentionally or unintentionally leak the transaction history of the whole network.

1.1.4 Scalability

As stated in Section 1.1.1 blockchain networks can not compete with centralized systems in terms of transaction throughput or latency. However, the consensus mechanism is only partly the bottleneck of these scalability issues. The overall transaction throughput is limited by the number of transactions per block and the rate at which blocks are produced. These parameters are predetermined by the given consensus protocol. Due to the chained data structure, blocks have to be produced sequentially, imposing this strict limit (Chauhan et al., 2018).

However, there are approaches to circumvent these limitations. The first one allows a parallel production of blocks, resulting in a data structure that is not a chain but a directed

acyclic graph (DAG) (Koens and Poll, 2018). A block can have multiple predecessors and successors in these data structures. Therefore, there is no strict ordering of blocks and, consequently, no ordering of transactions. The second approach to allow parallel block production is to partition a larger network into smaller networks that independently process their transactions. This method originates from distributed databases and is referred to as *sharding* (Cai et al., 2018). Having a sharded network has two significant implications. First, the shards have to coordinate with each other so that users can not make conflicting transactions on multiple different shards. Second, each shard must be sufficiently large not to be compromised or overruled by a single actor (Han et al., 2021).

Other scalability solutions do not rely on redesigning the blockchain data structure. Instead, these solutions provide a way to make multiple transactions without the blockchain. The blockchain is only used for a final settlement. This is the basic concept behind payment channels. Here, two participants lock funds in a smart contract to create a channel. Then the two participants can make arbitrary payments with these funds by just cryptographically signing the transactions. Only the last transaction is committed to the smart contract when the channel is closed, which settles the overall payments. This idea can be extended to linking multiple channels to a payment channel network. While in these networks, the transaction throughput can be drastically improved, the latency involved with opening and closing channels makes it unsuitable for some use-cases (Poon and Dryja, 2016).

In addition to the latency and transaction throughput, blockchains tend to have additional bottlenecks. For example, the block size limit directly limits the complexity of smart contract code that can be deployed to the blockchain and the amount of input data that can be sent to smart contracts. The block rate limits the maximum execution time of smart contracts (Wang and Malluhi, 2019).

1.2 Research Questions

Based on the four blockchain challenges, this thesis sheds light on each of the problem areas and searches for solutions for the areas. This leads to the following guiding research question (GRQ) of this thesis:

GRQ How do the four main blockchain challenges impact the current blockchain landscape, and how can problems be circumvented or solved?

A prerequisite for addressing GRQ is a deep understanding of the root causes of each key challenge. To achieve this, each of the four problems is considered individually, and appropriate solutions are developed.

This thesis aims to provide deeper insights into the problem areas and provide individual solution approaches instead of completely solving each issue. Instead, the scientific contributions are to be understood as suggestions for the potential development of blockchain

technology. To address each challenge individually, we pose one research question for each blockchain challenge. Finally, we provide an additional research question for the *scalability* issue to demonstrate the usage of scalability solutions in blockchain applications.

For the first challenge *inefficient consensus*, we focus on PoW, which is considered the least efficient consensus mechanism concerning resource consumption. While it is inefficient, it is considered the most secure consensus mechanism (see Section 1.1.1). Therefore, it is necessary to assess the security limitations in the constantly evolving blockchain landscape, which leads to the following research question:

RQ1: Which are the security limitations for the PoW consensus concerning centralization, attacks and forks?

Current solutions to improve the second challenge, *smart contract security*, still cannot provide a guarantee for error-free contract code. Additionally, they increase the complexity and cost of development. A promising way to solve this issue is to develop standardized components. Standardized components for critical functionalities are a measure to reduce complexity and cost simultaneously. To assess the standardization potential for smart contracts, it is necessary to classify state-of-the-art smart contract functionalities. Additionally, there is a need to identify which functionalities are often used in conjunction. This results in the following research question:

RQ2: How are current smart contracts structured and how can the structure be used to drive standardization?

The third problem area concerns *privacy* issues for blockchains. Here, much research was conducted on novel methods to make transactions on blockchains less transparent. Techniques such as ring-signature cryptography and zero-knowledge computation were applied to make transactions or even smart contract computations private. For businesses, the preferred method is to keep data private from the general public by utilizing a permissioned blockchain. Here, data is only shared between a set of predetermined stakeholders. However, each participant has to ensure the security of their blockchain node. Previous research did not identify the exact security implications resulting from one participant failing to secure their node properly. Therefore, we pose the next research questions:

RQ3: What information can an attacker extract from compromised nodes of a consortium blockchain?

Finally, the problem of blockchain *scalability* is one of the most researched fields for blockchain research. However, understanding the purpose and limitations of these solutions to find the correct one for a specific application can be a complex task. Most of the solutions

have prerequisites or dependencies with other solutions. There is currently a lack of guidance on applying scaling technology. Therefore, the fourth research question is:

RQ4: What are best practices to ensure scalability of blockchain applications, and how should they be applied?

Based on the developed best practices and guidelines to build scalable blockchain applications, we finally want to apply this guideline and demonstrate how scalability solutions can be used in practical applications. This results in the final research question:

RQ5: How can the scalability best practices be applied in practice?

1.3 Structure

To answer the overreaching research question, this thesis is composed of two parts, consisting of seven independent chapters that were published as research articles.

In the first part of this dissertation, we address each of the four weaknesses. Within the following four chapters, we answer the first four research questions. Afterwards, the second parts includes Chapters 6-8. These demonstrate the application of scalability solutions and answer RQ5. Figure 1.1 provides an overview of the scientific contribution of the main chapters and articles in this thesis.

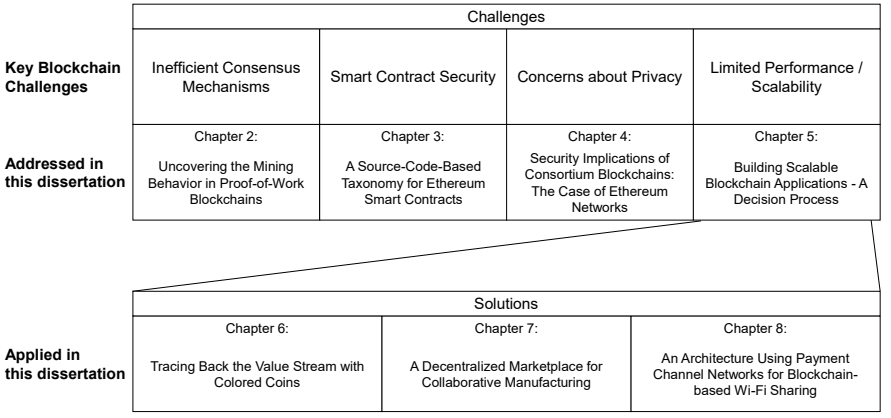


Figure 1.1: Overview of the Scientific Contribution

Chapter 2 addresses RQ1 by providing an in-depth study of the major PoW blockchains and the behavior of miners. The chapter investigates the distribution of the participants' power to examine whether the blockchains are at risk of attacks. Different factors that influ-

ence the mining behavior are taken into account, including time, mining reward and major blockchain forks.

To improve smart contract security, we address RQ2 in Chapter 3. To identify common functionalities of smart contracts, we analyzed the source code of over 100 Ethereum smart contracts. From this analysis, we structured the functionalities in a taxonomy. Based on this taxonomy, we identified seven archetypes of smart contracts and provided recommendations for further standardization and research efforts.

To address RQ3, consortium blockchains are analyzed from an adversarial perspective in Chapter 4. For this purpose, we first created an overview of misconfigured blockchain nodes. Based on this overview, four small networks were chosen, and data was extracted from the faulty nodes. The extracted data was used to reconstruct the transaction structure, reverse-engineer smart contracts and gain insights into the usage behavior of the network.

Chapter 5 provides an overview of scalability solutions for blockchain applications, answering RQ4. By classifying the solutions and identifying dependencies, we could provide a decision process for choosing the right scalability solution for a given application. Therefore, the chapter addresses RQ4 and provides a foundation for the remainder of this thesis.

Building on Chapter 5 the following three chapters focus on applying the decision process to potential blockchain applications and address RQ5.

In Chapter 6 we developed and demonstrated an efficient method to trace goods through a supply network. While previous approaches relied on complex smart contracts, the designed approach relies only on the intrinsic transaction structure of specific blockchains. Additionally, we demonstrated how product movement data could be extracted from information systems and transformed into the correct transaction format.

In addition to scalability issues, Chapter 7 also addresses privacy issues in the proposed solution for a decentralized marketplace for manufacturing capacities. The proposed solution uses a combination of secure multiparty computation and zero-knowledge proofs to enable decentralized, hidden order-book matching. The compute-intensive order matching is not performed on the blockchain itself. Therefore, operations that are impossible on most blockchain networks can be offloaded, and transaction inputs and outputs can be kept private.

The final paper demonstrates a prime use-case for payment channel networks in Chapter 8. First, the shortcomings of current WiFi-sharing solutions are mapped out. We derived design principles necessary to create a solution that addresses these shortcomings. Finally, we propose an architecture that addresses all these shortcomings, including a payment solution for WiFi-sharing based on payment channel networks.

In the final chapter, the results of this thesis are summarized and discussed. Additionally, we provide references for further research that served as a basis for some of the chapters of this thesis.

1.4 Preliminary Work

The following Section presents related publications that were preliminary work for the included publications but not included in the main part of this dissertation. Figure 1.2 displays the relevant publications and how they relate to the publications included in this thesis. The preliminary work that led to this thesis can be split into three distinct research streams that utilized similar methodologies and built upon each other.

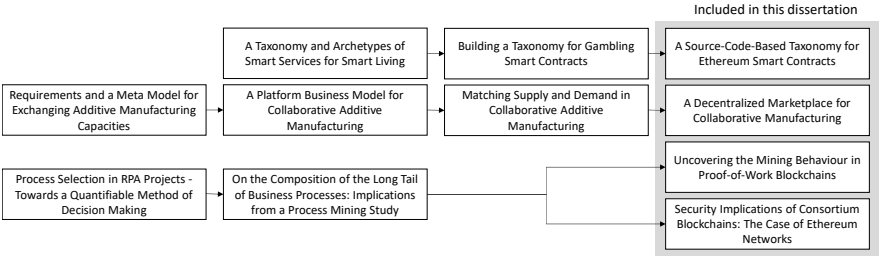


Figure 1.2: Preliminary Work for this Thesis

The first research stream is based on *Taxonomy Development*. The research in this area leads to the final taxonomy for Ethereum smart contracts. It started with a taxonomy for smart services. Table 1.1 summarizes the contents of this paper. The analyzed smart services exhibit similarity with smart contracts in that they execute commands autonomously and pose a potential security risk if exploited by a malicious actor.

Table 1.1: Summary: A Taxonomy and Archetypes of Smart Services for Smart Living

Title	**A Taxonomy and Archetypes of Smart Services for Smart Living**
Co-Authors	Marcus Fischer, David Heim, Christian Janiesch, Christoph Klima, Axel Winkelmann
Outlet	Electronic Markets (EM)
Abstract	Smart service integrates digital and physical competencies for automated service delivery in smart service systems to co-create value. Smart services envelop digital services delivered through smart products. The latter act as boundary objects to the consumer. Smart services are capable of learning, adapting, and decision-making based on communicated data through self-controlled functions. Due to the multidisciplinary discourse, there is a knowledge gap concerning common ground for central concepts, the transformative potential of smart products as well as evidence-based design knowledge derived from real-world services. In this paper, we apply conceptual research and data analysis to construct a taxonomy that supplies this common ground for smart service. The resulting taxonomy comprises 8 dimensions with 20 characteristics. Based on an empirical analysis of 100 smart services from the smart living sector, we performed a cluster analysis to derive five archetypes that classifies smart service as either monitor, command execution, diagnostics and automation, personal tracker, or trainable assistant using smart products as boundary objects for distinct purposes.
Method	Taxonomy Development

This motivated the development of a taxonomy for applications where security issues can not easily be patched, i.e., smart contracts. Before analyzing the whole set of smart

contracts included in this thesis, a preliminary study on a subset was conducted to check the feasibility of source-code analysis for taxonomy development. Therefore, a taxonomy for gambling smart contracts was developed and published as a research-in-progress paper. Table 1.2 summarizes the content of this paper. Based on the feedback from the scientific community, the methodology was improved to develop the final source-code-based taxonomy for Ethereum smart contracts.

Table 1.2: Summary: Building a Taxonomy for Gambling Smart Contracts

Title	**Building a Taxonomy for Gambling Smart Contracts**
Co-Authors	Julian Kolb, Luc Becker
Outlet	European Conference on Information Systems (ECIS)
Abstract	In recent years, the blockchain technology has matured and established new opportunities in the digital world. With the release of the Blockchain 2.0, the Ethereum Network and smart contracts, it is now possible to operate applications decentralized and independently. These applications promise lower transaction costs, better efficiency and higher security. However, there is still a lack of in-depth understanding and standardization within the variety of recently developed smart contracts. In addition, there is still no proper taxonomy that structures the technical elements of a smart contract and makes them comparable. Hence, we develop a smart contract taxonomy using an inductive research approach. Following Nickerson et al. (2013) we analyze the smart contracts of 47 gambling DApps to identify the 18 dimensions and 41 characteristics of your technical and code-based taxonomy. In future research, we will continue to expand the developed taxonomy and include other application areas. Finally, a general taxonomy for research and product development will be available to science and practice, ensuring a consistent and standardized implementation of smart contracts.
Method	Taxonomy Development

The second research stream was a design science research effort to enable companies to share their manufacturing capacities. These efforts finally resulted in the proposed architecture for a decentralized marketplace for manufacturing capacities. The design process began by analyzing the requirements for automated capacities sharing for additive manufacturing machines (see Table 1.3). From the requirements, a meta-model for the data exchange was derived. Many participants of this study highlighted that they are hesitant to share information about their current production capacity utilization with a third party. However, current manufacturing sharing platforms require this information to match supply and demand efficiently.

The requirements and the meta-model provided a technological basis for designing a marketplace for manufacturing capacities. Besides technological soundness, economic feasibility had to be demonstrated. A business model was developed in the next paper to show that such a marketplace is economically feasible. Table 1.4 summarizes the results of this paper. The paper showed that a crucial prerequisite for a successful marketplace is achieving a critical mass of users. For some niche manufacturing technologies, this is not possible. Therefore, sharing manufacturing capacities on such a centralized platform is impossible for these niche applications.

Table 1.3: Summary: Requirements and a Meta Model for Exchanging Additive Manufacturing Capacities

Title	**Requirements and a Meta Model for Exchanging Additive Manufacturing Capacities**
Co-Authors	Chiara Freichel, Marcus Fischer, Axel Winkelmann
Outlet	International Conference on Wirtschaftsinformatik (WI)
Abstract	In an environment shaped by digital transformation and globalization, manufacturers face increasing market dynamics, cost pressure, and more sophisticated customer requirements. As this demands flexibility and adaptability, enterprises rely on new solutions for collaboration. A marketplace for production capacities supports companies in reducing order risks and improving responsiveness to changing market conditions. We seek to define requirements for a marketplace that is capable of matching products with production processes. With an initial focus on additive manufacturing, we aim to build a blueprint for similar application scenarios in other industrial contexts. Therefore, we employ a qualitative research based on expert interviews. Our results suggest that a marketplace for production capacities must address various requirements, which can be grouped under the categories of technologies, machines, and products. We further build a conceptual meta model that sets the groundwork for the matching and thus facilitates the implementation of the marketplace in practice.
Method	Design Science Research

Table 1.4: Summary: A Platform Business Model for Collaborative Additive Manufacturing

Title	**A Platform Business Model for Collaborative Additive Manufacturing**
Co-Authors	Chiara Freichel, Isabel Ernst, Axel Winkelmann
Outlet	Hawaii International Conference on System Sciences (HICSS)
Abstract	Modern manufacturing is caught in a trade-off between maximizing efficiency and staying flexible in dynamic markets. Inter-organizational sharing of manufacturing capacities on a digital marketplace could contribute to gain flexibility, reduce cost and capital employed as well as provide further business opportunities. Although current research has already prepared the ground for its technical conceptualization, research on such a marketplace's implementation in a business model is scarce. However, since an efficient matching of supply and demand requires a sufficient number of platform users, attracting corporate customers with a suitable business model is crucial. The present research aims to address this problem by developing and evaluating a business model for a marketplace provider, illustrated for the case of additive manufacturing.
Method	Design Science Research

In the final paper of this research stream, the previous results are combined into the prototype of a marketplace for additive manufacturing capacities, with automated matching of supply and demand. The proposed marketplace would solve the problem of sharing manufacturing capacities adequately. However, it still suffers from two problems. First, participants must share information about their machine utilization with a third party. Second, this marketplace would not be profitable for niche manufacturing technologies. The architecture for a decentralized marketplace for manufacturing capacities was developed to address these shortcomings.

Table 1.5: Summary: Matching Supply and Demand in Collaborative Additive Manufacturing

Title	**Matching Supply and Demand in Collaborative Additive Manufacturing**
Co-Authors	Chiara Freichel, Axel Winkelmann
Outlet	International Journal of Conceptual Modeling (EMISAJ)
Abstract	Due to an increasing individualization of products, additive manufacturing is often seen as a solution to cater for more sophisticated customer requirements. In order to fulfill customer needs, manufacturers have to rely on collaboration to distribute risk and improve the utilization of their resources. In this paper, we used qualitative interviews to define requirements for a marketplace that allows the automatic exchange of additive manufacturing capacities. From these requirements, we derived a conceptual model that matches orders to sales offers while taking specific product requirements, such as quality, into account. Additionally, we implemented a demonstrator to evaluate the model with potential buyers and sellers of additive manufacturing capacities. Our research showed that most requirements could be implemented in a marketplace. However, we could show specific limitations for particular requirements.
Method	Design Science Research

The final research stream is based on data mining. The following papers applied data analytics for single companies and processes on process-related data. Both studies involved the development of unique KPIs and gathering an understanding of how the data relates to real-world events. The first paper utilized process mining methods to identify the automation potential for business processes. It is summarized in Table 1.6.

The second paper of this research stream applied similar methods to condense complex process features into manageable KPIs. These KPIs were used to prove the existence of a hypothesized *long tail* distribution for business processes.

Condensing information into manageable units is crucial when dealing with blockchain data. The graph and social network analytics tools used in process mining could be directly transferred to analyze the transaction graph of blockchains. They were used to analyze the mining behavior of the well documented public PoW blockchains. Additionally, even the undocumented and blindly extracted data from consortium blockchains could be analyzed with these techniques.

Table 1.6: Summary: Process Selection in RPA Projects

Title	**Process Selection in RPA Projects – Towards a Quantifiable Method of Decision Making**
Co-Authors	Jonas Wanner, Marcus Fischer, Florian Imgrund, Christian Janiesch, Jerome Geyer-Klingeberg
Outlet	International Conference on Information Systems (ICIS)
Abstract	The digital age requires companies to invest in value-creating rather than routine activities to drive innovation as a future source of competitiveness and business success. Thus, many companies are reluctant to invest in large-scale, costly backend integration projects and seek adaptable solutions to automate their front-office activities. Bridging artificial intelligence and business process management, robotic process automation (RPA) provides the promise of robots as a virtual workforce that performs these tasks in a self-determined manner. Many studies have highlighted potential benefits of RPA. However, little data is available on operationalizing and automating RPA to maximize its benefits. In this paper, we shed light on the automation potential of processes with RPA and operationalize it. Based on process mining techniques, we propose an automatable indicator system as well as present and evaluate decision support for companies that seek to better prioritize their RPA activities and to maximize their return on investment.
Method	Design Science Research / Process Mining

Table 1.7: Summary: On the Composition of the Long Tail of Business Processes

Title	**On the Composition of the Long Tail of Business Processes: Implications from a Process Mining Study**
Co-Authors	Marcus Fischer, Florian Imgrund, Christian Janiesch, Axel Winkelmann
Outlet	Information Systems
Abstract	Digital transformation forces companies to rethink their processes to meet current customer needs. Business Process Management (BPM) can provide the means to structure and tackle this change. However, most approaches to BPM face restrictions on the number of processes they can optimize at a time due to complexity and resource restrictions. Investigating this shortcoming, the concept of the long tail of business processes suggests a hybrid approach that entails managing important processes centrally, while incrementally improving the majority of processes at their place of execution. This study scrutinizes this observation as well as corresponding implications. First, we define a system of indicators to automatically prioritize processes based on execution data. Second, we use process mining to analyze processes from multiple companies to investigate the distribution of process value in terms of their process variants. Third, we examine the characteristics of the process variants contained in the short head and the long tail to derive and justify recommendations for their management. Our results suggest that the assumption of a long-tailed distribution holds across companies and indicators and also applies to the overall improvement potential of processes and their variants. Across all cases, process variants in the long tail were characterized by fewer customer contacts, lower execution frequencies, and a larger number of involved stakeholders, making them suitable candidates for distributed improvement.
Method	Data Mining / Process Mining

Chapter 2

Uncovering the Mining Behavior in Proof-of-Work Blockchains[1]

2.1 Introduction

The Internet and the World Wide Web (WWW) were the major decentralizing forces of the last two decades. However, these technologies led to the rise of tech giants, like Google, Amazon, and Facebook, who nowadays control large amounts of the internet (Möller and Rimscha, 2017). While this centralization of power makes using the Internet more convenient, it also implies negative effects on privacy and security (De Filippi and McCarthy, 2012). Lately, the blockchain technology has received much attention as a possible driver of decentralization. Primarily through the hype over cryptocurrencies in January 2018, the technology became more and more popular (Carson et al., 2018). The concept of having a fully decentralized system without central authorities such as governments, regulatory institutions or companies, is highly attractive for building trust among untrusted parties and saving costs by eliminating intermediaries (Davidson et al., 2016).

Due to its network structure, it is predicted that blockchains share the same fate as the Internet or the WWW. The decentralized network has already started to centralize with new mining technologies, commercial crypto mining, and large trading platforms for cryptocurrencies. Unlike the Internet, centralization in blockchains could pose an existential threat to the networks by circumventing the trustless mechanism that it is based on. Therefore, centralization poses a threat to the security and usability of the technology (Lin and Liao, 2017; Hurlburt, 2016). There are many reasons why power can shift and centralize, starting at the development and programming of the mechanisms themselves. In this study, however, we

[1] This chapter was published in *ECIS 2020 Research Papers* as Hofmann et al. (2020) and co-authored by Fabian Schatz and Axel Winkelmann.

focus on *mining*, the primary consensus mechanism of most blockchains and arguably the most important source of trust (Wang et al., 2018a).

Risius and Spohrer (2017) conducted a literature review on the current state of blockchain research. They also identified unequal power structures as a possible weakness of blockchains and proposed the following research question: "Which consensus mechanisms can blockchain platforms deploy to avoid monopolization of power?" Answering this question requires a deep analysis of the multitude of different consensus mechanisms used. The constant development of new mechanisms or modifications of old ones makes this a challenging task. To take the first step in this paper, we focus on Proof-of-Work (PoW), the most widely used consensus mechanism for public blockchains (Bach et al., 2018).

So far, only a few studies have tackled this topic. Mingxiao et al. (2017) identified mining centralization as a major challenge for the PoW consensus model. Beikverdi and Song (2015) further researched the centralization in Bitcoin mining through the formation of mining pools. It was found that after 2011 the mining process centralized until 2014 and was predicted to further centralize in the following years. It was also noted that the constant lowering of the block reward would change the mining behavior unpredictably. Therefore, it is important to research whether there are overall trends in mining behavior and sudden changes for specific events. There is a lot of other research on mining, which focuses on the overall network structure of miners, and its centralization, but not on the distribution of mining power (Gencer et al., 2018; Miller et al., 2015). To fill this gap, we pose the following research questions:

RQ1: How is mining power distributed over time?
RQ2: What are the possible causes of centralization?
RQ3: What are possible causes that prevent a monopolization of power?

To answer these research questions, we utilized a mix of big data analytics and classical econometric scores to calculate the inequality in mining power distribution. In the following section, we summarize how a consensus is reached through mining in networks. We further introduce different concepts, such as mining difficulty, mining pools, and mining rewards. The next section gives an overview of our methodology and the data we used to conduct our analysis. In Section 2.4, we present the analysis results and briefly discuss the findings. The final section wraps up the paper and gives an outlook for future research.

2.2 Consensus in Blockchains – Proof-of-Work

The concept of blockchains stems from the idea and implementation of a trustless distributed ledger by Nakamoto (2008). Here, transactions of any kind are shared among participants in a network and checked for validity in a decentralized fashion.

PoW is the consensus protocol first introduced in the Bitcoin network and later adapted by other blockchains (Nakamoto, 2008). In a decentralized network, someone must be selected to record the transactions. The easiest way would be a random selection. However, the random selection makes the system vulnerable to so-called Sybil attacks, where a lot of fake identities are created to increase the chance of being selected (Douceur, 2002). Thus, if a node wants to publish a block of transactions, a lot of work has to be done to prove that the node is not likely to attack the network. Generally, the work implies computer calculations. Hence, the computations necessary to vote for a block are the proof of work.

Most blockchains use a PoW based on cryptographic hashes (Loe and Quaglia, 2018). This entails finding a *nonce value* in a way that when hashed with additional block parameters, such as the previous block hash and the transactions, the value of the hash must be smaller than or equal to the current *target value*. The target value is automatically adjusted according to the total computational power in the network to keep the time necessary to mine new blocks consistent (Nakamoto, 2008). When one node reaches the target value, it would then broadcast the block to the network and all other nodes must mutually confirm the correctness of the hash value. If the block is validated, other nodes append this new block to their own copy of the blockchain. The process of producing new blocks is referred to as *mining*, the mining target value is dependent on the current network is called *difficulty*. Miners get rewarded for their expense of computational resources with a *block reward* (Carlsten et al., 2016).

Most blockchains use a PoW based on cryptographic hashes (Loe and Quaglia, 2018). This entails finding a *nonce value* in a way that when hashed with additional block parameters, such as the previous block hash and the transactions, the value of the hash must be smaller than or equal to the current *target value*. The target value is automatically adjusted according to the total computational power in the network to keep the time necessary to mine new blocks consistent (Nakamoto, 2008). When one node reaches the target value, it broadcasts the block to the network, and all other nodes must mutually confirm the correctness of the hash value. If the block is validated, other nodes append this new block to their own copy of the blockchain. The process of producing new blocks is referred to as *mining*, the mining target value is dependent on the current network is called *difficulty*. Miners get rewarded for their expense of computational resources with a *block reward* (Carlsten et al., 2016).

Since the selection process of the miner is still random, multiple miners may cooperate to increase the chance of mining a block. This is done via *mining pools*. The reward of the block is then shared among the members of the mining pool according to the computational power they provided (Dev, 2014). This does not increase the reward per miner. They often have to pay a small fee to the owner of the mining pool (Salimitari et al., 2017). However, pooled mining enables a more predictable mining reward since a large pool finds blocks more frequently and shares the reward. In contrast, solo miners are less likely to find a block but get to keep the whole reward. This makes it particularly attractive for smaller miners to join a mining pool.

As the block reward usually consists of new coins added to the network, the overall coin supply is steadily growing. Since this causes monetary inflation and a theoretically infinite coin supply, blockchain networks reduce the block reward frequently. Since most protocols reduce the reward by fifty percent, this mechanism is referred to as halving. Most networks have a programmatic reward halving included in the protocol (Kroll et al., 2013). This means that the total supply follows a geometric series, resulting in an overall capped supply of coins. Even though the reward is not strictly halved for Ethereum based blockchains (it was reduced from 5 ether to 3 to 2), we will still refer to it as halving.

The PoW mechanism encourages honest behavior since it is expensive to provide enough computational work to overrule the rest of the network. Therefore, the network is considered secure as long as no single mining node (or mining pool) has strictly more than half of the computational power in the network. Gaining the majority of computational power is called 51% attack (Lin and Liao, 2017). This majority can also be achieved if multiple malicious miners cooperate.

Supposing the network does not reach a consensus on the next block, a blockchain split may occur, meaning each group uses the blocks they see as valid. This is often referred to as a *blockchain fork*. This phenomenon occurs when new rules that large part of nodes do not want to implement are enforced in the blockchain protocol. An example is a proposed update for Bitcoin to increase the maximum block size from 1MB to 8MB. The update was not accepted by a majority and led to a block larger than 1MB, which was rejected by the Bitcoin network and split the Bitcoin blockchain into what is now Bitcoin and Bitcoin Cash.

2.3 Material and Methods

For our research methodology, we follow the guidelines and principles of big data analytics as introduced by Müller et al. (2016). The data we collected can be classified as big data because it satisfies the 4 Vs of big data: volume, velocity, variety, and veracity (Buhl et al., 2013). The *volume* of our initial data-set consists of 526 GB of compressed blockchain data. Since the transactions and blocks are a constant stream of user-generated data and block times are as low as 15 seconds (Buterin, 2014), the *velocity* is also given. *Variety* of the data is given since theoretically arbitrary data can be written to the blockchain (Matzutt et al., 2018). This is particularly true for the data used to identify the miner of a block. Additionally, the data collected is entirely user-generated and can be analyzed with fine granularity on a per-user level. This requires advanced algorithms to filter and preprocess the data (Hedman et al., 2013).

One of the main challenges that come with big data is the trade-off between interpretability and accuracy of the resulting models (Müller et al., 2016). Since this study is more of an observatory nature and does not build predictive models, we do not rely on a high precision of the models. Therefore, we focused on analyses that are easy to interpret, such as linear regressions and the focus on single events in a time series.

Since cryptocurrencies today are the most widespread applications of blockchain technology, we used the three largest PoW cryptocurrencies by market capitalization as well as their major forks. While Litecoin and Bitcoin and their derivatives are primarily used as investment or payment, Ethereum and Ethereum Classic provide blockchain platforms for applications like tokens, smart contracts, and games.

2.3.1 Data Collection

Since one main feature of public blockchains is availability, the data collection was done directly via the official software provided by the individual blockchains. In particular, we use the blockchains retrieved with *bitcoind, multigeth* and *litecoind*. The blockchain software of these blockchains was downloaded and installed on a server in the first step. For Bitcoin and its derivatives, the *bitcoind* software was used in three different instances to get the three versions of the blockchain. For Ethereum, the *mulitgeth* project enabled obtaining Ethereum as well as the Ethereum Classic blockchains. To access the transaction data, the blockchains must be fully synchronized. This way, we can ensure to have the correct data and have a firsthand data source. The analyzed data is summarized in Table 2.1. Note that the date of the forks' first block equals the moment the chain was forked, not to be confused with the creation date of the original chains genesis block.

Blockchain	Market Capitalization	Hard Fork	First Block	# Blocks
Bitcoin	$122,751,501,000	No	2009-01-02	601,000
Ethereum	$15,025,642,000	No	2015-07-30	8,843,000
Bitcoin Cash	$3,620,986,000	Yes (Bitcoin)	2017-08-01	128,000
Litecoin	$2,822,582,000	No	2011-10-07	1,729,000
Bitcoin SV	$1,771,769,000	Yes (Bitcoin Cash)	2018-11-15	50,000
Ethereum Classic	$423,825,000	Yes (Ethereum)	2016-07-20	7,180,000

Table 2.1: Blockchains Chosen for Analysis (CoinMarketCap, 2019)

We came across other minor forks such as Litecoin Cash during our research. Although it is the largest fork of Litecoin, we did not take it into consideration since its volume is not large enough to represent a majority of miners. Litecoin Cash only has a market capitalization of $4,444,000 at the time of writing (CoinMarketCap, 2019).

2.3.2 Data Preparation

To make the blockchain data easier to process, only the block headers' relevant information was extracted and written into a database. The goal is to structure the data and remove unnecessary metadata. We limited the time frame and only looked at transactions of each blockchain to the 31st of October 2019. For forked blockchains, we did not start at the genesis block but at the block where the chain was forked. The complete data preparation process is depicted in Figure 2.1 and described in detail in the following section.

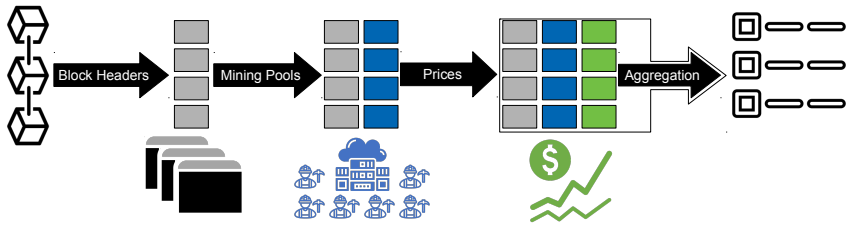


Figure 2.1: Data Preprocessing Pipeline

First, we queried the blockchain software for the block headers of each block to obtain including the *time stamp*, *difficulty*, and *block reward*. For the Ethereum-based blockchains, the *miner* is directly stated in the block header. For Bitcoin-based blockchains and Litecoin, the miner can be extracted from the so-called coinbase transaction. However, this is unreliable because the transaction can be sent to multiple receivers. Instead, we used the *coinbase_data* which contains extra data that miners can add to the block. Mining pools use this field to write the pool's name in the block. While there are lists of known mining pools available online, we found those incomplete. For unknown pools, we, therefore, used an iterative approach to map *coinbase_data* with mining pools. The remaining blocks are likely mined by either solo miners or unknown mining pools and are regarded as a single entity. We managed to assign over 90% of the blocks to mining pools. While this approach is a limitation of our analysis, we observed that adding smaller mining pools to our list did not change the centralization scores significantly.

Since money is the main incentive for miners, we augmented the blockchain data with the pricing data of the coins. For each block, we multiplied the reward with the average price of the mined coins for that day in US Dollars. Together with the average number of blocks mined per day, the mining profitability can be estimated. We could not account for hardware and electricity costs since they do not only vary strongly per region but also because some mining farms produce their own electricity (Oliver, 2019).

Finally, we aggregated the block data per week to avoid false correlations due to an excessive granularity (Müller et al., 2016). In the aggregation, we calculated the average difficulty and reward per block, the total amount of blocks per week as well as the total count of blocks mined by each miner per week. We can use this data to calculate inequality scores for each week and create a time series for each blockchain.

To measure the inequality in mining power distribution, we use scores commonly used in economics to measure inequality in wealth distribution. The *Gini Coefficient* has found wide application in measuring inequality in many fields of economics (Sen, 1976). In the following x_i is the number of blocks, which miner i produced in any given week. The total amount

of miners, which found blocks in that week is given by n. The Gini coefficient is therefore calculated as

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n \sum_{i=1}^{n} x_i} \tag{2.1}$$

While there is a straightforward geometric interpretation of the Gini Coefficient, an interpretation from an economic perspective is not easy. Therefore, we additionally use the *Theil Index* as the second measure of inequality (Theil, 1967). The Theil Index can be seen as the probability of a random block mined by one specific user. It is calculated as

$$T = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{x_i}{\mu} \ln \frac{x_i}{\mu} \right) \tag{2.2}$$

with μ being the mean value

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2.3}$$

Finally, we used the uniformity measure to assess the centralization in mining as done by Beikverdi and Song (2015), to verify and compare their results:

$$U = \frac{\sqrt{\frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n}}}{\mu} \tag{2.4}$$

where μ is the mean value. This uniformity measure is a scaled statistical variance of the mining distribution. For all the inequality measures, 0 means perfect equality, whereas 1 means perfect inequality. For the uniformity measure, 0 means perfect uniformity but has no maximum value. Therefore it is hard to interpret the different values.

2.4 Data Analysis and Results

This section first takes a macro look at the mining power distribution over time. We thus compare each centralization score's development for the blockchains and possible influences on the mining behavior. Subsequently, we investigate the effects of singular events such as reward halvings and blockchain forks.

The mining behavior in the very first years of Bitcoin has already been researched. Mining started very centralized due to the low adaption, developed a stronger decentralization, and then centralized again when first mining pools started forming (Beikverdi and Song, 2015). The same effect could be observed with Litecoin, as it was also one of the first public blockchains. This behavior is well understood. We, therefore, focused on mining behavior after the state-of-the-art in blockchain mining was established.

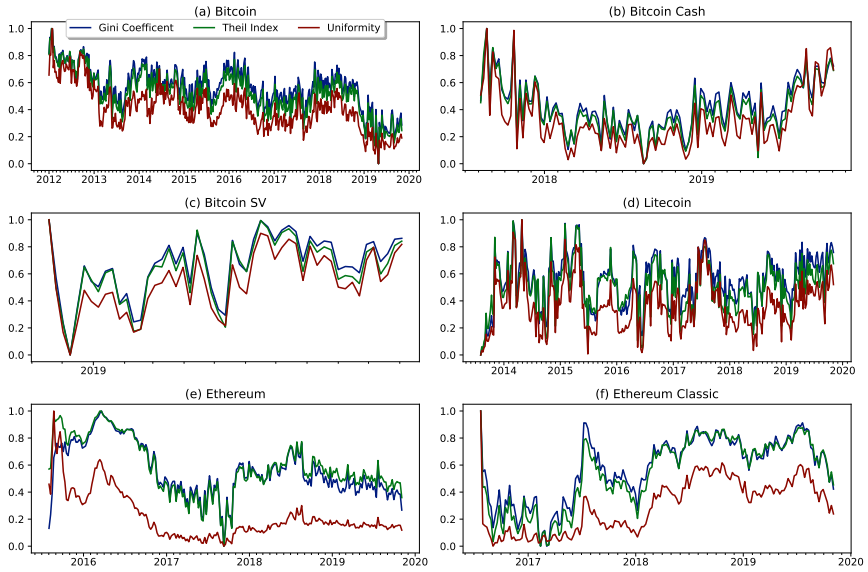


Figure 2.2: Development of the Centralization Scores for each Blockchain. The Scores have been Scaled to the Interval $[0, 1]$.

Figure 2.2 shows the three centralization scores over time. Notice that in this graph, we normalized the scores using a min-max-scaler. We lose the actual values of the scores but can showcase the overall development over time and the strong correlations of the scores. While no overall trend is visible for all blockchains, it is interesting to observe a wave pattern for all cases. It should be noted that even though Ethereum (Classic) is not mined on specialized hardware like the other blockchains, there is no fundamental difference visible in the evolution of mining. Another interesting aspect is the strong fluctuations, especially for Litecoin. We could not explain this phenomenon even after a deeper inspection of the underlying data. Litecoin is the only blockchain in our sample that uses the *scrypt* hashing algorithm for its proof of work. The reasoning behind the fluctuations could hide inside the *scrypt*-based mining ecosystem, which also entails niche coins such as Dogecoin and Feathercoin (Kuanyshbayev et al., 2013; Chohan, 2017).

The scaled scores hide that the scores are generally much higher for Ethereum and Ethereum Classic compared to the other blockchains. We attribute this to the differences in the data format the blockchains provide us with. As we stated in Section 2.3.2, the Ethereum and Ethereum Classic data format provides us directly with information about the miner, while it had to be guessed according to the *coinbase_data* string for the other samples. For the Ethereum-based blockchains, many small miners were therefore accounted in the analysis, which negatively impacts the scores. To verify this claim, we want to refer to Figure

2.3 showing the share of blocks mined in a week by the largest miner. It can be seen that the share of the largest miner in the network is quite similar for all blockchains. The largest miner produces about one-third of the blocks for each blockchain. Therefore, the total value of centrality scores does not contain much additional information. As we are more interested in the development over time than in total centralization, this does not have any drawbacks.

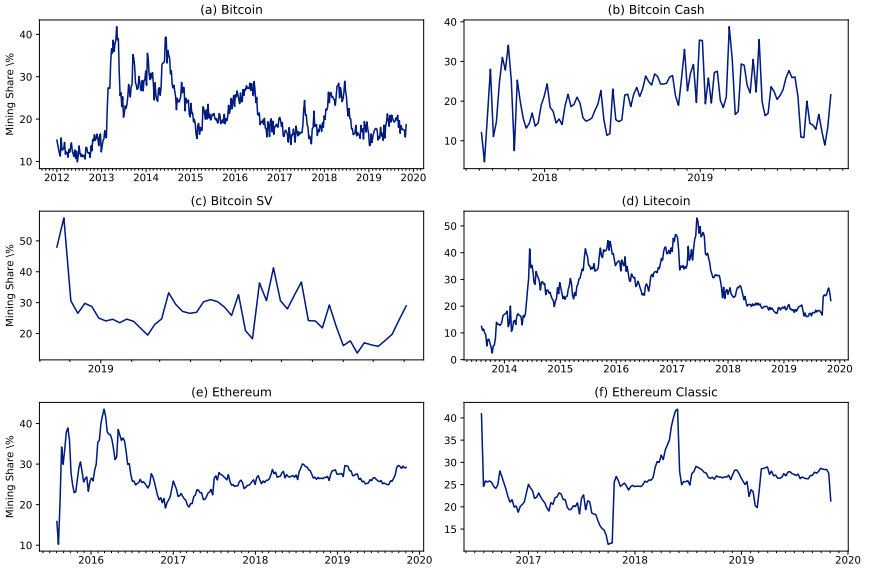


Figure 2.3: Share of Blocks found by the Largest Miner per Week

Taking a closer look at Figure 2.3, it can be seen that the largest miner rarely reaches more than 50% of the total mining power. The only two instances when this occurred were Bitcoin SV shortly after its fork and Litecoin in June 2017. It is interesting to observe that after these peaks are reached, the mining power of the largest pool harshly drops. Similar declines can be seen in the other blockchains shortly before reaching the 50% mark. The data shows that the drop did not occur because the rest of the network increased the mining power as a defense mechanism. Instead, the mining pool decreases its own power. So miners leave a large mining pool before it becomes too powerful and join a smaller one to keep the network balanced. The explanation for this behavior is straightforward. A blockchain with monopolized mining is not trustworthy anymore. Since trust is the only thing that gives cryptocurrencies value, miners have to do anything in their power to maintain the trust in their network. Performing a 51% attack on your own network is not viable since the coins acquired by such an attack have no value afterward.

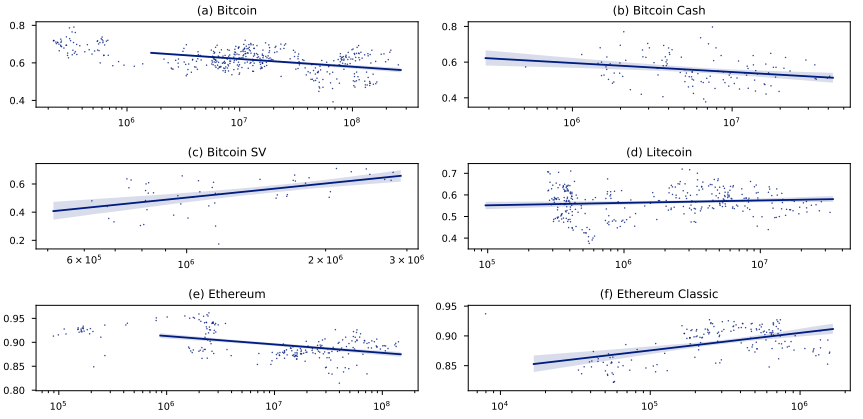


Figure 2.4: Relation between Mining Profitability and Centralization

We are still looking for an explanation for long-term fluctuations in the power distribution. The overall downward trend observed in Bitcoin and Ethereum could be explained by the rising price of the respective coins over the last years. The reasoning behind this is that if mining is profitable, it is lucrative for small miners to join the market. As profitability sinks, however, it can be unfeasible for some miners to participate in the process as their electricity costs exceed the mining reward. We, therefore, studied the correlation between mining profitability (i.e., average mining reward in USD per week) and centralization. The relation between these two indicators is depicted in Figure 2.4. To get an overview of how these figures develop over time, see Figure 2.5. There seems to be a positive correlation for the large cryptocurrencies while the trend reverses for smaller ones. These correlations are not likely to imply causation. Even if our assumption is true, small individual miners tend to join the mining pools with the most benefits, which is not necessarily a small one. For different blockchains, this could be either a large mining pool, which would increase centralization or a small one, which would result in the opposite. To gain further insights into this phenomenon, the structure of mining pools has to be further dissected.

While overall profitability seems to have no unified effect on centralization, sudden changes in profitability may have an impact. Since it was already hypothesized that lowering mining rewards could have unpredictable consequences on mining behavior and we could not identify long-term effects, we now look at short-term effects. In particular, we observe mining behavior around reward halving events. Figure 2.5 shows the development for the mining profitability compared to the centralization.

