Spring 4-29-2019

PUBLIC BLOCKCHAIN SCALABILITY: ADVANCEMENTS, CHALLENGES AND THE FUTURE

Amritraj .

PUBLIC BLOCKCHAIN SCALABILITY: ADVANCEMENTS, CHALLENGES AND THE FUTURE

A Thesis Presented to

The Faculty of the Department of Software Engineering and Game Development

by

Amritraj

In Partial Fulfillment

of Requirements for the Degree

Master of Science in Software Engineering

Kennesaw State University

May 2019

PUBLIC BLOCKCHAIN SCALABILITY: ADVANCEMENTS, CHALLENGES AND THE FUTURE

Approved:

Dr. Reza M. Parizi

Your Advisor's Name

Dr. Chi Zhang

Department Chairperson's Name

Dr. Jon Preston

Dean's Name

In presenting this thesis as a partial fulfillment of the requirements for an advanced degree from Kennesaw State University, I agree that the university library shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish, this thesis may be granted by the professor under whose direction it was written, or, in his absence, by the dean of the appropriate school when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from or publication of, this thesis which involves potential financial gain will not be allowed without written permission.

Amritraj

____

Notice to Borrowers

Unpublished theses deposited in the Library of Kennesaw State University must be used only in accordance with the stipulations prescribed by the author in the preceding statement.

The author of this thesis is:

Amritraj

1000 Chastain Road,
Kennesaw GA 30144
USA

The director of this thesis is:

Dr. Paola Spoletini,

Department of Software Engineering and Game Development
Kennesaw State University
Building J, Office – 375
1100 South Marietta Pkwy
Marietta GA 30060
USA

Users of this thesis not regularly enrolled as students at Kennesaw State University are required to attest acceptance of the preceding stipulations by signing below. Libraries borrowing this thesis for the use of their patrons are required to see that each user records here the information requested.

Name of user	Address	Date	Type of use (examination only or copying)

PUBLIC BLOCKCHAIN SCALABILITY: ADVANCEMENTS, CHALLENGES AND THE FUTURE

An Abstract of

A Thesis Presented to

The Faculty of the Department of Software Engineering and Game Development

by

Amritraj

Bachelor of Technology in Electrical and Electronics Engineering,
Visvesvaraya National Institute of Technology, 2016

In Partial Fulfillment

of Requirements for the Degree

Master of Science in Software Engineering

Kennesaw State University

May 2019

Abstract

In the last decade, blockchain has emerged as one of the most influential innovations in software architecture and technology. Ideally, blockchains are designed to be architecturally and politically decentralized, similar to the Internet. But recently, public and permissionless blockchains such as Bitcoin and Ethereum have faced stumbling blocks in the form of scalability. Both Bitcoin and Ethereum process fewer than 20 transactions per second, which is significantly lower than their centralized counterpart such as VISA that can process approximately 1,700 transactions per second. In realizing this hindrance in the wide range adoption of blockchains for building advanced and large scalable systems, the blockchain community has proposed first- and second-layer scaling solutions including Segregated Witness (Segwit), Sharding, and two-way pegged sidechains. Although these proposals are innovative, they still suffer from the blockchain trilemma of scalability, security, and decentralization. Moreover, at this time, little is known or discussed regarding factors related to design choices, feasibility, limitations and other issues in adopting the various first- and second-layer scaling solutions in public and permissionless blockchains. Hence, this thesis provides the first comprehensive review of the state-of-the-art first- and second-layer scaling solutions for public and permissionless blockchains, identifying current advancements and analyzing their impact from various viewpoints, highlighting their limitations and discussing possible remedies for the overall improvement of the blockchain domain.

PUBLIC BLOCKCHAIN SCALABILITY: ADVANCEMENTS, CHALLENGES AND THE FUTURE

A Thesis Presented to

The Faculty of the Department of Software Engineering and Game Development

by

Amritraj

In Partial Fulfillment

of Requirements for the Degree

Master of Science in Software Engineering

Advisor: Dr. Reza Parizi

Kennesaw State University

May 2019

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

In the last decade, blockchain has emerged as one of the most influential innovations in software architecture and technology. Ideally, public blockchains (such as Bitcoin (Nakamoto, 2008) and Ethereum (Wood, 2014)) are designed to be architecturally and politically decentralized (Robinson, 2018), similar to the Internet. They enable trusted transactions among several untrusted participants on a network without a need for a trusted central authority or a third party. As a result of this, blockchains are now employed in various computing and business domains such as cloud computing, supply-chains, Internet of Things (IoT), finance, and many others (Miller, 2018), (Fiaidhi, Mohammed, & Mohammed, 2018), (Zhou, Wang, Sun, & Lv, 2018), (Mylrea & Gourisetti, 2018). Alongside its industrial counterpart, academic research in the domain is also increasing rapidly, especially in applying blockchain technology for developing decentralized solutions and applications (S. Yu et al., 2018), (Lou, Zhang, Qi, & Lei, 2018), (Kan et al., 2018), (Robinson, 2018). Additionally, research in recognizing technical challenges in the blockchain domain (Reza M Parizi, Amritraj, & Dehghantanha, 2018), (Atzei, Bartoletti, & Cimoli, 2017), (Giaglis et al., 2017) has also been growing steadily in the recent years along with studies that provide possible solutions to these challenges including, formal verification of smart contracts (Bhargavan et al., 2016), (Amani, Bégel, Bortin, & Staples, 2018), (Abdellatif & Brousmiche, 2018), scalability improvement of blockchains (Dennis, Owenson, & Aziz, 2016) and defining atomic cross-chain swap protocols (Herlihy, 2018).

1.1 <u>The Problem</u>

This growth in interest from both the enterprise and research communities in blockchain related technologies has seen a major stumbling block in recent years in the form of scalability, which has quickly become the core problem surrounding blockchains. Scalability of a system or network is defined as its capacity to grow in size and manage increased demand from its user-base (Duboc, Rosenblum, & Wicks, 2006). In other words, scalable systems can be efficiently enlarged to accommodate increased usage and activity from their user-base.

State-of-the-art blockchains are hindered by scalability due to the following two reasons: 1) There are limits on the number of transactions that a blockchain network can process and 2) blockchains are designed to provide solutions to a specific problem, they often tend to be vanilla in nature and hence, generally lack many features that traditional state-of-the-art centralized systems offer out of the box. For instance, a centralized database system can be built to provide several functionalities at once such as supply chain tracking, financial payments, and remote shopping, whereas on the other hand, a blockchain such as Bitcoin is built to provide only one functionality, i.e. to facilitate trustless peer-to-peer financial transactions within its network. Hence, it cannot store supply-chain information or provide the comforts of remote shopping to a user on its network by itself. In fact, it is extremely difficult if not impossible to implement a universal blockchain that could solve all problems with the current technology.

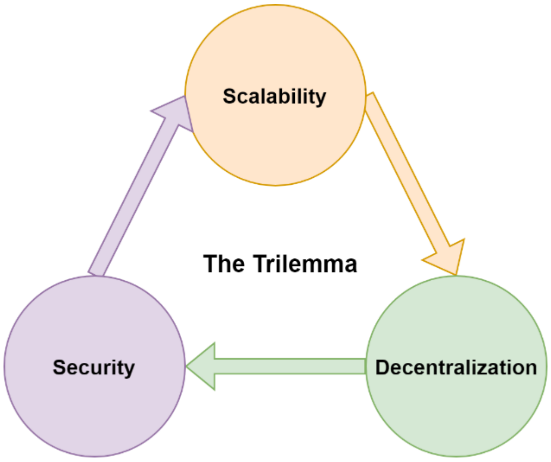


Figure 1: The Blockchain Trilemma

State-of-the-art blockchains face a trilemma of scalability, security, and decentralization (Figure 1). Blockchains can only have two of these three attributes:

- **Scalability** concerns with the ability of a blockchain to process transactions in bulk. If public blockchains are to become mainstream, then they need to be able to handle the scenario in which there are millions of users on the network.
- **Security** is concerned with the immutability of the blockchain and its robustness to attacks such as Sybil (Douceur, 2002), Distributed Denial of Service (DDoS) (Feinstein, Schnackenberg, Balupari, & Kindred, 2003) and 51% attacks[1].
- **Decentralization** is the core tenant upon which the blockchain community is built upon which provides censorship resistance and allows any user to participate in a decentralized environment without prejudice.

Public and permissionless blockchains such as Bitcoin and Ethereum were designed around decentralization and security as core features. However, this came at an expense of scalability as both Bitcoin and Ethereum have extremely low throughput when it comes to transaction processing rates. For instance, Bitcoin can only process approximately 7 transactions per second[2] and Ethereum can process approximately 15 transactions per second[3]. When compared to their centralized counterparts such as VISA which can process approximately 1,700 transactions per second[4], these public blockchains do not post impressive numbers.