It can be seen that profitability halves immediately after the events for most blockchains (note the logarithmic scaling on the y axis). The only exception is the Ethereum blockchain.

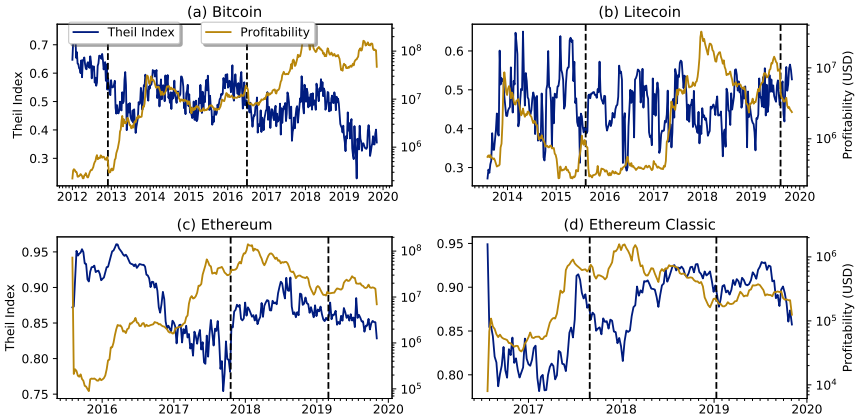


Figure 2.5: Development of Centralization and Mining Profitability over Time. Horizontal Lines Indicate Reward Halving Events.

This is because it was originally planned to sharply increase the mining difficulty to switch to a different consensus algorithm. This mechanism is referred to as the *difficulty bomb* (Fairley, 2018). However, this plan was dismissed as the technology was not ready, and the difficulty was adjusted back to normal at the time of the reward reduction. This happened at both halving events and actually increased mining profitability since even though the reward per block decreased, the rate at which blocks were created increased sharply.

However, there is no unified effect on centralization. While Ethereum experiences a sharp increase in centralization at the first halving, this is not the case at any other point in time. The increase seems to be a correction after centralization decreased when the difficulty bomb was active. Otherwise, there is no other pattern visible at the time of a reward halving. Since reward halving is a foreseeable event programmed into the blockchain protocol, miners can adapt in advance to these changes and plan accordingly. Therefore, the effects are not as unpredictable as Beikverdi and Song (2015) expected them to be.

A more unpredictable event is the fork of a blockchain. While there are clear signals that a majority will not accept a protocol change, it is hard to predict whether this results in a fork. Figure 2.6 shows the centralization of forked blockchains next to each other. The graph shows very well that the centralization of the smaller fork fluctuates strongly after the fork occurs and stabilizes over time. This is an effect that is expected since mining pools have to be newly formed while it is unforeseeable how many miners follow the forked protocol and in which mining pools they will participate. Only after the fork is active for some time, the mining pools can be organized to keep the network safe. The strong centralization and the week with over 50% mining power in Bitcoin SV immediately after the fork results from a reorganization that was not fast enough.

It can be observed that forks have no visible effect on the blockchain being forked since the vast majority of miners remain in the bigger network. When Bitcoin Cash started, the Bitcoin network stayed stable. The same happened when Ethereum was forked. While there were some fluctuations in the difficulty of Bitcoin Cash after the Bitcoin SV fork, the centralization was kept relatively stable. There was a slight uptick in the week after the fork, and the centralization increased marginally. This is because the portion of miners that left Bitcoin Cash for Bitcoin SV is significantly larger than usual for forks. This means that many miners leaving the network destabilize the network more.

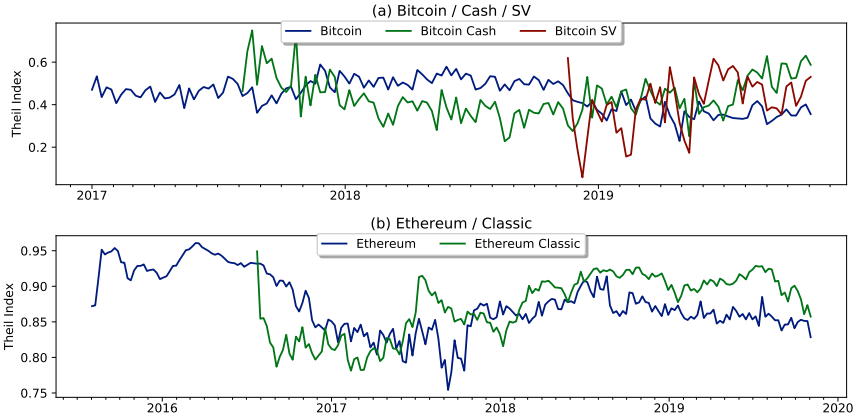


Figure 2.6: Centralization over Time for Bitcoin, Ethereum and Forks

While forks have unpredictable effects on centralization, miners take only a short time to adapt and stabilize the networks. It is also interesting to see that there are no cases of large miners switching to a smaller network to make it unusable. This would make sense to get rid of the competing protocol derived from the own chain. However, if miners would attack a smaller blockchain, this could severely weaken the trust in the attacked blockchain and the whole technology itself. Again, since trust gives cryptocurrencies value, this would harm the attackers themselves. This is a beneficial side effect protecting smaller blockchains from attacks. Therefore, performing a 51% attack on a different network can severely harm your network value and is thus economically not viable.

Overall, there are no prolonged periods where centralization is exceptionally high or when a miner owns a majority of power. While there were rare cases of actual 51% attacks, the system seems to be a self-maintaining ecosystem. For business use cases, a blockchain with a long history and a large network is the most reliable and harder to attack. It would cost, for example, over $\$500,000$ to perform such an attack on the Bitcoin blockchain for one hour (crypto51, 2019). If protocol changes are being implemented and a fork is likely, it is

particularly important to check if the protocol changes are relevant for the business. If a majority rejects the changes, it could be unwise to switch to the forked chain before the fork has stabilized. Unfortunately, this means that all transactions have to be paused until this point, which is not always possible.

2.5 Discussion

Since the key selling point of blockchains is their decentralized governance, the power distribution in blockchain networks is a major factor for the security of the technology. This leads to the question, which consensus mechanism provides an equal power distribution among the network nodes. To tackle this broad question step by step, we first focused on PoW, the most prominent consensus mechanism. We found that prior research on this topic is dated and highlights effects not relevant to current ecosystems. There was a particularly strong focus on mining pools as they were regarded as a major problem. To mitigate this problem, some solutions were suggested, which allow reward sharing without centralized pools (Cong et al., 2020). However, other effects on mining centralization, such as forks, have been ignored so far, even though they were identified to have an impact on blockchain governance (Risius and Spohrer, 2017). We analyzed major PoW blockchains for their mining behavior to better understand these effects.

We, therefore, posed the first research question on how the distribution of mining power changes over time. We noticed that no further centralization did occur after the formation of mining pools. Instead, the centralization follows a wave pattern, constantly correcting power imbalances. To answer our second research question, we looked for possible centralization causes. Therefore, we investigated the influence of profitability, reward halvings, and blockchain forks. While the first two had no unified effect, forks can have unpredictable short-term effects on centralization. This is contradictory to the hypothesis posed by Beikverdi and Song (2015).

Along with this research, we found evidence for a working incentive mechanism, which prevents monopolization of power. Not only in large blockchains, which are hard to attack but also in smaller blockchains, which could be easily overruled. Overall, the criticism of centralizing behavior is unjustified, at least for the mining behavior.

Even though the results matched our expectations, it is essential to highlight the limitations of this research. Using the data of only six different blockchains might not be sufficient to generalize the observations. While our data set covers a large share of the mining community, we ignored smaller blockchains. Although this limits our research, it was necessary since the preprocessing and analysis of these data sets is very time-consuming. The smaller PoW blockchains could provide additional insights into miners' overall decision process, especially since there are mining protocols in use that do not allow pooled mining. This makes it possible to analyze a miner's individual behavior closely. Limiting the external data sources to pricing data could hide many effects that we did not uncover. Additional data sources,

such as news articles on mining bans in certain countries, could explain some developments. Even though we could not pinpoint individual behavior to certain events, we gained a much better overall understanding of how the mining ecosystem works.

While the PoW mechanism seems to work very well for the blockchains we analyzed, it has a lot of other problems. Wasting computational resources and electricity to find a consensus along untrusted peers is an often criticized problem of the current blockchain economy (Vranken, 2017). This has led to other mechanisms, such as Proof-of-Stake, which does not rely on computational power but on capital investment to establish consensus. These mechanisms have to be analyzed similarly to answer the broader research question posed by Risuius. We want to encourage other researchers to adapt our approach to compare alternative consensus mechanisms. For example, it is hypothesized that Proof-of-Stake blockchains will have a more substantial effect on the riches getting richer (Zheng et al., 2018). Additionally, these mechanisms have a bigger problem in case of a 51% attack since it cannot be reversed without a fork (Nicolas, 2014). Therefore, we propose to research this mechanism next. Overall, the research gave many essential insights into blockchain centralization and made an important step towards an overall centralization theory, which ultimately can answer how to avoid monopolization of power.

Chapter 3

A Source-Code-Based Taxonomy for Ethereum Smart Contracts[2]

3.1 Introduction

Blockchain technology has emerged in multiple applications and thus became a disrupting technology in both information systems research and industry since its conceptualization and realization through Nakamoto (2008) at the end of the 2000s. Starting as a simple state-machine application (Bitcoin), blockchain quickly extended into an environment that allowed decentralized Turing-complete computations, today known as the Ethereum network. The already existing idea of so-called smart contracts published by Szabo (1997) was enhanced into decentralized smart contracts to become a substantial driver for automation. Despite repeatedly being confused with contracts in the legal sense, a smart contract may also execute arbitrary program code to carry agreements and their implications between contractors. Based on this, a smart contract built like an application and fitted with a user interface (front-end) is called a Decentralized Application (DApp) (Antonopoulos and Wood, 2018).

In contrast to a regular app, a DApp runs its back-end code in a decentralized peer-to-peer (P2P) network instead of central servers. The underlying program is published on the blockchain as byte-code to ensure compatibility and acceptable performance. Once published, the smart contract code can not be changed, and errors can only be fixed by deploying a new code version. However, the old version will always be accessible. Therefore, programmers have to ensure contract correctness and avoid unforeseen functional issues (Zheng et al., 2020).

[2]This chapter was published in *ICIS 2021 Proceedings* as Hofmann et al. (2021c) and co-authored by Julian Kolb, Luc Becker and Axel Winkelmann.

From a conceptual and practical viewpoint, creating and evaluating a smart contract is a complex task. To counteract these issues, developers rely on standardized functionalities that have proven to be efficient and secure. However, still among the most desired improvements for the Ethereum ecosystem, which is to date the most-used smart contract platform, are "more general-purpose libraries" and "more standard interfaces" (Zou et al., 2019). Therefore, the main concerns for developers are that it is hard to guarantee the security of smart contracts and the lack of powerful tools that support the development and testing of the smart contracts (Zou et al., 2019).

This paper aims to guide and standardize the development by identifying common patterns, functionalities, and their relations in state-of-the-art smart contracts, which can serve as a basis for discussion for the development of standards and libraries. Therefore, we analyze Ethereum smart contract development and categorize smart contracts based on mutual code patterns. We summarize the resulting constellations in the form of taxonomy to "provide a structure and an organization to the knowledge of a field, thus enabling researchers to study the relationships among concepts" (Nickerson et al., 2013). Understanding the code patterns and their relationship can help researchers and developers focus on specific areas, especially where standardization is lacking or external libraries are being widely used but not yet standardized. Therefore, we focus on the following research question:

RQ: Which common code patterns are used to develop smart contracts, and which archetypes of smart contracts can be distinguished based on these patterns?

Several blockchains are using the Solidity programming language. However, we focus our research only on smart contracts deployed to the Ethereum network since the source code of individual smart contracts is mostly publicly available. While this poses a limitation on the generalizability of the results, we argue that for a first, focused discussion, the Ethereum blockchain is an ideal candidate, as it is the most popular blockchain for practitioners and researchers alike. Additionally, the Ethereum blockchain hosts smart contracts for a wide range of applications from the areas of decentralized finance (DeFi), games, collectible assets, and social networking. We contribute to smart contract research and development by answering our research question while creating a taxonomy of smart contracts to support structuring the scientific discussion.

This paper organizes as follows to answer the research question: In the next section, we present the previous work related to our research, followed by our research approach for the taxonomy derivation and evaluation in our future research. Subsequently, we introduce the processes of defining meta-characteristics and ending conditions, data collection, taxonomy building, and the final taxonomy. We then proceed with a cluster analysis to identify relations of common patterns and summarize them into archetypes of smart contracts. Lastly, we examine the primary findings, their implications for research and practice, limitations, and future research building on this study.

3.2 Foundations and Related Work

Due to the initial use case of Bitcoin as the first blockchain application, the blockchain landscape had mainly consisted of cryptocurrencies. This landscape diversified with the development of more complex blockchain use-cases and programmable smart contracts. The first attempt to structure the various potential use-cases emerged in March 2015, three months before the initial release of the Ethereum platform, ultimately leading to the first wave of DApps (Buterin, 2014; Glaser and Bezzenberger, 2015). In the years to follow, more and more blockchain applications were published, which lead researchers to further structure those applications in some manner.

As of today, some taxonomies on blockchain platforms and applications already exist (Sarkintudu et al., 2018; Tasca and Tessone, 2019; Wieninger et al., 2019). Sarkintudu et al. (2018) and Tasca and Tessone (2019) provided taxonomies on blockchain platforms. The identified characteristics ranged from fundamental features, such as the underlying consensus mechanism, to specific considerations, such as the programming language and feature-set for smart contracts running on the platform. The taxonomy of Tasca and Tessone (2019) is more detailed in these regards and can be adopted as a solid basis for designing novel blockchain platforms.

In more specialized insights, taxonomies are focusing on the consensus mechanisms and provide profound insights (Yeow et al., 2017). However, the researched consensus mechanisms are not limited to blockchain networks but also contain other forms of distributed ledger structures, such as directed acyclic graphs. To understand these structures and design options, we refer to the taxonomy of Ballandies et al. (2021). Here, the research focuses on the data structure, transaction structure, and consensus of distributed ledgers.

So far, the research on smart contracts taxonomies is scarce. Tönnissen and Teuteberg (2018) have built a smart contract taxonomy and identified nine dimensions closely related to the ones found in legal contracts based on literature. This taxonomy is geared towards conceptualizing a smart contract based on business requirements. In contrast to this approach, we analyze smart contracts on a source code level to support the construction of smart contracts on a technical level.

Smart Contracts on the Ethereum Blockchain are primarily programmed in Solidity, a high-level programming language similar to JavaScript, making the transition easy for web developers. While Solidity allows writing maintainable and understandable code, this code can not run directly on the Ethereum blockchain. To ensure compatibility and high performance, the source code is optimized and compiled into byte-code, that can be run on the Ethereum Virtual Machine (EVM). This compiled code gets submitted to the Ethereum network and published. After publishing, the byte-code can be viewed by anyone who holds a copy of the Ethereum ledger or uses a blockchain explorer. However, this compiled code is not human-readable, and users are reluctant to interact with smart contracts with obfuscated functionalities. Therefore, developers have the option to publish the human-readable smart

contract so that users can transparently verify the claimed functionalities. In this paper, we aim to categorize smart contracts based on this published source code.

There have been other approaches to analyze smart contracts on a technical level. Though, their goal was not to classify and group code patterns. Wöhrer and Zdun (2018) for instance, utilized a similar source-code-based approach to analyze design and security patterns (Wohrer and Zdun, 2018). However, the authors did not show how those are combined in real-life smart contracts. Bartoletti and Pompianu (2017) also propose a taxonomy focused on the contracts' usage area and application categories. The authors also suggest some high-level design patterns. However, from a technical perspective, the provided patterns and categories are too superficial to provide meaningful guidelines for developing secure smart contracts.

3.3 Methodology

Our research follows a two-step approach that combines the qualitative and quantitative research methods as illustrated in Fig. 3.1 (Bryman, 2006). In the first two stages (A + B), we use an inductive taxonomy development approach according to Nickerson et al. (2013) to classify the properties and core elements of smart contracts. Especially technologies which are the main focus of research, such as blockchain technology, can be better explained and understood with the help of taxonomies (Oberländer et al., 2019). Taxonomies offer a set of dimensions with differentiated and unique characteristics, with each entity having exactly one suitable attribute for each dimension (Nickerson et al., 2013). To create a rigorous taxonomy, we followed the seven steps framework published by Nickerson et al. (2013), which is well established within this area of information systems research (Fellmann et al., 2018; Rizk et al., 2018; Tönnissen and Teuteberg, 2018). However, even with this approach, there is no guarantee that this is the optimal taxonomy since the research process is characterized by qualitative influences and by subjective decisions of the researchers (Nickerson et al., 2013).

In the first stage (A1-A4), we already used a limited set of data to develop an initial version of our taxonomy for gambling smart contracts (Kolb et al., 2020b) to outline the topic. Our previous paper analyzed the smart contracts of gambling DApps to identify 18 dimensions and 41 characteristics of your technical and code-based taxonomy of gambling smart contracts. We chose these types of contracts because they can utilize diverse functionalities and standards. The analysis provided an ideal starting point to validate our approach of analyzing source code to categorize smart contracts. In fact, in this first research, we could identify many fundamental concepts still present in our final taxonomy, such as token standards, the usage of helper functions, or ownership handling of contracts (Kolb et al., 2020b).

The first step was a rigorous and reliable data collection process (A1). We then determined the meta-characteristics and the necessary ending conditions (A2). In the third step, the actual taxonomy was defined (A3). According to the taxonomy building guidelines of Nickerson et al. (2013), we used an iterative process to create, check, and modify dimensions

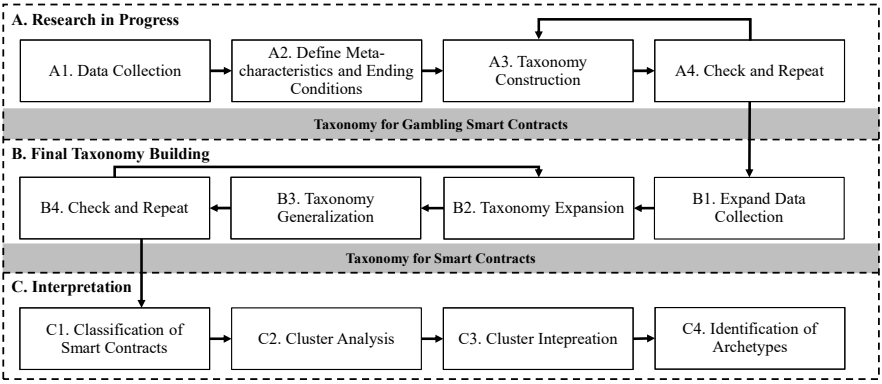


Figure 3.1: Overview of Research Methodology

and characteristics until the satisfaction of ending conditions (A4). We only outline these steps in this paper. For a detailed description, we refer to Kolb et al. (2020b).

It is now necessary to provide general guidance to contribute to the development and design of smart contracts in business usage. In the paper at hand, we expand the data to a broad set of different smart contracts. To achieve this, we extend and finally generalize the taxonomy still following the same guidelines of Nickerson et al. (2013) as shown in Figure 3.1 (stage B). In stage C, we extend the purely descriptive analysis by clustering the smart contracts and identifying archetypes. To do so, we classified the smart contracts at hand with the final taxonomy (C1) and afterwards performed a cluster analysis to identify different archetypes to determine similarities and overarching patterns (C2). We then analyzed the clusters and interpreted the findings (C3) to ultimately identify archetypes of smart contracts (C4). These archetypes should help researchers and practitioners to understand the smart contract landscape better. For example, developers who want to build new Ethereum applications can use the archetypes to classify their applications and use the taxonomy to guide best practices for applications in these categories. Furthermore, it guides in identifying possibilities to improve the tools, libraries, standards, and techniques currently used to develop smart contracts on the Ethereum blockchain by highlighting the most used functionalities, which still lack standardization or toolsets. Ultimately, future developers can use our findings to streamline their results and efficiently learn from other projects.

3.4 A Taxonomy for Smart Contracts

The following section describes our process of taxonomy building. The final taxonomy will be presented in the section ahead.

3.4.1 Data collection

Our data collection process involves identifying appropriate resources and tools for gathering the data itself. We have conducted our inductive research approach by collecting data from the following databases: dapp.com[3] and stateofthedapps[4]. Both databases offer to filter by blockchain technology (Ethereum, EOS, Steem, ...) and categorization of the listed DApps. To limit our dataset, we filtered it for Ethereum based applications. In contrast to previous research, we did not limit our research to one category but analyzed all available categories (art, exchange, finance, gambling, game, high risk, tools, social, and others). We then added the to our previous gambling dataset. In the next step, we excluded all DApps that had no user activity in mid-2020 to ensure the current significance of the data. After this step, the dataset consisted of 183 DApps. Ultimately only DApps that allowed access to the underlying source code were taken into account. Although the other smart contracts are available as byte code on the Ethereum blockchain, this code is difficult to analyze as coherent wording, crucial to the researchers' understanding, is missing. While there are methods to reverse engineer and decompile these smart contracts, they have not proven reliable enough for rigor analyses (Eevm, 2020). This is a limitation of the research at hand, which can not be circumvented in the foreseeable future. The whole process results in a final dataset of 101 DApps, as depicted in Figure 3.2.

It should be noted that these DApps are often based on more than one smart contract. For the taxonomy, we treat these as one single contract. The reason for splitting the logic into multiple contracts that interact with each other can be easier readability and maintainability of the source code. The set of source code documents analyzed consisted of 150 source code files.

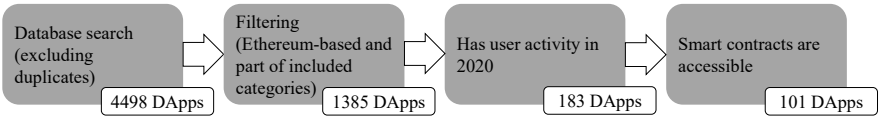


Figure 3.2: Data Collection Process

3.4.2 Meta-characteristics definition and ending conditions

The second step will define the scope, goals, meta-characteristic, ending conditions, and the taxonomy structure itself. The intention is to structure the utilized smart contracts to identify core patterns of the source code. The ultimate goal is to provide the necessary groundwork for further standardization and development of future smart contracts. The meta-characteristic of the taxonomy is *smart contract classes, functions (i.e. procedures) and code patterns*, and it can be directly derived from our research question.

[3] `https://www.dapp.com`
[4] `https://www.stateofthedapps.com`

The ending conditions for the taxonomy development process are divided into objective and subjective ending conditions (Nickerson et al., 2013). The objective conditions are:

OC1 - A representative sample of objects has been examined. The set of 101 smart contracts is small compared to the estimated one million smart contracts currently deployed on the Ethereum network. However, we argue that the sample is on the upper limit of what is reasonable to analyze. Additionally, to ensure that the sample is representative, we used a stricter objective condition 2.

OC2- In the last five iterations, no characteristics or dimensions are combined, divided, or added. While Nickerson et al. (2013) suggest that in the *last* iteration, no dimensions should be combined, divided, or added, we expand this to the last five iterations. We do so to counteract the possibility of having a too-small sample. We examined a set of 5 smart contracts in each iteration, so if the taxonomy changes in the last five iterations, we have to expand our dataset by including less popular smart contracts. If the taxonomy does not change for five iterations, we can be confident that the dataset is representative enough.

OC3 - At least one object is classified under every characteristics of every dimension. We ensured that this ending condition is met after each iteration by only adding characteristics, with at least one object classified.

OC4 - Every dimension is unique and not repeated, every characteristic is unique within its dimension. Again, we ensured that this ending condition was met after each iteration. While we do have some characteristics, that are not unique (e.g., *implemented* and *Non-Implemented*), they are unique within their respective dimension.

The subjective conditions were, that the taxonomy has to be *concise*, *robust*, *comprehensive* and *extendable*.

SC1 - Concise. This condition encourages the taxonomy to be meaningful without being overwhelming. Note that Nickerson et al. (2013) postulated 5-9 dimensions as an adequate range (Nickerson et al., 2013). While our taxonomy exceeds these recommendations, it was not justifiable to delete or combine dimensions in the last iterations. We argue that this extensive taxonomy is necessary to grasp the possible granularity of design decisions on a code-level basis.

SC2 - Robust. The characteristics have to provide a sufficient differentiation among the objects. Differentiation is an important condition since it is important for clustering. If the characteristics do not differentiate enough, the clustering will not be meaningful.

SC3 - Comprehensive. All objects within the domain of interest must be able to be classified, all dimensions of interest must be identified. This condition is linked to OC2. We ensured that each new smart contract could be classified completely with the current state of the taxonomy for each iteration.

SC4 - Extendable. New dimensions and characteristics can be added to the taxonomy. While taxonomy is already extensive, there is still room to extend it. There is a constant stream of new applications deployed to the Ethereum blockchain with new trends and stan-

dardization efforts. When new patterns in smart contracts gain popularity, they can be added easily to the current taxonomy.

3.4.3 Taxonomy Building

After having defined the meta-characteristics and the ending conditions, we started to extend our taxonomy. The source code of the smart contracts was queried from Etherscan[5] and stored as text documents. The analysis of the source code was twofold: two researchers started highlighting the functions in every contract. While doing so, they derived a coding scheme in an iterative procedure using MAXQDA as supporting software. The coding scheme was only a supporting tool to enable the two researchers to have a standardized way to label functions and classes according to their purpose in the source code. Therefore, the coding system was not checked for inter-coder reliability. We then compared the results of both researchers and discussed them after each iteration within a panel of experts.

Overall we analyzed over 120,000 lines of source code and coded over 1,100 passages in this source code with a final set of 70 different codes. After each iteration, similar codes were grouped if possible and large coding categories were split if possible. Additionally, each researcher created a category "unknown" in each iteration where he marked passages in the source code that were not similar to the other categories or where the functionalities were unclear. Both researchers analyzed these together to decide how they should be labeled. The codes from the labeling process were used to derive the dimensions and characteristics of the taxonomy. Characteristics were added, removed, or merged depending on the number of occurrences in the smart contracts and their importance in the source code. This process was guided by the subjective conditions SC1-SC4.

3.5 Smart Contract Taxonomy

Our analysis of 101 smart contracts yielded 64 characteristics from 28 dimensions, grouped into six categories. The complete taxonomy is presented in Figure 3.4-3.7 and is described in the following chapter.

3.5.1 DApp Design

The category DApp Design contains dimensions and characteristics related to the basic architecture of the examined application. Our research has shown that DApps differ in their quantity of employed smart contracts, which we describe as *smart contract quantity*. Our dataset identified two distinct types of applications: DApps that combine all functionalities in one single smart contract and DApps that split functionality among multiple smart contracts.

[5] `https://etherscan.io/` - Ethereum (ETH) Blockchain Explorer

Table 3.1: Dimensions and Characteristics of *DApp Design*

Dimensions	**Characteristics**	
Smart Contract Quantity	Single	Multiple

3.5.2 Core Functionality

In the second category, we examined how smart contracts process their actual core functionality in the smart contracts and defined it as Core Functionality.

Most smart contracts within our research are including, what we call *Core Logic* within their code. These functions comprise game rules in gambling smart contracts or auction mechanisms in exchange or finance DApps. Other applications exclude these functionalities and access external software via various interfaces, returning results to the smart contract.

Smart contracts also deal differently with their *Usage Fee*. Some smart contracts provide a changeable fee, which the owner or administrator of the contract may modify. Other applications use an initially fixed usage fee or do not use such a fee at all.

When we first looked at gambling smart contracts, *Asset Handling* was a big issue there. However, this functionality is also widely used in other application areas and is part of the core functionality. Asset Handling mechanisms help to manage, transferring or proving ownership of various tangible or intangible objects. Those can be digital objects in games, real estate, or share certificates, among others. We could differentiate between smart contracts, which use transferable assets, or which only provide non-transferable assets.

Table 3.2: Dimensions and Characteristics of *Core Functionality*

Dimensions	**Characteristics**		
Usage Fee	Changeable	Fix	None
Core Logic	Included	Excluded	
Asset Handling	Transferable	Non-transferable	

3.5.3 Helpers

Many smart contracts require augmented features on top of the implemented core functionality. These features can be hard to implement within the contract, or sometimes data outside the smart contract needs to be accessed. For this purpose, many smart contracts include libraries, which provide various functions. In total, we have identified five different helper libraries frequently used in smart contracts: *Byte Helpers*, *String Helpers*, *Math Helpers*, *Oracles*, and *Interfaces*. While the helper functions provide functionalities that programmers are familiar with from other programming languages, Oracles and Interfaces are different. The former are used to retrieve data from non-blockchain sources. The ladder are used to interact with the smart contract from outside the blockchain.

Math Helper libraries provide different functionality regarding mathematical operations. From simple functions that determine the minimum or maximum of two (or more) numbers,

those libraries may extend to functions that implement square root, logarithmic, or exponential functions. Since these are based solely on the limited mathematical capabilities of the Ethereum byte-code, they often require an iterative approach to calculate the desired value based on the basic mathematical operations (division, multiplication, addition, and subtraction). This approach makes some of the functions costly to execute. Correspondingly, some math libraries carry as annotation the following warning: "This is where your gas goes.".

Similarly, *String Helpers* provide functionalities used to manipulate strings. Among the most common functions implemented are checking the lengths of a string, slicing it into substrings, and concatenating or comparing two strings. Since Ethereum does not provide a string datatype, the bytes datatype is used for this purpose. The bytes are always interpreted as UTF-8 encoded strings in the contracts we examined.

Byte sequences that do not represent strings but store arbitrary data are also common to the contract code. *Byte Helper* libraries are used to manage this very flexible data type. Like the string functions, they often allow slicing and concatenating byte sequences, but they are also used to transform bytes into other data types like unsigned integers or Ethereum addresses. Unlike the previous libraries, the Byte functions often use inline assembly code to manipulate storage efficiently.

An *Oracle* is a service that allows importing data into a DApp or smart contract from an external source like the Internet (Xu et al., 2016a). These Oracles are mainly used to query the results of external events (like sports matches, real estate data, or stock exchanges) or to include an external source of randomness (especially in gambling and high-risk contracts). Querying Oracles is possible through services like Provable™, that supply their libraries to interact with the Oracles (Provable, 2020). Oracles have some criticism since they rely on a centralized source of truth, which can be manipulated on an otherwise very secure network.

Interfaces are needed to interact with smart contracts from outside the blockchain. While every smart contract provides callable public functions, certain conventions allow interaction with a contract in a standardized way. While many token standards provide their standardized interface, querying whether a contract supports or not is difficult. Therefore, the ERC165 standard can be used to query a specific contract for its available standard interfaces.

Table 3.3: Dimensions and Characteristics of *Helpers*

Dimensions	**Characteristics**	
Math Helpers	Implemented	Non-implemented
String Helpers	Implemented	Non-implemented
Byte Helpers	Implemented	Non-implemented
Interfaces	ERC16	Others
Oracles	Implemented	Non-implemented

3.5.4 Contract Management

We noticed that smart contracts differ very clearly in their ability to be controlled. Among others, this includes roles, ownership handling, rebranding, updating, and killing smart contracts. We have summarized dimensions in this area in the Contract Management category.

First, we propose a differentiation between different *Roles*. The roles *Token Owner* and *Admin* are usually available. Both can occur alone or in combination. In addition, some smart contracts define individual roles, which we have not included as a further characteristic because they differ from application to application. Using roles permits the smart contracts to control various user functionalities: the right to execute transactions, change the contract, or generally access certain functions. Some smart contracts may also not use roles at all.

In addition to the Admin and Token Owner roles already introduced, there is the contract owner. The owner of a smart contract has significant rights, like killing and pausing a contract. We identified various ways of how smart contracts deal with *Ownership Handling*. First, some smart contracts do not allow ownership and are therefore non-ownable. Opinions differ here as to whether an owner has a positive or negative effect on a smart contract. However, we observed that about one-third of the smart contracts do not utilize the implementation of an owner and thus fully support the principle of decentralization in a blockchain. We also found out that not all smart contracts provide a function to transfer ownership. This can have multiple reasons, yet the predominant is that a smart contract should not be transferred at all. We also investigated that in some contracts, a transfer can be renounced or must be actively accepted. We did not expect this functionality at the beginning of our investigation and were surprised about this feature. We assume that it is used as a security feature preventing an accidental transfer of ownership in some cases.

Another dimension in the Contract Management category is the ability to *Rebrand* a smart contract. Only 5% of the examined smart contracts implemented this function, but we are very critical about it. If such a possibility is implemented, the admin or owner can rename the smart contract and present it differently to the outside world. We assume that this function is used mainly by dubious applications, allowing scamming more users under different names.

While the contract code is unchangeable once a smart contract has been deployed to the blockchain, there still exist ways to make the smart contract *Upgradable*: Via proxy contracts. They delegate the contract call to the current version of the contract. If a new version is deployed, a variable in the proxy contract is changed to the address of the new version.

Finally, we distinct between smart contracts that are *Killable* and those that are not. About 15% of the examined smart contracts provide a function, which specifies the end of life of a smart contract and ends all pending transactions.

3.5.5 Safety Functions

Since smart contracts are characterized by a decentralized organization and usually do not have a central controlling authority, some Safety Functions are necessary to ensure proper

Table 3.4: Dimensions and Characteristics of *Contract Management*

Dimensions	**Characteristics**			
Roles	Admin and Token owner	Token owner	Admin	None
Ownership Handling	Non-ownable	Ownable, Transferable and Non-renouncable	Ownable, Transferable and Renouncable	Ownable and Non-transferable
Rebrandable	Rebrandable		Non-rebrandable	
Upgradable	Upgradable		Non-upgradable	
Killable	Killable		Non-killable	

operation. Their use can be to check transactions and validate them first to prevent false and inadvertent transactions. Some functions are employed to control and limit the influence of individuals within the system.

A *Check Address* function is used in smart contracts to check the validity of wallet addresses and other smart contracts before executing a function. This check can prevent permanent loss by transferring user tokens or other objects to an invalid address or smart contract. These functions can be either implemented or not.

The official Solidity guidelines recommend implementing a *Default Function* in a smart contract. This function gets executed when sending a transaction to the smart contract without input data and can be used to load funds into a contract. Additionally, it is often used as a safety measure to prevent users from accidentally sending Ether to the contract. In this case, the default function reverts the transaction. Often, however, the default function is implemented without any additional functionality.

By using the *Default Function*, non-specific requests to the smart contract are processed and, if necessary, rejected. Some smart contracts implement them without additional logic, while some throw error messages or execute supplementary code. Sometimes, a Default Function is not implemented at all.

Contrary to the *Math Helpers*, the *Safe Math* functionality is present in most contracts that require even the most simple calculations. The basic math functions in Ethereum do not check for overflow or underflows of the variables and can therefore yield wrong results. *Safe Math* functions monitor additions, subtractions, multiplications for overflows and underflows while additionally implementing integer division.

In 2016 over 3.6 million Ether were stolen when the contract of the popular DApp TheDAO was hacked. The attackers used a reentrancy attack to funnel the funds out of the contract. As a countermeasure, many smart contracts implement a *Reentrancy Guard*, that prevents this type of attack.

Especially token sales try to prevent single users from acquiring lots of tokens in an early sale stage. These measures are summarized as *Anti Early Whale* protocols.

If a contract shows unexpected behavior or is under attack by a malicious party, some contracts have a *Pause Contract* functionality. This function can pause and unpause the complete functionality of the contract until normality is restored.

Another method to deal with unexpected behavior is the option to *Refund Users*. Users can request a refund that is to be approved by contract owners or administrators if the request is justified.

In some cases, a transaction may get stuck in an automated contract (pending) and cannot be processed further. In this case, around 20% of the contracts in our dataset offer the possibility to *Withdraw Pending Transactions* to the user. In this case, the transaction is cancelled, and affected tokens are credited back to the user.

Table 3.5: Dimensions and Characteristics of *Safety Functions*

Dimensions	**Characteristics**		
Check Address	Implemented	Non-implemented	
Default Function	Implemented	Additional Logic	Non-implemented
Refund User	Implemented	Non-implemented	
Safe Math	Implemented	Non-implemented	
Reentrancy Guard	Implemented	Non-implemented	
Anti Early Whale	Implemented	Non-implemented	
Pause Contract	Implemented	Non-implemented	
Withdraw Pending Transaction	Implemented	Non-implemented	

3.5.6 Tokens

Tokens are the components that create an economic incentive in blockchain technology. They are units of local values and are primarily used to foster the operation of a blockchain (Shin et al., 2019). For example, nodes in a blockchain receive rewards for their work in the network using different protocols. In some cases, some marketplaces allow these tokens to be traded with each other and being exchanged for fiat money, thereby assigning them a monetary value (Hülsemann and Tumasjan, 2019).

These tokens can be characterized by different criteria, as demonstrated in the following. For example, Euler (2021) and Hülsemann and Tumasjan (2019) divide the intention of token use into cryptocurrencies, network tokens, and investment tokens. In our dataset, we could not detect any significant differences. Therefore, we concentrate on the technical features (*Token Usage*). Hülsemann and Tumasjan (2019) classify them as blockchain-native tokens, non-native tokens, and DApp tokens. Within our analysis, however, we only recognized native tokens or DApp Tokens.

When implementing a token into a blockchain network, the developer can choose between different *Token Standards* or create a new one: The ERC20 standard served as the groundwork for more recent standards such as ERC223, ERC667, ERC721, and ERC777 developed within the Ethereum network (Victor and Lüders, 2019). These differ according to various characteristics such as fungibility or other individual features. Among the examined DApps, the characteristics for the dimension token standard could be almost exclusively identified as ERC20 (40%) or ERC721 (10%) tokens. In our first study, which took place about six months earlier, the ERC721 standard was still very rare. In the meantime, however, it

obviously established itself and is now used in about 10% of the contracts examined. Approximately 40% of the smart contracts studied do not use tokens at all. The remaining 10% either use Multiple standards or rely on a Modified token.

As a further dimension, we distinguished between Contracts where new tokens can be created or not. Mintable tokens allow the user to mint a token or stop the process, such as mint() and finishMinting(). Most mintable tokens are ERC20 based, which include an additional function that helps to increase the stock. This means that the supply is not fixed, although you can specify the initial stock level in the contract. According to our results, tokens can be mintable or are already created initially.

The opposite of *Mintable* tokens are tokens that are *Burnable*, which means that tokens can be destroyed and never used again. These two features do not exclude each other. While some contracts implement burnable tokens accidentally by allowing tokens to be sent to an invalid address, the *Burnable* feature is an intentional property of the token.

Finally, we can describe the characteristics of *Trading* and *Accounting* for the tokens in use. First, tokens can either be *sold* and bought, or the functionality is not implemented. Furthermore, either deposit and withdrawal are possible, or only withdrawal is permitted. As before, this functionality may even not be implemented at all.

Table 3.6: Dimensions and Characteristics of *Token*

Dimensions	**Characteristics**				
Token Usage	Native Token			DApp Token	
Token Standard	ERC721	ERC20	Multiple	Modified	None
Burnable	Burnable		Non-burnable		
Mintable	Mintable		Non-mintable		
Token Trading	Buy and Sell		None		
Token Accounting	Deposit and Withdrawal		Withdrawal only	None	

3.6 Archetypes of Smart Contracts

Our descriptive analysis provides an overview of real-life examples of smart contract characteristics and code structures. However, it does not shed light on the combined characteristics of distinct types of smart contracts that act as boundary objects in the Ethereum smart contract ecosystem. Therefore, we performed a cluster analysis to determine these latent functional clusters of smart contracts that represent smart contracts archetypes. Additionally, a meaningful result of the clustering validates the meaningfulness of the taxonomy to some extent. We used agglomerative clustering hierarchical clustering (Wards Method) with the Jaccard distance because this method is well suited for clustering categorical data, as seen in the analysis of other taxonomies (Gimpel et al., 2017; Fischer et al., 2020). Choosing the appropriate number of clusters is always a challenging task. To support our decision, we analyzed three common metrics: the Elbow-Criterion (Madhulatha, 2012), Silhouette coefficient (Devaraj et al., 2007) and Dunn index (Devaraj et al., 2007). Figure 3.3 displays the three

scores concerning the number of clusters chosen. While according to the Elbow criterion, the optimal number of clusters should be five or seven, the other scores suggest a much higher number of clusters. As with previous research, there is a trade-off between interpretability and accuracy of the clustering. We, therefore, opted for seven clusters since it yields a suitable basis for interpretation as it shows a significant decrease in within-cluster variance. In our opinion, it represents the most comprehensive yet manageable solution to distinguish smart contracts.

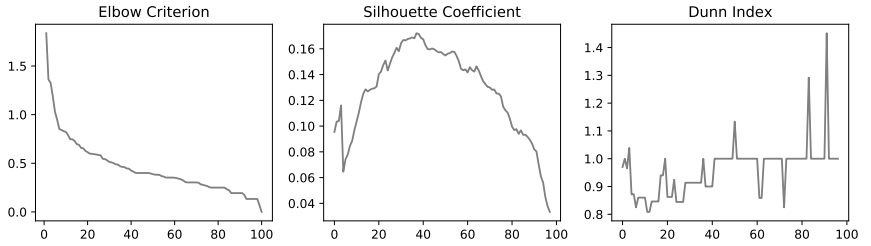


Figure 3.3: Clustering Scores for Optimal Choice of Clusters

Due to distinct functional differences, the number of observations comprised in each cluster varies significantly. The cluster sizes reach from 9 to 25 smart contracts per cluster. However, most hold between 10 to 15 contracts. In Figure 3.4-3.7 we summarize the clustering result and visualize the smart contract features in addition to the full taxonomy. In the rest of this section, we describe the identified clusters, draw conclusions on development structures and provide actionable insights for future smart contract development.

We describe the clusters not in the same order as they are shown in Figure 3.4-3.7 to highlight similarities and crucial differences between the clusters.

3.6.1 Archetype 4: "Bets" on Off-Chain Events

The contracts in this category have two notable similarities: using Oracles to interact with off-chain data and a fully implemented default function to receive Ether. The contracts are mostly classified into the *Gambling* and *Exchange* categories. The contracts allow bets on real-life events with Ether (may it be sports or financial bets).

While most contracts relied on a library for interaction with oracles, the specifics of the implementations often differed. As Zou et al. (2019) noted, developers wish for easier interaction with off-chain data. We agree that there is a lack of standardization. Notably, there exists an Ethereum Improvement Proposal (EIP) to tackle this issue and provide a standardized oracle interface (Lu, 2021). It should be noted that two gambling contracts combined Oracles with string helper functions to process the retrieved data. This parallelism

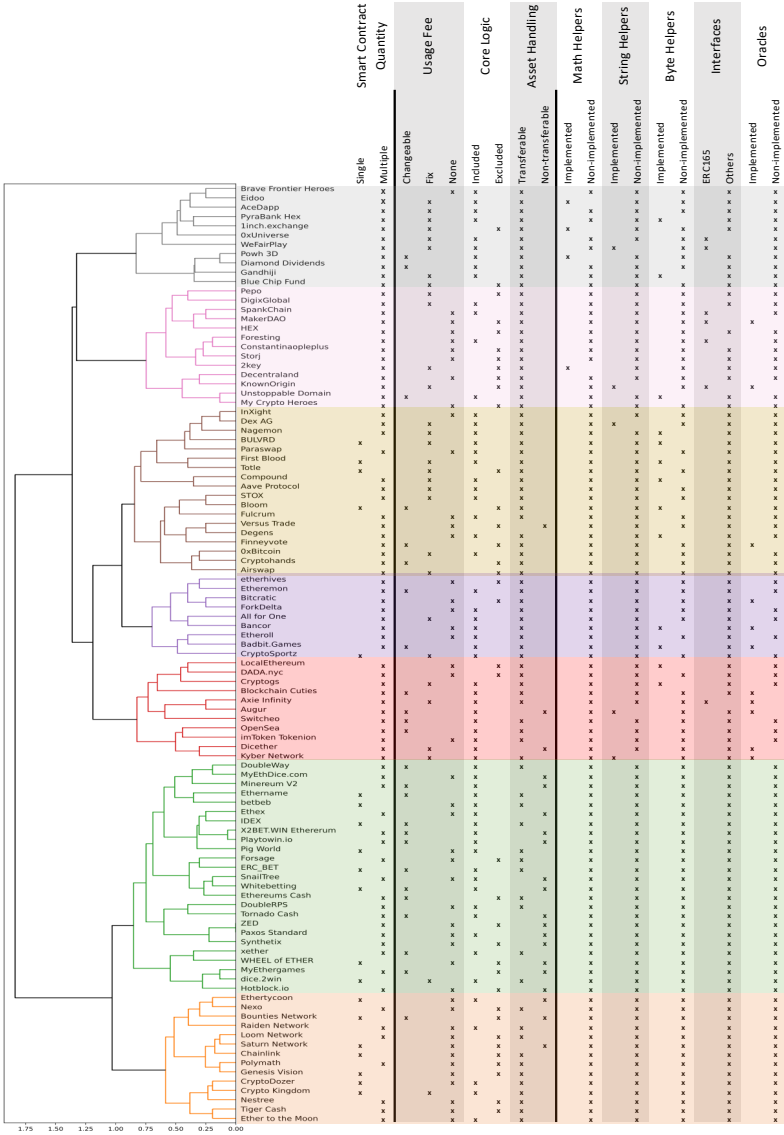


Figure 3.4: Clustering and Visualization of Smart Contract Characteristics (1/4)

Figure 3.5: Clustering and Visualization of Smart Contract Characteristics (2/4)

Figure 3.6: Clustering and Visualization of Smart Contract Characteristics (3/4)

Figure 3.7: Clustering and Visualization of Smart Contract Characteristics (4/4)

should be kept in mind when standardizing oracles so that developed string libraries or even native data types are compatible with the oracle standard.