[1] https://medium.com/chainrift-research/bitcoins-attack-vectors-51-attacks-a96deac43774
[2] https://blockexplorer.com/blocks
[3] https://etherscan.io/
[4] https://altcointoday.com/bitcoin-ethereum-vs-visa-paypal-transactions-per-second/

1.2 Solutions Proposed by the Community

Realizing this hindrance in the growth and further adoption of blockchains for building advanced and complicated software systems, both the research and enterprise communities have proposed first- and second-layer scaling solutions for blockchains.

1.2.1 First-layer Scaling Solutions

First-layer scaling solutions are the ones that require changes to the source code of a blockchain (Dolce, 2018). These solutions propose enhancements to the core characteristics and features of a blockchain. Some examples of first layer solutions include increasing the block size limit of Bitcoin from 1MB to 10 MB or reducing the block creation time from 10 minutes to 5 minutes. Some other ways in which first layer scaling solutions could be implemented are as follows:

- **Segregated witness:** Segregated Witness (Dolce, 2018) is a proposed first-layer scaling solution for the Bitcoin protocol that changes the way data is stored on the Bitcoin blockchain. The proposal is to remove the signature data from each transaction of a block to free up space for more transactions to be included in Bitcoin's current 1 Megabyte block size. In its current implementation the signature data in Bitcoin takes up almost 70% of the block space which leaves behind little space for transactions. Therefore, removing it would save tremendous space that allows more transactions to be included in the block.

- **Sharding:** Sharding (Dolce, 2018) proposes the breaking down or dividing blockchains into smaller manageable parts called *shards*, that run simultaneous (parallel) to one another. Each shard is in-charge of processing transactions within the group, thereby increasing processing output across the board. Fragmenting the network into many different small parts allow the Ethereum blockchain to function as the sum of its parts, rather than being limited by the speed of each individual node.

1.2.2 Second-layer Scaling Solutions

Second-layer scaling solutions are the ones that propose the implementation of secondary protocols on top a pre-existing primary blockchain called the primary chain or the mainchain (Dolce, 2018). This reduces network congestion and saves space as the transactions are off-loaded onto the secondary protocols. Second-layer solutions involve the proposal of sidechains:

- **Sidechains:** are secondary blockchains which are connected to other blockchains by means of a two-way peg. A two-way peg is a mechanism that allows the bidirectional transfer of assets between the main chain and the sidechain at a fixed or pre-deterministic exchange rate. Sidechains may have their own protocol and implementation which can be completely different from the main blockchain. Such adjustability provides users of a network with the flexibility to access various other functionalities and features offered on a sidechain by using the assets they already own on the main blockchain. Furthermore, sidechains are isolated from the main blockchain in such a way that in the case of a cryptographic break (or a maliciously designed sidechain), the damage is entirely confined to the sidechain itself.

1.3 Contributions of this Work

Although promising, the existing scaling solutions are still in the state of infancy and to this date, little is known or discussed regarding factors related to design choices, feasibility, limitations and other issues in adopting these scaling solutions. Moreover, there is a lack of comparative and empirical studies both in academic and industrial environments to analyze such protocols and multi-blockchain systems in a comprehensive manner. Hence, the motivation of this research is to provide the first comprehensive analysis of the first- and second-layer scaling solutions for public and permissionless blockchains to understand the design choices, advancements, use cases, and limitations of such solutions. The specific contributions of this research are as follows:

- Provide a thorough analysis of first and second-layer blockchain scaling solutions.
- Analyze the most common design choices for these scaling solutions by highlighting their advantages and disadvantages
- Provide a comprehensive review of current state-of-the-art scalability enhancing platforms based on their use cases, consensus mechanisms, asset transfer protocol and limitations with horizontal comparison
- Provide an overview of new and upcoming innovative scaling solutions and frameworks
- Identify open issues and discuss possible solutions to mitigate those issues with state-of-the-art blockchain scaling solutions

The rest of this thesis is structured as follows: Chapter 2 describes the research protocol and methodology used and the research questions answered by this work. Chapter 3 investigates the first layer scaling solutions and answers the research questions raised for such proposals. Chapter 4 discusses the second layer scaling solutions and answers the research questions raised for such solutions. Chapter 5 sheds light on open issues and limitations while proposing future directions and possible solutions to mitigate these issues and limitations. Finally, Chapter 6 provides the conclusion of this work.

Chapter 2. Research Methodology

In this work, we analyze the state-of-the-art first- and second-layer scaling solutions and platforms proposed for improving the scalability of public and permissionless blockchains including Bitcoin and Ethereum. For this work, we adopted an Systematic Literature Review (SLR) based information and data gathering approach which helped us in identification, evaluation and interpretation of all available research, solutions or platforms relevant to one or more research questions which are mentioned below (Kitchenham, 2004), (Kitchenham & Charters, 2007). Based on these research questions (RQs) (see section 2.2) we designed a custom data gathering protocol for the identification of relevant resources and platforms for this research.

2.1 Research Protocol

An important characteristic of every research work is the identification of a problem and a protocol to solve that problem. Figure 3 outlines the protocol for the completion of this thesis.

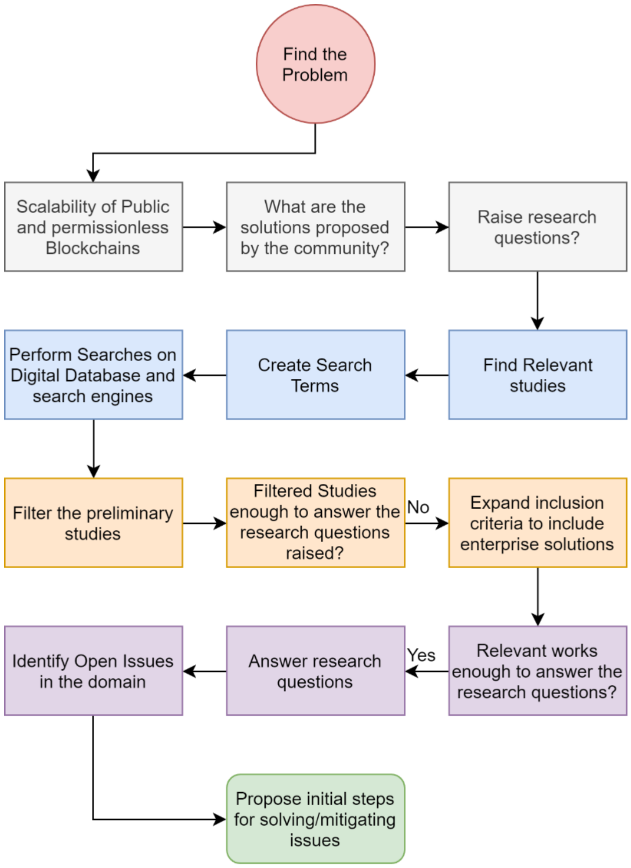

Find the Problem
Scalability of Public and permissionless Blockchains
What are the solutions proposed by the community?
Raise research questions?
Find Relevant studies
Create Search Terms
Perform Searches on Digital Database and search engines
Filter the preliminary studies
Filtered Studies enough to answer the research questions raised?
No
Expand inclusion criteria to include enterprise solutions
Relevant works enough to answer the research questions?
Yes
Answer research questions
Identify Open Issues in the domain
Propose initial steps for solving/mitigating issues

Figure 2: Thesis Protocol

It can be seen from the figure that our work started with the identification of a problem (in this case, scalability of public and permissionless blockchains), which led us into raising research questions regarding the state-of-the-art solutions. We then collected, filtered and expanded our inclusion criteria to include all relevant research resources and proceeded to answer the raised research questions which allowed us to identify open issues and ultimately, propose initial steps to solve or mitigate these issues. In the following sections of this chapter, we will enlist the research question raised and our data gathering strategy to answer these questions.

2.2 Research Questions (RQs)

The most compelling motivation behind this research is to answer the following research questions (RQs) based on the proposed first- and second-layer scaling solutions and platforms.

2.2.1 First Layer Scaling Solutions Research Questions (RQs)

- **RQ1:** What are the Design Choices available for implementing first layer scaling solutions?
- **RQ2:** What are the limitations of these designs?
- **RQ3:** What are the problems associated with implementing First Layer scalability solutions in public blockchains?

2.2.2 Second Layer Scaling Solutions Research Questions (RQs)

- **RQ1:** What are the available design choices for implementing two-way pegs?
- **RQ2:** What are the advantages and limitations of these design choices?
- **RQ3:** Which state-of-the-art platforms are implementing sidechains?
    - ➢ **RQ3a:** What are the use cases of these platforms?
    - ➢ **RQ3b:** How does asset transfer take place on these platforms?
    - ➢ **RQ3c:** What consensus mechanism do these platforms utilize?
    - ➢ **RQ3d:** How do these platforms impact the scalability of their mainchain?
    - ➢ **RQ3e:** What are the limitations of these platforms?