It is noteworthy that the usage of oracles is somehow very limited to the use-case of betting. We are unsure about the reason behind this but hope that standardization can help adopt the concept to broader applications.

3.6.2 Archetype 2: Technically Secure Implementations of Financial Applications

This archetype includes mainly contracts that include functionalities of *initial coin offerings* (ICOs) and general token sales. There is a high usage of ERC20 tokens with the Burnable function (e.g., burning unsold tokens). Often they also include the functionality to mint additional tokens. However, the main functionality of these tokens is that they can be bought, sold, and transferred through the smart contract. While ICOs generally yield a financial risk for investors, the high usage of standardized functionalities, implementation of security features, and in some cases, usage of helper libraries indicates that developers and issuers of these ICOs are aware of risks. This use is undoubtedly motivated by the implications of losing the revenue of the ICO due to technical errors.

Here, we can show that there is enough knowledge in the developer community to build secure financial applications. However, the knowledge should be accumulated to implement standard libraries for this security functionality.

3.6.3 Archetype 1: Tokenized Asset Contracts without user management

The tokens in this cluster all used one or multiple token standards. However, the tokens can not be bought, sold, deposited, or withdrawn, which is the main differentiator between this Archetype and Archetype 2. A similarity, however, is that almost all of them have the functionality to mint or burn tokens. In this case, this is to create or destroy assets in an otherwise self-contained system. Half of them implement the functionality to check the recipient address to prevent accidentally losing assets by transferring them to an invalid address. The contracts track things like game assets or register (domain) names on the blockchain.

3.6.4 Archetype 3: Asset Centered Contracts with User Management

This cluster is very similar to Archetype 1. However, in this cluster, most contracts handle their ownership, and the roles of users. The question arises whether these functionalities were not included in contracts of Archetype 1 because they do not provide an additional benefit or are too complex to implement for added value. It can be hypothesized that if ownership and role management are standardized, Archetypes 1 and 3 merge into one since their functionalities get implemented in almost every contract. Here, further research on the standardization of governance in Ethereum blockchains is needed to provide a viable solution. We propose

to conduct further research in this area along with the research framework for governance in blockchains developed by Beck et al. (2018).

3.6.5 Archetype 0: High Value Asset Management

The smart contracts in this cluster have much functionality regarding the ownership of the contract. Additionally, some of them provide role management. The rest of the functionality is quite mixed, and the cluster is quite unstructured. The contracts in this cluster use many tokens. However, they are not standardized. However, almost all of them offers users the functionality to withdraw pending transactions, a feature that is rarely seen in other archetypes. The contracts often handle collectables. Therefore, losing one because of a faulty transaction or insufficient gas can yield high losses. We argue that the functionality of withdrawing pending transactions should be implemented more often with non-fungible tokens such as ERC721, which can represent ownership of valuable assets.

3.6.6 Archetype 5: Simple Contracts and Miscellanous

This archetype does not offer much standard functionality with other contracts. Contracts of this archetype do not implement any helper functions, role management and rarely any standardized function. This can either mean that the contracts have a simple structure. However, many contracts are comprised of multiple contract files. Therefore, this cluster is a catchall category for otherwise uncategorized contracts.

3.6.7 Archetype 6: Simple Contracts and Miscellaneous with Asset-Based Governance

Like Archetype 5, this category is not as clearly defined as the other archetypes as very few patterns are present in this cluster. However, unlike Archetype 5, the contracts employ more asset handling functionalities and define roles for token owners. The core functionality of some contracts is far from standard functionalities. For example, solutions that increase the transaction rate on the Ethereum blockchain or make it interoperable with other blockchains are in this category. Standardization of these contracts is quite challenging. However, these are rare use-cases, and we, therefore, do not see a necessity for standardization.

3.7 Conclusion

Smart contracts and DApps are part of the disruptive blockchain technology, facilitating peer-to-peer transactions and decentralized applications. They have, therefore, become an increasingly important topic in information systems research. However, their technical and functional characteristics are not yet well understood and not yet standardized. To structure

the smart-contract landscape on a technical level, we developed a source-code-based taxonomy for Ethereum smart contracts.

We used a data-driven method of taxonomy building to provide descriptive knowledge and a structure that had been missing in the previous discussion and development of smart contracts. We, therefore, collected empirical data on 101 DApps comprising 150 smart contracts and used an iterative research approach following Nickerson et al. (2013). This approach led us to define a final taxonomy consisting of 28 dimensions with 64 characteristics based on the six meta-categories. As defined by its nature, a taxonomy is never complete and should be expandable in its dimensions and characteristics as new objects emerge (Nickerson et al., 2013). However, it turned out that challenging our first taxonomy, which focused on applications in gambling, only resulted in minor changes and extensions. No new meta-categories were created or significantly changed. After the final taxonomy was created, we classified the examined DApps for evaluation. All 101 DApps could be fully mapped in the taxonomy. We then clustered the smart contracts and identified seven archetypes of smart contracts. We could identify some patterns that show room for improvement, especially regarding standardization and the development of additional libraries.

By providing a taxonomy for smart contracts, which can be used for future research and development projects, we contribute to current research and practice. Uncovering the technical characteristics within usage categories may help researchers better classify and analyze smart contracts in depth. New blockchain applications can now be conceptualized on a platform level using higher-level taxonomies such as the one from Tasca and Tessone (2019). Then the smart contracts can be conceptualized on a business level based on the work of Tönnissen and Teuteberg (2018). Finally, our taxonomy can then be used to specify technical details for the programmers.

By analyzing the clusters, we showed that applications have strong technical similarities while sometimes serving different purposes. The development of new contracts can use our results to consider architectural designs and standardize functionalities. For example, we identified similar functionalities used by secure financial applications that could be merged into a financial application framework. This would enable programmers to bundle their resources and reuse secure functionalities.

Qualitative research work such as source code analysis is usually subject to the subjectivity of the researchers involved. While source code does not leave much room for interpretation, we have considered this problem by integrating additional researchers into the process. The coding was carried out, verified, compared and discussed by a total of four people. Nevertheless, even after this generalization of the taxonomy, it is still possible that our coding is incomplete, incorrect or subjective.

Additionally, this research is focused on only one blockchain network. Some challenges go beyond the Ethereum ecosystem. Therefore, we suggest expanding the research to other platforms. It should be started with technologies that also use the Ethereum Virtual Machine as a technological basis, such as the Ethereum test networks, the fork Ethereum Classic or

other unrelated networks like Avalanche Ecosystem (Team Rocket, 2018). By doing so, many patterns and standards can be reused. In a second step, the research should be expanded to technologies that use different programming languages. Adapting the research methodology and concepts to these platforms is challenging but is needed to lay a basis for cross-chain standardization and tooling.

Chapter 4

Security Implications of Consortium Blockchains: The Case of Ethereum Networks[6]

4.1 Introduction

Blockchain technology has sparked interest in a variety of industries. Even after the initial Bitcoin hype, blockchain as a technology is still regarded to have the potential to drive decentralization and disintermediation. The cryptographic primitives and consensus mechanisms make storing and transferring of data not only secure and resistant against manipulation but also not reliant on a trusted third party (Nakamoto, 2008). Consequently, many consider the potential of this technology immense and disruptive.

Most commercial blockchain applications rely on a private or a consortium blockchain. The purpose of this sort of blockchain is only to allow a select group of participants to read or write data from or to the ledger. Customer-focused solutions, such as the Diem[7] cryptocurrency, use this approach to keep customer transaction data private (Diem Association, 2020). However, depending on the protocol's configuration, blockchain nodes share data with every other node on the network. The distributed nature of blockchains makes them more fail-safe and resistant to manipulation. Attacks such as 50+1 percent attacks and selfish mining, therefore, are well researched. However, with each additional node that joins the network, simultaneously its attack surface for data theft increases. This implies that, even for large networks, only one misconfigured node can leak the whole blockchain data to malicious ac-

[6]This chapter was published in *Journal of Intellectual Property, Information Technology and Electronic Commerce Law* as Hofmann et al. (2021b) and co-authored by Fabian Gwinner, Christian Janiesch and Axel Winkelmann.

[7]Formerly known as *Libra*

tors. In business contexts, information about internal structures can be leaked to competitors. For private use-cases, information about the individual transaction structures can give deep insights into personal behavior and contain the most sensitive information.

To assess the severity of a data breach on one node of the network, we conducted a study to determine how information can be extracted and visualized to gain as many insights into a private blockchain as possible. Thus, our study reverse engineers parts of blockchain networks to gain the necessary information. Reverse engineering a system is typically used to infer how an underlying mechanism works. The difficulty of reverse engineering systems is determined by the number of their components and the interdependence of their components as well as the number of their settings (Lee et al., 2017). For our work, we chose the Ethereum platform as a framework and a popular part of the blockchain universe. Inspired by the Internet Census (Carna Botnet, 2012), our approach relies on data reverse-engineered from a security issue in a faulty configuration of Ethereum. Starting there, we conducted four small case studies on different implementations of the Ethereum platform to identify stakeholders and mechanisms of these networks. Building on this, we want to address the following research questions (RQ) in this study:

RQ1: Which methods and tools are required to reverse engineer Ethereum networks?
RQ2: How much information can be extracted from consortium blockchains with one misconfigured node?

Our paper addresses managers, lawmakers and scientists who are interested in a more technical evaluation of the security of private blockchains. In this paper, we contribute methods used in the process of reverse engineering, as well as the results of the evaluation. Additionally, we provide the insights we gained from the reverse engineering of blockchain networks and the implications they provide for the adoption of the technology. The rest of the paper is structured as follows: In the next section, we lay the foundations by discussing relevant literature and previous work. We then introduce the methodology as well as the data we used for the analysis. The following chapter contains our main research results, by first providing an overview of the technological side of the market and then a detailed analysis of four different blockchains and their use. The final chapter summarizes and concludes the research.

4.2 Foundations and Related Work

In its very basics, the blockchain is a distributed ledger of transactions autonomously managed by a consensus mechanism. Technically, it can be pictured as a growing chain of linked blocks, from where its name originates. The blocks of a blockchain are stored distributed by the participants, the so-called nodes (Nakamoto, 2008; Beck et al., 2017). This distribution also brings the advantage that no single party could manipulate already stored

data and that the storage is resilient against outages of nodes. The blocks of a chain consist of a block header and a list of transactions. In the Ethereum blockchain, each transaction has one sender and one recipient. Today, it is possible to not only store transactions in the blockchain, but also data objects and small programs, which is how (smart) contracts are implemented (Delmolino et al., 2016). In Ethereum, this is often used to realize user-defined tokens. There are many smart contract-based tokens, often standardized by Ethereum Request for Comments (ERC) standards, which define their characteristics and interface.

Given all transactions in a network, naturally, a graph can be built to model the interactions of the participants. The nodes of this graph do not necessarily have to correspond to the nodes of the blockchain network and must not be confused. One physical node of the network could, for example, host multiple Ethereum accounts and therefore represent several nodes in the transaction graph. Additionally, the nodes of the transaction graph can be smart contracts as well. There has been a lot of prior research on the technical analysis of blockchains. This research strongly focuses on large public blockchains, analyzing the transaction structure of public blockchains and the usage patterns therein. First analyses were used to deanonymize Bitcoin users (Reid and Harrigan, 2012). In the early years of blockchain, it was still possible to dissect the whole transaction graph of the first cryptocurrencies (Ron and Shamir, 2013). Due to Bitcoins' transaction structure, it was necessary to apply advanced heuristics to reconstruct and analyze the user graph of the Bitcoin network (Di Francesco Maesa et al., 2018). There have been fewer studies on the public Ethereum networks (Chan and Olmsted, 2017; Anoaica and Levard, 2018). These studies could only link nodes if Ether (the currency of the Ethereum networks) were sent. To consider all transactions, it would be necessary to include the additional network structure that is built by interacting with smart contracts. Studies researching transaction networks of ERC-20 tokens partially deconstructed those structures (Victor and Lüders, 2019; Somin et al., 2018). Interaction networks within smart contracts can be researched in a similar fashion.

The limited existing research regarding the programming interface (JSON-RPC) of a network focuses mostly on the possible attack surface it provides, such as stealing mining reward and denial-of-service attacks (Wang et al., 2018b), or the use of blockchain-based applications (Lee, 2019; Ko et al., 2018). So far, we could not find any studies that use this interface to map transaction networks or reverse engineer the users and use-cases of private blockchains.

In contrast to other security or software engineering related topics, we focus on extracting knowledge for a more research-driven goal. Therefore, our motivation was led by the "Internet Census" of 2012, where the authors used a security vulnerability to create the first full "map" of the internet. Several researchers used this as a foundation, regarding the provided knowledge as well as the used methods, to get insights in other technologies or security-related issues (Heidemann et al., 2008).

4.3 Materials and Methods

To answer our research questions, we used a multiple case study approach. The case study research design consists of the study's *questions*, its *propositions*, *units of analysis*, the *logic linking of the data to the propositions*, and *the criteria for interpreting the finding* (Yin, 2017). We already posed the research questions in the introduction of this paper. As units of analysis, we chose the block headers and transaction data, as well as the network node data for different blockchains. To identify potential blockchains for a more in-depth analysis, we first created an overview of the Ethereum platform landscape.

To do so, we used Shodan, a search engine for Internet-connected devices. We searched the search engine by the query "port:8545" for Ethereum nodes with an active RPC interface[8]. We additionally searched for the string "Ethereum RPC enabled" but considered the results nearly identical. We exported the 3,042 found IP addresses and metadata from Shodan in CSV format. Each IP address represents a node in an Ethereum blockchain network, with an exposed RPC interface. Technically, this gives everyone the possibility to not only extract data from the whole blockchain but also to manipulate the node. It should however be noted that each node in our dataset is for some reason not configured according to the official recommendations, as the RPC interface should never be exposed openly to the internet. Therefore, we only cover blockchains where at least one node was not configured properly.

To build our overview dataset on the operation of nodes, we queried the RPC interface of each of the 3,042 nodes. We extracted the chain version, genesis block (i.e., the first block of a blockchain), and information on whether the node was mining or not. To determine the age of each blockchain, we additionally queried the second block of each chain. We decided not to use the timestamp provided in the genesis block since it often provided a zero value in the timestamp. For nodes that are running on the Ethereum main network, we also queried block number 1,920,000 at which the chain splits into Ethereum and Ethereum Classic. We used this as a mechanism to check how valid our data was and how representative our sample of blockchain nodes was.

Our final overview dataset consists of 2,063 active Ethereum nodes, of which 1421 nodes are used in 621 unique blockchain networks and 622 nodes are connected to the Ethereum main network. The network size of the entire Ethereum main network is at the time estimated at 6,900 nodes according to ethernodes.org (ethernodes, 2021). As a result, our dataset covers about 9% of the Ethereum main network. Additionally, we compared how many nodes of the mainnet[9] are operated in different countries and arrived at a very similar distribution, as shown in Figure 4.1. We did this estimation with other known networks, such as the various Ethereum test networks, which we extracted from an open-source repository for known networks (atlas, 2021). We arrived at similar results, which lets us conclude that our dataset covers the overall landscape of the Ethereum platform comprehensively.

[8] `https://www.shodan.io/report/VwRYVIqq`
[9] Mainnet refers to live blockchain where tokens are in use

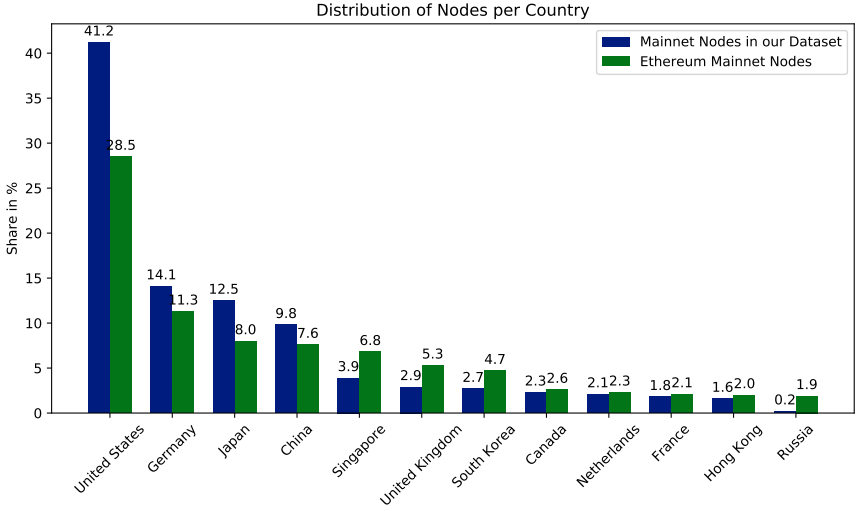


Figure 4.1: The Distribution of the Mainnet Nodes in our Dataset Compared to all Mainnet Nodes

We used the final overview dataset to provide high-level insights into the Ethereum landscape. Additionally, we used this data to identify potential candidates for our case studies. We chose the blockchains according to the number of active nodes, length, and age of the blockchain as well as the distribution of nodes. The goal was to get a diverse set of blockchains to study and draw generalized conclusions. For the chosen blockchains, we extracted account holders for each node and the complete blockchain record of transactions. To identify usage patterns, we used social network analyses on the transaction networks to identify commonly used smart contracts. We extracted and decompiled the smart contracts with the Panoramix decompiler (Eevm, 2020) to find out what their role in the blockchain is. While this is a state-of-the-art approach, the decompilation of Ethereum contracts is still in an experimental stage and does not guarantee success. Therefore, we were not able to decompile and analyze all relevant smart contracts. We summarize the overall data extraction process in Figure 4.2. The mix of source code analysis and social network analysis allowed us to reverse engineer use cases and interaction patterns with the blockchains, and hence provide a suitable way to investigate the proposition.

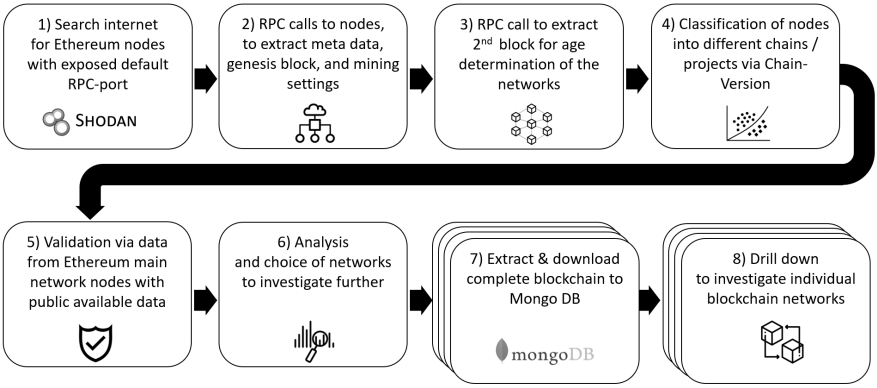


Figure 4.2: Overall Data Collection Process

4.4 An Analysis of Business Blockchains within the Ethereum Landscape

The primary analysis of this paper consists of two parts. First, we describe the overall landscape of the Ethereum protocol using the overview dataset. From there, we can draw the first conclusions, before providing a more in-depth analysis of four case studies for Ethereum-based blockchains.

4.4.1 Mapping out the Ethereum Landscape

To get an overall view of the Ethereum Landscape and map our findings, we analyzed the metadata from the collected dataset. For further analysis, we have chosen different dimensions, which contribute to our overall goal and give us first useful insights in the Ethereum universe to determine the potential case study candidates later.

As a first dimension, we analyzed the hosting of the different nodes. Figure 4.3 (left) shows that almost 75% of all nodes are hosted by major hosting or cloud providers. With over half of all nodes, the big cloud providers Amazon, Digital Ocean, Microsoft, Google, and Alibaba are claiming a large piece of the Ethereum hosting. This shows that the Ethereum technology shows great potential for business adoption since the cloud setup process is a fast solution to get started. It is an advantage over other technologies, which currently rely on specialized mining hardware that is not widely available.

We were surprised by the large share of cloud providers since one of the main advantages of blockchain applications is its distributed topology that affords the technology security and resilience advantages. These advantages are strongly mitigated, when the majority of nodes use the same hosting provider or same data center (Li et al., 2017). To use the full potential of

decentralization, blockchain nodes should be hosted on-premise. We assume to see a smaller share of cloud providers in the dataset, once the technology is more adopted.

As another dimension, we analyzed the country where the nodes are operating. This analysis should give us a picture where most of the Ethereum projects are implemented and may be used as a hint in which country the technology receives most attention. However, since the nodes are mostly cloud-based, this metric can be skewed. Additionally, because nodes of the same chain can operate in different countries, it was not possible to normalize our analysis.

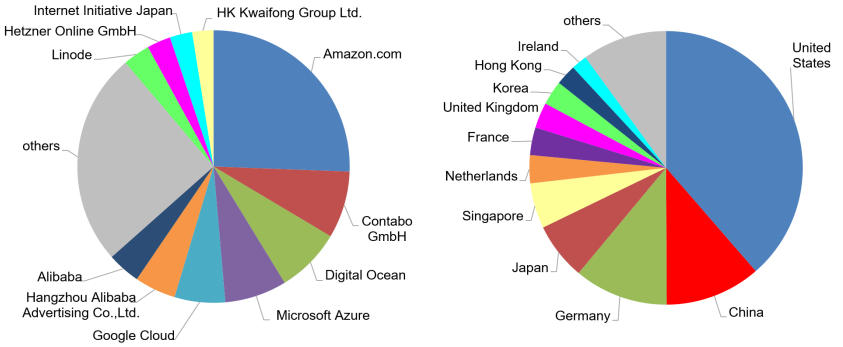


Figure 4.3: Distribution of Nodes per Hoster (left) and per Country (right)

Instead, we have decided to include all nodes in this distribution (Figure 4.3 (right)) to give a weighted analysis of origin. Therefore, blockchains operating with more nodes increase the respective share of a country. With this knowledge, the chart becomes an activity analysis, showing which country is more active and may have advanced further in the process of adopting Ethereum technology. Yet from this point of view, it is not possible to determine if there are more projects or just networks with more nodes that determine the share of a country.

To determine the state of the different chains and thereby to gain knowledge about the phase in which these projects are, we analyzed the length of the different chains. Figure 4.4 (left) shows that there are many very short chains. After analyzing and exploring some random samples of these short chains, it showed that these were purely test setups, either with only some test data, partly with less than ten transactions or even completely empty. Extracting information form these projects does not advance this study, and, therefore, we did not consider them in our analyses further. To achieve better knowledge of potential chains, which we could use for further analysis, we analyzed the age of the different implementations. Figure 4.4 (right) shows the distribution of age, based on the first block. That the initiation of most chains was less than a year ago leads to the conclusion, although the technology is not

new anymore, that either projects implementing it are still in an experimental state or that only projects in an early stage still have misconfigured nodes.

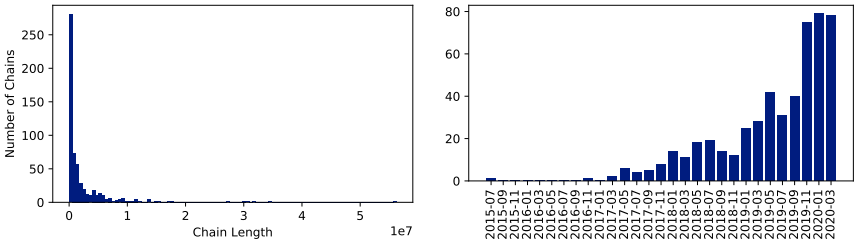


Figure 4.4: Distribution of Blockchain Length (left) and Number of Networks over Time (right)

To consolidate our findings, we put the length of chains in relation to their age, illustrated in Figure 4.5. Newer but longer chains are either configured with a shorter time per block (block time) or represent fast-growing chains. Older but shorter chains were more mature blockchains such as the Ethereum main- and testnets as well as other public Ethereum-based projects. There is a visible forming of "beams" originating from the lower right corner. All networks on the same beam have the same configuration for the block time. There seem to be only a few main variants for this configuration, which could indicate that many of the private Ethereum networks only use a few boilerplate projects as setup. Considering just the distribution and the aggregation of a line in the center, we assume these represent chains with the default configuration. Additionally, increasingly short block times (indicated by a strong negative slope) are introduced in the last years. This could be either due to the need for higher transaction throughput and lower latency or due to the increase in computation power and network speed. A common criticism of the blockchain technology is the high computational overhead and the resulting lack of performance (Kim et al., 2018). Blockchains running at a lower block time are less performance-intensive and are less likely to become out of sync. Additionally, when using the proof-of-work consensus mechanism, shorter block times indicate a lower difficulty, and therefore, a higher risk of double-spending attacks in the network. However, since most private blockchains are not based on this mechanism, we do not research this phenomenon further in this paper.

4.4.2 Detailed Analysis of Consortium Blockchains

As shown in the previous section, most of the networks are either not mature enough to research or are inactive. We identified many blockchains with only one active node and some networks with less than ten transactions over the last two years. For our case studies, we chose four blockchains, that all have more than ten active nodes as well as more than 1 million blocks. Additionally, we excluded the large public blockchains, like the Ethereum

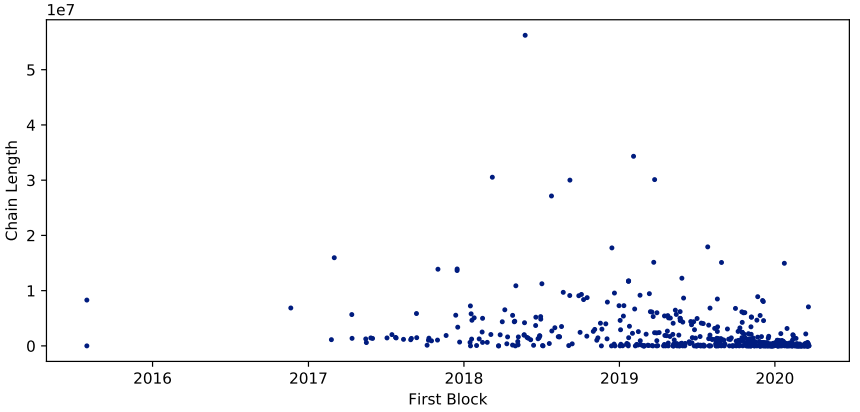


Figure 4.5: Blockchain Length in Relation to Age

mainnet and the various public test networks. Table 4.1 summarizes the networks chosen for analysis.

Table 4.1: Blockchains for Case Studies

Case	Network ID	First Block	Length	Number of Nodes	Number of Transactions
1	10	2019-11-03	1,400,000	16	29,000
2	1337	2019-10-22	7,500,000	20	804
3	2894	2018-11-04	3,200,000	13	2,700,000
4	159	2019-08-18	10,500,000	19	34,000,000

Case Study 1: Network ID 10

We chose the first blockchain we analyzed for its unique properties. It uses the chain version 10, which could indicate that it uses the Quorum variant of Ethereum. Quorum is being developed by JP Morgan Chase as a blockchain, particularly for financial transactions, and offers additional features for this purpose. The Quorum protocol is designed as a permissioned or private blockchain (JP Morgan Chase, 2018). The analysis of the transactions revealed an unusual transaction graph. Only 102 addresses were creating a one-to-one pairing of senders and receivers as displayed in Figure 4.6 (left). More precisely, half of these addresses only sent transactions to a single address, and the other half received transactions from a single address. In all following graphs, accounts are colored blue and smart contracts are colored red. The width of the edges indicates the number of transactions sent from one node to another.

This structure led to the assumption that the receivers are all smart contracts with a single user each. We hence queried the nodes for the contract code of the addresses, downloaded,

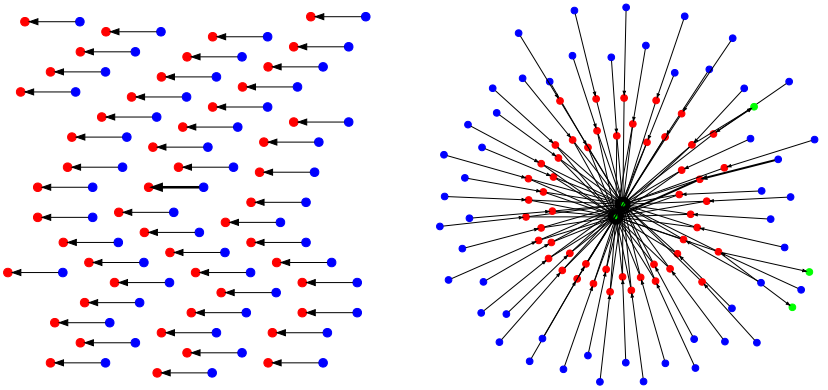


Figure 4.6: Complete Graph without (left) and with Proxy Contracts (right)

and decompiled the code. The contract provided 22 public functions, most of which are used to manage ownership and access to the smart contract. However, the transactions called only one of those functions named *execute*, which takes two parameters as input. The first parameter is an address of the contract, which the call is delegated to. The second parameter are the parameters of that contract call. This means that the smart contracts, we identified initially, are so-called proxy-contracts that are used to call other contracts. We expanded the transaction graph by the contracts that were called by the proxy contracts. We show the resulting full transaction graph in Figure 4.6 (right). The added contracts are colored in green. It can be seen that there are two very central contracts that contain the actual logic, and that every user interacts with. Unfortunately, we were not able to decompile these contracts, and therefore were unable to find out what the purpose of this blockchain network is. However, the overall structure lets us assume that the centralized contracts only accept calls from the proxy contracts and that the proxy contracts are used to manage user access. It should also be noted that the calls to the smart contract are not associated with any cost. Normally deploying or calling a smart contract would cost the user gas[10], which is paid for in Ether. However, the accounts all have a balance of zero Ether and there are no transaction fees in this network. This, along with the fact that the central smart contracts were too complex to decompile, could imply that the developers test a novel use-case that exceeds the current computational limits of standard Ethereum configurations.

From a social network perspective, the graph seems very decentralized. Since each user interacts with only one proxy contract, which in turn interacts with at most two other contracts, the out-degree centrality of the nodes is equally distributed between the users. It

[10]Gas measures the amount of work of miners to include transactions in a block

should be noted that one user sent 87.6% of all transactions. Additionally, we examined how many blocks were mined by each individual miner. With 85.4% of all blocks, we do not consider this a secure network, since this miner has over 50% of mining power (Nakamoto, 2008). With this much power for one node, it should be reevaluated if a centralized solution could be a better alternative (Wüst and Gervais, 2018). However, if the network is indeed only a test setup, the security implications are not as important.

Case Study 2: Network ID 1337

The second blockchain we identified exhibits a different kind of centralization. While the nodes are distributed all over the world, they are all hosted in the Microsoft Azure cloud. This centralization to a single provider gives a single entity immense power over the network, since it could completely shut down all nodes or simply block access to the nodes on short notice (De Filippi and McCarthy, 2012).

Furthermore, we noticed that many contracts deployed on the blockchain use smart contracts developed by Ambisafe (ambisafe, 2021). Ambisafe offers a blockchain quickstart platform that lets users easily build a blockchain by using preconfigured modules. We identified an EToken2 contract, which offers advanced token functionality but is compatible with the ERC20 interface. Additionally, we identified contracts for identity management (ERC725) and claim management (ERC735). Again, we found proxy smart contracts, but in this case, they were not for access management, but they made contracts upgradeable.

The overall network structure looks distributed, as shown in Figure 4.7 (left). There is one centralized node that interacts with a lot of smart contracts. Approximately a third of these contracts are EToken2 contracts. Each of these contracts corresponds to a contract deployed by the same address that allows transfers of EToken2 to ICAP addresses. These are addresses that are compatible with the IBAN bank account numbers. Another very central node is the smart contract in the upper cluster. This smart contract is a claim management contract. While this looks like the architecture of a decentralized exchange, there is little to no interaction of different accounts with each other, either direct or via smart contracts. Figure 4.7 (right) shows the transaction graph with a dot layout (Ellson et al., 2001), which indicates that the transactions all flow in only one direction. In addition to this unidirectional transaction flow, the root node holds an overwhelming majority of Ether with approximately 1032 Ether. In comparison, the second largest account holds 18.7 Ether, while most accounts hold less than one.

We conclude that this is an experimental setup that is used for testing or demonstration purposes only, or possibly a network that is currently being built and the funds are being distributed to the nodes according to their needs.

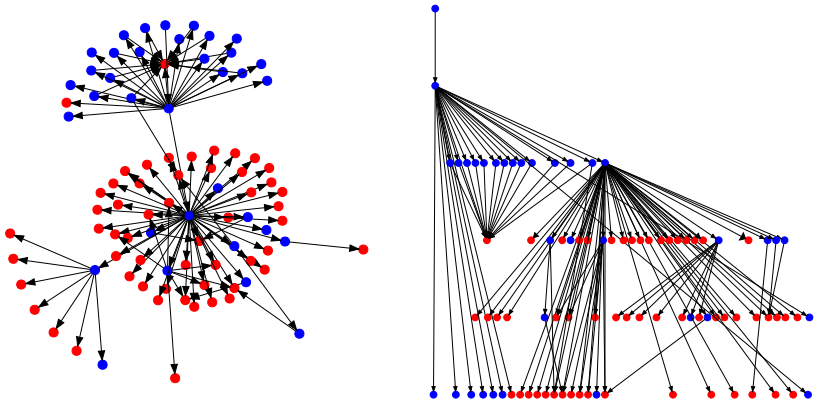


Figure 4.7: Transaction Graph in Neato Layout (left) and Dot Layout (right)

Case Study 3: Network ID 2894

The first insight of our analysis was that there are no smart contracts deployed in this network. This means that the transactions transfer Ether. In fact, the transactions in the network carry on average 2,176.3 Ether. The overall transaction graph is much larger than the previous blockchain. The network consists of 15,489 addresses. This size makes it too complex to display completely. Therefore, we chose the representation of the graph as an approximation in Figure 4.8 (left) by only displaying edges where there were more than 1,000 sent transactions with the corresponding nodes. The second representation we chose was a transaction graph that only displays those transactions that have data attached in addition to the transaction value, as shown in Figure 4.8 (right). We could not identify what this data represents since the data seemed to be in the form of arbitrary numbers not correlated with the transaction value. However, there were three different types of numbers: small numbers between 1 and 256, medium numbers around 106, and extremely large numbers in the order of magnitude 1056.

Even though the number of nodes is much larger than other networks, the graph is much more centralized. Figure 4.9 (left) shows the indegree and outdegree centrality for each node. Note that we had to use a logarithmic scale due to the massive differences in centrality. These differences could be as a result of an initial token distribution process. Additionally, the distribution of mining power is not distributed equally either. Figure 4.9 (right) shows that two miners mined a disproportionally large share of the blocks. While this might not be an immediate problem, if those two miners cooperate, they could overrule the rest of the network. Finally, the distribution of Ether is unequal among the nodes, but it is not nearly as unequal as seen in the previous case study. A large portion of the nodes have one to 108

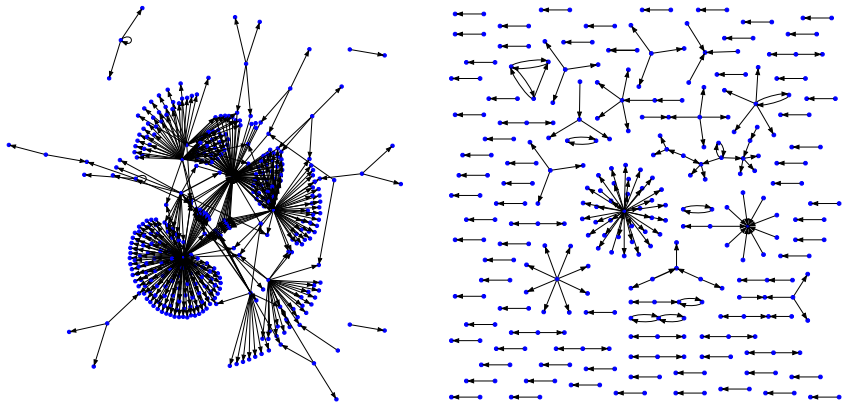


Figure 4.8: Transaction Graph with Nodes with more than 1,000 Transaction (left) and with Attached Data (right)

Ether, but the majority have less than one. The centralized transaction network and mining, as well as the unequal distribution of Ether, are phenomena that can be seen in large public blockchains, in particular because larger networks tend to centralize. This network, despite its use as a pure accounting network, is the most used network in our dataset.

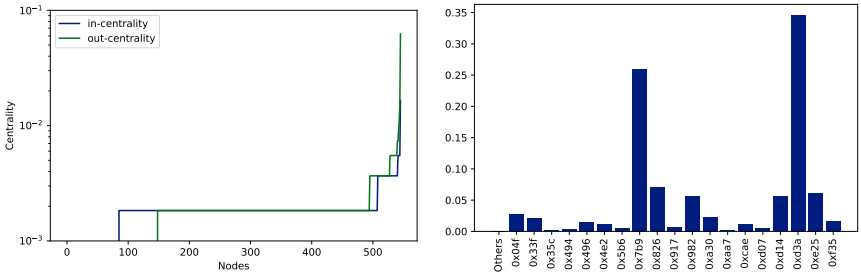


Figure 4.9: Centrality Scores per Node (left) and Share of Mined Blocks per Miner (right)

Case Study 4: Network ID 159

Our last case study concerns a network that has a massive number of transactions. Since it was launched, the network has about 20% of the public Ethereum mainnet transactions. The Ethereum mainnet is used by thousands of users. However, we noticed a very centralized contract in the network, as shown in Figure 4.10 (left). We identified it as a TomoChain BlockSigner smart contract (TomoChain R&D Team, 2018), which is used as an alternative

consensus mechanism. In fact, all smart contracts we identified are used for this mechanism, and the transactions therein are not relevant to the actual transaction network structure. Therefore, we also analyzed the network structure of the remaining network separately as shown in Figure 4.10 (right). The resulting graph only considers 895 transactions.

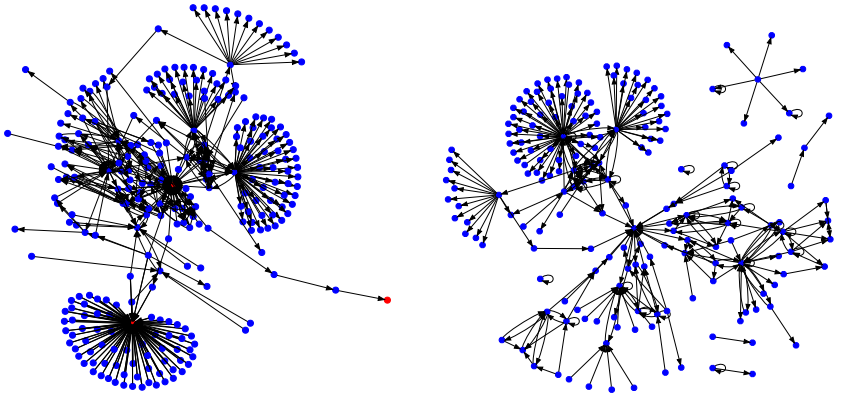


Figure 4.10: Transaction Structure with (left) and without Smart Contracts (right)

This transaction graph is not fully connected. There are some small islands with unidirectional transactions. The main island consists of a few larger clusters of outgoing transactions. Again, this could indicate an initial token distribution process. Since this network is not as old as the previous network we analyzed, it could show much more activity in the future and build a similar transaction graph. Since a smart contract handles the block generation process, we could not easily identify the miners of the blocks, and hence could not analyze the distribution of mining power.

Upon further investigation through the IP addresses of the nodes, we found out that the network is connected to the Caelum Project, which is not accessible anymore. It is described as a decentralized storage solution, to secure digital crypto assets[11] with inheritance functionalities (Caleum, 2021).

4.5 Conclusion

Past research on blockchain security has focused mainly on the prevention of fraudulent transactions. However, with the rise of private and consortium blockchains, data privacy has become another important topic, lacking extensive research. Against this backdrop, in this

[11] Crypto assets are "a new type of asset recorded in digital form and enabled by the use of cryptography that is not and does not represent a financial claim on, or a liability of, any identifiable entity" (Ecb, 2021)

paper, we analyzed the exploitation potential of misconfigured private blockchains. Our approach consisted of reverse engineering actual implementations of the Ethereum platform for individual use-cases to analyze the transaction structure and smart contract implementations, to gain insights into the usage patterns and stakeholders of the networks.

In our first research question, we asked, which methods and tools are required to reverse engineer Ethereum networks. Our approach consisted of using a port-scanning dataset and enriching it with additional data that the listed nodes provided. Using social network analyses and source code analyses, we additionally conducted small case studies on selected networks. The social network analysis proved to give useful insights into the actual usage of the network but fell short of revealing the whole structure without the source code analysis of the smart contracts. The smart contract analysis was a very successful approach for some networks, while for others, we could not retrieve the source code of the smart contracts by decompiling them. The main improvement we would suggest for future research would be a "magical" decompiler that can retrieve the original commented source code from Ethereum bytecode. Additionally, it should be checked whether some of the analyses can be automated, to give a quick overview of all networks fast and not rely on analyzing them step by step.

Our second research question was how much information can be extracted with only one misconfigured node. We could identify that our approach is not able to paint the full picture of the networks but can give valuable insights. For some networks, we could link IP addresses and specific smart contract structures with publicly available data to get insights of stakeholders. For other networks, we had to rely on the transaction structure and could only identify entities by their cryptographic addresses. Especially for Ethereum networks, each node holds a full copy of the ledger. Therefore, all analyses were based on a maximum of available data. In further research, other structures such as the Hyperledger project should be examined, where the network is segmented into channels. Here, attacking only one node should only provide partial information about the network and would hence call for more elaborated analysis techniques.

Due to the availability of data, our research focused on organizational entities rather than individuals. However, the results indicate that for our analysis of the data from an analytical point of view, it does not matter whether the data is of organizational or personal nature. Network structures and agreements can be derived or inferred be it the one or the other. Therefore, we think that the results can be transferred to blockchain networks comprising end users sharing personal data. Thus, our study also raises the very relevant question as to whether (private) blockchain networks can reach a consensus without sharing all data between nodes and what data distribution strategies would defend best against weak links in the chain that exposes private information of individuals.

Our dataset consists of over 621 unique blockchain networks, of which we were only able to analyze four for more detailed insights. The process of retrieving and analyzing the entire blockchain for many networks is extremely time consuming, but we are sure that analyzing a larger portion of it would give even better insights into information extraction processes.

Overall, improving the systems and tools needed for the reverse engineering as well as a full analysis for the network information, can therefore be future work.

The research provided us with an exciting puzzle that is still not assembled completely. We, therefore, hope that the approach is adopted for other blockchain technologies such as Hyperledger or even other unrelated technologies to improve current tools.