To answer these questions, we designed a custom protocol to search and identify all relevant resources such as journal articles, conference papers, workshop articles, etc. in the realms of blockchain scalability. In the subsequent sections, we describe the various steps that we performed for filtering the relevant resources to accurately answer the research questions in this section.

2.3 Search Strategy

Once we determined the digital libraries and search engines to be used for gathering relevant resources, we constructed several search terms to be used on these libraries and search engines based on our research questions. Some examples of our search terms are mentioned in Table 1.

Table 1: Search string and terms

Terms	Blockchain, Scalability, Sharding, Segregated witness, Sidechains, Smart Contracts, Bitcoin, interoperability, Ethereum, etc.
Search Terms Used	Ethereum Sharding, Bitcoin scalability, Blockchain Scalability, smart contracts scalability, blockchain interoperability, etc.

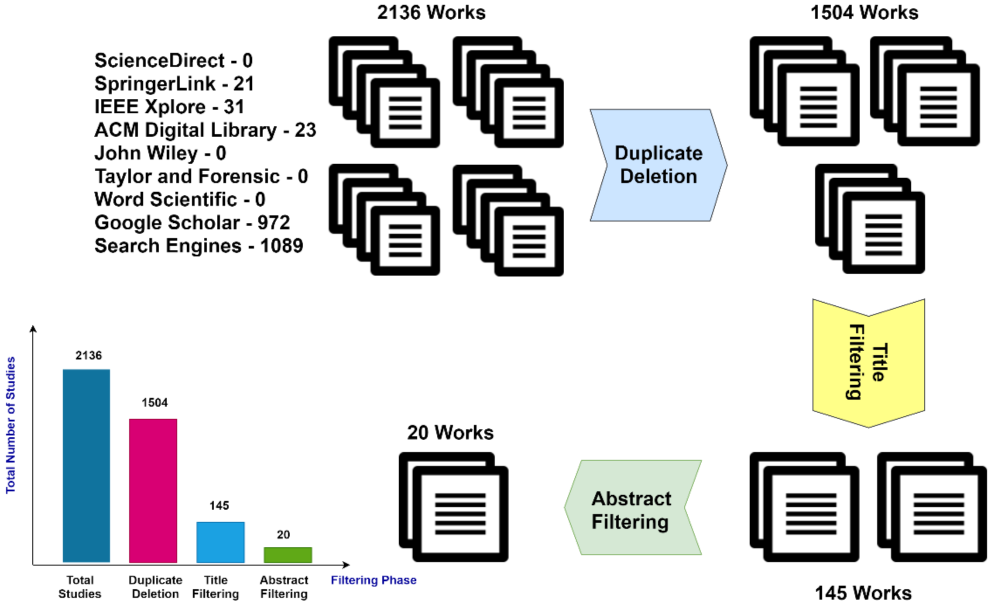


Figure 3: Initial filtering of the relevant works

Next, we performed manual searches with several combinations of search terms on digital libraries and search engines such as Google, Duck-Duck Go and Yahoo which yielded the results as shown in Figure 2. This provided us with the unfiltered preliminary set of works on blockchain scalability.

2.4 Preliminary set of works

As mentioned in the previous section, we conducted manual searches based on search terms and keywords on several identified digital libraries and search engines to identify and collect the preliminary set of works. The results from these searches are summarized in Figure 2, which shows the total number of preliminary studies acquired from each database and search engine. We obtained a total of 2136 preliminary studies from our search. These studies were carefully chosen based on the inclusion and exclusion criteria mentioned in Table 2.

Table 2: Inclusion and Exclusion Criteria for relevant works

Inclusion Criteria	Exclusion criteria
Be published online digital databases such as IEEE, ACM Digital Library, SpringerLink, etc.	Resources not published in English
Studies are in the domain of blockchain scalability	Resources from unreliable online sources
Studies offer technical quality in the presentation of ideas and reviews	Studies with poor presentation quality
Studies used current technical quality aspects	Grey literature, studies with incomplete ideas and poor explanation of concepts

Out of these 2136 studies, only 1047 were from online digital databases namely ScienceDirect, IEEE Xplore, ACM Digital Library, SpringerLink, John Wiley, Taylor and Forensic, Word Scientific and Google Scholar. The remaining 1089 resources were from search engine results such as Google, Yahoo and Duck-Duck Go. It is important to mention at this time the results obtained in the search engines were considerably larger than just 1089 studies but, most of these results covered repetitive topics or were not from trustworthy sources. Hence, after we collected the 2136 preliminary set of works, we started the initial filtering phase where, the collected studies were carefully removed based on duplicate removal, title filtering, and abstract filtering. The studies remaining after each filtering stage is shown in Figure 2.