Chapter 5

Building Scalable Blockchain Applications – A Decision Process[12]

5.1 The Need for Scalable Blockchain Applications

Blockchain technology is seen as one of the leading drivers of digitalization and decentralization. One of the fundamental limitations of blockchains for various use-cases is scalability. Cryptocurrencies, the most well-known application for blockchains, suffer from several scalability issues that are often criticized. A popular example is the low transaction rate of 7 transactions per second for Bitcoin compared to centralized solutions like VISA™reaching up 47,000 transactions per second (Trillo, 2019). Another point for criticism is the high transaction latency of up to 41 hours that occurred at especially high network loads in January 2018 (bitinfocharts.com, 2019). Even though some experts state that technical issues such as scalability and security are not an issue for the adaption and diffusion of blockchain technology (Post et al., 2018), we argue that for some possible applications, scalability is an essential factor. For example, blockchain is often seen as a device for trustless machine-to-machine transactions. CISCO™predicts that by 2020 about 50 billion devices will be connected to the internet (Evans, 2011) and even if only a fraction of these devices communicated via a blockchain, the network load would be orders of magnitude higher than with current cryptocurrencies. While some researchers assume that blockchain provides a scalable peer-to-peer communication protocol, this is not yet the case (Kouicem et al., 2018). In contrast to peer-to-peer transactions in a trusted environment, the process of verifying transactions on a trustless distributed ledger is inherently inefficient. With the introduction of programmable smart contracts and distributed applications (DApps) that are executed and verified in the

[12]This chapter was published in *Lecture Notes in Computer Science* as Hofmann (2020). This chapter is single-authored. The authors use of "we" is for consistency.

same manner, the problem of scalability became even more complex (Worley and Skjellum, 2018; Chauhan et al., 2018).

There have been many contributions that developed solutions to those problems. However, it is hard for developers to find suitable solutions for their applications. Therefore, the goal of this paper is to provide a standardized process for developing scalable blockchain applications that developers can use as guidance. To do so, we propose the following research questions:

RQ1: Which technologies exist that solve scalability issues in blockchain applications?
RQ2: What decisions have to be made to include these solutions in blockchain applications?

To answer these questions, we analyze the scalability problems that blockchains bring, accumulate existing solutions and develop a decision process that helps build scalable blockchain applications.

The following section gives an overview of the state-of-the-art in scalability research, followed by the research process of this article. In Section 5.4 we provide an overview of existing solutions and introduce the decision process as the core of this article. The resulting artifact consists of a four-step process that guides developers while developing their applications. Section 5.5 explains our evaluation process, followed by the final section, concluding this article by discussing the results.

5.2 Foundations and Related Work

Distributed ledger technology has a variety of scalability issues. The literature can be classified into five categories: transaction throughput, transaction latency, the number of nodes, storage, and computational complexity.

The majority of research focused on the *transaction rate*. This comes from the fact that the most popular blockchain applications are cryptocurrencies (especially Bitcoin), whose transaction rate is often compared to the transaction rate of centralized transaction systems (Lacity, 2018).

In addition to the overall throughput, the *latency of the transactions* was criticized by a majority of researchers. On the one hand, this problem was linked to the low throughput by stating that a long transaction queue will delay transactions (Dinh et al., 2017). On the other hand, transactions are not considered secure until a few additional blocks have been mined. Therefore blockchains like Bitcoin with block times of approximately ten minutes do not enable instant transactions (Dinh et al., 2017).

Four authors acknowledged that in contrast to classical peer-to-peer systems, some blockchains do not scale well with an increasing number of nodes. This is because the scaling properties with the number of nodes in the network heavily depend on the consensus mechanism implemented. This is further discussed in Section 5.4.1. Independent from consensus

mechanisms, the additional nodes are likely to contribute to a higher transaction count. This makes some blockchains without special scaling solutions unsuitable for networks with many nodes, such as Internet-of-Things (IoT) applications.

For *storage and I/O* one common remark was that the whole blockchain has to be stored on every node. This makes it hard to use, especially for cheap devices with limited storage, like IoT devices. Another aspect is how storage is accessed in smart contracts. Blockchains like Ethereum or Hyperledger only support key-value-based storage. This makes it difficult to efficiently store unstructured data such as images or database-like structures that can easily be queried. Additionally, large data sets cannot be stored in single blocks due to block size limitations. While it has been shown that arbitrary data can be stored and accessed in blockchains, it is often unfeasible to do so (Sleiman et al., 2015).

Only four authors directly acknowledged the *computational complexity* of the decentralized calculation of smart contracts. The problem of computational complexity and the cost of it is two orders of magnitude higher for some applications than in centralized systems (Rimba et al., 2018). The global limitations of executing smart contracts and the limits of single executions play a role when developing applications. Since smart contracts are executed on every mining node, the number of operations has to be limited (Dinh et al., 2017). This means that some applications are impractical and even impossible to implement on specific blockchain architectures.

5.3 Research Design

To construct the decision framework, we use a design science research process and follow the guidelines for design science research proposed by Hevner et al. (2004). It is stated that IS research has to cater to business needs defined by the *environment* to assure the research's relevance. We do not consider a specific business case but the general need for the scalability of blockchain applications. Additionally, existing foundations from a broad *knowledge base* are to be applied to the research to achieve rigor. With these side conditions, the actual *research* can be conducted through *building* and *evaluating* an artifact.

5.3.1 Ensuring Rigor

A literature review is conducted to get a better understanding of the current landscape of the most relevant scalability issues in IS research (Vom Brocke et al., 2009). We conducted a literature review on articles from IS journals and conferences as well as literature from the fields of computer science and engineering. This allowed access to novel ideas and solutions, which is crucial in a fast-evolving field like blockchain research. However, the literature was still focused on scholarly research and excluded white papers, blog entries, or news stories, even though we acknowledge that they have been influential in the blockchain industry as well as in academia. The searched databases were: AIS Library, EBSCOHost, SpringerLink,

and Google Scholar. We limited the Google Scholar search to the first 200 articles since there were over 14,000 results overall.

The search terms depended on the scalability issue they were supposed to solve. The search term was *"blockchain AND scalability AND ((transaction AND rate) OR (transaction AND throughput) OR latency OR bandwidth OR storage OR computation)"*. We filtered the papers by outlet, title, and abstract. Then, we conducted a forward and backward search resulting in a set of 90 papers. The remaining literature was read and papers sorted for a final set of 48 papers.

5.3.2 Development and Evaluation

We have a solid base to develop an artifact with the overview of existing solutions. In this step, the decision process is built on the foundation of the available solutions and evaluated with respect to the utility provided in solving scalability issues as well as the usability of the process. We examined all available solutions and categorized them in a first step. Since some solutions work similarly and have the same advantages and disadvantages, we grouped them to avoid redundancies. We then examined further similarities, differences, dependencies, and contradictions between the solutions to build the decision process. The development of the final process included multiple adjustments due to the dependencies and contradictions of some solutions.

The decision process falls in the category of *methods* since it *guides how to solve problems* (Hevner et al., 2004). The evaluation of the artifact is carried out in a *descriptive* manner by testing it against multiple distinct scenarios to demonstrate its utility.

5.4 Building Scalable Blockchain Applications

This section presents the designed artifact in the form of a decision process that recommends suitable scalability solutions. The first part gives a short overview of the literature and the available solutions. In the second part, the final process is described.

5.4.1 Available Solutions

The solutions presented in the existing literature can be split into two main categories, depending on which aspect of the blockchain they target.

The first type of solution targets the *blockchain layer*, the second type the *application layer* (Xu et al., 2016a). In mainstream blockchain literature, solutions targeting the blockchain layer are referred to as layer 1 solutions, while solutions targeting the application layer are called layer 2 solutions (Buterin, 2018). For the sake of brevity, we adhere to this naming convention. This distinction between the layers is important for building blockchain applications since changing the blockchain infrastructure to implement layer 1 solutions is not

trivial and requires a consensus among the network participants. Layer 2 solutions mostly provide scalability by interacting with the blockchain as little as possible and only using it as a final source of truth or as a settlement layer. Finally, all articles were grouped by the way the presented solutions work. We grouped the solutions further into six categories. Layer 1 solutions include the *consensus mechanism*, *architecture*, *sharding* and *parameters*, while layer 2 solutions can be categorized as *off-chain protocols* and *decentralized storage solutions*.

The effect of different *consensus mechanisms*, especially on transaction throughput and latency, has been discussed in 16 of the analyzed articles. Mostly Proof-of-Work (PoW), Proof-of-Stake (PoS), practical byzantine fault tolerance (PBFT), and Proof of Elapsed Time (PoET) were evaluated as mechanisms for the decision process. While we acknowledge a wider variety of consensus mechanisms, we focused on the most widespread to keep the decision process manageable.

This category *architecture* fits all solutions that do not use traditional blockchains to store the distributed ledger. Among those are articles proposing data structures other than a blockchain such as directed acyclic graphs (DAG), e.g., a hashgraph or tangle (El Ioini and Pahl, 2018). Since these are not blockchains and have often had entirely different properties incompatible with other solutions we describe, they are excluded from the decision process. Other solutions, such as sidechains and multichain architectures, are too complex to fit into the scope of this paper but should be evaluated in future research.

Sharding is a process first introduced to distribute databases. The idea is to split the data into smaller data sets (shards) and spread them among multiple servers. For blockchains, the principle is similar. The blockchain network is divided into smaller communities that validate transactions with classical consensus mechanisms (Feng et al., 2018).

Adjusting the *blockchain parameters* is the most straightforward solution to increase the transaction rate or improve latency. However, it is only mentioned in six articles. It is stated that latency, as well as transaction throughput, can be improved by decreasing the average time that is needed to confirm a block (Croman et al., 2016). Additionally, the increased block size can improve the transaction throughput and the latency that comes from waiting queues at high network loads. This solution can also enable smart contracts that rely on storing many variables.

The most discussed scalability solution is moving transactions to secure *off-chain* communication channels. Payment channels describe a class of techniques that enable users to conduct multiple transactions without committing single transactions to the blockchain. In the case of purely bidirectional transactions, state channels constitute bilateral agreements between two parties. Against this backdrop, multiple users can build networks, which allow unconnected users to conduct transactions by routing them over intermediaries (McCorry et al., 2016a). A similar technique to offloading transactions from the blockchain, complex computations can also be done with minimal interaction with the blockchain itself. While in the current literature presented protocols do secure off-chain computation (Molina-Jimenez et al., 2019), they still impose some limitations. Newer solutions support efficient on-chain

verification of off-chain computation results, solving some shortcomings. A downside of this procedure is that every party that relies on the computation to be correct has to set up the verification contract. However, other less efficient solutions do not rely on a trusted setup process (Galal and Youssef, 2018).

The means of *data storage* for blockchains is often limited to storing key-value pairs. While this is a good solution for a wide range of applications, it may be useful to have the availability of mass storage for media files or to have a database structure that can be queried. Against this backdrop, several solutions have been introduced like the interplanetary file system (IPFS) (Cucurull et al., 2018) for decentralized, trustless storage or decentralized cloud solutions based on smart contracts (Xue et al., 2018). The choice of database solutions ranges to the scale of Hadoop-based Big Data databases (Sahoo and Baruah, 2018).

5.4.2 Decision Process

Before the decision process is started, it should be decided whether a blockchain is needed to solve the problem at hand. The requirements to decide whether a decentralized, trustless ledger is suitable for an application were already researched (Wüst and Gervais, 2018). Additionally, there is existing research on whether a different architecture such as hashgraphs or tangles is suitable for an application (Koens and Poll, 2018).

We derived a four-step process with an additional evaluation step to choose the right solutions for the desired applications.

In the first step, it should be decided whether to implement an own blockchain or build the application on an existing one. While it is much easier to use an existing blockchain that is stable and trusted, it must be considered that all applications running on this blockchain compete for the same limited resources of the blockchain. The second step is choosing the proper consensus mechanism for the blockchain. Then the parametrization and layer 1 solutions for the blockchain have to be defined. If an existing blockchain is chosen, it should fit the desired parameters and layer 1 solutions as close as possible. In the fourth step, layer 2 solutions will be considered and chosen according to the scalability needs. Additionally, we recommend an evaluation as a final step. Here it should be checked whether the chosen solutions are compatible with each other and, if needed, adjust the decisions. Figure 5.1 provides an overview of the complete process.

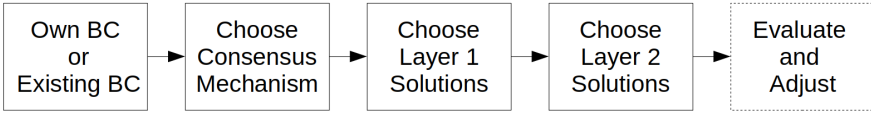


Figure 5.1: Overview of the Decision Process

Choosing a Consensus Mechanism

The choice of the consensus mechanism is independent of the choice of other layer 1 solutions, and the choice of the consensus mechanism alone can result in a scalable solution for some use-cases. The decision tree to follow in order to choose the right mechanism is depicted in Figure 5.2.

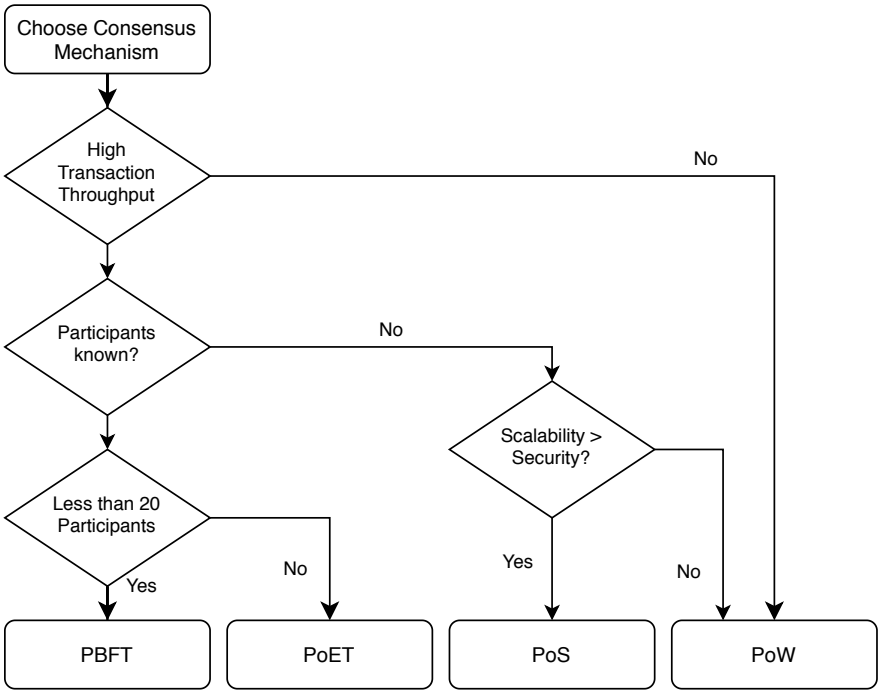


Figure 5.2: Flowchart for Choosing a Suitable Consensus Mechanism

If the transaction throughput of the application is limited, the Proof-of-Work consensus is the most tested and stable consensus mechanism available. Typical PoW blockchains can handle around 25 transactions per second. If the application strongly exceeds this limit, another mechanism should be chosen.

If the participants of the blockchains (i.e., users of the application) are known, a permissioned blockchain can be utilized. Permissioned blockchains can utilize more efficient consensus mechanisms since they are resistant to Sybil attacks. If no permissioned blockchain can be utilized, the PoS consensus mechanism should be considered if the scalability is worth a security trade-off. If not, the PoW consensus should be chosen. Scalability can still be achieved with other solutions.

If a permissioned blockchain is suitable, it should be checked if the number of nodes is not too high for the PBFT. Studies showed that the protocol does not work efficiently in practical applications if the number of nodes exceeds 20. If it does, the PoET consensus is recommended.

Choosing Layer 1 Solutions

After the consensus mechanism was considered, the parameters block-time and block-size must be assigned and if sharding or other architectural decisions are suitable and needed. In this step, multiple or even all solutions can be chosen.

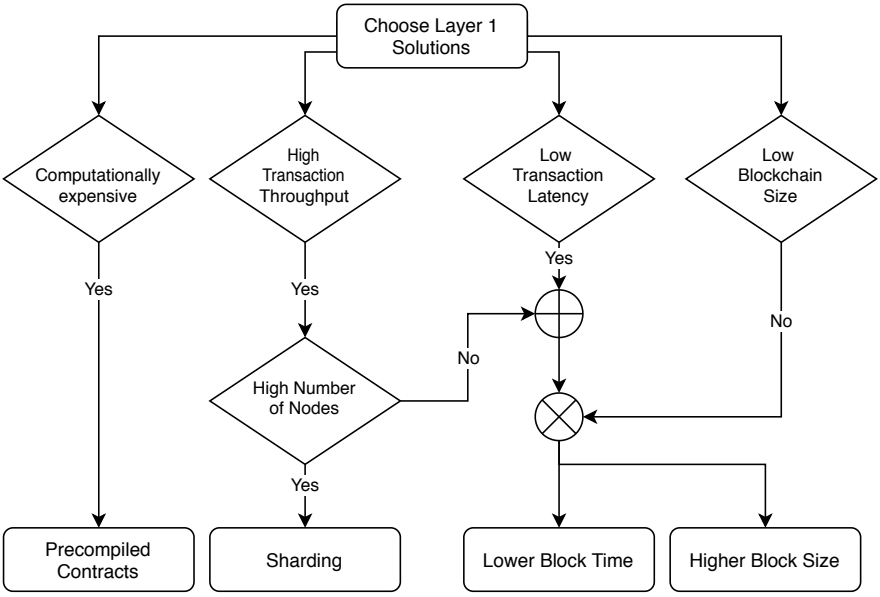


Figure 5.3: Flowchart for Choosing Suitable Layer 1 Solutions

Often-used functions of smart contracts or those that are expensive to execute, such as cryptographic functions, should be precompiled into machine language so that they do not have to be executed with blockchain bytecode. Precompiling elliptic curve pairings is a prerequisite for using efficient zkSNARK verification on blockchains.

If a high transaction throughput is necessary, it should be checked whether the number of nodes is big enough to allow sharding. The transaction throughput scales linearly with sharding nodes, but too many nodes make the partial networks too small and, in consequence, not secure (Cai et al., 2018).

Therefore if the number of nodes does not allow this solution *or* a low latency is required, the block time should be minimized and the block size maximized. However, this is only possible if the overall blockchain size is not of concern.

Choosing Layer 2 solutions

The last step is to choose layer 2 solutions.

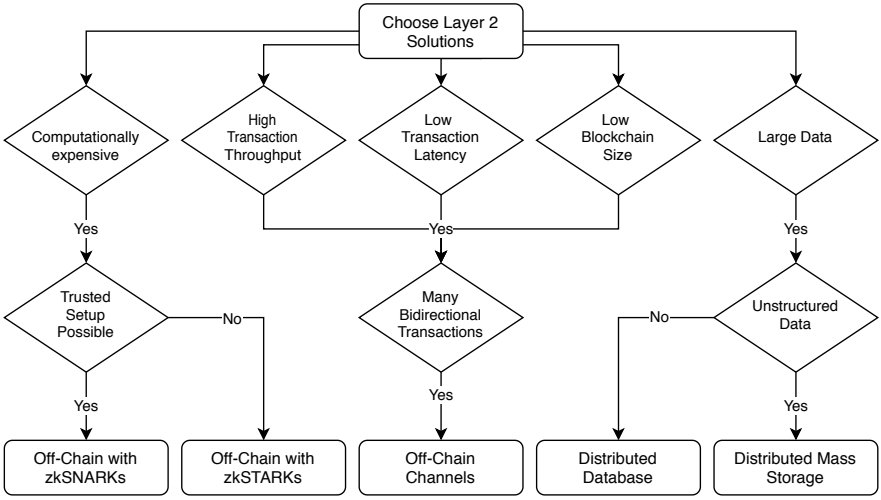


Figure 5.4: Flowchart for Choosing Suitable Layer 2 Solutions

If the application is computationally expensive, off-chain computation with on-chain verification should be considered. Off-Chain computations that can be proven to be correct on-chain are only useful if the verification of the proof is cheaper than the actual computation (Galal and Youssef, 2018). Verifying a zkSNARK on the Ethereum blockchain costs around 1.8 million gas (Eberhardt and Tai, 2018). Every function that requires less gas should not be considered for an off-chain computation. Additionally, the setup of the verification contract is not trivial and should include all parties that need to trust the verification process. If such a trusted setup is not possible, zkSTARKs should be considered for on-chain verification. However, verification is computationally orders of magnitude more complex and therefore not the preferred option (Ben-Sasson et al., 2019).

If either a high transaction throughput, low latency, or a small blockchain size are required, it is recommended to offload most of the blockchain transactions into payment channels. Once a payment channel has been created, it offers instant payments and high throughput. Since the transaction to open and close the channel underlies the same limitations as regular blockchain transactions, these channels should only be used if it is likely that the par-

ties involved will perform multiple bidirectional transactions. It should also be noted that, for security reasons, transactions with a high transaction value or with important data should be handled directly on the blockchain. The extension to a network of channels is only useful if the network is dense enough to provide a path between the parties involved.

If the application relies on data that is too big to fit into single blockchain blocks, the data must be stored differently. Here we have to differentiate between structured data that can be stored in a database and unstructured data, such as image or movie files, that requires mass storage. The usage of distributed databases is recommended when the underlying blockchain architecture is insufficient for storing or querying the needed data sets. We do not differentiate between SQL-like databases or databases capable of processing big data since the differences are well researched and discussed in other fields (Trillo, 2019). For unstructured data, solutions such as IPFS ensure data integrity as well as data availability among all participants.

5.5 Evaluation

To evaluate the framework designed here, we utilized a scenario-based evaluation (Peffers et al., 2012). We evaluated the process for its ease of use, reproducibility of the results, and if it serves its purpose. We, therefore, supplied three fellow researchers with the decision process and the use-cases and let them choose the appropriate scalability solutions. They all chose the same scalability solutions and claimed that the usage was straightforward. We implemented the use-cases and tested them against implementations without scalability solutions to test whether the process served its purpose. We hence tested it by implementing three use-cases, each with different challenges for scalability. The first scenario was a *distributed voting process* that needs to handle a large burst of transactions in a short voting period. The second use-case was *tracing of goods in a supply network*. Here a few parties track a large number of items. The final use-case was a *blockchain-based chess game*. There have been several attempts to implement such a game without a trusted third party, most of which failed because it is too computationally expensive to check whether moves are legal.

This example showed that choosing the scalability options needed to implement the application was straightforward. While the implementation itself was complex, the decision process in each scenario was simple and led to the desired results.

5.6 Conclusion and Future Work

Blockchain applications widely suffer from scalability issues that prevent widespread popularity. While many solutions to those problems exist, there was no structured process to decide how and when to apply them. We managed to extract those solutions for scalability limitations by reviewing state-of-the-art literature from IS, computer science, and engineering, structured them, and provided an overview, which answered our first research

question. By building on this knowledge, we answered our second research question and constructed a decision process that helps developers, businesses, and researchers to build scalable blockchain applications. The provided process is short enough to be executed before each development project and comprehensive enough to give developers an idea on which solutions to focus. The provided process is not without limitations. There is still a wide variety of scalability solutions that had to be excluded from this work for complexity reasons. The process can be extended to give a more granular decision process in future research.

Additionally, the process can be evaluated in more practical scenarios. An evaluation and extension with domain experts are also possible. Another important consideration for every type of application is the underlying technology's maturity and ease of use. Many of the technologies presented here are new and not sufficiently tested in productive use. The quality of the documentation and the existence of a community that can help develop the technology are also essential. This is especially the case if the developers of an application lack the resources needed to fix or improve immature software in order to use it in their projects. These issues became very clear when implementing some of the evaluation scenarios. We encourage that the maturity of the presented technologies should be researched. The methodology used in this paper is also suitable to develop decision processes for objectives other than scalability, such as privacy. Overall this paper opens up new ways for further research on decision processes and the development of blockchain applications.

Chapter 6

Tracing Back the Value Stream with Colored Coins[13]

6.1 Introduction

In recent decades, supply chains and manufacturing processes have become increasingly complex and distributed (Andersen et al., 2019). To organize them as efficiently as possible, companies have increasingly implemented lean methods and enterprise resource planning (ERP) systems to achieve competitive advantages (Powell, 2013; Mendling et al., 2018). In the future, new technologies such as machine learning, process mining, and blockchain will be implemented in the IT landscapes and influence the human factor in business processes (Mendling et al., 2018; Knoll et al., 2019). In addition to the increasing use of technologies, customers and companies are getting more aware of sustainability. They are increasingly demanding information about the production, logistical processes, and sources of products. Due to its decentralized nature, blockchain technology can track and trace products even through complex supply chains (Wang et al., 2019). We see the terms "tracking" and "tracing" address two different target groups in this context. Where customers would like to see end-to-end traceability and sustainability, companies use this term to pursue traceability to optimize their supply chains (Mondal et al., 2019; Tian, 2016). In current solutions, the digital twins of products are managed by smart contracts piloted by service providers like IBM, Cisco, and SAP (Wang et al., 2019). However, we noticed that these smart contracts become increasingly complex if the supply chain includes production steps that combine goods to create a new, different good. Apart from smart contracts, we identified an often overlooked concept to represent real-world assets in the blockchain called "colored coins" (Anand et al.,

[13]This chapter was published in *ICIS 2020 Proceedings* as Pytel et al. (2020) and co-authored by Norman Pytel and Axel Winkelmann.

2016). Therefore, our motivation is to reduce the complexity of the technology to make it understandable and practical for future researchers and practitioners. This could facilitate the adoption and accelerate the diffusion of blockchain technology in supply chain use-cases (Agarwal and Prasad, 1997).

Finally, we found that the current literature lacks deeper insights on how exactly the movement of goods and production processes are mapped from current ERP Systems to supply blockchains. We, therefore, answer the following research questions:

RQ1: How can lean methods, ERP, blockchain and the human factor be connected in future projects?
RQ2: How can colored coins be used for traceability for logistic and production processes in an end-to-end scenario?

To answer these research questions, we will use a lean method in our paper to achieve traceability for customers in a blockchain and utilize state-of-the-art techniques from value stream mapping (VSM). VSM is a Lean Management technique that provides, in a modified form, a structured way of analyzing data points in information systems. On the management side, it visualizes all relevant information that should flow along a supply chain. Therefore, we use state-of-the-art information systems and modified colored coins on the technical side.

In the following section, we give an overview of the work done so far in this area. We then present the traceability solutions, conceptual architecture, and methodology to map the data to the blockchain. After that, we demonstrate the solution and show how companies could implement it in their environment. Finally, we briefly evaluate the methods used and show ways for future improvements. The last section summarizes the results and concludes this paper.

6.2 Foundations and Related Work

A challenge for the industry is to get real-time data along a supply chain (Heng, 2014). For manufacturers, Buer et al. (2018) summarizes numerous studies from the past that use RFID technology to map real-time value streams. The authors notice that there is still a lack of knowledge of implementing them in a lean manufacturing environment. Past research also deals with how digitization and lean principles can be combined. According to Lorenz et al. (2019) lean can be used to support digitization. We also note that lean methods are increasingly considering sustainability aspects (Brown et al., 2014; Faulkner and Badurdeen, 2014; Garza-Reyes et al., 2018). Nevertheless, Buer et al. (2018) notice it is still unclear which practices and technologies can be combined.

VSM is a fundamental tool to record the material and information flow of products from customers to suppliers (Rother and Shook, 2003). It thus offers a good visualization for end-to-end production and logistic processes. Various industries have already adopted the

method to detect waste and improve their processes (Shou et al., 2017). In an empirical assessment of German companies, it was pointed out that traditional paper-based methods provide only a "snapshot" of the production process. Therefore, it should be implemented with interfaces into information systems (e.g., ERP/MES) (Lugert et al., 2018). To harmonize processes and data from information systems, recent research offers different approaches from the hospital or industrial environment to record the movement of materials and information in a modern industrial environment (Heger et al., 2020; Hartmann et al., 2018). The presented extended VSM notation from Hartmann et al. (2018) has already been successfully tested by Meudt et al. (2017) in eight industrial enterprises, so that we have adapted the notation for our food manufacturing process.

An efficient control system for visualizing material and information flow is fundamentally needed in production and logistics to monitor the origin and quality of end-to-end processes in supply chains. It is especially necessary for produced batches in the food and pharmaceutical industry, as non-conform products can endanger customers' health. For ensuring traceability in supply chains, provider-specific ERP programs exist that allow top-down or bottom-up batch analysis (Doller, 2013). In this case, the sovereignty of data is owned and driven by the companies needs. Our approach offers an understandable way for companies to implement a blockchain solution based on extended lean principles and state-of-the-art information systems so that sovereignty can be transferred to customers.

The most widespread approach to represent real-world assets on blockchains is in the form of tokens based on Ethereum request for comments (ERC) standards (Ethereum Foundatiom, 2015). The most used tokens are ERC20 for fungible tokens and ERC721 for non-fungible tokens (Fröwis et al., 2019). While fungible tokens represent objects that are arbitrarily interchangeable such as stocks, non-fungible tokens represent uniquely identifiable objects such as real estate (Ethereum Foundatiom, 2015). Only non-fungible tokens can be used to trace objects along a supply chain. However, they are not suited for production processes in which goods are combined and transformed into other goods. The original goods can get destroyed in the process, and a new good is created. Therefore, old tokens must be destroyed, new tokens of a different type must be generated, and the old tokens must be referenced in the new ones. This issue was tackled in the literature by utilizing a token recipes model or token compositions (Westerkamp et al., 2018; Westerkamp et al., 2019). This approach adds another layer of complexity to the tracking smart contracts, making the decentralized computations more expensive and possibly hindering scalability (Scherer, 2017). Instead of using tokens, many supply chain solutions rely on different smart-contract-based solutions, such as logging movements of goods on the decentralized ledger (Bocek et al., 2017; Lu and Xu, 2017).

The idea to use tokens to represent and track real-world goods existed before ERC token standards were introduced. The first concept to represent assets on the blockchain was so-called colored coins (Anand et al., 2016). The usage of colored coins is not very widespread since the transaction structure needed to represent colored coins is based on unspent trans-

action outputs (UTXO), which Ethereum-based blockchains do not use (Buterin, 2014). The basic idea is to assign native transactions a property (the color), which indicates the asset it represents. For example, each Satoshi (the smallest possible value of Bitcoin) can represent a different asset for the Bitcoin blockchain. This concept is mainly used to track ownership of the tokens and, therefore, the assets. Since the transactions can be combined or split into new transactions, and the color can easily be changed after each transaction, we see great potential to utilize colored coins as an efficient means for tracing in production environments. Finally, the transactions can easily be visualized and analyzed with existing tools, such as blockchain explorers (Kuzuno and Karam, 2017).

6.3 Conceptual Framework and Methodology

The following section describes the underlying blockchain architecture and transaction structure we suggest for efficiently tracing goods along the supply chain. After that, we introduce the process to map the logistics and production processes from traditional enterprise software to the blockchain.

6.3.1 The Blockchain Architecture

As stated before, we suggest mapping tokens as colored coins on a blockchain that supports these colored coins. This technology works on blockchains that record the current state by keeping track of UTXOs. As the name suggests, UTXOs are transaction outputs that were not yet spent. When a new transaction is created, it takes at least one UTXO as input and creates new outputs. In this process, the output has to be equal to the input (disregarding the transaction fee), and inputs have to be spent entirely. If only a part of the input should be spent, one output can be assigned to a change address, which receives the surplus coins. This address can be the same as the sender's address. This mechanism links transactions together and allows the tracking of individual coins. Therefore, each coin can be marked with a value that indicates which asset it represents. This is referred to as the color of the coin. Once a coin is colored, wallets that support the feature can differentiate between the coins and transfer them like any other token.

So far, colored coins do not differ much from other non-fungible tokens. However, since transactions can have multiple inputs, we suggest a model that can mix colors, representing a step in a production process. For example, a transaction can have tokens representing four wheels, an engine, and a chassis as input and mix the colors to represent a car as the output. If the customer receives the car and the transaction representing the car, he can look at the input transactions and see which wheels were used as the input. Then each wheel can be traced back to the rubber used and so forth. An obvious downside is that the car transaction's output has at least six coins since it had six input transactions. Therefore, it has to be specified somewhere in the transactions, besides the color, how many coins represent

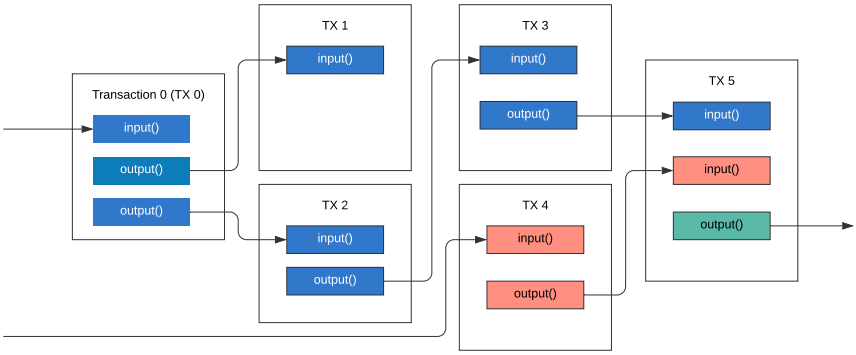


Figure 6.1: Transaction Flow of Colored Coins for a Multi-Stage Production Process

one car. However, this additional information can be written next to the color in the extra transaction data. Figure 6.1 visualizes how colored coin transactions would represent a multi-stage production process.

Such a system's benefit is that it works on simple existing technology and allows easy top-down and bottom-up analysis of the material movements. For a bottom-up analysis, the transaction output of a produced batch serves as a starting point. The transaction graph can easily be traversed by checking in which transaction the output was taken as an input. The resulting graph is a tree, and the leaves are the customers with goods associated with the batch or serial number.

6.3.2 Mapping the Value Stream

To achieve a harmonized material and information flow, Busert and Fay (2019) propose six steps based on extended VSM. We adapted the approach to build a traceability solution for customers using different state-of-the-art information systems. The six steps are displayed in Figure 6.2.

We have gone through the steps using information systems that support primary and extended production processes and material movements. In our view, this includes supply chain planning (SCM) systems, ERP, manufacturing execution systems (MES), and warehouse management systems (WMS).

Step 1 – Traceability Stream Analysis:

(1) – Select Goods / Product Family:

To reach traceability in the supply chain, we started selecting a concrete product or product family. These are materials for which a customer requires transparent traceability of physical movements and production processes. These include perishable goods that are processed

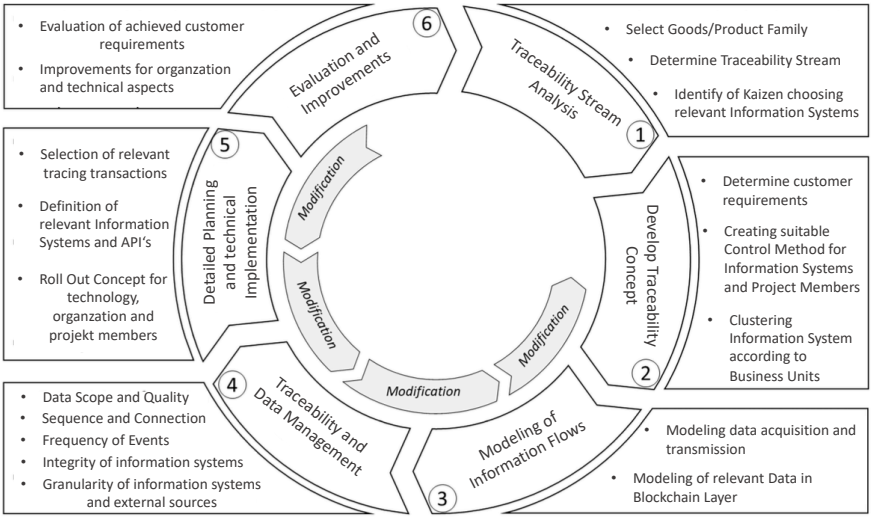


Figure 6.2: Execution Steps of the Extended VSM Method (Busert and Fay, 2019)

along the supply chain. Since companies in practice have numerous routings and bills of materials in ERP systems, this perspective is significantly simplified. Past research has summarized problems regarding the complexity of products (Dal Forno et al., 2014). With a perspective on classical value streams and a manageable number of products with low complexity Rother and Shook (2003) have already provided a way of how product families can be clustered.

(2) – Traceability Stream Mapping:

For getting a basic understanding of the product, the production area has to be determined. In our case, this involves one customer, one production plant, and one vendor. We have gone through the production process virtually from the sink (customer) to the source (supplier) and determined all relevant production processes (workplaces) and storage locations that are relevant for material movements. From the customer to the supplier, we use the master and transaction data generated by the ERP system to determine the exact amount of data required from the customer. We only use classic symbols in the extended value stream notation to present these visually. As this might be time-consuming to visualize material flow activities between storage locations/sections, Knoll et al. (2019) has already presented the possibility of using process mining techniques in a complex production area for internal logistics from goods receiving to the assembly line.

(3) – Identification of Kaizen choosing relevant Information Systems:

The first improvements are already in place following the documented tracing stream. Information systems improve productivity but also generate much data. As this leads to un-

necessary waste due to the costs of executed process steps in blockchain applications (Rimba et al., 2017), we exclude production systems and data that are not relevant tracing products through a production plant.

Step 2 – Concept Development for Blockchain Work Area:

Determine customer requirements, Control Methods, and Clustering:
Information systems contain individual database tables and modules to store information about business processes. For choosing End-to-End Processes, we used the SCOR model. It is a standard Supply Chain Framework to simplify intra-and inter-company processes, which, therefore, offers the main steps *plan*, *source*, *make*, *deliver* (Huan et al., 2004). Since we investigate material movements and batches between these steps, we add two more layers to the concept, as shown in Table 6.1. To uniquely identify data tables in the blockchain, we have marked them with consecutive numbers. Since we demonstrate a logistic and production process based on SAP, we have marked the information systems and modules.

Table 6.1: Concept of Relevant ERP-Module-Types and Tables

Business Unit	SAP System and Modules	Data Type	
Planning	SAP APO	Transaction/Master Data	P1
Sourcing	SAP ERP - MM	Transaction/Master Data	T1/M1
Manufacturing	SAP ERP - PP	Transaction/Master Data	T2/M2
Sales	SAP ERP - SD	Transaction/Master Data	T3/M3
Inventory Management	SAP ERP - IM/(WM)	Transaction/Master Data	T4/M4
Batches	Across several Modules	Transaction/Master Data	T5/M5

Step 3 – Modelling of Information Flows:

Modeling Data Acquisition, Transmission, and Blockchain Layer:
Based on the scope determined in Step 2, the information flow for the production process must be derived. This step is essential to determine the data from the ERP system for a correct token implementation. Busert and Fay (2019) recommend a division into three layers — first, the process layer, which represents all activities belonging from customers to vendors. The second layer contains all information systems (storage media) used in an end-to-end process. To determine the optimal scope of information, we started from the customer's goods receipt, representing the source of a finished product. Especially in the food and pharmaceutical industry, logistics and production processes require special customizing in ERP systems (Doller, 2013). We, therefore, differentiate between information systems and user activities. Our research did not find a detailed representation that would overall represent current information systems using extended VSM. Therefore, we divided them up, as shown in Figure 6.3. The third and last layer displays the processed information. It is equivalent to our demonstrated blockchain layer. It processes information from different ERP database tables to achieve a logically connected chain of material and production processes of batch

materials. An example of the information flow is depicted in Figure 6.3. The relevant information flows from blockchain to ERP differentiate between actual output and associated information. Besides, we investigate if there is a need for harmonization between ERP systems using blockchain or if no other programming in standard code is necessary to display end-to-end material movements.

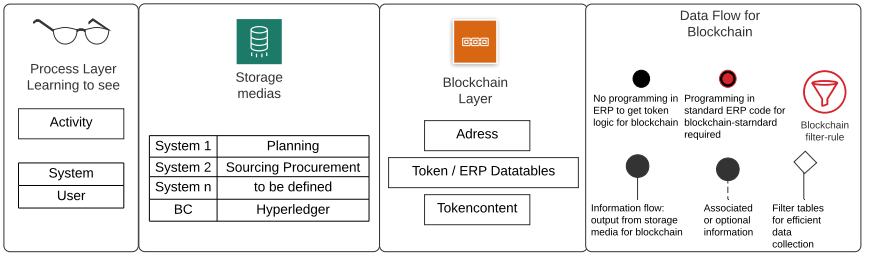


Figure 6.3: Enhanced Value Stream Mapping for a Blockchain Token Approach

Step 4 – Traceability and Data Management:

To harmonize the collected data and reach a high data quality level for material movements, we traversed the tables from the sink (customer) to the source (vendor) to determine the required information's optimal scope. To harmonize activities and information from the ERP system, we have developed five dimensions that must be present as a minimum to achieve a top-down or bottom-up analysis.

(1) – Information Quality:
Master data is used to differentiate between internal and external companies. The material flow for logistics and production must be transparent. The mark contains information about the quantity of material produced, the batch, and the expiration date. We thus fulfill the information required by Global Traceability Compliance for food products to organize supply chains (Mager et al., 2016).

(2) – Sequence and Connections of Tokens:
The individual tokens need information about goods movement, or production orders and batches, to establish a relationship between the sender and receiver. Therefore, we link only data automatically generated by the ERP system to avoid incorrect entries by users.

(3) – Frequency of Events:
The frequency of data pushed or pulled from Information System to the blockchain depends on the business process's requirements. Excellent quality in the blockchain relies on the quality of the ERP system. Therefore, cancellation events in case of incorrect material movements occur in, e.g., ERP Systems. To achieve good data quality, these must be excluded in the best possible way.

(4) – Integrity between Information Systems and Blockchain:
If, besides this precaution, wrong material movements are written to the blockchain, the data must be corrected afterward. Since the blockchain transaction cannot be deleted, a new transaction must be committed to the blockchain, which reverses the wrong material movement before the new, correct transaction can be sent. This cancellation transaction should be marked as such, to not display it in later tracing steps.

(5) – Granularity of Information Systems and External Sources:
Different information systems, database tables, and data fields lead to a different granularity of information. When selecting ERP modules or other information systems, the target is to achieve a top-down and bottom-up analysis. It also requires taking customer needs into account.

Step 5 – Detailed Planning and Implementation:

(1) – Selection of relevant Tracing Transactions:
The ERP transaction data must be selected individually by employees manually (after production order confirmation or transactions on material movements) or automatically by systems. Only relevant data should be transferred to the blockchain.

(2) – Definition of relevant Information Systems and API's:
To choose relevant data for an end-to-end process, ERP systems provide various database tables. Modern ERP systems use numerous APIs to integrate data into a blockchain. For legacy information systems, it may be necessary to program them subsequently. Overall, the blockchain's wallet addresses should correspond to storage locations in ERP Systems to represent a sender or receiver address.

(3) – Concept Roll-Out:
A roll-out concept must be developed individually for technology, organization, and project members. It is essential to involve value stream managers having an overview of the selected end-to-end process (Rother and Shook, 2003). Because of the increasing use of technologies in end-to-end scenarios, value stream managers will be confronted with technical aspects of information systems and goods flow. Therefore, there should be project members that can merge both material and data flow.

Step 6 – Evaluation of the Effectiveness of the Concept:

(1) – Evaluation of achieved customer requirements:
The achievement of objectives depends on the complexity of the product in terms of production and logistics. Products can pass through only one production process or numerous production steps in different plants. In this respect, it is essential to evaluate end-to-end information according to customers' added value.

(2) – Identifying Improvements:
Further improvements can include both organizational and technical aspects.

6.4 Demonstration of the Concept

We demonstrate our approach based on an SAP ERP System we have in place at our chair. The setup is used for students to learn how business processes are mapped to enterprise software in a simulated scenario. We followed the steps from the previous section to map the logistics and production processes to blockchain transactions.

Step 1 – Trace Stream Analysis:

(1) – Selecting Goods / Product Family:
In consultation with the customer and project members, we chose one finished product, which should be traced for demonstration in our food supply chain. To obtain the correct master data for the products, we used the tables MAST, STPO, and MAKT.