Table 3: Search results from digital databases

Digital Database	Year	Article Name	Reference
arxiv	2018	Requirements for private Ethereum Sidechains	(Robinson, 2018)
IEEE	2018	The Blockchain for Domain Based Static Sharding	(Yoo, Yim, & Kim, 2018)
IEEE	2018	A Scale-Out Blockchain for Value Transfer with Spontaneous Sharding	(Ren et al., 2018)
IEEE	2018	OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding	(Kokoris-Kogias et al., 2018)
IEEE	2018	A Scalable and Extensible Blockchain Architecture	(Y. Yu, Liang, & Xu, 2018)
IEEE	2018	Challenges and Pitfalls of Partitioning Blockchains	(Fynn & Pedone, 2018)
IEEE	2018	A Game-Theoretic Analysis of Shard-Based Permissionless Blockchains	(Manshaei, Jadliwala, Maiti, & Fooladgar, 2018)
IEEE	2017	A Prototype Evaluation of a Tamper-Resistant High Performance Blockchain-Based Transaction Log for a Distributed Database	(Aniello et al., 2017)
IEEE	2018	Chameleon: A Scalable and Adaptive Permissioned Blockchain Architecture	(He, Su, & Gao, 2018)
IEEE	2018	Blockchain and Scalability	(Chauhan, Malviya, Verma, & Mor, 2018)
IEEE	2018	ProductChain: Scalable Blockchain Framework to Support Provenance in Supply Chains	(Malik, Kanhere, & Jurdak, 2018)
ACM Digital Library	2016	A Secure Sharding Protocol For Open Blockchains	(Luu et al., 2016)
ACM Digital Library	2016	Bringing Secure Bitcoin Transactions to Your Smartphone	(Frey, Makkes, Roman, Ta\"\iani, & Voulgaris, 2016)

ACM Digital Library	2018	RapidChain: Scaling Blockchain via Full Sharding	(Zamani, Movahedi, & Raykova, 2018)
ACM Digital Library	2018	Towards Solving the Data Availability Problem for Sharded Ethereum	(Sel, Zhang, & Jacobsen, 2018)
ACM Digital Library	2016	Bringing Secure Bitcoin Transactions to Your Smartphone	(Frey et al., 2016)
SpringerLink	2018	Pruneable sharding-based blockchain protocol	(Feng et al., 2018)
SpringerLink	2017	Short Paper: Service-Oriented Sharding for Blockchains	(Gencer, van Renesse, & Sirer, 2017)
SpringerLink	2018	A Decentralized Sharding Service Network Framework with Scalability	(Cai, Yang, & Ming, 2018)
SpringerLink	2016	On scaling decentralized blockchains	(Croman et al., 2016)

The initial filtering stages reduced the relevant works to just 20 studies. These studies are shown in Table 3. The next step in the filtering process was content filtering, where we removed studies with irrelevant content in regard to this research by carefully and thoroughly reading each of the 20 studies. This process further reduced the relevant studies into single digits. This research focuses on public and permissionless blockchain such as Ethereum and Bitcoin’s scalability and most of the 20 preliminary studies fell out of the scope of this study as they focus on the scalability of private or permissioned blockchains.

At this stage, we were forced to turn our attention towards the enterprise blockchain community and other online resources published by credible sources and individuals such as CoinDesk, Vitalik Buterin, and other well-established platforms such as the Lightening Network[5] for Bitcoin, etc. Hence, we have answered our research questions based on the results of our thorough investigation and analysis of both resources from online digital databases and other

[5] https://lbtc.io/

credible technical sources including but not limited to white papers, conference presentations, technical talks, etc.

Chapter 3. First Layer Scaling Solutions

First layer scaling solutions are referred to as the scaling solutions that require changes to be made onto the codebase of the blockchain. This entails enhancing the core features and characteristics of the blockchain. Some examples of first layer solutions include increasing the block size limit of Bitcoin from 1MB to 10 MB or reducing the block creation time from 10 minutes to 5 minutes.