Table 6.2: Standard SAP Tables for Bill of Material and Material Description

Tables/Fields	STLNR	low level code	POSNR	MATNR/IDNRK	MAKTX
MAST MAKT	00000190	0	10	BB-F12	1kg Blueberry Cereals
STPO MAKT	00000190	1	10	BB-R02	Blueberries
STPO MAKT	00000190	1	20	BB-R05	Wheat
STPO MAKT	00000190	1	30	BB-R06	Oats
STPO MAKT	00000190	1	40	BB-P01	Large Box (1kg)
STPO MAKT	00000190	1	50	BB-P02	Large Bag (1kg)

(2) – Trace Stream Mapping:
To get a basic understanding of the value stream, we have defined the production area. We identified relevant customers in Table LIKP, LIPS, and MSEG that consume finish materials. To reduce the demonstration's complexity, we have only used the finished product BB-F12 from step 1. From source to sink, we identified customer plant 4712, production plant 4711, and vendor plant 4710. Plant 4711 provides storage location 02 for finish products and storage location 89 for raw materials. Three different raw materials are delivered from supplier plant 4710 Storage Location 01 to plant 4711 Storage Location 89.

(3) – Identification of Kaizen choosing relevant Information Systems:
We identified the first improvement tasks at the start of implementation. Therefore, to avoid waste, we have only implemented material that should be relevant for tracing. Mainly this means we only included food components R-02, R-05, R-06, that have a batch requirement in material master data and are used by bill of material and routing, which have the status "released for production". Therefore, a trace stream analysis was only carried out for materials with batch management requirements. Furthermore, we have excluded the SCM system, as this only performs planning heuristics in the production context of batches and therefore does not provide any added value in terms of the traceability of products for the customer (Doller, 2013).

Step 2 – Concept Development for Blockchain Work Area:

Target Definition and selection of suitable control methods:
Since the focus is on top-down and bottom-up analysis for batch material, we consider only physical movement data and no stock data for traceability. To work out a defined concept, we examined database tables of the SAP system and clustered them according to business units and ERP modules. Further, necessary database tables contain the master data of customizing, which is necessary for mapping organizational structures and the unique identification of a stream of products. To get an understandable representation for our end-to-end example, we changed our simulated production plant in standard field WERKS from "BB" to "4711" and added Values 4710 and 4712 for plants and 01 and 03 for storage locations to include values for our inter-company process. To reduce the complexity of plant 4711, we consider only one production step between storage locations 89 and 02.

Table 6.3: SAP Standard Customizing Tables

Table	Field	Example	Description
T001L	WERKS	4710;4711;4712	Plant
T001L	LGORT	89;01;02;03	Storage Location
T156	BWART	101	Goods Receipt; (+)
		261	Goods Issue for Production; (-)
		601	Goods Issue to Customer; (-)
T156	SHKZG	S	(+) receipt
		H	(-) issue

The SAP system offers different database tables for standard transactions in the sourcing, manufacturing, and sales areas. We initially included database tables for targeted purposes, representing our requirements and which we have collected over time as relevant, optional, or purely informative for further requirements. Relevant batch information is integrated into several processes and standard SAP tables. For sourcing, these concerns, for example, the table EBAN, for the manufacturing area the table AUFM, for Inventory Management table MSEG (Doller, 2013). Tables LIPS and LIKP contain additional data on delivery information. Detailed data on the creation and expiry date are located in table MCH1. As we do not require any stocks for batches, we excluded table MCHB. Tables KNA1 and LFA1 provide comprehensive information on customers and vendors. However, we have not used these in the demonstration. Since the new S/4HANA system offers a different model for representing vendors and customers, table BUT000 is alternatively relevant for SAP and blockchain developers in future integration. Therefore, our control method is the presentation to connect business units, ERP consultants, different information systems/releases, and blockchain developers. We have used the LEANX table documentation [14] to help standardize the descriptions of table names.

[14] `http://leanx.eu/en/sap/table/search`

Table 6.4: Chosen SAP Tables for Tracing Products with Tokens

Business Unit	SAP ERP Modules	Table name	Data Type	Relevant	BC_Ident
Sourcing	SAP MM	EKKO	Transaction Data	optional	T1
Sourcing	SAP MM	EKPO	Transaction Data	optional	T2
Sourcing	SAP MM	EBAN	Transaction Data	optional	T3
Sourcing	SAP MM	LFA1	Master Data	ptional	M1
Sourcing	SAP MM	ADRC	Master Data	optional	M2
Batches	SAP MM	MCH1	Transaction Data	x	T4
Batches	SAP MM	MCHB	Transaction Data		
Manufacturing	SAP PP	MAST	Master Data	x	M3
Manufacturing	SAP PP	STPO	Master Data	x	M4
Manufacturing	SAP PP	MAKT	Master Data	x	M5
Manufacturing	SAP PP	AUFK	Transaction Data		
Manufacturing	SAP PP	AFPO	Transaction Data		
Manufacturing	SAP PP	AFRU	Transaction Data	optional	T5
Manufacturing	SAP PP	AFFW	Transaction Data		
Manufacturing	SAP PP	AFWI	Transaction Data		
Manufacturing	SAP PP	RESB	Transaction Data		
Manufacturing	SAP PP	AUFM	Transaction Data	optional	T6
Sourcing/Sales	SAP MM/SD	BUT000	Master Data	optional	M6
Sales	SAP SD	LIPS	Transaction Data	optional	T7
Sales	SAP SD	LIKP	Transaction Data	optional	T8
Sales	SAP SD	KNA1	Master Data	optional	M7
Customizing	SAP IM SPRO	T001L	Customizing Data	x	C1
Customizing	SAP IM SPRO	T156	Customizing Data	x	C2
Inventory Management	SAP IM	MSEG	Transaction Data	x	T9
Inventory Management	SAP IM	MARD	Transaction Data		
Customizing	SAP WM SPRO	T300	Customizing Data	optional	C3
Customizing	SAP WM SPRO	T301	Customizing Data	optional	C4
Customizing	SAP WM SPRO	T302	Customizing Data	optional	C5
Inventory Management	SAP WM	LAGP	Master Data	optional	M7
Inventory Management	SAP WM	LTAK	Transaction Data	optional	T10
Inventory Management	SAP WM	LTAB	Transaction Data	optional	T11
Inventory Management	SAP WM	LTBK	Transaction Data	optional	T12
Inventory Management	SAP WM	LTBP	Transaction Data	optional	T13

As this is our project clustering and working with SAP ERP System, we know that companies organize their information systems differently. Nevertheless, a subdivision can look very individual and must be developed specifically for each project and ERP system. We will investigate whether the SCOR model applies to other ERP systems. Optional database tables can be redundant or enrich the process with additional information. To select only relevant data needed for a tracing solution, we limit ourselves to database tables that offer added value for the customer.

Step 3 – Modelling of Information Flows:

The modeling of the information flow requires the integration of several project participants. Figure 6.4 shows a finished model for the information flow in our use-case. The complete model consists of three layers. The layers are built sequentially from top to bottom.

Layer (1) – Modelling of the process: Business units and lean experts must first record material movements and workplaces of selected materials. Since the defined target is tracing, no time and inventory recording are required.

Layer (2) – Define relevant storage media: The challenge is to match the representation with existing information systems (e.g., ERP) logic. This requires project members or consultants with an integrative understanding of the business areas and ERP modules to ensure an end-to-end perspective. Once we had found the processes and data, we connected the activities with horizontal lines to the corresponding information systems. Since this concept requires a larger team, we view this approach critically concerning small and medium-sized companies, as they may lack lean and technical competence.

Layer (3) – Plotting data points and creating a Blockchain layer: The interaction points with the storage media during the process are marked with a dot. Activities that cause data transfer to the blockchain are marked similarly.

Compared to the BPMN notation we found in research (Jæger et al., 2019; Lu and Xu, 2017), the extended VSM offers a more detailed view on business and technical aspects regarding industry processes that use complex ERP Systems and colored coins. It provides a control method using a standardized language of business units, lean experts, ERP consultants, different information systems, and blockchain developers. Our research project can identify possible problems corresponding to data quality or integrity can be addressed using the horizontal lines and dots. Therefore, we provide a compliant way to organize the supply chain according to the required GS1 standards, in which documentation about structure, responsibilities, and procedures regarding safety hazards crisis should exist (Mager et al., 2016).

Step 4 – Traceability and Data Management:

(1) – Information Quality:

According to Busert and Fay (2019), ideal data quality in practical production environments is unrealistic. We started from the customer's incoming goods department to determine the optimal scope of information. To ensure that batch numbers are always entered in the goods receive or production order, the setup of ERP needs a mandatory entry of batches for physical movements. Incorrect booking records, like posting consumptions without stock, are restricted for blockchain entries. Therefore they are not transferred to the blockchain until a business unit corrects them (SAP Transaction COGI / Table AFFW/AFWI). Each token needs a real physical partner to trace movements in a blockchain. Therefore, we used organizational master data from ERP Customizing to ensure a realistic representation of a plant and the

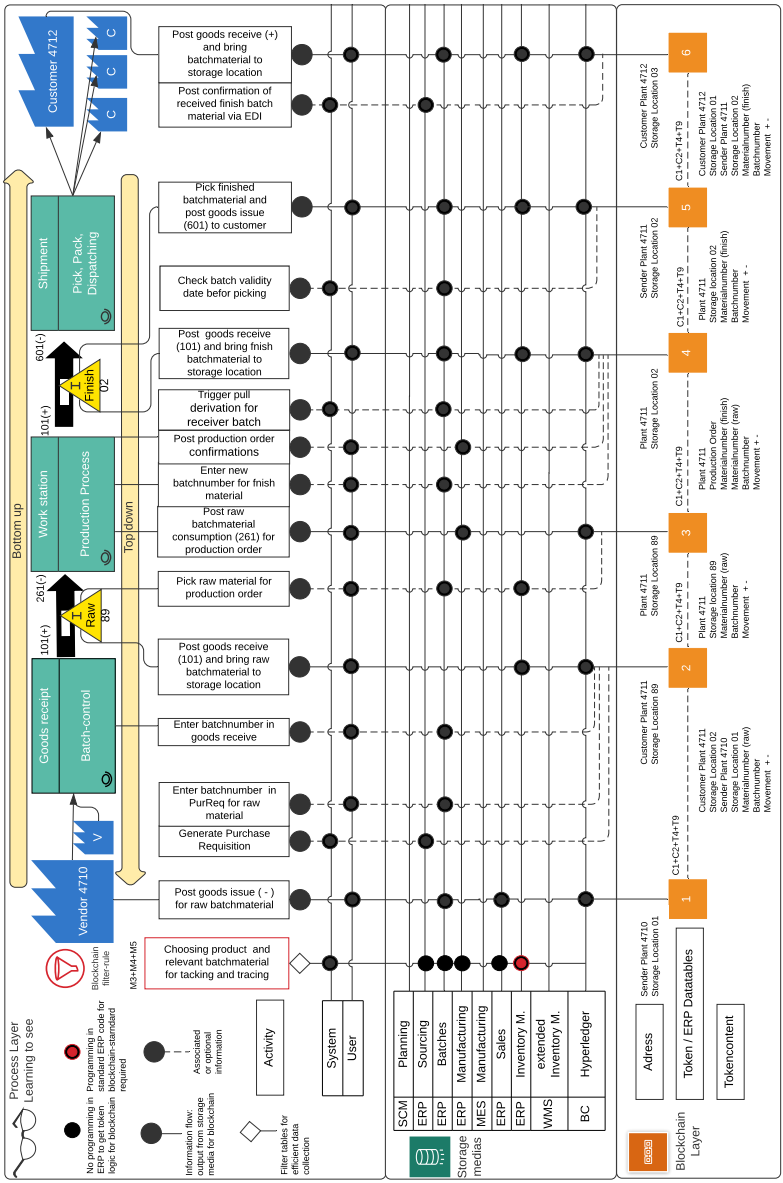


Figure 6.4: Data Acquisition of Information Systems and Batch Processes

corresponding storage locations (Table T001L). Besides, ERP systems offer specific logic for physical movements in customizing, as shown in Table 6.3. Therefore additional effort might be needed to adapt a blockchain standard for physical movements between plant and storage locations to achieve data integrity between different non-SAP-systems.

(2) – Sequence and connections of tokens:
To define a sequence of tokens, we followed the information flow from supplier to customer. To get a coherent chain, all tokens need information about goods movement, vendor number, supplier number, production orders, and identifier (batches). Therefore we link only data automatically generated in material documents (MBLNR) by the ERP system. In our demonstration, we manually enter batch numbers for raw and finish after extracting data from SAP. Besides, generated goods movements that do not find a predecessor in the ERP system are not considered for tracing. We avoid goods issues to the customer being pushed into the chain before a production order is confirmed.

(3) – Frequency of the pushing events from ERP to Blockchain:
The frequency of the events recorded in the blockchain depends on the points described in (1) and (2). We have configured a script outside the ERP system to write only user-released transactions to the blockchain in intervals of one hour. This way, we try to prevent unwanted transactions that result from cancellation movements. This solution causes a "real-time" analysis of the movements is no longer possible. Since this is not the focus of our demonstration, this form of data provision is sufficient.

(4) – Integrity between Blockchain and ERP:
If, despite the measures taken in (3), cancellation movements and subsequent component changes must be logged to the blockchain. It can be relevant for the use of serial numbers in various industries. The integrity is no longer given at the current implementation stage since these transactions are not yet implemented. Otherwise, there is little room for error due to automated scripts that generate transactions for the user in the background.

(5) – Granularity of Information Systems and External Sources:
MES and WMS provide a higher granularity of information regarding production orders and material movements. Since necessary information about production orders and batches is available in the SAP ERP system, we have not integrated the MES. On the other hand, the WMS is an integrated module and provides a higher granularity in goods movements between different warehouse locations (Table T300) and storage bins (Table LAGP). Since our demonstration consists only of one production step, we have marked this in Table 6.4 as an optional path in the data recording in further development.

Step 5 – Detailed Planning and Implementation:

(1) – Selection of relevant Tracing Transactions:
To achieve an end-to-end token process, we have connected ERP information from tables C1+C2+T4+T9 to the presented six tokens in Figure 6.4, to achieve a top-down and bottom-

up analysis. Figure 6.5 shows how ERP System generates the transaction for production order 1000002 in Table MSEG and how the inputs and outputs correspond to the data in Table 6.5.

Table 6.5: Example Production Order in SAP

MBLNR	BWART	MATNR	WERKS	LGORT	CHARG	SHKZG	MENGE	MEINS	AUFNR
4900000116	101	BB-F12	4711	2	WUE_GR_Finish_4711	S	24.000	ST	1000002
4900000018	261	BB-P01	4711	89		H	24.000	ST	1000002
4900000018	261	BB-P02	4711	89		H	24.000	ST	1000002
4900000018	261	BB-R02	4711	89	WUE_GR_4710_R02	H	7.200	KG	1000002
4900000018	261	BB-R05	4711	89	WUE_GR_4710_R05	H	8.400	KG	1000002
4900000018	261	BB-R06	4711	89	WUE_GR_4710_R06	H	8.400	KG	1000002

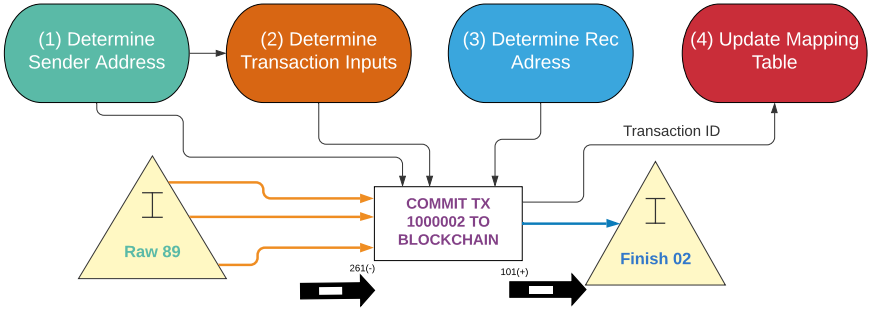


Figure 6.5: Process of Mapping of Inputs and Outputs for a Production Order and Token Transactions

(2) – Definition of relevant Information Systems and API's:
For our demonstration on a blockchain, we used Hyperledger Fabric since it supports UTXO based transactions and is flexible enough to implement the extended colored coin mechanisms. In addition to the blockchain itself, we used middleware scripts that map the ERP data to transactions that can be executed on the blockchain.

(3) – Concept Roll-Out:
We started with one material to achieve a controlled roll-out and successfully integrated it into the platform. Once we verified that the transaction structure worked for one product and could be tracked bottom-up and top-down, we integrated more complex products.

Step 6 – Evaluation of the Effectiveness of the Concept:

We evaluated the solution with practitioners to gain further feedback. We achieved the defined target of tracing goods with lean, ERP, and blockchain technology. The bottom-up and top-down tracing worked well, provided that we used simulated data and no illegal transactions booked into the ERP System. One common criticism was that this type of blockchain approach could make the supply chain too transparent and allow competitors to gain deeper insights into a companies structure. Therefore we suggest the token-based approach could be extended with zero knowledge protocols to shield transaction inputs and outputs

(Hopwood et al., 2020). Only when needed, such a solution, a tracing request, must be issued, and the actual transactions must be revealed. It is, however, far out of the scope of this paper.

With a vision of future applications, they should be evaluated from several aspects. Our evaluation plan, therefore, considers two dimensions:

(1) – Improvements for Organization and Technical Aspects:
Since our prototype is created in a controlled setting, we demonstrated that colored coins could be used in production and logistics processes. We will connect more heterogeneous ERP systems to achieve a more authentic test environment as a further target.

(2) – Evaluation Plan on Performance Aspects:
One premise of our approach was that colored coins are less complex than smart-contract-based solutions. Therefore, we assume that the usage of colored coins is more efficient, especially in large-scale supply chains. However, we have to verify this claim in the future by benchmarking comparable solutions.

6.5 Conclusion

In this paper, we would like to inspire different ways of integrative thinking. Overall, there is more than one perspective to consider when combining new technologies, lean, ERP, or other information systems. A lean perspective on different complexity drivers can influence how industrial companies organize future blockchain roll-outs. Other aspects, such as sustainability, the customer perspective, and small and medium-sized enterprises, should also be considered by practitioners and researchers, as we believe they will become a more integrated part of a modern industrial ecosystem. For the future challenges of modern industry, an integrative perspective will be required to reduce complexity and increase value for both companies and customers.

Our first research question investigated how it is possible to trace an entire supply chain and transfer it in a structured way from ERP systems into a blockchain. Following Buer et al. (2018) the solution for the future sometimes lies in the past. The SCOR model developed in 1996 and the Lean Principles of Rother and Shook (2003) also provide a basis for the development of use cases to present increasing complexity understandably. We furthermore show not only how to integrate information systems but also how to integrate humans and existing lean principles. We have documented this with a simple and understandable example and oriented ourselves on existing standards provided by the GS1 (Mager et al., 2016). Our elaborations show how business units, lean experts, ERP consultants, and blockchain developers can develop a common standardized language in blockchain projects. Integrating different information systems into a blockchain across company borders remains a challenge. It is still essential to develop an overview of the end-to-end process and limit it to activities that create added value for customers. In our eyes, the ERP system is a database capable of mapping end-to-end processes. However, all production and logistics steps must be mapped in information systems (e.g., ERP, MES, WMS, or external sources). Since we have considered

a simple make-to-stock process with batch identification, colored coins offer further research potential for various processes that contain unique references (e.g., purchasing and sales documents numbers or their deliveries, documents or production orders that are labeled with customer orders, processes with serial numbers, handling units used by warehouse management systems, or transport units used by transport management systems).

Additionally, we observed that tracking and tracing along a supply chain become increasingly more complex. Against this backdrop, we asked how a simple token-based approach can achieve tracing in supply networks. To answer this research question, we used traditional VSM from the customers' perspective and introduced a concept of modified colored coins that can change colors when combining different coins in a production step.

The data set for the presentation was limited to one ERP system. We did not implement a fully integrated scenario using standard ERP-API in our attempt, as we could only extract data from our system via the transaction se16/se16n and added batches afterward. We used standard fields but could not use the standard coding for batches of the SAP System. Further research direction is to use other information systems such as Navision, weclapp, Xentral, Sage, Infor, Godesys, Oracle, myfactory, or Odoo to determine whether a simple solution using colored coins applies to heterogeneous IT supply chain networks. The presented production and logistic processes were simplified to visualize an uncomplicated end-to-end process of a small supply chain scenario.

Further research on using our token approach is the integration of colored coins in multiple operations of production orders or specialized processes such as subcontracting, third-party order processing, rework, returns, joint production, and recursive processes. Besides, this includes cross-plant transportation and production scenarios. As we show process steps on a small scale, we would encourage research to provide detailed information about logistic and production processes to assess blockchain applications specifically.

Chapter 7

A Decentralized Marketplace for Collaborative Manufacturing[15]

7.1 Introduction

Uncertainty arising from future demand or machine availability poses significant challenges to production planning and can result in overcapacity or capacity shortages. Both cases decrease profitability by either excess cost or lost sales. Simultaneously, modern manufacturing is becoming more digitized and enables new types of collaboration, such as intra- and inter-organizational sharing of production capacities.

The concept of connecting manufacturing capacities between companies is not new. However, in recent years research moved from production networks (Wiendahl and Lutz, 2002) to platform models for collaborative manufacturing. Schmitt et al. (2015) introduced a marketplace concept to enable inter-organizational sharing of production capacities. The sharing economy's overall concept has since proven to become more relevant in business-to-business environments (Ocicka and Wieteska, 2017).

These platforms often take the form of centralized marketplaces. However, centralized platforms often suffer from trust issues (Hawlitschek et al., 2016; Nofer et al., 2017). Especially when detailed information about a company's production capabilities and its economic situation has to be shared with a third party. Additionally, solutions for niche production technologies or small production networks are not available since the low market volume does not provide a viable business case for centralized providers. Finally, the involvement of an intermediary can introduce additional cost for each transaction.

[15]This chapter was published in *ECIS 2021 Proceedings* as Hofmann et al. (2021a) and co-authored by Chiara Freichel and Axel Winkelmann.

Blockchain technology has been driven by the vision to eliminate these intermediaries (Nakamoto, 2008). Accordingly, numerous proposals have been suggested to use the technology in electronic markets (Richter et al., 2018; Alt, 2020; Zhang and Wang, 2017; Notheisen et al., 2017; Noll and Alt, 2020). The concept of blockchain-based markets has opened a new area of research (see Section 7.2). The markets can be either operated on large public networks or in a small consortium. However, most market approaches inherit the property of complete transparency from the underlying blockchain architecture. Since some users are unwilling to share information about their production capacities with a third party, they are unlikely to share this information on a transparent public ledger.

Against this backdrop, we seek to incorporate other trusted decentralized computation paradigms into a marketplace solution. In recent years, zero-knowledge proofs, specifically, zk-SNARKs (Pinto, 2020) and secure multiparty computation (Zhong et al., 2020), have shown significant synergy effects with blockchain technology. With these technologies, information can be kept private while still being used to match supply and demand on a decentralized platform efficiently. Therefore, our research focuses on the blockchain market perspective *infrastructure for electronic markets* proposed by Alt (Alt, 2020), by developing a new decentralized marketplace model without an intermediary. To develop the new infrastructure, we set the following research questions:

RQ1: Which technologies can be used to solve trust issues in collaborative manufacturing?
RQ2: How should an infrastructure for a decentralized marketplace for production capacities be designed?

We applied a design science research (DSR) approach to develop the novel infrastructure to answer the research questions. The rest of this paper is structured as follows: In Section 7.2 we provide an overview of related literature and previous work on decentralized marketplaces, followed by a description of the methodology for developing and evaluating the artifact in Section 7.3. The proposed infrastructure is introduced in Section 7.4. It consists of a description of the three architectural layers and a sequence model describing the actors' interaction on the marketplace and the layers. Section 7.5 summarizes the results of the expert interviews to explore the artifact's performance in its real environment. The final section discusses the results and concludes this paper.

7.2 Foundations and Related Work

Our paper tackles topics from the three areas: digital markets and platforms, production networks, and distributed ledger technology. In the following, we present previous research that combined at least two of these areas, as shown in Figure 7.1. Out of these combinations, we identified the areas' intersections from literature: 1. production capacity sharing, 2. decentralized marketplaces, and 3. supply network blockchains. We will describe the related literature in the order shown in Figure 7.1. Finally, we discuss the approaches that combine all areas as well as the research gap.

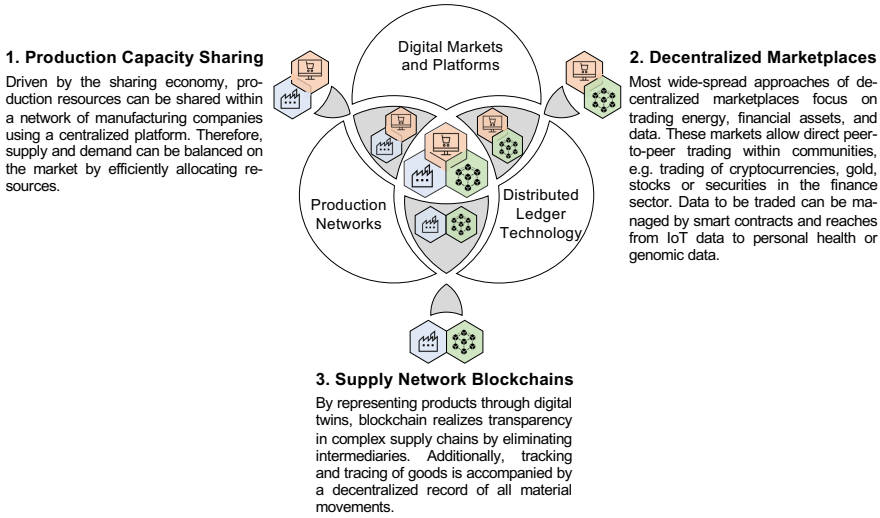


Figure 7.1: Areas of Related Literature and Research Gap

Research on production capacity sharing combines the areas of digital markets and production networks. Mainly driven by the sharing economy trend, the idea of a capacity marketplace for sharing production resources emerged (Schmitt et al., 2015). Within an integrated network of manufacturing companies and inter-organizational production planning, machines can be higher utilized, and supply and demand can be balanced on the market (Schmitt et al., 2015; Freichel et al., 2019). Most approaches propose to realize the marketplace by using a centralized platform to distribute manufacturing capacities. While there are concepts for sharing agricultural machines and woodworking equipment (Daum and Birner, 2020), we focus on stationary machines that cannot easily be moved to a different manufacturing facility. Freitag et al. (2015) modeled and simulated different sharing mechanisms in production networks. Specifically, in additive manufacturing, Stein et al. (2019) developed a market mechanism that efficiently allocates resources utilizing stochastic optimization. This

process can further be automated by allowing an automated matching of products with additive manufacturing machines.

Several approaches pursue the development of decentralized marketplaces. The first and most widespread approaches focus on the energy sector. The decentralized market approach is adaptable in this context since electricity is an extremely homogenous good (Kirpes et al., 2019). Additionally, electricity generation becomes more decentralized if households produce their own power. Furthermore, total transparency is not an issue in this market. The overall trend towards local energy markets allows local communities to direct peer-to-peer energy trading without relying on large energy exchanges (Richter et al., 2018; Mengelkamp et al., 2018). However, even large energy exchanges, like the European Energy Exchange, have discovered the potential of decentralized markets and assessed the benefits of a blockchain-based market (Alt, 2020). The next trend in decentralized markets can be summarized with the term decentralized finance (DeFi). While DeFi spans other areas such as decentralized lending, DeFi is mostly concerned with decentralized trading of various assets, most notably cryptocurrencies (Zhang and Wang, 2017). However, the underlying technologies allow trading of other assets that can be represented as tokens such as gold, stocks, or securities (Notheisen et al., 2019). The final large area is decentralized data trading platforms. Trading data on decentralized marketplaces can be very efficient since the access to the data can be managed by blockchain smart contracts (Noll and Alt, 2020). The data traded on these platforms reaches from internet of things (IoT) data to personal health or genomic data (Xu et al., 2019; Jin et al., 2019). Apart from these more extensive areas, numerous articles exist on other industries, where a decentralized market, with its added transparency and disintermediation, offers advantages over a centralized solution. For example, Zavolokina et al. (2020) presented a marketplace for used cars. Finally, Notheisen et al. (2017) presented a market engineering approach for blockchain-based markets. We used this approach as an orientation for our market design.

The research on blockchain applications in supply chains and supply networks mainly focuses on increasing transparency in complex supply chains. The main focus is on enabling traceability of goods along the supply chain to their source (Kurpjuweit et al., 2021). Due to the immutability and disintermediation, the blockchain contains a decentralized record of all material movements (Banerjee, 2018). The two most common use cases control perishable goods and prevent illegal sourcing of natural resources, especially in the pharmaceutical and food industry (Bocek et al., 2017; Lu and Xu, 2017). With a similar concept, blockchain helps prevent counterfeit products' circulation in the post supply chain. Each product is represented by a digital twin on the blockchain that gets transferred when the products' ownership changes (Toyoda et al., 2017). The disintermediation of information is another significant application domain for blockchains, especially in cross-country supply chains (Hull, 2017). The technology is used for cross-border payments (Guo and Liang, 2016), business-to-government information sharing (van Engelenburg et al., 2019), and automation of business processes with smart contracts (Weber et al., 2016). For a more detailed overview of

blockchain technology application in supply chains, we refer to the comprehensive literature review by Wang et al. (2019).

We could only identify one project that combines all three areas. The Decentralized Industry Marketplace is developed by the IOTA Foundation (Sobolev and Schneider, 2019) and is designed to enable machine data exchange, similar to Noll and Alt (2020). However, the functionality is focused on industry environments. Unlike our approach, the marketplace is not designed to trade production capacities. Therefore, this paper encompasses all areas shown in Figure 7.1 to develop a decentralized marketplace for collaborative manufacturing.

7.3 Methodology

In the following section, we describe the methodology we used to develop the decentralized marketplace infrastructure. We use a design science approach to guide the development and evaluation of the artifact. As we seek to solve a practical problem, we follow the guidelines proposed by Hevner et al. (2004). Additionally, we follow the DSR process, according to Peffers et al. (2007). Finally, blockchain-specific approaches exist to guide the development of blockchain applications (Xu et al., 2016b, 2017; Wüst and Gervais, 2018). We extend the technologies used by other decentralized, trustless mechanisms, such as zero-knowledge proofs and secure multiparty computation, as proposed by Hofmann (2020).

7.3.1 Development of the Artifact

The framework shown in Figure 7.2 guides our methodological procedure, according to Hevner et al. (2004). The authors propose that the artifact has to fulfill the environment's business needs to ensure relevance. The artifact is designed to connect producing companies of various sizes. Especially small and medium enterprises (SME) suffer more from fluctuations in demand, and a marketplace for production capacities can help them become more flexible and robust towards these fluctuations (Freitag et al., 2015; Stein et al., 2019). We assume that the companies use information systems to plan their production and that they can be extended to purchase or sell production capacities automatically. If this is not the case, the proposed solution can still work but offers slightly less utility. Some steps must be performed manually, especially transferring data about physical machines to the blockchain-based assets.

Furthermore, the artifact must be based on existing theories, models, and methodologies to ensure scientific rigor. We base our approach on existing models, methods, and instantiations of decentralized systems, especially marketplaces (see Section 7.2). We use the design process by Peffers et al. (2007) and methodologies for qualitative interviews to ensure methodological rigor.

To ensure a high utility of the developed artifact, we applied the following seven guidelines for the application of DSR introduced by Hevner et al. (2004).

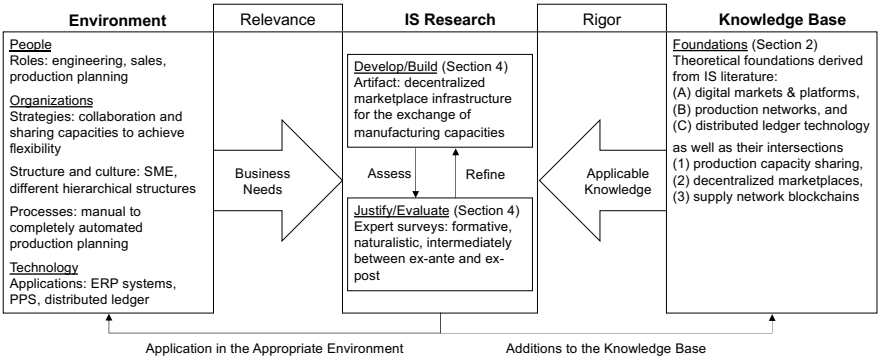


Figure 7.2: Design Science Research Framework (based on Hevner et al. (2004))

Design as an artifact: The research result is a proof-of-concept infrastructure for a decentralized marketplace for production capacities. The design allows users to buy and sell production capacities without revealing their capacities to third parties.

Problem relevance: The designed mechanisms allows connecting supply and demand in a decentralized ecosystem. Additionally, it eliminates previous decentralized approaches' main weakness, namely the transparency of purely blockchain-based approaches.

Design evaluation: We use semi-structured expert interviews to evaluate the proposed solution's utility, usefulness, and practicability. We further describe the evaluation methodology in Section 7.3.2.

Research contribution: We extend the knowledge base by introducing an approach to represent manufacturing capacities with fungible tokens that can be traded with existing technologies. By constructing a complete infrastructure, we build the foundation for the implementations of fully decentralized and secure marketplaces.

Research Rigor: To ensure scientific rigor, we take DSR frameworks as well as blockchain-based frameworks into account in the design process. Additionally, we base our research on prior scholarly publications focused on decentralized marketplaces and disintermediation.

Design as a research process: To find a solution that fits the problem, we use the iterative process proposed by Peffers et al. (2007). We evaluate existing concepts and solutions from the scientific literature to build a preliminary infrastructure during this process. Additionally, we use an ex-ante evaluation to incorporate requirements from practitioners and refine the infrastructure.

Communications of research: We describe the infrastructure and the interactions necessary to exchange capacities on the marketplace. During the evaluation, we secondarily focused on making the description of the solution understandable for practitioners and scholars to maximize the range of the contribution.

7.3.2 Evaluation of the Artifact

Following the definitions of Venable et al. (2016), we perform a formative evaluation in-between ex-ante and ex-post evaluation. We employed the evaluation with qualitative expert interviews to explore our artifact's performance in its real environment (Venable et al., 2016). We rely on an exploratory research approach by integrating multiple stakeholders' perceptions with extensive experience in the domain of interest. In the following paragraphs, we describe the selection of interview partners, the semi-structured questionnaires' design, and the procedures for data collection and analysis. The interviewees all have knowledge about platform business models as well as blockchain technologies. We use criteria such as market position, experience in the research field, and industry and company size for the selection. The final set of interviewees consists of three blockchain experts of different companies (experts A, B, and C). A short overview of the interviewees is displayed in Table 7.1. Even though the group consists of practitioners, all experts have a research background and are currently involved in blockchain research projects. This small panel of highly specialized experts was suitable to quickly improve the model within this first evaluation. We plan technical evaluations with an implemented marketplace prototype and a broader evaluation with potential users. However, this is out of the scope of this conceptual work.

Experts	Industry	Position in Company	# Employees
Expert A	Software Development	Scientist in Residence	300
Expert B	BC Development and Consulting	Product Manager	70
Expert C	Software Development	Business Development Manager	100

Table 7.1: Classification of Interviewees

We designed semi-structured questionnaires as interview guidelines with questions that can all be answered openly. We provide the opportunity to include emerging concepts and ideas (Edwards and Holland, 2013; Paré, 2004). Besides essential, factual, or direct questions that address this study's key topic, structuring questions were used to guide the interview progress. The questionnaire starts with a preamble explaining the goals and scope of this study. The preamble is followed by general questions asking for advantages and disadvantages of central marketplaces to prepare the interviewees for the following, more complex questions about decentralized marketplaces. The following two parts of the questionnaire include descriptions of the developed layer model and sequence model for decentralized capacity exchange and questions concerning the models' understandability. In the last section, we investigate the minimum requirements and exclusion criteria for using the marketplace. Finally, we ask for issues with the proposed design and improvement potential for these issues. To ensure appropriate clarity, structure, and length of the questionnaire, we conducted a pilot study with two independent researchers (Berg, 2001). We did only apply minor changes to the final questionnaire. The interviews were conducted by phone and documented as audio recordings. For data analysis, we applied the process suggested by Kuckartz (2018) and Green et al. (2007). We transcribed the recordings literally, adapted the sentence structure,

deleted filling words, and anonymized the final transcript. In the next step, we coded the transcripts by keywords to aggregate relevant information. We analyzed the frequency of specific keywords and their synonyms to determine relevant codes with sub-codes. Finally, we grouped the codes into higher-level categories. For each category and code, we collected all text segments to aggregate and compared relevant statements. After this process, the coded information was used to evaluate and improve our artifact, described in the following section.

7.4 Infrastructure for Decentralized Collaborative Manufacturing

In the following section, we introduce the proposed infrastructure. First, we describe the overall architectural model and its layers and components. Then, we explain buying and selling capacities on the marketplace and the interaction between the layers. In both subsections, we directly integrated our findings gained from the evaluation.

7.4.1 Architectural Model

We use a three-layer model in Figure 7.3, consisting of a (1) machine layer, (2) token layer, and (3) matching layer. First, on the machine layer, the real-world machines of type are represented. Free capacities of these machines correspond to a number of capacity tokens on the token layer. For example, one capacity token represents one hour of machine time. Additionally, value tokens on the token layer are used to pay for capacities. These value tokens could be a stablecoin initially (e. g., the value of one token corresponds to one dollar) but could later be freely tradable if used in a stable ecosystem.

The underlying token technology should use privacy-preserving features such as zero-knowledge proofs or ring signature algorithms to hide transaction inputs and outputs to mitigate transparency issues (Hopwood et al., 2020; Noether et al., 2016). To enable trading of capacities for value tokens, buyers and sellers must submit their orders to the decentralized order book. Buyers of capacities order capacities in the form of capacity tokens for a maximum price of value tokens. In contrast, sellers order a minimum amount of value tokens for their amount of capacity tokens. Additionally, both buyers and sellers must submit a deadline until which the offer must be processed. Finally, within the matching layer, both sides' offers are matched to maximize the revenue on the platform. In the following, we further explain each layer in detail.

Machine Layer

The machine layer represents the physical/actual machines that are present at each manufacturing company. The machines are indexed by type, and machines of each type are

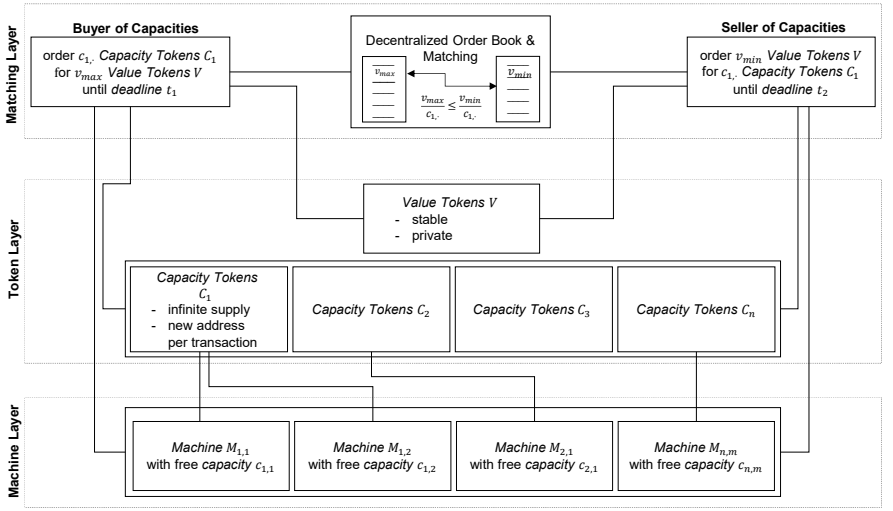


Figure 7.3: Layer Model for Decentralized Capacity Exchange

enumerated. A manufacturer can have multiple machines that are all the same type of CNC machine ($M_{1,1}, M_{1,2}, ...$), and several different models of additive manufacturing machines ($M_{2,1}, M_{3,1}, ...$). For example, in additive manufacturing, different models can utilize different techniques to manufacture and bond layers or differ in the maximum build dimensions. These differences influence the type of goods produced on the machine and the cost to produce it. Additionally, it can be important to differentiate between individual machines, as individual machines of a model can have different characteristics despite the same manufacturing process, such as the material that is being processed. We assume that each machine type has a finite production capacity allocated by a production planning system.

Token Layer

The token layer is connected to the machine layer, as each type of machine has a corresponding token on the token layer. As previously stated, each token represents, e.g., one hour of free capacity on a given machine. The concept of tokenization of real-world assets is well established and lays the foundation of the so-called token economy (Lee, 2019). Since capacity is mutually interchangeable, the capacity tokens should be representable as a fungible token. To prevent one user from holding all capacity tokens of one kind, we propose not to limit the tokens' supply. On the technical side, this can be achieved with a token based on the ERC20 token standard with variable supply (Ethereum Foundatiom, 2015). If suppliers want to hide their transaction history of sold capacities, they can choose to use a new wallet

address for each time they sell tokens on the marketplace. However, sellers of capacities can also use the same address as a marketing device to show their positive record.

We propose an approach to issue different tokens for each combination of machine, material, and certifications. This would not change the matching process than only associating tokens with machine types, but only the number of different tokens. A higher granularity allows better differentiation between the different offers resulting in different pricing for each token. We show examples, how different tokens for these combinations would correspond to ERC20 tokens in Figure 7.4.

Figure 7.4: Each Combination of Machine, Material and Certification Corresponds to a Unique Token

Figure 7.5 shows how the user choices would translate to an order that can be transacted to the matching layer. This process can be automated if the capacities and resources, such as materials, are planned with an information system. If the planning is done manually, choosing the correct token to sell or buy is also a manual process and complicates this step.

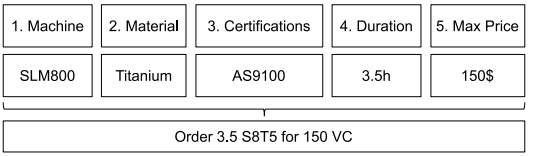


Figure 7.5: Automatic Generation of Orders based on the Users Input

The degree of differentiation of the tokens is highly dependent on the underlying manufacturing process. We are, therefore, looking forward to implementing a prototype in different manufacturing scenarios to validate the approach further.

To allow trading the capacity tokens against money, we introduce a second type of tokens, the value tokens. From a technological perspective, we recommend a privacy-preserving token technology as seen in cryptocurrencies such as Zcash or Monero (Pinto, 2020; Zhong et al., 2020). This is important to minimize transparency on market activities and hide sensitive information about sales from other participants. On the economic side, it should be chosen whether the token should be priced according to the market or if it should be a so-called stablecoin (Mita et al., 2019). This means that the token's value is pegged to a national

currency, through active monetary policy, introducing an intermediary who actively regulates the price. However, the intermediary would not have any insights into the individual transaction behavior or production capacities. Additionally, if the motivation to introduce a decentralized marketplace is the small market size, this intermediary would not interfere with the objectives of the marketplace.

The token layer can be implemented directly on a blockchain. The proposed solution is designed for manufacturers who do not want to share information about their capacities with a centralized third party or for smaller production networks to make centralized solutions available or affordable. For the first scenario, the blockchain can be either completely public or powered by a consortium blockchain operated by a few larger stakeholders. The solution of a consortium blockchain is also suitable for the second scenario. Here, all stakeholders should operate the network. This network architecture allows for much more efficient consensus algorithms and a much lower cost for operating the network compared to a public network (Hofmann, 2020). This consensus can eliminate many of the inefficiencies that are often associated with blockchain technology.

Matching Layer

The matching layer provides the primary mechanism behind the proposed marketplace design. We rely on a dark pool protocol to enable automatic pricing, such as described by Zhang and Wang (2017). The protocol allows the trade of arbitrary crypto-assets, like cryptocurrencies (value tokens) and ERC20 tokens (capacity tokens), and is, therefore, ideal for our use case.

Trades are placed on a decentralized, hidden order book and are matched through a distributed matching engine. The order matching works like a centralized exchange based on the bid and ask prices for the capacities. The matching engine uses multiparty computation, which provides order execution without exposing sensitive information such as price and volume at a particular position. This prevents other market participants from taking advantage of the transparency of an open order book by methods such as frontrunning (Zhang and Wang, 2017).

For the matching, orders are split up into fragments transmitted to different nodes in the network. The fragments are not a fraction of the orders' value, but a separation of sensitive data of the underlying order. Each node performs order matching computations on fragments of different orders. The result is combined with the result of other nodes that computed different fragments.

This protocol does not run directly on the blockchain as a smart contract. Instead, it additionally requires the operation of the decentralized dark pools. The participants of the network can operate these nodes. Since the protocol relies on a trusted party, such as a blockchain, we propose to run it alongside the blockchain nodes.

This matching process has three advantages compared to other, purely blockchain-based approaches (Zhang and Wang, 2017). First, the full order can only be reconstructed if over half

of the fragments are combined, making the network resistant against order reconstruction attacks. Second, since only half of the fragments are required, the network is also resilient to denial-of-service attacks and nodes' failures during the computations. Finally, not all nodes have to perform all computations to perform a secure matching. Performing expensive computations securely outside of the main blockchain network and only proving the results to the blockchain is an excellent way to improve the scalability of blockchain applications (Hofmann, 2020).

7.4.2 Process Model

Figure 7.6 shows the complete process for buying and selling capacities using a UML sequence diagram. The three layers and the buyer and seller are the entities represented at the top of the diagram. Rectangles represent times in which an entity is performing an operation. The sequence of operations is performed from top to bottom. Arrows represent messages that are triggered by the entities. If the arrows are dashed, the message is a response to a previous message.

To buy manufacturing capacities, the buyer must own value tokens to exchange on the marketplace. These tokens can be bought via the token layer as a first step. Then the offer can be placed on the matching layer. The seller must first check how much capacity is still available on different machines. For the capacity to be sold on the marketplace, the seller must request capacity tokens for the respective machine. These capacity tokens can then be offered on the matching layer. The matching layer matches offers in real-time and, if two offers can be matched, automatically swaps the corresponding tokens on the token layer. This token swap ensures that the seller can not offer the same capacity again on the token layer. The only possibility to accidentally double book the sold capacity would be for the seller to sell the capacity on a different platform.

As soon as the buyer receives the capacity tokens, the production information must be sent to the seller, who can then start manufacturing the ordered items. The information can be encrypted with the public key of the token sender to protect the content from unauthorized access (Menezes et al., 1996). As soon as the goods are manufactured, they are shipped to the buyer, verifying that the order is complete and meets the quality criteria. Finally, the seller of the capacities can sell the value tokens via the token layer to exchange them for fiat currency.

7.5 Evaluation Results

During the evaluation, Expert A stated that the quality of products, even if produced on the same type of machine, can differ dramatically. Therefore, the expert proposed to include meta-information about the machine in the tokens. Expert B suggested solving this issue by incorporating information about the processed material and manufacturers' certifications into the tokens. Since this is not trivial for the proposed type of tokens and it is hard to consider

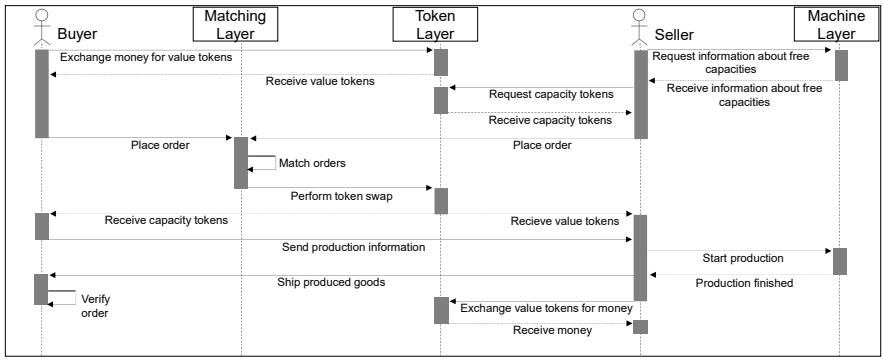


Figure 7.6: Sequence Model for Decentralized Capacity Exchange

this information in the matching process, we propose an approach to issue different tokens for each combination of machine, material, and certifications. We further discussed this idea with expert B, who stated that this idea only works if the choice of the correct token to buy and sell is automated, based on the users' choices of machines, material, and certifications. Expert C confirmed that companies would not adopt a marketplace that is too complicated to use. We demonstrated that it is not difficult to associate capacities with tokens. However, this can only work fully automated if the artifact can directly be integrated into a production planning system.

Concerning the value tokens, expert A argued that the marketplace could only be successful if it is not subject to speculation. Using a stablecoin would minimize the risk of using the value token as a speculative asset.

All experts stated that information leaks in the matching process and network failures are the main exclusion criteria that would prevent users from joining the marketplace. The properties of the chosen protocol, therefore, fit the requirements of the experts.

Expert C criticized that the process does not include a mechanism to modify or cancel an order once it is matched. Since all transactions are unchangeable after they are committed to the blockchain on the token layer, this could lead to fraud in the marketplace. Attackers could offer capacities on the marketplace that are not available and immediately sell the value tokens after the token swap. Both could be mitigated by introducing an escrow smart contract into the process. Asgaonkar and Krishnamachari (2019) introduced such a mechanism, specifically to enable cheat-proof delivery and payment of goods. However, including this in our proposed marketplace is out of this paper's scope and subject for future research. The idea behind such a smart contract is that the tokens are not swapped immediately to the buyers' and sellers' wallet but are locked into a smart contract and can only be released if both parties agree to it, i.e. that the real world transaction produced goods were produced and

met the buyers' quality criteria. If the two parties cannot agree, the case has to be disputed internally, or legal measures must be taken.

Finally, all experts stated that fraudulent behavior or insufficient product quality could be prevented if only trusted companies could join the marketplace. This could either be achieved by using a consortium blockchain and establish a precise legal setting. Alternatively, a public blockchain can be used with decentralized governance and a decentralized know-your-customer solution, such as proposed by Parra-Moyano et al. (2019).

7.6 Conclusion

The presented concept of a marketplace for manufacturing capacities allows companies to use the advantages of sharing economy since modern manufacturing is becoming more digitized and enables new types of collaboration. Intra- and inter-organizational sharing of capacities within a production network allows buyers of capacities meeting demand and sellers to utilize available and unused capacities. First approaches have been proven successful and realize this concept as centralized marketplaces. However, we could identify trust issues and transaction costs as the main limitation.

Against this drawback, we identified blockchain technology as a suitable solution since it eliminates intermediaries, keeps information private if combined with new cryptographic technologies, and is successfully used in comparable electronic markets scenarios. Therefore, we could answer our first research question of which technology we could use to solve trust issues in collaborative manufacturing. To answer our second research question, we applied the DSR approach to design an infrastructure for a decentralized marketplace for production capacities consisting of two models. The three-layer model describes the interaction of the machine layer, token layer, and matching layer. The complete process of decentralized capacity exchange between buyer and seller is designed as a sequence model.

The presented infrastructure combines well-established concepts from blockchain technology with new cryptographic primitives to enable secure and private trading. Hereby, capacities are represented by ERC20 tokens with variable supply that can be traded against a stablecoin on a decentralized exchange. The capacity tokens can represent complex combinations of machines, materials, and certifications to allow users to specify their orders precisely.

This research provides a scientific as well as practical contribution. From a scientific point of view, it contributes to the emerging research on sharing production capacities. The research complements existing articles on an infrastructure perspective and, thus, expands the concept of a marketplace for production capacities on new design elements. Second, our results contribute to general, practical research on digital markets. With the theoretically grounded design science approach, the study can contribute to a more precise elaboration of blockchain use cases. Especially on an abstract level, the proposed infrastructure could be used for other mutually interchangeable goods.

However, our research is not without limitations and leaves room for future research. We are aware that the blockchain technology introduces a complex solution to simple problems. While there is a research stream that searches for solutions to this complex problem, we argue that at the current state, the proposed marketplace is not ready to be implemented in practical environments. However, we view the proposed architecture as a reference for future implementations.

As this study is exploratory, the final artifact design is influenced by the chosen methods and the derived implications. Since we evaluated our artifact with blockchain experts between ex-ante and ex-post, we will add an ex-post summative evaluation with blockchain experts as well as potential users in further research. Especially the evaluation with potential users will show how well the representation of capacities with tokens is suited for their use case, since it only takes time into account, not material consumption or wear and tear of the tools. Moreover, our evaluation showed further improvement potential, such as the inclusion of escrow smart contracts to secure the marketplace against fraud.

Nevertheless, the artifact design provides a basis for future research tackling concrete implementation in a real-world inter-organizational production network applying the concept of a decentralized marketplace for sharing manufacturing capacities in practice.

Chapter 8

An Architecture Using Payment Channel Networks for Blockchain-based Wi-Fi Sharing [16]

8.1 Introduction

Wi-Fi sharing has become a topic of interest in research and practice (Camponovo and Cerutti, 2005; Frangoudis et al., 2011; Cao et al., 2015). It yields various benefits, including ubiquitous Internet access, lower utilization of mobile network capacities, and reduced need for maintenance due to decentralization and self-regulation. For instance, despite 5G availability and free Wi-Fi initiatives in some major cities, still the intrepid traveler often faces steep fees for data access once he or she leaves free roaming coverage. A global decentralized Wi-Fi sharing network with low entry barriers for both users and operators can be a remedy.

For operators, such solutions can improve the perceived network and service quality by extending their services' coverage and capacity (Dimatteo et al., 2011). To date, several initiatives have established public Wi-Fi infrastructures, so-called hotspots, thereby, providing individuals with the opportunity to share their private broadband connection with public guests. For example, Fon is an international company that offers a Wi-Fi community network with over 21 million hotspots around the world (see fon.com).

However, current Wi-Fi sharing concepts have several constraints, such as user authentication or illegal behavior, and lack coverage, participation, and scalability (Cao et al., 2015; Leroy et al., 2011). This is partly due to a one-sided dependence on network operators, who

[16] This chapter was published in *ACM Transactions on Management Information Systems* as Janiesch et al. (2021) and co-authored by Christian Janiesch, Marcus Fischer, Florian Imgrund and Axel Winkelmann.

not only control price structures and terms of use, but also determine the network's availability through their own customer reach and area coverage (Shi et al., 2017). While most users are concerned about security issues and potential decreases in their private network performance, current solutions lack adequate incentives or benefits to compensate for these risks and, thus, to facilitate their participation in Wi-Fi sharing networks (Shi et al., 2017; Mamatas et al., 2010).

Addressing these shortcomings, we propose a fast, reliable, and scalable reference architecture for Wi-Fi sharing based on blockchain technology and payment channel networks. The concept fundamentally builds upon two complementary components. First, a blockchain provides a distributed database for saving and securing transactions and building mutual trust among users within a network. Second, payment channel networks provide users with the means to conduct transactions without committing each of them to the blockchain, thus, enabling high network performance at low costs. Consequently, Wi-Fi sharing becomes uncoupled from traditional network operators and users face more incentives to participate in the network.

With our research, we contribute to research on the effects of blockchain on networked business models in particular considering trusted third parties. We summarize our research questions as follows:

RQ1: What are the requirements for secure and reliable Wi-Fi sharing networks and how are they addressed by current approaches and concepts?

RQ2: Based on these requirements, what are design principles for the design of a reference architecture that facilitates the development of scalable, efficient, and secure Wi-Fi sharing networks?

We employ a design science research (DSR) approach to develop our contribution. Consequently, this research centers on designing and developing an artifact in the form of a reference architecture for blockchain-based Wi-Fi sharing networks.

We organize this paper as follows: In Section 2, we introduce the theoretical foundation on the concepts of blockchain and payment channel networks. Subsequently, we explicate our research method in Section 3. After collecting and analyzing the requirements for workable Wi-Fi sharing networks by detailing related work in Section 4, we develop 12 design principles for workable Wi-Fi sharing networks in Section 5. We instantiate them in a multilayer reference architecture for Wi-Fi sharing networks in Section 6 and detail our evaluation efforts in Section 7. Section 8 concludes this research with a summary of findings, limitations, and future research potentials.

8.2 Theoretical Foundations

8.2.1 Blockchain

The blockchain describes a distributed transaction ledger that is duplicated across all participants in a network (Beck et al., 2016). Transactions made on the blockchain are verified, grouped, and chronologically stored as a chain of data blocks. Blockchains can process different types of data and, unlike traditional networks, do not require trusted intermediaries due to the use of cryptography and game theory (Nakamoto, 2008). Initially viewed as an alternative for the bank-centered financial system, research and practice have recently introduced various blockchain application scenarios, which span across different sectors and industries including electronic markets (Alt, 2018).

From a technical perspective, blockchain-based systems build upon a decentralized database, cryptographic security measures, and consensus mechanisms, which provide the means for decentralized time stamping and agreement among multiple distributed participants (Gipp et al., 2015). Based on so-called smart contracts, blockchains can evaluate transactions against a set of programmable rules and, thus, enable parties, who do not fully trust each other, to interact (Tschorsch and Scheuermann, 2016). In general, a blockchain represents an immutable distributed ledger in which transactions are recorded publicity as blocks chained in a chronologic order (Fanning and Centers, 2016). Each block is assigned with a unique identifier in the form of a hash, which is produced by running contents of a block through a cryptographic hash function (Sikorski et al., 2017). To ensure immutability, changes to the original data incur extensive and seemingly uncorrelated changes to the hash and require altering all data entries subsequently recorded on the blockchain (Rogaway and Shrimpton, 2004). As the blockchain is mirrored across all peers of a network, it provides full transparency regarding transactions and facilitates mutual trust and security (Risius and Spohrer, 2017).

Besides resolving conflicts among interacting agents in a network, the technology is capable of reducing information asymmetries without establishing a central instance (Beck and Müller-Bloch, 2017; Yli-Huumo et al., 2016). In practice, consensus mechanisms vary regarding their application scenarios. For example, public and anonymous blockchains require that mining new blocks is linked to a sufficient amount of cost to prevent the distribution of malicious content (Beck and Müller-Bloch, 2017; Swan, 2015; Derks et al., 2018). Proof-of-work (PoW) and proof-of-stake (PoS) are the most widespread and most researched consensus mechanisms today. These mechanisms demand high computational resources or high monetary resources, respectively, and can yield centralization and high costs (Beikverdi and Song, 2015).

To append a new block to the blockchain in PoW-based networks, the participants have to find a specific value (referred to as nonce) that is combined with the transaction data of the block and the hash of the previous block. The value has to be chosen such that the hash of the combined data starts with a string of zeros. The number of zeros is determined

by the current difficulty of the protocol. Due to the use of cryptographic hash functions, the nonce cannot be calculated, but must be found through brute-force search, which is a massive computational effort and consumes large amounts of energy. This makes PoW-blockchain transactions expensive but very secure. If two different blocks are broadcasted simultaneously to the network, each node must choose which should be appended, by using this blocks hash for the calculation of the next block. After a few blocks, one version of the chain will be longer than the other, since more nodes agreed on this version and the other chain will be orphaned. Consensus here means, that the nodes agree on the longest chain, that is the chain with the most computational effort.

PoS can be seen as a virtualized form of PoW. Here, the resources are not denoted by computational power, but monetary resources in the form of tokens on the blockchain. For each block, a validator is selected, based on the number of tokens they possess. PoS is generally considered less secure than PoW, since it has one major flaw: when two blocks are broadcasted simultaneously, nodes do not have to choose which one to keep. They can use their stake to produce blocks for each of the blocks to maximize their reward, resulting in a constantly forked blockchain. This is referred to as the nothing-at-stake problem.

In a controlled environment, such as a private or permissioned blockchain network that consists of unique and known participants, computational load can be significantly reduced based on identity-based authentication schemes such as practical byzantine fault tolerance (PBFT) (Li et al., 2015; Bellare et al., 2009). Here, each participant votes for the next valid block. Since each participant has a unique identity, the system cannot be flooded with votes from fake identities. Furthermore, the voting process is conducted over multiple rounds to account for network errors and ensure correctness. Furthermore, several hybrid mechanisms exist for niche applications. However, all mechanisms rely on the appropriateness of predefined rules. Hence, it is important to ensure their correctness, reliability, and accuracy (Ahangama and Poo, 2016).

Although the number of mostly disruptive visions has grown tremendously in recent years, Avital et al. (2016) argue that neither research nor practice has fully grasped the technology's true potential. In fact, most solutions remain premature, and implementations are limited to a preliminary proof of concepts. By conducting a comprehensive literature review, Risius and Spohrer (2017) reveal that the current body of research has mostly focused on technological questions of design and features, while neglecting aspects associated with the application, value creation, and governance of blockchain solutions.

8.2.2 Payment Channels Networks

Payment channels describe a class of techniques that enable users to conduct multiple transactions without committing single transactions to the blockchain (Malavolta et al., 2017). In the case of purely bidirectional transactions, payment channels constitute bilateral agreements between two parties. To establish a new connection, unconnected parties must con-

stantly negotiate and agree over multiple aspects, thus, yielding high transaction costs and reducing performance and scalability. Against this backdrop, multiple users can build payment channel networks, which allow unconnected users to conduct transactions by routing payments over intermediaries (Rohrer et al., 2017). These networks typically draw upon Hashed Timelock Contracts (HTLC), as a special class of smart contracts that is established between parties of a transaction and transferred to the blockchain for execution (Decker and Wattenhofer, 2015; Poon and Dryja, 2016). While research and practice have introduced a variety of payment channel network concepts, this paper builds upon Poon-Dryja payment channels, which are implemented to conduct Bitcoin transactions in the Lightning Network (Poon and Dryja, 2016).

Joining a payment channel network requires users to create a new channel that is connected to a network participant as well as to make a funding payment, which equals the overall transaction's value (McCorry et al., 2016b). Both parties must then agree to a set of rights and obligations to conduct a transaction. Initially, the network blocks the sender's funding transaction until the receiver secures an equivalent refund transaction, which equals the outstanding amount (McCorry et al., 2016b). This mechanism constitutes a money-back guarantee and ensures secure transactions, even if one partner is non-cooperative or seeks to conduct fraudulent behavior (Poon and Dryja, 2016). The blocking time also determines the closing of the corresponding payment channel (Poon and Dryja, 2016). We summarize and illustrate the functioning of payment channel networks with the example in Figure 8.1.

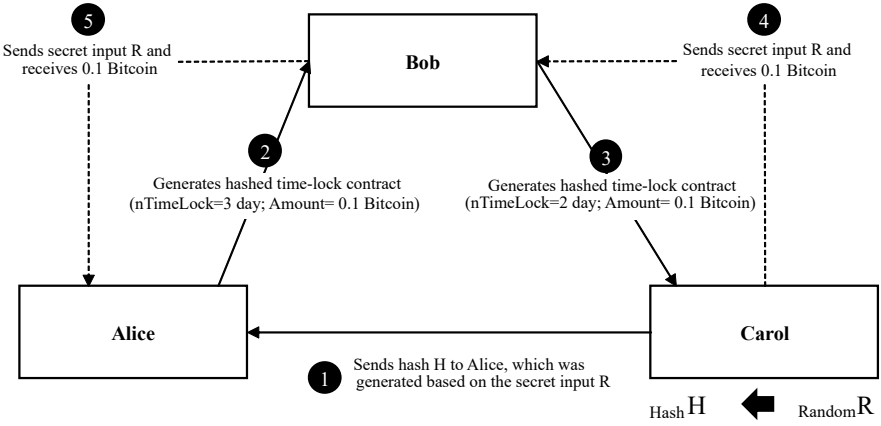


Figure 8.1: Transactions in Payment Channel Networks

In this scenario, Alice sends 0.1 Bitcoin to Carol, while both are connected to each other through the intermediary Bob. Thereby, Carol creates hash H based on the secret random number R and sends it to Alice (1), who establishes an HTLC with Bob (2). The contract

allows Alice to send 0.1 Bitcoin to Bob and requires both partners to agree on the following aspects:

1. If Bob can create the known hash H from the random number R and send it to Alice within 3 days, Alice will compensate Bob with the amount of 0.1 Bitcoin.
2. After three days, the contract is voided, and payments can neither be send nor requested.
3. Subject to approval of Bob and Alice, the established contract can be closed prior to this time limit and withdrawals of any amount can be made.
4. If Bob or Alice breach any of these obligations, the full transaction amount is transferred to the counterparty.

Subsequently, Bob and Carol must establish an equivalent HTLC that enables Carol to receive 0.1 Bitcoin from Bob (3). The contract requests Carol to create another hash H from the random number R and to transfer it to Bob within two days (4). For a transaction between Carol and Alice, Bob transfers to Alice the random number R and demands 0.1 Bitcoin as a compensation (5).

8.3 Research Design

In this study, we apply a problem-centered DSR approach as suggested by Peffers et al. (2007). Typical outcomes of DSR activities are artifacts, which include constructs, models, methods, and instantiations (Hevner et al., 2004). Being experts in the domain of practice-oriented applications of blockchain technology, we have noticed a lack of concepts for the efficient and secure sharing of private broadband capacity based on Wi-Fi sharing. We address this important unsolved problem in a unique and innovative way by developing two novel artifacts. First, we collect various requirements for Wi-Fi sharing and derive a set of design principles for solutions that resolve the weaknesses of current approaches and concepts. Second, we design an integrated reference architecture for Wi-Fi sharing networks, which fosters efficiency and security by combining the blockchain technology with payment channel networks. We demonstrate its applicability by describing how its main components interplay to enable fast and secure transactions between multiple users in shared Wi-Fi networks. With blockchain and payment channel networks, we use and integrate two concepts whose research and application are still at an early stage. We therefore rely on descriptive methods to evaluate the applicability and usefulness of the resulting artifact (Venable et al., 2012, 2016). Consequently, we consider this research as conceptual by nature, yielding far-reaching implications for future research and practice.

We summarize the applied DSR approach as well as complementary methods in Figure 8.2.

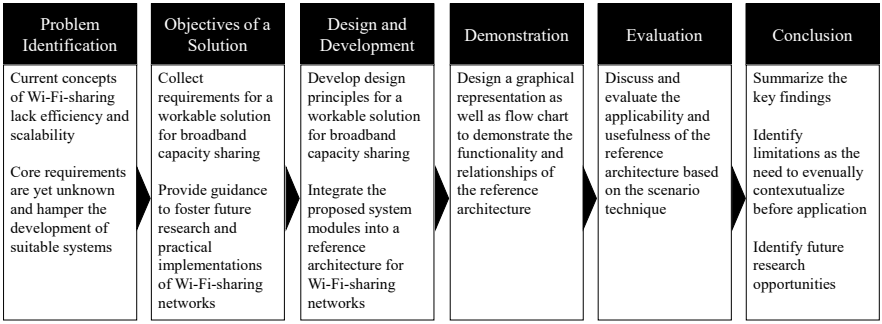


Figure 8.2: Overview of the DSR Approach (based on Peffers et al. (2007))

In following a staged process, which allows for multiple iterations of the design principles to evolve, we aim at developing design principles, which describe a class of systems as a means for implementing Wi-Fi sharing networks. In an initial iteration, we identified and carved out the problem to be solved in discussions with an expert for business process management and an enterprise architect from a large German Internet service provider. During conceptual development, we formulated initial design principles and refined them in an iterative process of discussion and reflection with researchers as well as said business professionals, which resulted in further challenges and perspectives to consider in the next iterations. Our research built upon and benefited from this exchange with industry. We formulated our design principles according to Chandra et al. (2015)'s proposal for effective formulation, including materiality, action, and boundary conditions. We have provided the consolidated results of our research to academic as well as professional experts. We have incorporated the recommendations from academia and have not received any negative feedback from practice.

8.4 Requirements for Wi-Fi Sharing and Current Approaches

8.4.1 Potential Risks and Threats in Wi-Fi sharing Networks

Due to a growing demand for mobile Internet applications, telecommunication infrastructures are at their capacity limit and cannot always deliver high performance during peak hours (Clarke, 2014; Cisco, 2020). Simultaneously, network operators must cope with a growing competition as well as with declining revenues and constantly increasing requirements for network performance and quality (Khan et al., 2011; Seufert et al., 2013). Addressing these challenges requires them to invest into expanding current infrastructure or to identify and implement mechanisms to increase effectiveness. Thereby, both research and practice point to the vast potentials of accessing private landline broadband capacities in Wi-Fi sharing net-

works, which can reduce the overall usage of mobile network infrastructures (Camponovo and Cerutti, 2005; Frangoudis et al., 2011; Cao et al., 2015).

To facilitate user participation and cost-effective operations, Wi-Fi sharing networks require an adequate system architecture that ensures security, efficient accounting, and service quality. In this research, we draw upon Leroy et al. (2011) who reduce the wide range of requirements to the three categories of security, administration, and accounting. We provide an overview of these requirements and specify corresponding risks and threats in Figure 8.3.

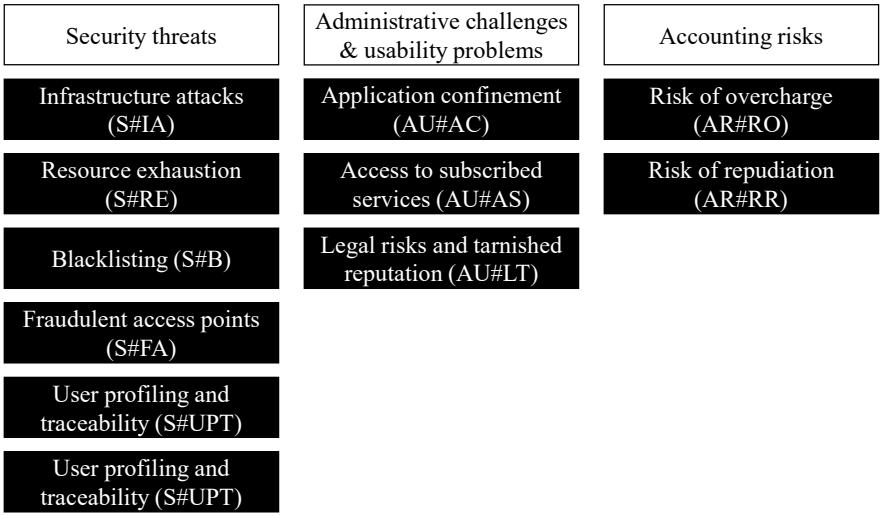


Figure 8.3: Risks and Threats in Current Wi-Fi Sharing Networks (based on Leroy et al. (2011))

Regarding the dimension of security threats, a Wi-Fi sharing network must facilitate cooperative user behavior and sanction fraudulent actions respectively (Cao et al., 2015; Leroy et al., 2011; Sastry et al., 2007). This entails preventing network infrastructure attacks (S#IA) as well as discouraging users from conducting malicious actions using resource exhaustion (S#RE), which can result in access points becoming blacklisted (S#B) by external service providers (Leroy et al., 2011). To further avoid phishing of sensible user data, the architecture must account for the various risks imposed by fraudulent access points through the emulation of fake Service Set Identifier (SSID) (S#FA), which can be used to intercept connections between users and access points (Sastry et al., 2007). Ultimately, Leroy et al. (2011) note that data processing must comply with presently enacted data protection laws, which prohibit various techniques for data analysis and interpretation, such as user profiling and activity tracing (S#UPT).

Administration challenges and usability problems refer to a network's capabilities to support users in achieving quantified objectives with effectiveness, efficiency, and satisfac-

tion. Besides facilitating the solution's perceived ease of use and intuitiveness, the category includes all functionalities, rules, and restrictions that point to application confinement and potentially hamper user adoption (AU#AC) (Leroy et al., 2011). It also regulates the accessibility of subscribed services (AU#AS), which are made available unintentionally through the Internet Protocol of the access point. Ultimately, the category addresses risks imposed by illegal actions of network users, which can yield losses in reputation or even legal implications (AU#LT) (Leroy et al., 2011).

The category of accounting risks incorporates risks that emerge from service downtimes, that is the risk of user repudiation (AR#RR) or failure of service invoicing, in particular the risk of overcharging (AR#RO).

In addition to the lack of non-corruptible invoicing mechanisms, Leroy et al. (2011) describe the absence of a trusted intermediary for a secure and liable payment handling as a major weakness of current Wi-Fi sharing networks. Considerably hampering user participation, this leads to a reduced network coverage and, thus, to decreases in the perceived usefulness of the service. While all categories are important for building functioning Wi-Fi sharing networks, we consider adequate accounting mechanisms as their most essential component, as they facilitate mutual trust and provide users with incentives for participation.

8.4.2 Shortcomings of Current Wi-Fi sharing Networks

In general, we can distinguish between trust-based and security-based approaches. Trust-based approaches are mainly framed by the work of Cao et al. (2015), Seufert et al. (2013), and Lafuente et al. (2011). Besides using intermediaries to facilitate trust among network participants, these approaches typically build upon authentication mechanisms from online social networks (OSN). Having logged in over an OSN, users can use a host's broadband connection by either accessing his or her private network or a designated user network, which has been established for this specific purpose and is regulated by strict policy guidelines (Vidales et al., 2009). We summarize the main properties of trust-based Wi-Fi sharing networks in Figure 8.4.

Cao et al. (2015) develop a Wi-Fi sharing network, which enables users to automatically discover and authenticate nearby networks that are operated by befriended people from social networks. Thereby, users can gain unrestricted access to a host's private broadband connection by proving his or her identity over a relationship that has been established in an OSN. Disclosing a user's identity can not only reduce the risk of malicious actions, but also provide incentives for participating in Wi-Fi sharing communities. Intended to be non-commercial, the service is not subject to risks associated with service invoicing. Based on the findings of Daraghmi and Yuan (2014), we argue that implementing the approach is only feasible and beneficial if access points are also made available to friends of friends and, thus, beyond the scope of direct connections. As sharing private connections with further degrees

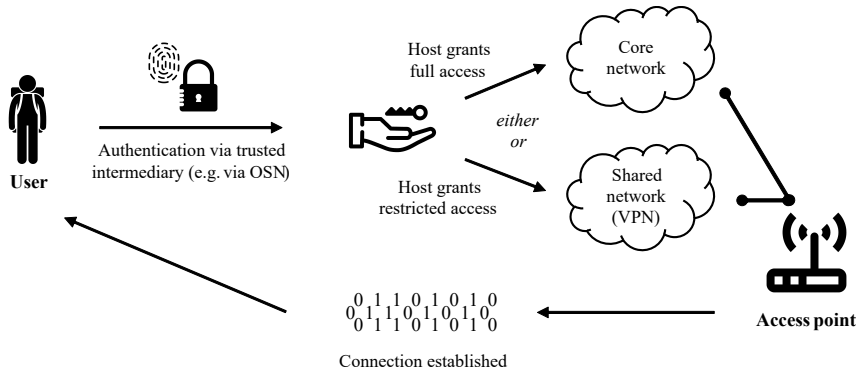


Figure 8.4: Trust-based Approach for Wi-Fi Sharing Networks

of friends can reduce the network's degree of trust, the approach suffers from a trade-off between security and reach.

Seufert et al. (2013) introduce a similar approach. They use OSN primarily as socially aware traffic management systems to authenticate user identities. Users can provide additional information, which is used as meta-data to manage localization and access within the network. The approach supports rewarding or sanctioning user behavior with a trust score, which provides hosts with the opportunity to prevent user groups from accessing their shared network. In general, users can only access a network upon request and hosts manually decide whether to share an access point or not. Nevertheless, the approach allows authorized users to gain access over a separately managed virtual private network (VPN), which is established and ran independent from the private network's infrastructure. Controlled by strict policy guidelines and separated from the network, non-authorized users can access the network over virtual access points. This discourages users from the unauthorized use of the host's subscribed services and supports hosts in preventing infrastructure attacks, resource overloads, and service backlisting. The approach is a non-commercial service and fosters user participation.

Lafuente et al. (2011) propose a service for Wi-Fi password sharing, which enables authorized users to access a shared network directly. To ensure data security, it requires hosts to approve all incoming connection requests manually. Communication and data transfer between user and host are further secured by encryption mechanisms, which prevent attacks that seek to obtain sensible user information (Lafuente et al., 2011). The authors further draw upon the concept of computational trust management Trček (2018) to ensure that passwords are only shared among trusted users. Although the proposed approach cannot fully prevent malicious actions, it facilitates cooperative user behavior.

In summary, most trust-based approaches lack mechanisms to prevent malicious actions of non-cooperative users. By failing to address the requirements from Figure 8.3, however,

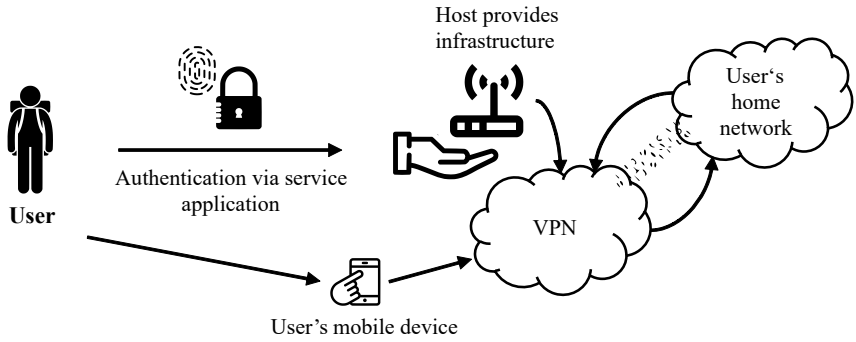


Figure 8.5: Security-based Approach for Wi-Fi Sharing Networks

these approaches pose manifold risks for hosts and users. This applies especially for the case of unsecured connections, which provide users with unrestricted access to private networks and all subscribed services. All concepts use authentication mechanisms from OSN to verify user identities. While this ensures trustable connections in many cases, it is not applicable when dealing with fake profiles that have been created to bypass such security barriers. Although Seufert et al. (2013) seek to address this issue by computing a user-specific trust score, their approach only yields adequate results if all users in a network have been identified by a trusted intermediary. Consequently, the feasibility of trust-based approaches relies strongly on the availability of intermediaries.

By contrast, security-based approaches use a host's infrastructure as an access point, over which a user establishes a VPN connection to its own private network. Secured by cryptography, these connections resolve host-sided security concerns and provide user with a full Internet access that is not restricted by external policy guidelines. We summarize the main properties of security-based Wi-Fi sharing networks in Figure 8.5.

Sastry et al. (2007) introduce a Wi-Fi sharing network that builds upon VPNs to establish Internet connections for trusted users within a network. This entails that users can use a host's access point to connect to their own private network, which then processes the session's entire Internet traffic. Besides yielding increased network latencies as well as broadband restrictions of 200 kbps in the case of asymmetric connections (Lakshminarayanan and Padmanabhan, 2003), the proposed concept can fully resolve latent trust dependencies between involved parties. As users gain access to the Internet over their own network, it can further overcome common security issues and usability restrictions. By using cryptography to encode communication and data transfer, users also benefit from higher security and trust. However, Sastry et al. (2007) neglect the risks imposed by fraudulent access points and build upon the assumption of generally cooperative network participants. Furthermore, the authors primarily sketch out the approach's applicability for scenarios that entail a lin-

ear increasing resource consumption for communication encoding, which is due to network latencies in long distance connections.

Leroy et al. (2011) augment the approach of Sastry et al. (2007) by using VPNs to establish encoded connections between a host's and a user's access point. The authors implement Roaming Authentication and Key Exchange (RAKE) for identity authentication. Furthermore, RAKE accounts for establishing and organizing the connection and determines explicit parameters necessary for authentication and encryption. By employing a lightweight accounting protocol similar to the Transmission Control Protocol (TCP) slow-start approaches, the network can dynamically manage shared bandwidths and close connections in the case of fraudulent behavior to reduce financial impacts.

While security-based approaches provide the means to address requirements related to security and administration, they lack adequate solutions for accounting. Although the approach of Leroy et al. (2011) supports hosts in minimizing monetary impacts, which can result from the early closure of a connection due to a user's fraudulent behavior, it builds upon TCP and, thus, entails significant performance reductions in fast scaling networks. Furthermore, it requires the protocol's implementation on all communicating routers to ensure the reliability and security of transferred data. Against the users' preferences for high mobility and flexibility, the protocol lacks efficiency and responsiveness especially in the case of high round-trip times. As the round-trip times and, thus, the transfer of data between a host and a user's private network can take up to several seconds, moving users that rapidly establish and close connections to hotspots are quickly out of the network's reach. Although these mechanisms are considered typically to provide suitable means for establishing and securing bilateral communication channels between hosts and users, they hardly conform to the requirements of Wi-Fi sharing networks, which require multi-channel-based communication opportunities.

To date, research has only paid limited attention to using the capabilities of blockchain technology for Wi-Fi sharing. Shi et al. (2017) suggest implementing smart contracts to establish a system capable of processing micro-transactions as payments for used capacities in Wi-Fi sharing networks. The authors motivate their approach by noticing that most data-sharing services lack user participation due to insufficient incentives. By drawing upon Leroy et al. (2011), they develop an accounting mechanism that uses a protocol that rewards cooperative users with a linearly growing bandwidth. The concept further enables hosts to terminate connections with non-cooperative transaction partners, which results in a complete loss of the transaction's content. Implementing the approach requires small adaptations to the hosts' access points as well as the installation of java-based application on the user's device. Thereby, the protocol establishes a connection to a blockchain network (e.g., the Bitcoin network) and uses the corresponding infrastructure to conduct micro-transactions. Consequently, Shi et al. (2017) demonstrate the potentials of using blockchain technology for conducting micro-transactions in Wi-Fi sharing networks. Significant benefits arise from the implementation of payment channels to clear fine-grained data services incrementally, as corresponding accounting protocols neither require the existence of a trusted intermediary,

nor demand the use of complex consensus mechanisms for transaction approval. In order to use smart contracts to conduct micro-transactions instantaneously, the Wi-Fi sharing network must register each contract stored within the blockchain. Thus, the entire payment logic is stored within the smart contract itself and executed on a local connection between users and hosts.

Table 8.1 summarizes these approaches and links them to the respective risks and threats of Figure 8.3. Furthermore, it clarifies that trust-based approaches are excellently suited to minimize or address security-relevant and administrative problems. Due to their low flexibility toward scalability, however, respective approaches do not meet the demands of a highly available and widely accessible solution, which is a prerequisite for the viability of Wi-Fi sharing networks in practice. Security-based approaches, on the other hand, do not have the usual limitations of scalability that result from the lack of reliable authentication mechanisms. Due to their single-channel-based communication semantics, however, hopping to and from another router, as is required in Wi-Fi sharing to not being bent to the local range of a particular terminal, is not efficient. Consequently, users encounter interruptions or extended waiting times when physically moving forward.

As explicated in Table 8.1, the current state-of-the-art addresses most requirements linked to the first two categories. In fact, the use of VPNs can increase security and facilitate cooperative user behavior. Despite constraints regarding their resource consumption and limited performance, corresponding approaches yield multiple benefits, as users connect to their own private network and, thus, do not face accessibility restrictions or risks imposed by data security. Thus, users cannot only hide their browsing habits but also eliminate the possibility of being tracked or profiled by third-party providers.

Table 8.1: Summary of Requirement Coverage in Current Solutions for Wi-Fi Sharing Networks[17]

Reference	Cao et al. (2015)	Seufert et al. (2013)	Lafuente et al. (2011)	Sastry et al. (2007)	Leroy et al. (2011)	Shi et al. (2017)	Our Approach
S#IA	O	O	O	X	X	/	X
S#RE	O	X	O	X	X	/	X
S#B	O	X	O	X	X	/	X
S#FA	O	O	/	/	/	/	O
S#UPT	/	/	/	X	X	/	X
AU#AC	/	/	/	X	X	/	X
AU#AS	/	X	/	X	X	/	X
AU#LT	O	X	/	X	X	/	X
AR#RO	/	/	/	/	X	X	X
AR#RR	/	/	/	/	X	X	X
Approach	Trust-based	Trust-based	Trust-based	Security-based	Security-based	Blockchain-enhanced	Blockchain-enhanced

8.5 Design Principles for Secure and Reliable Wi-Fi Sharing Networks

Addressing the abovementioned requirements, our approach inheres the benefits of Leroy et al. (2011)'s work by implementing a security-based approach, while bypassing its scalability issues by transferring communication and session management to a blockchain-enabled authentication mechanism as initially proposed by Shi et al. (2017). We further increase the flexibility of payment processing, which Shi et al. (2017) have identified as still problematic, by implementing payment channel networks for micro-payments. As there is already robust research on how to set up and implement security-based Wi-Fi sharing networks, our referenced solution focuses on providing a scalable mechanism that enables intermediary-free payment processing. Based on relying on the effectiveness of blockchain-enabled payment channel networks, we claim our approach to superior to current implementations.

We derive four central design principles for secure and reliable Wi-Fi sharing networks based on the requirements derived from our survey of risks, threats, and related work:

DP1: Provide the system with a module for hosts to manage and organize the provided bandwidth in order for the system to provide access to the Internet.
DP2: Provide the system with a module for users to initiate and maintain a private network without sharing secret keys in order for the system to prevent decoding the connection.
DP3: Provide the system with a module to provide only bandwidth to the user while users are routed to their private network even if the user's identity is known or has been approved by identity authentication mechanisms in order for the system to prevent users from conducting fraudulent actions.
DP4: Provide the system with a module to restrict user access to the host's private network infrastructure even if the user's identity is known or has been approved by identity authentication mechanisms in order for the system to prevent users from conducting fraudulent actions.
DP5: Provide the system with a module to identify access points clearly in order for the system to prevent security-related threats, such as eavesdropping users' traffic or DNS server phishing.

However, purely security-based concepts lack the means to cope with the requirements of the accounting category sufficiently. Due to their inability to implement immutable and non-corruptible payment protocols, corresponding concepts lack feasibility and user participation, as services provided by participants cannot be accurately billed or compensated (Leroy et al., 2011; Shi et al., 2017). This can sustainably reduce the usefulness of the entire Wi-Fi sharing network. Leroy et al. (2011) further notice that asymmetric communication channels cannot

[17]X: addressed directly by the approach used; O: addressed indirectly by the approach used; /: Not addressed by the approach used.

guarantee fairness regarding the billing of services without including a trustable intermediary or implementing expensive hardware for using complex consensus mechanisms. In addition to the three core elements of working Wi-Fi sharing networks, we can derive eight further design principles for the implementation of adequate accounting mechanisms:

DP6: Provide the system with a mutually trusted intermediary requiring a transaction history, which facilitates transparency by recording a user's behavior in terms of data traffic, resource consumption, and incurred costs, in order for the system to ensure that connection data cannot be manipulated or corrupted.

DP7: Provide the system with a module to keep transaction costs to a minimum in order for the system to prevent large numbers of micro-payments.

DP8: Provide the system with a module to forgo transaction costs for the execution of instant payments in order for the system to regulate the duration of the connection.

DP9: Provide the system with a module to set up the transaction in order for the system to enable host and user to mutually agree on the usage cost.

DP10: Provide the system with a module for pre-payment in order for the system to increase the quality of service and to prevent both risks imposed by overcharging and repudiation.

DP11: Provide the system with accounting mechanisms using a dynamic trust-score in order for the system to facilitate cooperative host behavior and ensure high service levels in terms of infrastructure accessibility by rewarding hosts for cooperative behavior or high availability.

DP12: Provide the system with a protocol that is platform-independent in order for the system to avoid potential lock-in effects and facilitate user adoption and scalability.

DP13: Provide the system with a protocol, which incrementally increases the provided bandwidth, and with instant payment functionalities, which ensure that outstanding payments are transferred immediately to unlock further resources in order for the system to prevent fraudulent behavior.

DP14: Provide the system with a protocol for users to initiate any number of connections and for hosts to simultaneously bill users with multiple connections in order for the system to ensure connections without the risk of overcharging or repudiation.

The collective boundary condition for all twelve design principles is "given that it shall be used to design secure and reliable Wi-Fi sharing networks".