In this chapter, we are going to discuss the two major First layer scaling solutions 1) Segregated Witness, as proposed for the Bitcoin blockchain and 2) Sharding, which is proposed for the Ethereum blockchain.

3.1 Segregated Witness

Segregated witness is a protocol upgrade for Bitcoin that changes the way and structure of how data is stored. By removing the signature data for each transaction, it frees up more space and capacity for more transactions to be stored in Bitcoin's 1MB-capacity blocks. The signature data – the digital signature that verifies the ownership and availability of the sender's funds – make up almost 70% of the entire space of a transaction. Therefore, removing it would save tremendous space that allows more transactions to be included in the block (Dolce, 2018).

3.2 Sharding

At the time of writing this thesis, Ethereum, the most prominent smart contract platform in the world, can only process approximately 15 transactions per second. This severe limitation, coupled with the popularity of the platform, leads to high gas prices (the cost of executing a transaction on the network) and long confirmation times. Although a new block is added every 10-20 seconds on Ethereum blockchain, the average time for a transaction to be added on to the blockchain is over 1 minute, according to ETH Gas Station[6]. Thus, low throughput, high gas prices, and high latency have rendered Ethereum unsuitable for building scalable services and applications.

Ethereum's low throughput is based on the fact that each node on the network has to process each transaction that occurs on the platform. To address this limitation, the blockchain community has proposed a few solutions which target the Ethereum protocol. Most of these solutions introduce central entities to process transactions at a high frequency. This is usually done by delegating all the computation to a small subset of powerful nodes. For instance, Thunder[7] runs a single node to process all transactions and claims to achieve approximately 1200 transaction per second which is 100 times faster than current Ethereum capabilities. Other examples of such solutions are Algorand[8], SpaceMesh[9], and Solana[10] who are all attempting to improve the consensus protocols and design of blockchains to process high volumes of transactions each second. In addition to decentralization, another limitation of these solutions is that they are all bounded by the processing capabilities of a single node and hence, are vulnerable to a complete shutdown in case of power failures, natural disasters, etc.

In contrast, the other proposed solution, Blockchain sharding, delegates work such that, each node on the network only performs a subset of the total amount of work in processing a transaction on the blockchain. Sharding is the solution being used by the Ethereum foundation for improving the scalability of the Ethereum platform.

The concept of sharding in the domain blockchains comes from the world of databases where it is used to make servers and databases more efficient. This is done by storing each shard

[6] https://ethgasstation.info/
[7] https://www.thundercore.com/
[8] https://www.algorand.com/
[9] https://spacemesh.io/
[10] https://solana.com/

which a horizontal chunk of a database on a separate server instance consequently, spreading the load on the server.

In blockchains, the idea is to have each node store only a part of the blockchain (called a shard in this context), instead of the entire blockchain itself. This means that a node that stores a shard only maintains information on that shard in a shared manner, thus, maintain decentralization. However, each node doesn't load the information on the entire blockchain, thus helping in scalability.

Proof of Work (PoW)[11] consensus algorithm cannot be used in conjunction with sharding, this is because all participant nodes cannot be involved in transaction validation as each node only has information regarding a particular shard i.e. the shard it belongs to. Thus, the ideas that have been proposed for blockchain sharding are based on consensus mechanisms like Proof of Stake (PoS)[12].

In Proof of Stake consensus mechanism transaction validation responsibilities are undertaken by specific designated nodes called "stakers". Stakers are required to stake their digital assets such as tokens to participate in transaction validation. A staker earns a part or the entirety of the transaction fees upon transaction validation. The number of transaction validations allowed for a staker is directly proportional to the amount and duration of their assets on stake. Additionally, the Proof of Stake consensus mechanism provides the following advantages over the Proof of Work:

- A subset of all nodes validates each transaction instead of the entire network nodes.
- Absence of mining eliminates the requirement for expensive special-purpose, high-performance hardware including CPUs, GPUs, and SSDs. This consequently decreases the energy costs.

[11] https://cointelegraph.com/explained/proof-of-work-explained
[12] https://blockgeeks.com/guides/proof-of-work-vs-proof-of-stake/

- It is easy to identify loyal and honest validators based on the amount and duration of the digital assets staked.

Each shard in a sharded blockchain identifies stakers who assume the transaction validation responsibilities for that shard. Since transaction validation is done by honest and loyal stakers, it is easy to presume that the security of the blockchain is still well preserved when compared to blockchains with Proof of Work mechanisms.