Drawing upon the results of Shi et al. (2017), current payment mechanisms can benefit by using the capabilities of the blockchain technology for invoicing in Wi-Fi sharing networks. Among others, the main advantages of such solutions include a decentral and non-corruptible database, which supports identity authentication, distributed transactions, and the generation of user protocols without the existence of a trusted intermediary. However, as their approach does not provide adequate accounting mechanisms, they lack the means to address the design principles DP1, DP4, DP6, and DP10. In the following, we introduce a reference architecture, which builds upon our twelve design principles and improves current Wi-Fi sharing concepts,

by addressing the requirements of security, administrability, and usability with adequate accounting capabilities.

8.6 A Reference Architecture Framework for Wi-Fi Sharing Networks Using Blockchain Technology and Payment Channel Networks

8.6.1 Multi-layer System Architecture

Subsequently, we design and develop the main artifact of this research: a multi-layer reference architecture for Wi-Fi sharing networks. For architecture development, we draw upon the work of Notheisen et al. (2017), who have introduced a market engineering framework for blockchain solutions. The system architecture comprises the four layers of agent, application, infrastructure, and environment.

The agent layer includes hosts and users participating in the network. Due to the large amount of different user devices and Wi-Fi hotspot hardware, the platform must be independent from specific systems or providers. While this is feasible for mobile devices, wireless routers often use proprietary technology, which is not accessible to third-party providers. This requires developing a generally accepted and compatible software solution.

The application layer manages all connections within a network and addresses many of the previously defined design principles. Hence, it complies with the various security requirements by implementing a so-called demilitarized zone to communicate with user devices, which protects the host's private network from malicious attacks. Furthermore, users establish VPNs to connect to their own private network, which increases data security and facilitates service accessibility. Blockchain technology provides a secure, trustable, and immutable solution for conducting transactions without the need for a mutually trusted intermediary. Additionally, payment channels ensure instant transactions at neglectable transaction costs. By prepaying for only a small timeslot, sunk cost as a results of unforeseen channel terminations or malicious user behavior can be kept to a minimum (Leroy et al., 2011). Monetary incentives further facilitate the provision of broadband capacities and very low expected payoffs prevent fraudulent behavior. Due to the implementation of payment channel networks, corresponding Wi-Fi sharing networks are highly scalable and allow a large number of participants to join and interact. The additional network overhead for the payment channel transactions is negligible. Since the size of a transaction is only a few kilobytes, payments can be sent every few seconds without noticeable impact on network performance, ensuring near real-time payments.

The infrastructure layer comprises a protocol layer and a hardware layer. Requirements on these layers are neglectable from a conceptual perspective and mostly addressed by the implementation of payment channel networks. Corresponding protocols must be capable of

accessing and interpreting smart contracts to conduct transactions on the blockchain. This is independent from underlying consensus mechanisms or cryptography concepts as long as the network allows for the secure processing of opening and closing operations. However, the transaction cost for opening and closing the channels can be dependent on the chosen blockchain architecture and consensus mechanism. While a smaller network structure such as a consortium blockchain can allow for more efficient consensus mechanisms such as PBFT (Hofmann, 2020), larger, public blockchains provide a larger user base that can participate in the payment channel network (Mercan et al., 2021). We argue that ensuring a successful routing of payments through the network outweighs the need for low transaction costs for funding transactions. Additionally, some types of payment channel networks, such as the Poon-Dryja payment channels, can be implemented only on public blockchains (Erdin et al., 2021). For public blockchains, we argue that the network size is more deciding than the underlying consensus mechanism, therefore, the choice between PoS and PoW is solely dependent on the choice of public blockchain. If a choice has to be made between two equivalent PoW and PoS networks, it can be argued, that the higher security that PoW provides is not significant for the comparatively small transaction values, therefore, a PoS network should be preferred. Reinforcing this recommendation, there is a trend towards integrating PoS in the major public blockchains such as Ethereum partly fueled by environmental concerns.

Although the network's context cannot be controlled, we account for its requirements with an environment layer, as data security laws or regulations determine the boundaries within in which a Wi-Fi sharing network can operate. We summarize the resulting framework in Figure 8.6.

8.6.2 Demonstration of Transactions on the Architecture

We demonstrate the reference framework's structure as well as its functionality using the sequence diagram in Figure 8.7.

To connect with a network, users must send a request for approval and transfer the required funding transaction to the payment channel network (create channels). Prior to saving the transaction on the blockchain, the network creates an identical refund transaction, which guarantees that users are reimbursed if they are affected by non-cooperative behavior conducted over the connected channel. The channel has been successfully established as soon as the funding transaction is recorded and approved by the network (send funding transaction). However, this requires saving the transaction data on the blockchain, which typically goes along with considerable latencies and processing times (save transaction to new block). Hence, the funding transaction should be conducted with enough time before a user intends to use a hotspot. To support a smooth processing of registration and approval requests, corresponding software solutions should directly connect with a service that provides users with the opportunity to buy or sell blockchain-based tokens with regular fiat money.

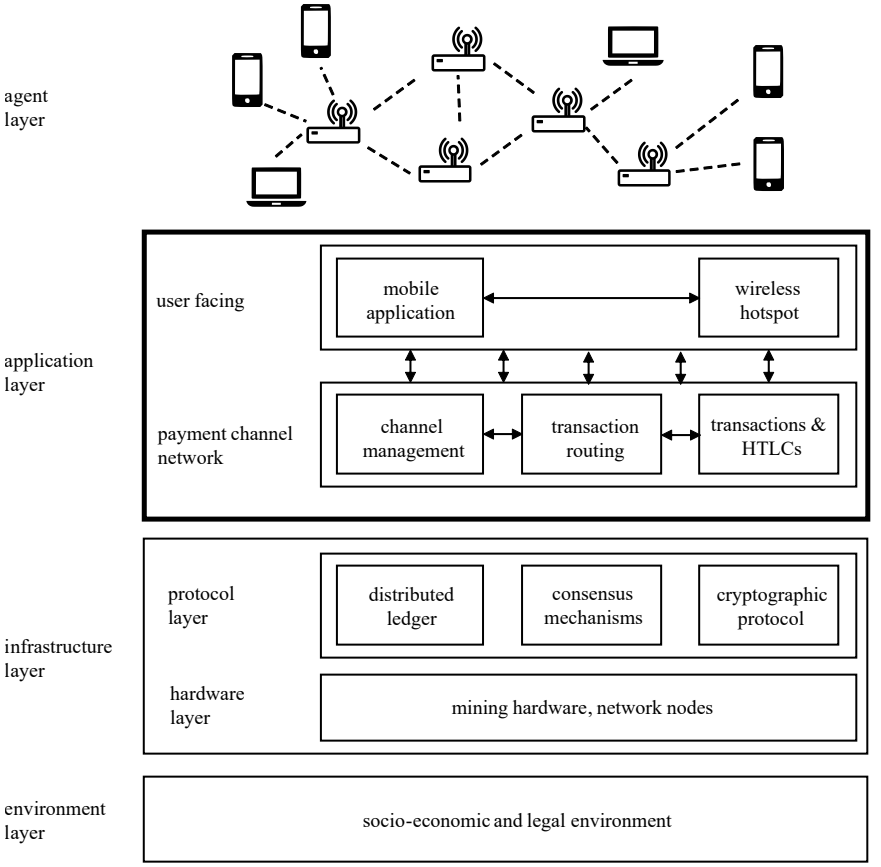


Figure 8.6: Reference Architecture Framework

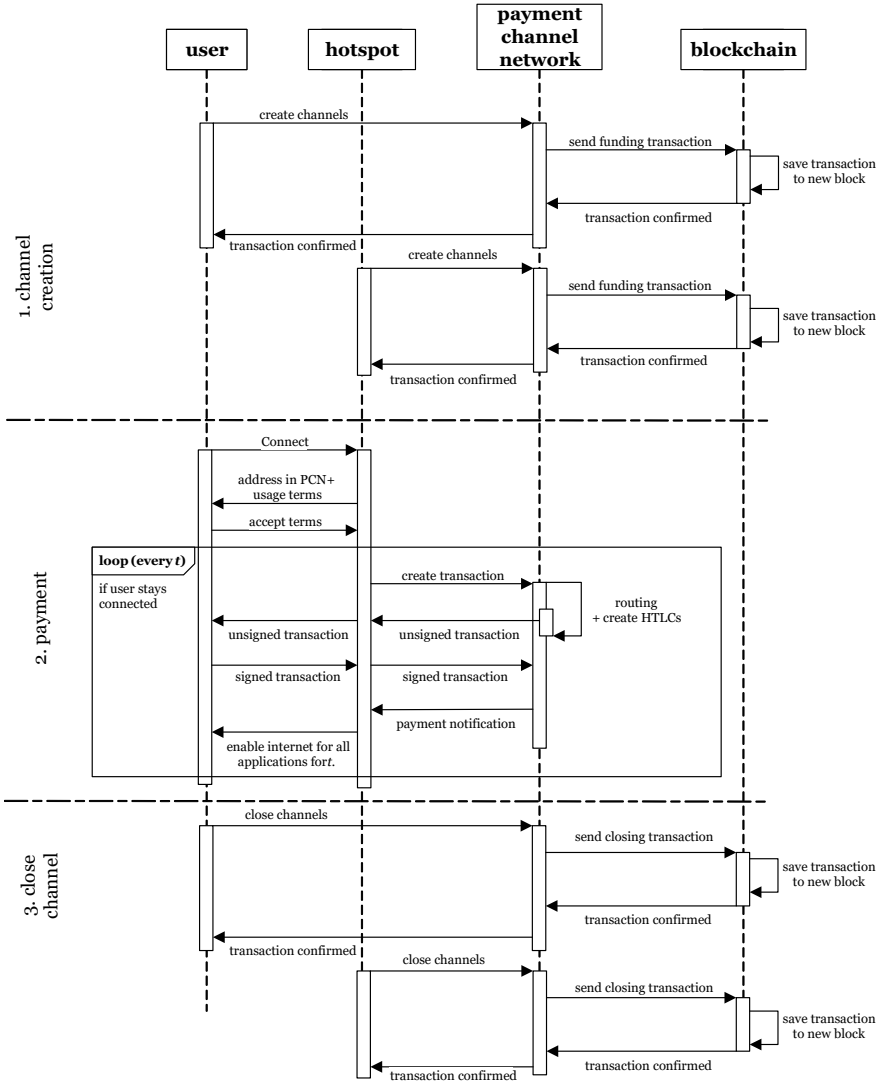


Figure 8.7: Overview of Architecture Functionality

Furthermore, hotspots must be assigned with a unique identifier that enables users to find them in a payment channel network. However, as hosts only receive but do not send payments, they are not required to conduct a funding transaction. In general, a network's stability, coverage, and performance correlates strongly to the amount of access points it comprises. Because it is often necessary that hosts act as intermediaries between two participants that are not directly connected, they should also be incentivized to conduct a small funding transaction to route payments through the network.

To access a host's private network, a user must ensure an active Wi-Fi connection. An access portal initially blocks most Web services and thus, only authorizes connections to the respective payment channel node provided by the router. Thereby, the user's mobile device and the host's router automatically negotiate terms of use, including minute-based fees for accessing a hotspot, the address of the payment channel, and the bandwidth provided by the network (address in PCN + usage terms accept terms). After their mutual agreement, the mobile device must approve the transaction's content and send the usage fee to the router (cf. loop). Thus, the node running on the Wi-Fi access point manages the routing of the transferred tokens and activates all services for access by the user. To use the payment channel, the user conducts an additional micro-transaction before the connection interrupts. This process repeats until no more micro-transactions are registered by the host's access points, which leads to the automated termination of the payment channel. Thereby, the user carries the risk of not receiving a compensation for the last completed transaction. As the value of these micro-transactions is neglectable, it is neither feasible nor economical to set up malicious hotspots with the purpose of taking advantage of these on-sided revenues. However, this may not apply to situations, in which malicious hotspots are used for profit generation by locating them at highly frequented places, such as for large events or at city sights. Consequently, the network must continuously collect and analyze data on the behavior of access points and sanction fraudulent actions, for example by blacklisting corresponding hosts. By contrast, if users conduct illegal actions or show fraudulent behavior, the host can terminate the connection at any time.

If a payment channel expires or both participants agree upon closing it (close channels), the last transaction is submitted to and saved on the blockchain (send closing transaction). Subsequently, hosts should immediately create new payment channels with either no funding or a small funding to contribute to the network's stability by routing transactions between unconnected participants. Tokens rewarded for sharing a private broadband connection should be periodically paid out as fiat money to minimize the risks imposed by fluctuating exchange rates. Further, to account for fluctuating exchange rates of cryptocurrencies, the reward could use a pegged exchange rate to one or multiple major currencies.

8.7 Evaluation

8.7.1 Scenario-based Evaluation

Evaluation is a central and essential activity in conducting rigorous DSR. Venable et al. (2016) note that without evaluation, DSR yields only unsubstantiated design theories or hypotheses. Peffers et al. (2007) divide the evaluation task into the activities of demonstration and evaluation. Thereby, demonstration proves that an artifact feasibly works to solve one or more instances of a problem. An evaluation then presents how well an artifact supports a solution in a formal and extensive way. Thereby, one can generally choose from multiple techniques, including observational methods (e.g. case studies or field studies), analytical methods (e.g. static analysis or optimization), experimental methods (e.g. controlled experiments or simulations), testing methods (e.g. functional testing and structural testing), and descriptive methods (e.g. informed argument and scenarios) (Hevner et al., 2004; Venable et al., 2012).

We have already demonstrated the capabilities of our reference architecture for conducting transactions in blockchain-based Wi-Fi sharing networks by describing its main components as well as their relationships in the sequence diagram in Figure 8.7. In the following, we evaluate the developed architecture by drawing upon Venable et al. (2016). We do so to clarify its usefulness, to control for undesirable consequences, and to identify existing improvement potentials. Due to the novelty of blockchain technology as well as of Wi-Fi sharing networks, we decided to evaluate our architecture based on a realistic scenario, not on a workable implementation. Thereby, we evaluate the proposed architecture with a buyer-sided focus (i.e., the user or guest of networks).

We define our evaluation scenario in the context of smart tourism (Gretzel et al., 2015), in particular the travel abroad for the purposes of vacation or business. While the demand for Internet availability is generally growing (Sastry et al., 2007), travelers face excessive costs for data access within foreign mobile communication networks. Besides yielding several benefits, including cost reductions, performance increases, and unrestricted data usage, shared-Wi-Fi networks also enable travelers to access important Web services, such as instant messaging or Web-based navigation. Hence, they can contribute to more convenient traveling by providing the means to access necessary information for various purposes. Similar requirements arise from scenarios in a traveler's home country, for example, when he or she visits an indoor location with poor mobile network coverage or an event location network congestion. Despite advances such as 5G networks, it is often more cost effective to set up Wi-Fi hotspots, which could be made available to the traveler using our architecture.

The architecture proposed in this study is fully capable of addressing the requirements of the introduced scenarios. Travelers typically require an ad-hoc information flow to find shortest routes, plan activities, and react to unforeseen events or to pass time while waiting. By using payment channel networks, only opening and closing a transaction is committed to and saved on the blockchain. After establishing the connection, this facilitates travelers to

gain Internet access quickly and to find necessary information without delays. Furthermore, transaction costs only incur when opening or closing a payment channel. Hence, travelers face neglectable costs for most transactions, which fosters their willingness to participate and use network infrastructures.

Because hotspots can not only establish bilateral connections with single travelers but also serve as intermediaries for participants that are not directly connected, the architecture ensures high connectivity, stability, and coverage. Consequently, the architecture enables travelers to connect to the Wi-Fi sharing network, even if they frequently change locations. Ultimately, travelers are typically cautious when using services in foreign countries or unknown locations. Besides transaction and data security, they demand trust-building mechanisms that curtail fraudulent behavior.

Our artifact addresses these requirements twofold. First, it draws upon security-based approaches and enables users to establish connections to their own private network. This ensures that communication and data traffic is routed over their own infrastructure and that sensitive data cannot be captured. Second, as the architecture provides mechanisms for incremental and simultaneous invoicing, fraudulent behavior leads to the instant termination of payment channels and travelers face little risks imposed by prepaying considerable amounts without receiving services as a compensation. Cf. also Table 8.2 for a summary of our argument.

Despite these benefits, evaluating the scenario also revealed shortcomings, which should be addressed by future research. First, opening payment channels requires submitting a transaction to the blockchain. As this can not only take up considerable time but also requires an active Internet connection, it seems problematic for travelers abroad. However, there are ways to mitigate these shortcomings: While on older blockchains such as Bitcoin, opening a payment channel and, therefore, joining a payment channel network takes around 20 minutes (Poon and Dryja, 2016), newer blockchain protocols offer much faster transaction times that only take a few seconds to be finalized. Additionally, the host network could provide a gateway to the blockchain network for every user that only serves the purpose to open and close payment channels. This would still mean that the user's blockchain wallet should always have sufficient funds to open a payment channel. The process of buying the needed cryptocurrency for fiat money in advance is, therefore, still an open issue, that must be solved until cryptocurrencies experience widespread adoption. Furthermore, many scattered mobile devices that simultaneously use network infrastructures can produce a significant overhead due to the creation of VPNs and, thus, reduce network performance as well as available bandwidth. While the increase in package size due to VPN overhead is only about two percent, the overhead does not affect the host network. However, the computational overhead for encrypting and decrypting traffic can negatively affect the performance for the end user, especially if he or she has multiple devices connected to the home network via VPN (Berger, 2006). Ultimately, network usage depends strongly on available payment methods. As the number of available currencies is constantly growing, determining a single payment

Table 8.2: Summary of Artifact Evaluation

Risk or Threat	Wi-Fi Sharing Network Using Blockchain and Payment Channel Networks
S#IA	Infrastructure attacks harm the user him- or herself, as he or she is forwarded directly to his or her own private network via VPN. There is no access to the host's network at any time. *Addressed by DP3 and DP4.*
S#RE	The host provides only a limited range of its bandwidth over which the user connects directly to his own network. Any exploitation of resources would therefore be at the user's own expense. An open technology stack can assist further in providing a scalable environment. *Addressed by DP1, DP2, DP3, DP12, and DP13.*
S#B	Similar to S#RE, blacklisting would be at the user's disadvantage, as he or she accesses the Internet via his or her own private connection and, thus, the user's (public) IP assigned by his or her Internet service provider. *Addressed by DP1, DP2, DP3 and DP4.*
S#FA	Since our reference architecture benefits from the immutability of blockchain networks, each access point has a unique, non-falsifiable ID associated with a dynamic trust score that ensures that the user only connects with trusted access points. We acknowledge that a further layer for trusted ID to prevent fraudulent actors from generating new IDs may be necessary to fully implement this solution. However, since first solutions for this problem such as, blockchain-based know-your-customer are getting implemented (Singhal et al., 2020), we have marked it as partially addressed in Table 8.1. *Addressed by DP5,DP6, and DP11.*
S#UPT	As with S#IA, S#RE, and S#B, user profiling and traceability (S#UPT) also benefits from the strict separation of the host's private network and the user's accessed private network. This way, neither the host can intercept the connected user's data or connection protocols, nor vice versa. Additionally, the usage of payment channel networks allows for better transaction privacy compared to on-chain transactions (Malavolta et al., 2017; Erdin et al., 2021). *Addressed by DP1, DP2, DP3 and DP4.*
AU#AC	The only application confinement the user might experience can occur due to limitations of local resources. Exemplarily, the user's hardware might run out of power, or the host might power off the only connected access point resulting in connection failures. Further, the user is likely to not have full bandwidth of his or her private Internet access, as he or she is restricted to the host's (shared) bandwidth. However, assuming high participation, at least in urban areas, the latter two limitations will fade as the Wi-Fi sharing networks are capable to establish multiple simultaneous connections. *Addressed by DP3, DP7, DP11, and DP12.*
AU#AS	The user will not experience any restrictions to his or her privately subscribed services. This is since the user connects via VPN to his private network and, thus, each resource the user accesses – including websites, infrastructure, or services – will treat the user as he were accessing from his home network. *Addressed by DP3.*
AU#LT	Hosts sharing his or her Internet access do not have to fear any risks in regard of legal infringements or tarnished reputation due to the user's misbehavior. Again, due to the VPN, any violation is committed directly by the user's network. *Addressed by DP1, DP2, DP3 and DP4.*
AR#RO	The user carries the risk of not receiving compensation for the last completed transaction. However, due to the instant initiation of payment channels as well as the low value of each micro-transaction, this cost is neglectable. As a consequence, malicious hotspots will not get economic benefits by misbehaving or trying to take advantage of on-sided revenues. Finally, we include a trust score that penalizes any detected misbehavior and, thus, clears the network from fraudulent access points. *Addressed by DP6, DP7, DP8, DP9, DP11 and DP13.*
AR#RR	Users and hosts do not have any risk of repudiation as we outsource any payment processing to payment channel networks operating on an immutable blockchain. Thus, there is no risk for users and hosts alike of paying too much or receiving less, respectively. *Addressed by DP5, DP9, DP10, DP13 and DP14.*

method can hamper user adoption as well as their participation willingness. This is mostly due to the need of maintaining multiple wallets on different platforms, which would increase management and transaction cost.

8.7.2 Assessment of Design Principle Expressiveness

Due to the nascent nature of our research and the absence of an instantiation, we employed the evaluation of our design principles (Venable et al., 2016) summative by conducting a workshop with experts in blockchain applications. In the workshop, we evaluated our design principles by employing Janiesch et al. (2020)'s assessment of design principle expressiveness based on Recker et al. (2011)'s test of ontological expressiveness.

That is, we discussed with the participants whether our design principles are free of principle deficit, principle redundancy, principle overload, and principle excess. In doing so, we tested whether "we do not miss principles to describe real-world phenomena, we do not provide more principles than required for a single phenomenon, we do not provide principles that can be used to describe more than one phenomenon, and we do not provide principles that are not relevant to describe phenomena" (Janiesch et al., 2020).

The workshop was held online with four participants from three organizations and lasted for more than an hour. We explained the assessment of design principle expressiveness and presented an iteration prior to the final design principles that we described in Section 8.4. Further, we detailed the architecture framework, before we discussed the design principles' expressiveness in light of the architecture framework. In this prior iteration, DP1 and DP2 were only recorded as DP1 and DP9 did not yet exist. All other DP remained the same except for minor wording changes. Table 8.3 summarizes the participants of the workshop.

Table 8.3: Workshop Participants

#	Role	Company Sector	Company Size
1	Senior Consultant R&D	Software Development	Small and medium-sized enterprise
2	Product Manager	Software Development	Small and medium-sized enterprise
3	Junior Software Developer	Software Development	Small and medium-sized enterprise
4	Researcher	Education & Research	Public research university

All participants are experts in the field of blockchain-based applications and are knowledgeable in software engineering. They were given a handout prior to the workshop with an excerpt of the paper. In the workshop, we explained the concept of design principles and design principle expressiveness before we detailed the actual design principles and discussed the architecture framework. All participants confirmed to have understood the concept of design principles and evaluating design principle expressiveness in terms of principle deficit, principle redundancy, principle overload, and principle excess.

Overall, the participants confirmed the design principles' expressiveness. In the discussion a few aspects emerged that required clarification. Most of those were related to the inner workings of payment channel networks and blockchains and, thus, unrelated to our design

principles. These issues could be clarified by providing further information about a suitable instantiation of the principles as proposed in Section 6.

Some comments related to the clarity of design principles. In particular, the participants agreed that the first design principle suffered from mild overload as a user and a host perspective was combined into one design principle. To improve clarity, we have split this design principle into DP1 and DP2 to reflect both perspectives even though the phenomenon for both design principles could be argued to be secure system access. Additionally, we used more precise wording for some design principles. Further, participant #1 noted that a dynamic trust score is not necessary for every role (DP11). We acknowledged the impreciseness and now refer to hosts rather than users. We checked the remaining design principles for inconsistent or ambiguous wording. In addition, participant #4 pointed out that "a user and a host need to explicitly agree on a cost structure to avoid overcharging". While this is implicitly available through payment channel networks, it may not be for other instantiations. Hence, we have included this as DP9. Lastly, participant #1 pointed out that DP12 could be considered excess or at least optional from a pure technical perspective. After careful consideration and discussion with the participants, we decided to retain the design principle due to its socio-technical importance for user adoption and acceptance.

8.7.3 Testable Propositions and Key Performance Indicators

Since our evaluation using the scenario technique was descriptive and thus of artificial summative nature, in the following we propose testable propositions to evaluate our artifact using either observational or experimental methods for a socio-technical evaluation and analytical or test methods for the technical evaluation (Hevner et al., 2004; Venable et al., 2012). This will enable a naturalistic evaluation of human risk and effectiveness (Venable et al., 2016).

Concerning the socio-technical aspects, we propose to perform a lab experiment and possibly at a later stage a field experiment to evaluate user satisfaction with our artifact as we expect that an instantiation of our artifact (i.e. the reference architecture) will result in better satisfaction of both consumers and providers of the Wi-Fi sharing network. We expect the results to be more significant when using mobile Internet services abroad. That is, the independent variable is the software support of the building process of a service platform. Thus, for further evaluation, we propose the following testable propositions:

P1: The use of the IT artifact that supports both, adequate accounting mechanisms as well as adequate security and performance, will result (a) in an improved user satisfaction of consumers using mobile broadband services and, thus, (b) in better user satisfaction of Wi-Fi sharing providers than using an IT artifact that only supports adequate accounting mechanisms.
Analogously:

P2: The use of the IT artifact that supports both, adequate accounting mechanisms as well as adequate security and performance, will result (a) in an improved user satisfaction of consumers using mobile broadband services and, thus, (b) in better user satisfaction of Wi-Fi sharing providers than using an IT artifact that only supports adequate security and performance.
As a baseline, we deem it necessary to test the following propositions as regards comparisons with approaches without any IT support as well:
P3: The use of the IT artifact that supports both, adequate accounting mechanisms as well as adequate security and performance, will result (a) in an improved user satisfaction of consumers using mobile broadband services and, thus, (b) in better user satisfaction of Wi-Fi sharing providers than using no Wi-Fi sharing.

One way to design an experiment for testing these propositions is to use a 2x2 factorial design along the dimensions of accounting and security with four groups of subjects, which will be in the following four treatments: (a) no Wi-Fi sharing, (b) an IT artifact that supports adequate accounting mechanisms, (c) an IT artifact that supports better security and performance, and (d) an IT artifact that supports both, adequate accounting mechanisms as well as adequate security and performance.

Concerning technical evaluation aspects, we propose to use analytical methods and test cases to measure the performance of our artifact to substantiate that its speed and security is at least on par with the state-of-the-art. Therefore, we propose a set of two primary indicators that can be measured: the transaction cost and the connection throughput. There are previous studies that examine both indicators. However, they are restricted to subsets of the proposed functionality. For example, there are studies on network and computational overhead for VPN connections (Berger, 2006), and studies for network and computational overhead for payment channel networks (Sivaraman et al., 2020).

To technically evaluate our architecture, we propose a cost model that combines these two costs. For the payment channel cost, we include the overhead for routing the payment through the network, which increases for the number of users in the network. However, the probability to route a payment successfully through the network increases for a larger number of users. If there is no way to route payments directly from the user of the Wi-Fi sharing network, an additional channel has to be created, which is associated with transaction cost. See Table 8.4 and Equation 8.1 for the operationalization.

Table 8.4: Cost Indicators for Evaluation

Symbol	Description
U	Number of users of the payment channel network
$C_{routing}(U)$	Cost of calculating the route through the payment channel network
$P_{routing}(U)$	Probability of finding a route between client and network operator
$C_{channel}(U)$	Cost of creating a new channel
$C_{VPN}(U)$	Cost associated with VPN overhead
$C_{total}(U)$	Total cost of using the network

$$C_{total} = (1 - P_{routing})(U) \times C_{channel} + C_{routing}(U) + C_{VPN} \tag{8.1}$$

Calculating the cost per throughput and comparing it with existing systems provides another means to judge the efficacy of the system. However, it must be put in relation with the testable proposition above as the security gains and user satisfaction factor in the overall assessment as well. Users may be content with a slightly lower performance if the security gains and accounting risks are improving substantially over existing solutions. Hence, at this point it is not only a technical issue but rather a socio-technical tradeoff of technology use and acceptance.

8.8 Conclusion and Outlook

Due to its capabilities to ensure ubiquitous Internet access and to reduce the utilization of mobile network capacities, the concept of Wi-Fi sharing holds many potentials. While numerous approaches have been introduced in the past, most of them cannot sufficiently address the diverse requirements of workable Wi-Fi sharing networks. While trust-based approaches require a trusted intermediary and cannot prevent malicious behavior conducted through fake profiles, security-based concepts lack adequate accounting mechanisms. Recent blockchain-based approaches provide the means to eliminate intermediaries and to build trust among users through immutability and transparency. However, they are hardly capable of realizing the technology's full potentials, as they lack performance and scalability and primarily support bilateral connections between participants.

Against this backdrop, we developed a reference architecture for fast, scalable, and reliable Wi-Fi sharing networks based on the combined use of the blockchain technology and payment channel networks. We collected requirements for workable Wi-Fi sharing networks and answered the first research question. To answer the second research question, we employed a DSR approach to develop design principles and an integrated architecture that comprises the layers of agent, application, infrastructure, and environment. We demonstrated and evaluated its applicability and usefulness by illustrating all phases of a payment channel lifecycle. Our results suggest that the proposed reference architecture can address the most significant shortcomings of established approaches and provides innovative means to conduct and route transaction without the need for a trusted intermediary. The applicability of the reference architecture is not limited to the case of Wi-Fi sharing networks, but can improve other network solutions, especially those involving micro-transactions between multiple independent participants.

Still, this research is not without limitations. Although our literature search procedure was designed to identify the most relevant and actual contributions, research on blockchain technology, payment channel networks, and Wi-Fi sharing is still at an immature level and scattered across multiple platforms and outlets. Furthermore, each research stream consti-

tutes a growing and dynamic field. Hence, we cannot eliminate the possibility that we missed single contributions that might have offered additional insights for our study. Furthermore, a more detailed and practice-oriented evaluation is necessary to provide more definite evidence into practical aspects, such as user adoption, network performance, and resistance to network and data security threats.

Further, our research does not explicitly cover the organizational implementation of the reference architecture making available Wi-Fi sharing networks to users. Open questions remaining to be answered are naturally centered on the governance of the system. That is, who is going to build it and operate it? Will utility or governance tokens assist ensuring a completely decentralized governance? Due to the focus of our research on the development of design principles and a reference architecture for an IT system, we have not covered these organizational aspects. Nevertheless, before making available Wi-Fi sharing networks based on our research, these questions must be asked and answered.

Chapter 9

Conclusion

This thesis addresses the main challenges that blockchain technology faces before experiencing widespread adoption. It is motivated by the four key challenges proposed by Kolb et al. (2020a): Inefficient consensus, privacy, smart contract security, and scalability. To tackle these issues, the goal was to answer the guiding research question:

How do the four main blockchain challenges impact the current blockchain landscape, and how can problems be circumvented or solved?

Because each of the four challenges provides an extensive research area, this thesis focused on one specific problem from each area in Chapters 2-5. Each chapter strengthens the current state-of-the-art understanding, develops solutions for specific challenges, or demonstrates how solutions can be applied.

First, Chapter 2 addresses the *inefficient consensus* of blockchains. The PoW consensus protocol is the least efficient consensus for blockchains. However, it is still popular due to its high security guarantees. In Chapter 2 we verified the security assumptions for the three largest PoW blockchains, as well as their major forks (RQ1). It was hypothesized that mining power centralizes over time. This phenomenon would make it possible that a few actors could cooperate to control the network. However, after an initial centralization period, the mining power seemed to remain stable or even decrease. Additionally, the networks exhibited a self-regulating effect, that if one mining pool gained too much power, miners would switch to a different pool to keep the network secure. This is a strong indicator for working economic incentives. Finally, it could be shown that disrupting events, such as a blockchain fork or reduction of the mining reward, only had temporary effects on the blockchains. Therefore, we concluded that the strong security guarantees of PoW blockchains are intact, as long as the network size is large enough.

In Chapter 3, we provided a basis for the standardization of specific smart contract features (RQ2). Standardization is a powerful tool to decrease complexity and increase *smart contract security*. Based on the analysis of over 100 Ethereum smart contracts, we identified 64 characteristics in 28 dimensions, grouped into six categories. We identified common security functionalities and helper functions besides already standardized implementations such as token standards. Additionally, we clustered the smart contracts based on their functionalities to identify seven archetypes of smart contracts. Based on these archetypes, we could identify dependencies between functionalities and recommend standardizing certain features. While some features, such as the safety functions, were quasi-standardized and many smart contracts used the exact same code to prevent calculation over- and underflows, we noticed a particular lack of standardization in the interaction with off-chain data. The chapter provides a solid basis to drive future standardization efforts.

To assess the *privacy* claims of consortium blockchains, we took the role of an adversary and conducted a case study in Chapter 4. The aim was to identify unsecured consortium blockchain networks, extract all the data, and gain as many insights as possible on these networks' structure, participants, and processes (RQ3). Overall, we were able to identify over 3,000 nodes with an incorrect configuration. Most were connected to one of the major public Ethereum networks. However, some were used by small enterprise consortia and consisted of only a few nodes. We conducted a case study on four of these networks. Although no personally identifiable information was found, we could develop tools to reconstruct the transaction graph for each network completely. Additionally, we could identify several smart contracts and reverse engineer the functionalities of some of them. Therefore, we proposed that consortium blockchains should use additional security measures used in public blockchains to ensure their data privacy.

Chapter 5 provided the basis for the remainder of this thesis by guiding the development of *scalable* blockchain applications (RQ4). Based on a literature review, four bottlenecks for scalability were identified: Computational complexity, transaction volume, transaction latency and storage capacity. Depending on the requirements and prerequisites of a given application, a five-step decision process was designed to assist developers in choosing the right scaling solution. The following three chapters demonstrated examples of scalable blockchain applications (RQ5).

The first example in Chapter 6 shows that scalability is not only dependent on the architectural design of the underlying blockchain. Utilizing native data structures of certain blockchains, we created an approach to trace goods through a complex supply network with so-called *colored coins*. Additionally, we provided a method to convert the transactional data of ERP systems to the transaction data structure for the blockchain. With these results, an implementation on Hyperledger Fabric was instantiated to demonstrate the feasibility of this solution.

The second example demonstrates a computationally complex problem that can still be executed in a blockchain environment. In Chapter 7, we provide an architecture for a de-

centralized marketplace for manufacturing capacities. The operations to match supply and demand are too complex to be run in a simple smart contract. Therefore, we propose the usage of secure multiparty computation and zero-knowledge proofs to offload the computations and only prove the final result to the blockchain, initiating the transactions.

Finally, we provide a use-case for payment channel networks in Chapter 8. This scalability solution provides a way for instant bidirectional payments. We built upon this idea to develop an architecture for WiFi sharing. Participants provide their network bandwidth to other users and get compensated for their services. Due to the usage of payment channels, the payments can be made as granular as necessary to prevent fraudulent behavior from hotspot providers.

While the limitations for each chapter were outlined, some common limitations must be highlighted. Specifically, the results based on the analysis of large data sets are especially limited by the resources spent on extracting and analyzing the data. It should be noted, that especially in Chapters 2, 4, and 3, the limitation was not the availability of data. Therefore, the developed methodologies and tools can expand the research on broader datasets to verify or generalize the results. However, in Chapter 5, the data was limited by the availability of high-quality research. The area of blockchain research is quickly evolving, and some results are already outdated by the time of publishing. Therefore, many researchers and blockchain developers publish their results without proper peer-review. A majority of state-of-the-art research is available only in the form of whitepapers and preprints. Some of the design patterns from this gray literature were adapted in the design-oriented chapters. On the one hand, this imposes a limitation for these chapters. On the other hand, these peer-reviewed chapters can provide a credible foundation for future design efforts.

The presented research can be extended by utilizing the developed methodologies. In Chapter 2, we provided the tools to extract, transform, and analyze the block generation data from public blockchains. These tools can be used to further analyze PoW blockchains, or extended to analyze alternative consensus algorithms. The method used to classify and cluster smart contracts from Chapter 3 provide an excellent basis to analyze smart contracts from other blockchains. A classification for multiple smart contract platforms is the basis for a source-code based comparison framework. Ultimately, the goal to provide standardized smart contracts is not limited to the Ethereum blockchain. The same principle applies for the methods developed in Chapter 4. While the study provided insights about the privacy issues in Ethereum-based blockchains, the overall purpose was to provide the tools and methods to analyze different blockchain systems. Only a comparison of different systems provides a basis for decision-making.

The decision process developed in Chapter 5 was a basis for three chapters of this thesis. Therefore, it has already proved useful to develop further use-cases for scalable blockchain applications. Additionally, the research should be extended by similar decision processes to build secure or privacy-preserving blockchain applications. Based on these guidelines, the proposed applications in Chapters 2-8 can further be refined and evaluated. Finally, a

practical implementation of the proposed solutions can help identify further limitations that may arise in practical use.

Despite the results achieved in this thesis, there is still room for improvement for each of the four blockchain challenges. There is a constant stream of new consensus protocols that promise better trade-offs between security, scalability, and decentralization. Additionally, new cryptographic primitives, such as homomorphic encryption, provide a technological basis to improve privacy in blockchain applications significantly. However, regulatory issues hinder the adoption of these technologies, mainly because of money laundering laws. Finding a balance between privacy and regulatory compliance is a new challenge and opens a new field for research. Therefore, while this thesis provided solutions to a wide range of technological challenges, a new field of socioeconomic challenges should be tackled.

Bibliography

Agarwal, R. and Prasad, J. (1997). The role of innovation characteristics and perceived voluntariness in the acceptance of information technologies. *Decision sciences*, 28(3):557–582.

Ahangama, S. and Poo, D. C. C. (2016). Credibility of algorithm based decentralized computer networks governing personal finances: The case of cryptocurrency. In *International Conference on HCI in Business, Government and Organizations*, HCI in Business, Government, and Organizations: eCommerce and Innovation, pages 165–176.

Alt, R. (2018). Electronic markets and current general research. *Electronic Markets*, 28(2):123–128.

Alt, R. (2020). Electronic markets on blockchain markets. *Electronic Markets*, 30(2):181–188.

ambisafe (2021). Ambisafe — Making financial markets universally accessible. `https://ambisafe.com/`. Accessed: 2021-01-11.

Anand, A., McKibbin, M., and Pichel, F. (2016). Colored coins: Bitcoin, blockchain, and land administration. In *Annual World Bank Conference on Land and Poverty*.

Andersen, R., Brunoe, T. D., and Nielsen, K. (2019). A framework for identification of complexity drivers in manufacturing companies. In *IFIP International Conference on Advances in Production Management Systems*, pages 392–399.

Andola, N., Raghav, Yadav, V. K., Venkatesan, S., and Verma, S. (2021). Anonymity on blockchain based e-cash protocols—a survey. *Computer Science Review*, 40:100394.

Anoaica, A. and Levard, H. (2018). Quantitative description of internal activity on the ethereum public blockchain. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5.

Antonopoulos, A. M. and Wood, G. (2018). *Mastering ethereum: building smart contracts and dapps*. O'reilly Media.

Asgaonkar, A. and Krishnamachari, B. (2019). Solving the buyer and seller's dilemma: A dual-deposit escrow smart contract for provably cheat-proof delivery and payment for a digital good without a trusted mediator. In *IEEE International Conference on Blockchain and Cryptocurrency*, pages 262–267.

atlas (2021). GitHub - ethereum-navigator/atlas: The single source of truth for all Ethereum networks. `https://github.com/ethereum-navigator/atlas`. Accessed: 2021-01-11.

Avital, M., Beck, R., King, J. L., Rossi, M., and Teigland, R. (2016). Jumping on the blockchain bandwagon: Lessons of the past and outlook to the future. In *ICIS 2016 Proceedings*.

Bach, L., Mihaljevic, B., and Zagar, M. (2018). Comparative analysis of blockchain consensus algorithms. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1545–1550.

Ballandies, M. C., Dapp, M. M., and Pournaras, E. (2021). Decrypting distributed ledger design—taxonomy, classification and blockchain community evaluation. *Cluster Computing*, 2021(1).

Banerjee, A. (2018). Blockchain technology: Supply chain insights from ERP. In *Advances in Computers*, pages 69–98. Elsevier.

Barker, E. and Roginsky, A. (2010). Recommendation for the transitioning of cryptographic algorithms and key lengths. *NIST Special Publication*.

Bartoletti, M. and Pompianu, L. (2017). An empirical analysis of smart contracts: platforms, applications, and design patterns. In *International conference on financial cryptography and data security*, pages 494–509.

Beck, R., Avital, M., Rossi, M., and Thatcher, J. B. (2017). Blockchain technology in business and information systems research. *Business & Information Systems Engineering*, 59(6):381–384.

Beck, R., Czepluch, J. S., Lollike, N., and Malone, S. (2016). Blockchain: The gateway to trust-free cryptographic transactions. In *30th European Conference on Information Systems*.

Beck, R. and Müller-Bloch, C. (2017). Blockchain as radical innovation: a framework for engaging with distributed ledgers as incumbent organization. In *Proceedings of the 50th Hawaii International Conference on System Sciences*.

Beck, R., Müller-Bloch, C., and King, J. L. (2018). Governance in the blockchain economy: A framework and research agenda. *Journal of the Association for Information Systems*, 19(10):1.

Beikverdi, A. and Song, J. (2015). Trend of centralization in bitcoin's distributed network. In *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 1–6.

Bellare, M., Namprempre, C., and Neven, G. (2009). Security proofs for identity-based identification and signature schemes. *Journal of Cryptology*, 22(1):1–61.

Ben-Sasson, E., Bentov, I., Horesh, Y., and Riabzev, M. (2019). Scalable zero knowledge with no trusted setup. In *Annual International Cryptology Conference*, pages 701–732.

Berg, B. L. (2001). *Qualitative Research Methods for the Social Sciences*. Allyn & Bacon.

Berger, T. (2006). Analysis of current vpn technologies. In *First International Conference on Availability, Reliability and Security (ARES'06)*, pages 8–pp.

Biryukov, A. and Tikhomirov, S. (2019). Deanonymization and linkability of cryptocurrency transactions based on network analysis. In *2019 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 172–184.

bitinfocharts.com (2019). Bitcoin block time historical chart. `https://bitinfocharts.com/comparison/bitcoin-confirmationtime.html`. Accessed: 2019-01-14.

Bocek, T., Rodrigues, B. B., Strasser, T., and Stiller, B. (2017). Blockchains everywhere-a use-case of blockchains in the pharma supply-chain. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 772–777.

Brown, A., Amundson, J., and Badurdeen, F. (2014). Sustainable value stream mapping (sus-vsm) in different manufacturing system configurations: application case studies. *Journal of Cleaner Production*, 85:164–179. Special Volume: Making Progress Towards More Sustainable Societies through Lean and Green Initiatives.

Bryman, A. (2006). Integrating quantitative and qualitative research: how is it done? *Qualitative research*, 6(1):97–113.

Buer, S.-V., Strandhagen, J. O., and Chan, F. T. S. (2018). The link between industry 4.0 and lean manufacturing: mapping current research and establishing a research agenda. *International Journal of Production Research*, 56(8):2924–2940.

Buhl, H. U., Röglinger, M., Moser, F., and Heidemann, J. (2013). Big data. *Business & Information Systems Engineering*, 5(2):65–69.

Busert, T. and Fay, A. (2019). Extended value stream mapping method for information based improvement of production logistics processes. *IEEE Engineering Management Review*, 47(4):119–127.

Buterin, V. (2014). Ethereum white paper: a next generation smart contract & decentralized application platform. `https://ethereum.org/en/whitepaper/`. Accessed: 2018-08-06.

Buterin, V. (2018). Ethereum scalability research and development subsidy programs. `https://blog.ethereum.org/2018/01/02/ethereum-scalability-research-development-subsidy-programs/`. Accessed: 2019-01-04.

Cai, S., Yang, N., and Ming, Z. (2018). A decentralized sharding service network framework with scalability. In *International Conference on Web Services*, pages 151–165.

Caleum (2021). Caleum project website. `https://web.archive.org/web/2020*/www.caelumproject.io`. Accessed: 2021-01-11.

Camponovo, G. and Cerutti, D. (2005). Wlan communities and internet access sharing: A regulatory overview. In *International Conference on Mobile Business (ICMB'05)*, pages 281–287.

Cao, Z., Fitschen, J., and Papadimitriou, P. (2015). Social wi-fi: Hotspot sharing with online friends. In *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 2132–2137.

Carlsten, M., Kalodner, H., Weinberg, S. M., and Narayanan, A. (2016). On the instability of bitcoin without the block reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 154–167.

Carna Botnet (2012). Internet Census 2012. `http://census2012.sourceforge.net/paper.html`. Accessed: 2021-01-11.

Carson, B., Romanelli, G., Walsh, P., and Zhumaev, A. (2018). Blockchain beyond the hype: What is the strategic business value. `https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/blockchain-beyond-the-hype-what-is-the-strategic-business-value`. Accessed: 2018-08-11.

Chan, W. and Olmsted, A. (2017). Ethereum transaction graph analysis. In *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 498–500.

Chandra, L., Seidel, S., and Gregor, S. (2015). Prescriptive knowledge in is research: Conceptualizing design principles in terms of materiality, action, and boundary conditions. In *48th Hawai'i International Conference on System Sciences (HICSS)*, pages 4039–4048.

Chauhan, A., Malviya, O. P., Verma, M., and Mor, T. S. (2018). Blockchain and scalability. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 122–128.

Chen, L., Xu, L., Shah, N., Gao, Z., Lu, Y., and Shi, W. (2017). On security analysis of proof-of-elapsed-time (poet). In Spirakis, P. and Tsigas, P., editors, *Stabilization, Safety, and Security of Distributed Systems*, pages 282–297.

Chohan, U. W. (2017). A history of dogecoin. *SSRN Electronic Journal*.

Cisco (2020). Cisco annual internet report (2018–2023) white paper. `https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/whitepaper-c11-741490.html`. Accessed: 2021-03-26.

Clarke, R. N. (2014). Expanding mobile wireless capacity: The challenges presented by technology and economics. *Telecommunications Policy*, 38(8):693–708.

CoinMarketCap (2019). Coinmarketcap. `https://coinmarketcap.com/`. Accessed: 2019-11-25.

CoinMarketCap (2021). Coinmarketcap. `https://coinmarketcap.com/`. Accessed: 2021-12-19.

Cong, L. W., He, Z., and Li, J. (2020). Decentralized Mining in Centralized Pools. *The Review of Financial Studies*, 34(3):1191–1235.

Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E. G., et al. (2016). On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125.

crypto51 (2019). Cost of a 51% attack for different cryptocurrencies. `https://www.crypto51.app/`. Accessed: 2019-04-25.

Cucurull, J., Rodríguez-Pérez, A., Finogina, T., and Puiggalí, J. (2018). Blockchain-based internet voting: Systems' compliance with international standards. In *International Conference on Business Information Systems*, pages 300–312.

Dal Forno, A. J., Pereira, F. A., Forcellini, F. A., and Kipper, L. M. (2014). Value stream mapping: a study about the problems and challenges found in the literature from the past 15 years about application of lean tools. *The International Journal of Advanced Manufacturing Technology*, 72(5-8):779–790.

Daraghmi, E. Y. and Yuan, S.-M. (2014). We are so close, less than 4 degrees separating you and me! *Computers in Human Behavior*, 30:273–285.

Daum, T. and Birner, R. (2020). Agricultural mechanization in africa: Myths, realities and an emerging research agenda. *Global Food Security*, 26.

Davidson, S., De Filippi, P., and Potts, J. (2016). Economics of blockchain. *Available at SSRN 2744751*.

De Filippi, P. and McCarthy, S. (2012). Cloud computing: Centralization and data sovereignty. *European Journal of Law and Technology*, 3(2).

de Vries, A. (2020). Bitcoin's energy consumption is underestimated: A market dynamics approach. *Energy Research & Social Science*, 70:101721.

Decker, C. and Wattenhofer, R. (2015). A fast and scalable payment network with bitcoin duplex micropayment channels. In *Symposium on Self-Stabilizing Systems*, Stabilization, Safety, and Security of Distributed Systems, pages 3–18.

Delmolino, K., Arnett, M., Kosba, A., Miller, A., and Shi, E. (2016). Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In *International conference on financial cryptography and data security*, pages 79–94.

Derks, J., Gordijn, J., and Siegmann, A. (2018). From chaining blocks to breaking even: A study on the profitability of bitcoin mining from 2012 to 2016. *Electronic Markets*, 28(3):321–338.

Dev, J. A. (2014). Bitcoin mining acceleration and performance quantification. In *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–6.

Devaraj, D., Valarmathi, K., Kanmani, J., and Radhakrishnan, T. (2007). Hybrid ga fuzzy controller for ph process. In *Computational Intelligence and Multimedia Applications, International Conference on*, volume 2, pages 13–18.

Dhillon, V., Metcalf, D., and Hooper, M. (2017). *The Hyperledger Project*, pages 139–149. Apress.

Di Francesco Maesa, D., Marino, A., and Ricci, L. (2018). Data-driven analysis of bitcoin properties: exploiting the users graph. *International Journal of Data Science and Analytics*, 6(1):63–80.

Diem Association (2020). White Paper — Diem Association. `https://www.diem.com/en-us/white-paper/`. Accessed: 2021-01-22.

Dimatteo, S., Hui, P., Han, B., and Li, V. O. K. (2011). Cellular traffic offloading through wifi networks. In *Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, pages 192–201.

Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., and Tan, K.-L. (2017). Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1085–1100.

Doller, A. (2013). *Chargenverwaltung mit SAP*. SAP : Logistik. Rheinwerk Verlag.

Douceur, J. R. (2002). The sybil attack. In Druschel, P., Kaashoek, F., and Rowstron, A., editors, *Peer-to-Peer Systems*, pages 251–260.

Eberhardt, J. and Tai, S. (2018). Zokrates-scalable privacy-preserving off-chain computations. In *IEEE International Conference on Blockchain. IEEE*.

Ecb (2021). Crypto-assets – trends and implications. `https://web.archive.org/web/2020*/www.caelumproject.io`. Accessed: 2021-01-11.

Edwards, R. and Holland, J. (2013). *What is Qualitative Interviewing?* Bloomsbury.

Eevm (2020). Panoramix. `https://github.com/eveem-org/panoramix`. Accessed: 2020-10-03.

El Ioini, N. and Pahl, C. (2018). A review of distributed ledger technologies. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 277–288.

Ellson, J., Gansner, E., Koutsofios, L., North, S. C., and Woodhull, G. (2001). Graphviz—open source graph drawing tools. In *International Symposium on Graph Drawing*, pages 483–484.

Erdin, E., Mercan, S., and Akkaya, K. (2021). An evaluation of cryptocurrency payment channel networks and their privacy implications. *ITU Journal on Future and Evolving Technologies*, 2(1):1–10.

Ethereum Foundatiom (2015). ERC-20 token standard. `https://ethereum.org/en/developers/docs/standards/tokens/erc-20/`. Accessed: 2020-02-21.

ethernodes (2021). Clients - ethernodes.org - The Ethereum Network & Node Explorer. `https://ethernodes.org/`. Accessed: 2021-01-11.

Euler, T. (2021). The token classification framework: A multi-dimensional tool for understanding and classifying crypto tokens. `http://www.untitled-inc.com/the-token-classification-framework-a-multi-dimensional-tool-for-understanding-and-classifying-crypto-tokens`. Accessed: 2021-05-04.

Evans, D. (2011). The internet of things: How the next evolution of the internet is changing everything. `https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf`. Accessed: 2019-11-11.

Fairley, P. (2018). Ethereum will cut back its absurd energy use. *IEEE Spectrum*, 56(1):29–32.

Fanning, K. and Centers, D. P. (2016). Blockchain and its coming impact on financial services. *Journal of Corporate Accounting & Finance*, 27(5):53–57.

Faulkner, W. and Badurdeen, F. (2014). Sustainable value stream mapping (sus-vsm): methodology to visualize and assess manufacturing sustainability performance. *Journal of Cleaner Production*, 85:8–18. Special Volume: Making Progress Towards More Sustainable Societies through Lean and Green Initiatives.

Fellmann, M., Koschmider, A., Laue, R., Schoknecht, A., and Vetter, A. (2018). Business process model patterns: state-of-the-art, research classification and taxonomy. *Business Process Management Journal*, 25(5):972–994.

Feng, X., Ma, J., Miao, Y., Meng, Q., Liu, X., Jiang, Q., and Li, H. (2018). Pruneable sharding-based blockchain protocol. *Peer-to-Peer Networking and Applications*.

Finneseth, J. (2021). Memecoin mania triggers triple-digit gains from binance smart chain-based altcoins. `https://cointelegraph.com/news/meme-coin-mania-triggers-triple-digit-gains-from-binance-smart-chain-based-altcoins`. Accessed: 2021-12-10.

Fischer, M., Heim, D., Hofmann, A., Janiesch, C., Klima, C., and Winkelmann, A. (2020). A taxonomy and archetypes of smart services for smart living. *Electronic Markets*, 30(1):131–149.

Frangoudis, P. A., Polyzos, G. C., and Kemerlis, V. P. (2011). Wireless community networks: an alternative approach for nomadic broadband network access. *IEEE Communications Magazine*, 49(5):206–213.

Freichel, C., Hofmann, A., Fischer, M., and Winkelmann, A. (2019). Requirements and a meta model for exchanging additive manufacturing capacities. In *International Conference on Wirtschaftsinformatik*, pages 2–16.

Freitag, M., Becker, T., and Duffie, N. A. (2015). Dynamics of resource sharing in production networks. *CIRP Annals*, 64(1):435–438.

Fröwis, M., Fuchs, A., and Böhme, R. (2019). Detecting token systems on ethereum. In *International Conference on Financial Cryptography and Data Security*, pages 93–112.

Galal, H. S. and Youssef, A. M. (2018). Succinctly verifiable sealed-bid auction smart contract. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 3–19. Springer.

Garza-Reyes, J. A., Torres Romero, J., Govindan, K., Cherrafi, A., and Ramanathan, U. (2018). A pdca-based approach to environmental value stream mapping (e-vsm). *Journal of Cleaner Production*, 180:335–348.

Gencer, A. E., Basu, S., Eyal, I., Van Renesse, R., and Sirer, E. G. (2018). Decentralization in bitcoin and ethereum networks. *arXiv preprint arXiv:1801.03998*.

Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., and Capkun, S. (2016). On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 3–16.

Gimpel, H., Rau, D., and Röglinger, M. (2017). Understanding FinTech start-ups – a taxonomy of consumer-oriented service offerings. *Electronic Markets*, 28(3):245–264.

Gipp, B., Meuschke, N., and Gernandt, A. (2015). Trusted timestamping using the crypto currency bitcoin. In *iConference*, pages 1–6.

Glaser, F. and Bezzenberger, L. (2015). Beyond cryptocurrencies-a taxonomy of decentralized consensus systems. In *ECIS 2015 Completed Research Papers*.

Gola, Y. (2021). Game over! squid game-inspired crypto scam collapses as price crashes from $2.8k to zero. `https://cointelegraph.com/news/game-over-squid-game-inspired-crypto-scam-collapses-as-price-crashes-from-2-8k-to-zero`. Accessed: 2021-12-10.

Green, J., Willis, K., Hughes, E., Small, R., Welch, N., Gibbs, L., and Daly, J. (2007). Generating best evidence from qualitative research: the role of data analysis. *Australian and New Zealand Journal of Public Health*, 31(6):545–550.

Gretzel, U., Sigala, M., Xiang, Z., and Koo, C. (2015). Smart tourism: foundations and developments. *Electronic Markets*, 25(3):179–188.

Guo, Y. and Liang, C. (2016). Blockchain application and outlook in the banking industry. *Financial Innovation*, 2(1):24.

Halpin, H. (2020). Deconstructing the decentralization trilemma. In *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications*.

Han, R., Yu, J., Lin, H., Chen, S., and Esteves-Veríssimo, P. (2021). On the security and performance of blockchain sharding. `https://ia.cr/2021/1276`. Accessed: 2021-11-29.

Hartmann, L., Meudt, T., Seifermann, S., and Metternich, J. (2018). Value stream method 4.0: holistic method to analyse and design value streams in the digital age. *Procedia CIRP*, 78:249–254. 6th CIRP Global Web Conference – Envisaging the future manufacturing, design, technologies and systems in innovation era (CIRPe 2018).

Hawlitschek, F., Teubner, T., Adam, M. T. P., Borchers, N. S., Möhlmann, M., and Weinhardt, C. (2016). Trust in the sharing economy: An experimental framework. In *ICIS 2016 Proceedings*.

Hedman, J., Srinivasan, N., and Lindgren, R. (2013). Digital traces of information systems: Sociomateriality made researchable. In *ICIS 2013 Proceedings*.

Heger, S., , Valett, L., Thim, H., Schröder, J., and and, H. G. (2020). Value stream model and notation – digitale transformation von wertströmen. In *WI2020 Zentrale Tracks*, pages 710–724. GITO Verlag.

Heidemann, J., Pradkin, Y., Govindan, R., Papadopoulos, C., Bartlett, G., and Bannister, J. (2008). Census and survey of the visible internet (extended). *Isi-tr-2008-649*.

Heng, S. (2014). Industry 4.0: Upgrading of germany's industrial capabilities on the horizon. `https://www.dbresearch.com/PROD/RPS_EN-PROD/PROD0000000000451959/Industry_4_0%3A_Upgrading_of_Germany%E2%80%99s_industrial_ca.PDF`. Accessed: 2020-04-03.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

Hofmann, A. (2020). Building scalable blockchain applications - a decision process. In Hofmann, S., Müller, O., and Rossi, M., editors, *Designing for Digital Transformation. Co-Creating Services with Citizens and Industry*, pages 309–320.

Hofmann, A., Freichel, C., and Winkelmann, A. (2021a). A decentralized marketplace for collaborative manufacturing. In *ECIS 2021 Research Papers*.

Hofmann, A., Gwinner, F., Winkelmann, A., and Janiesch, C. (2021b). Security implications of consortium blockchains: The case of ethereum networks. *JIPITEC*, 12(4):347–359.

Hofmann, A., Kolb, J., Becker, L., and Winkelmann, A. (2021c). A source-code-based taxonomy for ethereum smart contracts. In *ICIS 2021 Proceedings*.

Hofmann, A., Schatz, F. J., and Winkelmann, A. (2020). Uncovering the mining behaviour in proof-of-work blockchains. In *ECIS 2020 Research Papers*.

Hopwood, D., Bowe, S., Hornby, T., and Wilcox, N. (2020). Zcash protocol specification. `https://raw.githubusercontent.com/zcash/zips/master/protocol/protocol.pdf`. Accessed: 2020-01-20.

Huan, S. H., Sheoran, S. K., and Wang, G. (2004). A review and analysis of supply chain operations reference (SCOR) model. *Supply Chain Management: An International Journal*, 9(1):23–29.

Huang, Y., Bian, Y., Li, R., Zhao, J. L., and Shi, P. (2019). Smart contract security: A software lifecycle perspective. *IEEE Access*, 7:150184–150202.

Hull, R. (2017). Blockchain: Distributed event-based processing in a data-centric world. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems*, pages 2–4.

Hülsemann, P. and Tumasjan, A. (2019). Walk this way! incentive structures of different token designs for blockchain-based applications. In *ICIS 2019 Proceedings*.

Hurlburt, G. (2016). Might the blockchain outlive bitcoin? *IT Professional*, 18(2):12–16.

Jæger, B., Bach, T., and Pedersen, S. A. (2019). A blockchain application supporting the manufacturing value chain. In Ameri, F., Stecke, K. E., von Cieminski, G., and Kiritsis, D., editors, *Advances in Production Management Systems. Production Management for the Factory of the Future*, pages 466–473.

Jagati, S. (2021). Poly network hack exposes defi flaws, but community comes to the rescue. `https://cointelegraph.com/news/poly-network-hack-exposes-defi-flaws-but-community-comes-to-the-rescue`. Accessed: 2021-12-10.

Janiesch, C., , Rosenkranz, C., and Scholten, U. (2020). An information systems design theory for service network effects. *Journal of the Association for Information Systems*, 21:1402–1460.

Janiesch, C., Fischer, M., Imgrund, F., Hofmann, A., and Winkelmann, A. (2021). An architecture using payment channel networks for blockchain-based wi-fi. *ACM Transactions on Management Information Systems*. In Revision.

Jenkinson, G. (2021). Africrypt turns sour on investors: Founders flee as court cases build up. https://cointelegraph.com/news/africrypt-turns-sour-on-investors-founders-flee-as-court-cases-build-up. Accessed: 2021-12-10.

Jin, X.-L., Zhang, M., Zhou, Z., and Yu, X. (2019). Application of a blockchain platform to manage and secure personal genomic data: A case study of LifeCODE.ai in china. *Journal of Medical Internet Research*, 21(9).

JP Morgan Chase (2018). Quorum whitepaper. https://github.com/ConsenSys/quorum/blob/master/docs/QuorumWhitepaperv0.2.pdf. Accessed: 2021-02-10.

Khan, A., Kellerer, W., Kozu, K., and Yabusaki, M. (2011). Network sharing in the next mobile network: Tco reduction, management flexibility, and operational independence. *IEEE Communications Magazine*, 49(10):134–142.

Kim, S., Kwon, Y., and Cho, S. (2018). A survey of scalability solutions on blockchain. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1204–1207.

Kirpes, B., Mengelkamp, E., Schaal, G., and Weinhardt, C. (2019). Design of a microgrid local energy market on a blockchain-based information system. *it - Information Technology*, 61(2-3):87–99.

Knoll, D., Reinhart, G., and Prüglmeier, M. (2019). Enabling value stream mapping for internal logistics using multidimensional process mining. *Expert Systems with Applications*, 124:130–142.

Ko, K., Lee, C., Jeong, T., and Hong, J. W.-K. (2018). Design of RPC-based blockchain monitoring agent. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1090–1095.

Koens, T. and Poll, E. (2018). What blockchain alternative do you need? In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 113–129. Springer.

Kolb, J., AbdelBaky, M., Katz, R. H., and Culler, D. E. (2020a). Core concepts, challenges, and future directions in blockchain. *ACM Computing Surveys*, 53(1):1–39.

Kolb, J., Hofmann, A., and Becker, L. (2020b). Building a taxonomy for gambling smart contracts. In *ECIS 2020 Research-in-Progress Papers*.

Kouicem, D. E., Bouabdallah, A., and Lakhlef, H. (2018). Internet of things security: A top-down survey. *Computer Networks*, 141:199–221.

Kroll, J. A., Davey, I. C., and Felten, E. W. (2013). The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, volume 2013, page 11.

Kuanyshbayev, Z. M., Arpabekov, M., and Zhanabayeva, S. (2013). Confidence limits of cryptocurrency feathercoin in the intermodal connection. *Science And World*, page 49.

Kuckartz, U. (2018). *Qualitative Inhaltsanalyse. Methoden, Praxis, Computerunterstützung*. Beltz Juventa.

Kurpjuweit, S., Schmidt, C. G., Klöckner, M., and Wagner, S. M. (2021). Blockchain in additive manufacturing and its impact on supply chains. *Journal of Business Logistics*, 42(1):46–70.

Kuzuno, H. and Karam, C. (2017). Blockchain explorer: An analytical process and investigation environment for bitcoin. In *2017 APWG Symposium on Electronic Crime Research (eCrime)*, pages 9–16.

Lacity, M. (2018). Addressing key challenges to making enterprise blockchain applications a reality. *MIS Quarterly Executive*, 17(3):201–222.

Lafuente, C. B., Titi, X., and Seigneur, J. M. (2011). Flexible communication: A secure and trust-based free wi-fi password sharing service. In *10th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 706–713.

Lakshminarayanan, K. and Padmanabhan, V. N. (2003). Some findings on the network performance of broadband hosts. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 45–50.

Lee, J. Y. (2019). A decentralized token economy: How blockchain and cryptocurrency can revolutionize business. *Business Horizons*, 62(6):773–784.

Lee, S., Shin, S.-H., and Roh, B.-h. (2017). Abnormal behavior-based detection of shodan and censys-like scanning. *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 1048–1052.

Leroy, D., Detal, G., Cathalo, J., Manulis, M., Koeune, F., and Bonaventure, O. (2011). Swish: Secure wifi sharing. *Computer Networks*, 55(7):1614–1630.

Li, C.-T., Weng, C.-Y., Lee, C.-C., and Wang, C.-C. (2015). A hash based remote user authentication and authenticated key agreement scheme for the integrated epr information system. *Journal of Medical Systems*, 39(11):144.

Li, X., Jiang, P., Chen, T., Luo, X., and Wen, Q. (2017). A survey on the security of blockchain systems. *Future Generation Computer Systems*.

Lin, I.-C. and Liao, T.-C. (2017). A survey of blockchain security issues and challenges. *IJ Network Security*, 19(5):653–659.

Loe, A. F. and Quaglia, E. A. (2018). Conquering generals: an np-hard proof of useful work. In *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, pages 54–59.

Lorenz, R., Buess, P., Macuvele, J., Friedli, T., and Netland, T. H. (2019). Lean and digitalization—contradictions or complements? In Ameri, F., Stecke, K. E., von Cieminski, G., and Kiritsis, D., editors, *Advances in Production Management Systems. Production Management for the Factory of the Future*, pages 77–84.

Lu, A. (2021). EIP-1154: Oracle interface. `https://eips.ethereum.org/EIPS/eip-1154`. Accessed: 2021-05-04.

Lu, Q. and Xu, X. (2017). Adaptable blockchain-based systems: A case study for product traceability. *IEEE Software*, 34(6):21–27.

Lugert, A., Batz, A., and Winkler, H. (2018). Empirical assessment of the future adequacy of value stream mapping in manufacturing industries. *Journal of Manufacturing Technology Management*.

Madhulatha, T. S. (2012). An overview on clustering methods. url-http://arxiv.org/abs/1205.1117. Accessed: 2018-08-15.

Mager, L., Schmidt, P., Rezafard, A., Perry, R., Jacob, B., Moberg, D., Barcan, C., Rodgers, D., Mouton, O., Hoberg, P., Karlsson, C., Roberts, T., Eckhardt, D., Berthomieu, F., Le Hello, D., Krishnamurthy, V., Edison, C., Zottola, C., Bradley, A., Recham, H., Szabo, J., Schmid, S., Meissl, A., Maniero, A. P., Laur, R., and Sadiwnyk, M. (2016). *GS1 Global Traceability Compliance Criteria for Food - Application Standard*. Gs1 Aisbl.

Malavolta, G., Moreno-Sanchez, P., Kate, A., Maffei, M., and Ravi, S. (2017). Concurrency and privacy with payment-channel networks. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 455–471.

Mamatas, L., Psaras, I., and Pavlou, G. (2010). Incentives and algorithms for broadband access sharing. In *Proceedings of the 2010 ACM SIGCOMM Workshop on Home Networks*, HomeNets '10, page 19–24.

Matzutt, R., Hiller, J., Henze, M., Ziegeldorf, J. H., Müllmann, D., Hohlfeld, O., and Wehrle, K. (2018). A quantitative analysis of the impact of arbitrary blockchain content on bitcoin. In *Proceedings of the 22nd International Conference on Financial Cryptography and Data Security (FC). Springer*.

McCorry, P., Möser, M., Shahandasti, S. F., and Hao, F. (2016a). Towards bitcoin payment networks. In *Australasian Conference on Information Security and Privacy*, pages 57–76.

McCorry, P., Möser, M., Shahandasti, S. F., and Hao, F. (2016b). Towards bitcoin payment networks. In *Australasian Conference on Information Security and Privacy*, Information Security and Privacy, pages 57–76.

Mendling, J., Decker, G., Hull, R., Reijers, H. A., and Weber, I. (2018). How do machine learning, robotic process automation, and blockchains affect the human factor in business process management? *Communications of the Association for Information Systems*, 43(1):19.

Menezes, A. J., Katz, J., Van Oorschot, P. C., and Vanstone, S. A. (1996). *Handbook of Applied Cryptography*. CRC press.

Mengelkamp, E., Notheisen, B., Beer, C., Dauer, D., and Weinhardt, C. (2018). A blockchain-based smart grid: Towards sustainable local energy markets. *Computer Science - Research and Development*, 33(1-2):207–214.

Mercan, S., Erdin, E., and Akkaya, K. (2021). Improving transaction success rate in cryptocurrency payment channel networks. *Computer Communications*, 166:196–207.

Meudt, T., Metternich, J., and Abele, E. (2017). Value stream mapping 4.0: Holistic examination of value stream and information logistics in production. *CIRP Annals*, 66(1):413–416.

Miller, A., Litton, J., Pachulski, A., Gupta, N., Levin, D., Spring, N., and Bhattacharjee, B. (2015). Discovering bitcoin's public topology and influential nodes. `https://allquantor.at/blockchainbib/pdf/miller2015topology.pdf`. Accessed: 2019-04-03.

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., and Qijun, C. (2017). A review on consensus algorithm of blockchain. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2567–2572.

Mita, M., Ito, K., Ohsawa, S., and Tanaka, H. (2019). What is stablecoin?: A survey on price stabilization mechanisms for decentralized payment systems. In *International Congress on Advanced Applied Informatics*, pages 60–66.

Molina-Jimenez, C., Solaiman, E., Sfyrakis, I., Ng, I., and Crowcroft, J. (2019). On and off-blockchain enforcement of smart contracts. In Mencagli, G., B. Heras, D., Cardellini, V., Casalicchio, E., Jeannot, E., Wolf, F., Salis, A., Schifanella, C., Manumachu, R. R., Ricci, L., Beccuti, M., Antonelli, L., Garcia Sanchez, J. D., and Scott, S. L., editors, *Euro-Par 2018: Parallel Processing Workshops*, pages 342–354.

Möller, J. and Rimscha, M. (2017). (de) centralization of the global informational ecosystem. *Media and Communication*, 5(3):37–48.

Mondal, S., Wijewardena, K. P., Karuppuswami, S., Kriti, N., Kumar, D., and Chahal, P. (2019). Blockchain inspired rfid-based information architecture for food supply chain. *IEEE Internet of Things Journal*, 6(3):5803–5813.

Müller, O., Junglas, I., Brocke, J. v., and Debortoli, S. (2016). Utilizing big data analytics for information systems research: challenges, promises and guidelines. *European Journal of Information Systems*, 25(4):289–302.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`. Accessed: 2019-07-03.

Newar, B. (2021). Badgerdao reportedly suffers security breach, losing $120m. https://cointelegraph.com/news/badgerdao-reportedly-suffers-security-breach-and-loses-10m. Accessed: 2021-12-10.

Nickerson, R. C., Varshney, U., and Muntermann, J. (2013). A method for taxonomy development and its application in information systems. *European Journal of Information Systems*, 22(3):336–359.

Nicolas, H. (2014). It will cost you nothing to 'kill' a proof-of-stake crypto-currency. *SSRN Electronic Journal*.

Noether, S., Mackenzie, A., and Lab, T. M. R. (2016). Ring confidential transactions. *Ledger*, 1:1–18.

Nofer, M., Gomber, P., Hinz, O., and Schiereck, D. (2017). Blockchain. *Business & Information Systems Engineering*, 59(3):183–187.

Noll, D. and Alt, R. (2020). Internet-of-things marketplaces: State-of-the-art and the role of distributed ledger technology. In Abramowicz, W. and Klein, G., editors, *Business Information Systems. BIS 2020. Lecture Notes in Business Information Processing*, pages 337–350. Springer.

Notheisen, B., Hawlitschek, F., and Weinhardt, C. (2017). Breaking down the blockchain hype-towards a blockchain market engineering approach portugal, june 5-10, 2017. In *ECIS 2017 Research Papers*.

Notheisen, B., Willrich, S., Diez, M., and Weinhardt, C. (2019). Requirement-driven taxonomy development – a classification of blockchain technologies for securities post-trading. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*.

Oberländer, A. M., Lösser, B., and Rau, D. (2019). Taxonomy research in information systems: A systematic assessment. In *ECIS 2019 Research Papers*.

Ocicka, B. and Wieteska, G. (2017). Sharing economy in logistics and supply chain management. *Logforum*, 13(2):183–193.

Oliver, J. (2019). Wind energy used to mine cryptocurrency to fund climate research. https://julianoliver.com/output/harvest. Accessed: 2019-11-25.

Paquet-Clouston, M., Haslhofer, B., and Dupont, B. (2019). Ransomware payments in the bitcoin ecosystem. *Journal of Cybersecurity*, 1:11.

Paré, G. (2004). Investigating information systems with positivist case study research. *Communications of the Association for Information Systems*, 13(1):233–264.

Parra-Moyano, J., Thoroddsen, T., and Ross, O. (2019). Optimised and dynamic kyc system based on blockchain technology. *International Journal of Blockchains and Cryptocurrencies*, 1(1):85.

Peffers, K., Rothenberger, M., Tuunanen, T., and Vaezi, R. (2012). Design science research evaluation. In Peffers, K., Rothenberger, M., and Kuechler, B., editors, *Design Science Research in Information Systems. Advances in Theory and Practice*, pages 398–410.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77.

Pinto, A. M. (2020). An introduction to the use of zk-snarks in blockchains. In Pardalos, P., Kotsireas, I., Guo, Y., and Knottenbelt, W., editors, *Mathematical Research for Blockchain Economy. Springer Proceedings in Business and Economics*, pages 233–249. Springer.

Poon, J. and Dryja, T. (2016). The bitcoin lightning network: Scalable off-chain instant payments. `https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf`. Accessed: 2019-10-30.

Post, R., Smit, K., and Zoet, M. (2018). Identifying factors affecting blockchain technology diffusion. In *AMCIS 2018 Proceedings*.

Powell, D. (2013). ERP systems in lean production: new insights from a review of lean and ERP literature. *International Journal of Operations & Production Management*, 33(11/12):1490–1510.

Provable (2020). Provable - blockchain oracle service, enabling data-rich smart contracts. `https://provable.xyz/`. Accessed: 2020-12-15.

Pytel, N., Hofmann, A., and Winkelmann, A. (2020). Tracing back the value stream with colored coins. In *ICIS 2020 Proceedings*.

Recker, J., Rosemann, M., Green, P., and Indulska, M. (2011). Do ontological deficiencies in modeling grammars matter? *MIS Quarterly*, 35(1):57.

Reid, F. and Harrigan, M. (2012). An analysis of anonymity in the bitcoin system. In *Security and Privacy in Social Networks*, pages 197–223. Springer.

Richter, B., Mengelkamp, E., and Weinhardt, C. (2018). Maturity of blockchain technology in local electricity markets. In *International Conference on the European Energy Market*, pages 1–6.

Rimba, P., Tran, A. B., Weber, I., Staples, M., Ponomarev, A., and Xu, X. (2017). Comparing blockchain and cloud services for business process execution. In *2017 IEEE International Conference on Software Architecture (ICSA)*, pages 257–260.

Rimba, P., Tran, A. B., Weber, I., Staples, M., Ponomarev, A., and Xu, X. (2018). Quantifying the cost of distrust: Comparing blockchain and cloud services for business process execution. *Information Systems Frontiers*, pages 1–19.

Risius, M. and Spohrer, K. (2017). A blockchain research framework. *Business & Information Systems Engineering*, 59(6):385–409.

Rizk, A., Bergvall-Kåreborn, B., and Elragal, A. (2018). Towards a taxonomy for data-driven digital services. In *Proceedings of the 51st Hawaii International Conference on System Sciences*.

Rogaway, P. and Shrimpton, T. (2004). Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In *International workshop on fast software encryption*, pages 371–388.

Rohrer, E., Laß, J.-F., and Tschorsch, F. (2017). Towards a concurrent and distributed route selection for payment channel networks. In *European Symposium on Research in Computer Security*, Data Privacy Management, Cryptocurrencies and Blockchain Technology, pages 411–419.

Ron, D. and Shamir, A. (2013). Quantitative analysis of the full bitcoin transaction graph. In *Financial Cryptography and Data Security*, pages 6–24. Springer.

Rother, M. and Shook, J. (2003). *Learning to see: value stream mapping to add value and eliminate muda*. Lean Enterprise Institute.

Sahoo, M. S. and Baruah, P. K. (2018). Hbasechaindb–a scalable blockchain framework on hadoop ecosystem. In *Asian Conference on Supercomputing Frontiers*, pages 18–29.

Salimitari, M., Chatterjee, M., Yuksel, M., and Pasiliao, E. (2017). Profit maximization for bitcoin pool mining: A prospect theoretic approach. In *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*, pages 267–274.

Sarkintudu, S. M., Ibrahim, H. H., and Abdwahab, A. B. (2018). Taxonomy development of blockchain platforms: Information systems perspectives. In *AIP Conference Proceedings*, volume 2016, page 020130.

Sastry, N., Crowcroft, J., and Sollins, K. R. (2007). Architecting citywide ubiquitous wi-fi access. In *ACM Workshops on HotNets-VI*, pages 1–7.

Scherer, M. (2017). *Performance and Scalability of Blockchain Networks and Smart Contracts*. PhD thesis, Umeå University, Faculty of Science and Technology, Department of Computing Science.

Schmitt, R., Humphrey, S., Ellerich, M., and Groggert, S. (2015). Kapazitätsmarkt – Ressourcenhandel für die Produktion. Eine Cloud-basierte Plattform zum unternehmensübergreifenden Austausch von Produktionskapazitäten. *Industrie 4.0 Management*, 31(4):30–34.

Sen, A. (1976). Poverty: an ordinal approach to measurement. *Econometrica: Journal of the Econometric Society*, pages 219–231.

Seufert, M., Burger, V., and Hoßfeld, T. (2013). Horst - home router sharing based on trust. In *9th International Conference on Network and Service Management (CNSM 2013)*, pages 402–405.

Shi, F., Qin, Z., and McCann, J. A. (2017). Oppay: Design and implementation of a payment system for opportunistic data services. In *37th International Conference on Distributed Computing Systems (ICDCS)*, pages 1618–1628.

Shin, S. I., Kim, J. B., Hall, D., and Lang, T. (2019). What information propagates among the public when an initial coin offering (ico) is initiated? a theory-driven approach. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*.

Shou, W., Wang, J., Wu, P., Wang, X., and Chong, H.-Y. (2017). A cross-sector review on the use of value stream mapping. *International Journal of Production Research*, 55(13):3906–3928.

Sikorski, J. J., Haughton, J., and Kraft, M. (2017). Blockchain technology in the chemical industry: Machine-to-machine electricity market. *Applied Energy*, 195:234–246.

Singh, A., Parizi, R. M., Zhang, Q., Choo, K.-K. R., and Dehghantanha, A. (2020). Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities. *Computers & Security*, 88:101654.

Singhal, N., Sharma, M. K., Samant, S. S., Goswami, P., and Reddy, Y. A. (2020). *Smart KYC Using Blockchain and IPFS*, volume 643, pages 77–84. Springer.

Sivaraman, V., Venkatakrishnan, S. B., Ruan, K., Negi, P., Yang, L., Mittal, R., Fanti, G., and Alizadeh, M. (2020). High throughput cryptocurrency routing in payment channel networks. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 777–796.

Sleiman, M. D., Lauf, A. P., and Yampolskiy, R. (2015). Bitcoin message: Data insertion on a proof-of-work cryptocurrency system. In *Cyberworlds (CW), 2015 International Conference on*, pages 332–336.

Sobolev, A. and Schneider, L. (2019). Iota – industry marketplace technical documentation. `https://industry.iota.org/files/Industry_Marketplace_Technical_Documentation.pdf`. Accessed: 2020-10-13.

Somin, S., Gordon, G., and Altshuler, Y. (2018). Network analysis of ERC20 tokens trading on ethereum blockchain. In Morales, A. J., Gershenson, C., Braha, D., Minai, A. A., and Bar-Yam, Y., editors, *Unifying Themes in Complex Systems IX*, pages 439–450.

Stein, N., Flath, C. M., and Walter, B. (2019). Towards open production: Designing a marketplace for 3d-printing capacities. In *ICIS 2019 Proceedings*.

Sukhwani, H., Martínez, J. M., Chang, X., Trivedi, K. S., and Rindos, A. (2017). Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric). In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, pages 253–255.

Swan, M. (2015). *Blockchain: Blueprint for a New Economy*. O'Reilly.

Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*, 2(9).

Tasca, P. and Tessone, C. J. (2019). A taxonomy of blockchain technologies: Principles of identification and classification. *Ledger*, 4.

Team Rocket (2018). Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies. https://ipfs.io/ipfs/QmUy4jh5mGNZvLkjies1RWM4YuvJh502FYopNPVYwrRVGV. Accessed: 2018-12-04.

Theil, H. (1967). Economics and information theory. In *Studies in Mathematical and Managerial Economics*. Amsterdam: North-Holland.

Tian, F. (2016). An agri-food supply chain traceability system for china based on RFID blockchain technology. In *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–6.

TomoChain R&D Team (2018). Tomochain: Masternodes designtechnical white paper version 1.0. `https://tomochain.com/docs/technical-whitepaper--1.0.pdf`. Accessed: 2020-09-20.

Tönnissen, S. and Teuteberg, F. (2018). Towards a taxonomy for smart contracts. In *ECIS 2018 Research Papers*.

Toyoda, K., Mathiopoulos, P. T., Sasase, I., and Ohtsuki, T. (2017). A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain. *IEEE Access*, 5:17465–17477.

Trillo, M. (2019). Stress test prepares visanet for the most wonderful time of the year. `https://misc.visa.com/blogarchives/us/2013/10/10/stress-test-prepares-visanet-for-the-most-wonderful-time-of-the-year/index.html`. Accessed: 2019-01-14.

Trček, D. (2018). *Computational Trust and Reputation Management*, pages 21–54. Springer.

Tschorsch, F. and Scheuermann, B. (2016). Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials*, 18(3):2084–2123.

van Engelenburg, S., Janssen, M., and Klievink, B. (2019). Design of a software architecture supporting business-to-government information sharing to improve public safety and security. *Journal of Intelligent Information Systems*, 52(3):595–618.

Venable, J., Pries-Heje, J., and Baskerville, R. (2012). A comprehensive framework for evaluation in design science research. In *International Conference on Design Science Research in Information Systems*, Design Science Research in Information Systems. Advances in Theory and Practice, pages 423–438.

Venable, J., Pries-Heje, J., and Baskerville, R. (2016). FEDS: A framework for evaluation in design science research. *European Journal of Information Systems*, 25(1):77–89.

Victor, F. and Lüders, B. K. (2019). Measuring ethereum-based ERC20 token networks. In Goldberg, I. and Moore, T., editors, *Financial Cryptography and Data Security*, pages 113–129.

Vidales, P., Manecke, A., and Solarski, M. (2009). Metropolitan public wifi access based on broadband sharing. In *Mexican International Conference on Computer Science*, pages 146–151.

Vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., Cleven, A., et al. (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process. In *ECIS 2009 Proceedings*, volume 9, pages 2206–2217.

Vranken, H. (2017). Sustainability of bitcoin and blockchains. *Current opinion in environmental sustainability*, 28:1–9.

Wang, W., Hoang, D. T., Xiong, Z., Niyato, D., Wang, P., Hu, P., and Wen, Y. (2018a). A survey on consensus mechanisms and mining management in blockchain networks. *arXiv preprint arXiv:1805.02707*, pages 1–33.

Wang, X., Zha, X., Yu, G., Ni, W., Liu, R. P., Guo, Y. J., Niu, X., and Zheng, K. (2018b). Attack and defence of ethereum remote APIs. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6.

Wang, Y., Han, J. H., and Beynon-Davies, P. (2019). Understanding blockchain technology for future supply chains: a systematic literature review and research agenda. *Supply Chain Management: An International Journal*, 24(1):62–84.

Wang, Y. and Malluhi, Q. M. (2019). The limit of blockchains: Infeasibility of a smart obama-trump contract. *Commun. ACM*, 62(5):64–69.

Wang, Z., Jin, H., Dai, W., Choo, K.-K. R., and Zou, D. (2020). Ethereum smart contract security research: survey and future research opportunities. *Frontiers of Computer Science*, 15(2).

Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., and Mendling, J. (2016). Untrusted business process monitoring and execution using blockchain. In La Rosa, M., Loos, P., and Pastor, O., editors, *Business Process Management. BPM 2016. Lecture Notes in Computer Science*, pages 329–347. Springer.

Westerkamp, M., Victor, F., and Küpper, A. (2018). Blockchain-based supply chain traceability: Token recipes model manufacturing processes. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1595–1602.

Westerkamp, M., Victor, F., and Küpper, A. (2019). Tracing manufacturing processes using blockchain-based token compositions. *Digital Communications and Networks*.

Wiendahl, H.-P. and Lutz, S. (2002). Production in networks. *CIRP Annals*, 51(2):573–586.

Wieninger, S., Schuh, G., and Fischer, V. (2019). Development of a blockchain taxonomy. In *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 1–9.

Wöhrer, M. and Zdun, U. (2018). Design patterns for smart contracts in the ethereum ecosystem. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1513–1520.

Wohrer, M. and Zdun, U. (2018). Smart contracts: security patterns in the ethereum ecosystem and solidity. In *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, pages 2–8.

Worley, C. and Skjellum, A. (2018). Blockchain tradeoffs and challenges for current and emerging applications: Generalization, fragmentation, sidechains, and scalability. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1582–1587.

Wüst, K. and Gervais, A. (2018). Do you need a blockchain? In *Crypto Valley Conference on Blockchain Technology*, pages 45–54.

Xu, J., Xue, K., Li, S., Tian, H., Hong, J., Hong, P., and Yu, N. (2019). Healthchain: A blockchain-based privacy preserving scheme for large-scale health data. *IEEE Internet of Things Journal*, 6(5):8770–8781.

Xu, X., Pautasso, C., Zhu, L., Gramoli, V., Ponomarev, A., Tran, A. B., and Chen, S. (2016a). The blockchain as a software connector. In *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 182–191.

Xu, X., Weber, I., Zhu, L., Staples, M., Bosch, J., Bass, L., Pautasso, C., and Rimba, P. (2017). A taxonomy of blockchain-based systems for architecture design. In *IEEE International Conference on Software Architecture*, pages 243–252.

Xu, Y., Li, Q., Min, X., Cui, L., Xiao, Z., and Kong, L. (2016b). E-commerce blockchain consensus mechanism for supporting high-throughput and real-time transaction. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 490–496.

Xue, J., Xu, C., Zhang, Y., and Bai, L. (2018). Dstore: A distributed cloud storage system based on smart contracts and blockchain. In Vaidya, J. and Li, J., editors, *Algorithms and Architectures for Parallel Processing*, pages 385–401.

Yeow, K., Gani, A., Ahmad, R. W., Rodrigues, J. J., and Ko, K. (2017). Decentralized consensus for edge-centric internet of things: A review, taxonomy, and research issues. *IEEE Access*, 6:1513–1524.

Yin, R. K. (2017). *Case study research and applications: Design and methods*. Sage.

Yli-Huumo, J., Ko, D., Choi, S., Park, S., and Smolander, K. (2016). Where is current research on blockchain technology?—a systematic review. *PLOS ONE*, 11(10):1–27.

Zavolokina, L., Miscione, G., and Schwabe, G. (2020). Buyers of 'lemons': How can a blockchain platform address buyers' needs in the market for 'lemons'? *Electronic Markets*, 30(2):227–239.

Zhang, T. and Wang, L. (2017). A decentralized dark pool exchange providing atomic swaps for ethereum-based assets and bitcoin. `https://republicprotocol.github.io/whitepaper/republic-whitepaper.pdf`. Accessed: 2021-03-13.

Zheng, Z., Xie, S., Dai, H.-N., Chen, W., Chen, X., Weng, J., and Imran, M. (2020). An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, 105:475–491.

Zheng, Z., Xie, S., Dai, H.-N., Chen, X., and Wang, H. (2018). Blockchain challenges and opportunities: a survey. *International Journal of Web and Grid Services*, 14(4):352–375.

Zhong, H., Sang, Y., Zhang, Y., and Xi, Z. (2020). Secure multi-party computation on blockchain: An overview. In Shen, H. and Sang, Y., editors, *Parallel Architectures, Algorithms and Programming. PAAP 2019. Communications in Computer and Information Science*, pages 452–460. Springer.

Zou, W., Lo, D., Kochhar, P. S., Le, X.-B. D., Xia, X., Feng, Y., Chen, Z., and Xu, B. (2019). Smart contract development: Challenges and opportunities. *IEEE Transactions on Software Engineering*.