3.3 Answers to Research Questions (RQs)

In this section, we answer the research questions for first-layer scaling solutions raised in Chapter 2 to discuss the design choices for implementing such solutions, their impact on scalability and limitations. Finally, we will discuss the challenges associated with implementing first-layer scaling solutions in public blockchains such as Bitcoin and Ethereum.

3.3.1 RQ1: What are the Design Choices available for implementing first-layer scaling solutions?

To answer this research question, we will discuss multiple ways in which first layer scaling solutions can be implemented. We begin the discussion with a thorough look into the design of segregated witness:

3.3.1.1 Segregated Witness Design

To analyze the idea behind segregated witness, we need to first explain how a current transaction takes place on the Bitcoin network. This would allow us to demonstrate the potential impact of segregated witness on scalability, in particular, Bitcoin’s transaction throughput.

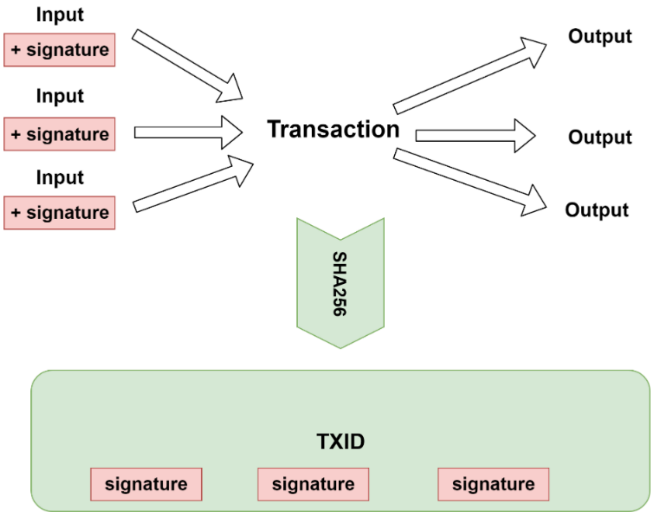


Figure 4: Non-segwit transaction

As shown in Figure 4, with non-Segwit transactions, the signatures needed to unlock the inputs are included along with the rest of the transaction data in the hash to get the transaction ID (TXID) (McManus, 2017). Non-segwit transactions include the signatures in the hash to get the TXID. These transactions are then included in each block up to the 1MB limit in structures called Merkle Trees.

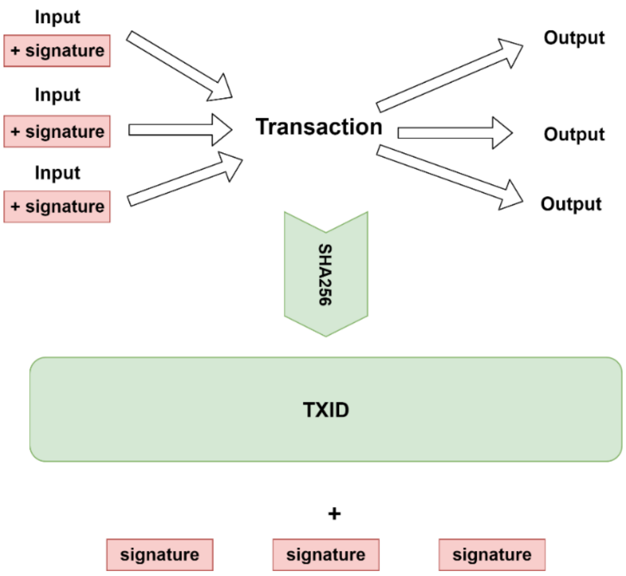


Figure 5: Segwit Transaction

On the other hand, as can be seen in Figure 5, with Segwit transactions, we have two fundamental changes. Segwit transactions do not hash the signature data. Signature data is stored as “witness” data in the block.

- The signature data is not included in the hash to form the TXID. Signatures are still stored in the block with the transactions as part of “witness” data, but they are longer included in the TXID hash.
- The block size limit is changed from 1MB (1,000,000 bytes) to a 4,000,000 “weight” limit, an arbitrary new metric. A normal byte in a transaction has a weight of 4 while a witness byte has a weight of 1.

Hence, there are two significant benefits of segwit transactions, which we will discuss now.

3.3.1.1.1 Transaction Malleability

With Bitcoin transactions before Segwit, there was a bug in the software called “transaction malleability”. As we know by now, the TXID pre-Segwit is the result of hashing the transaction data including the signatures. Although there were checks and balances to ensure that the inputs and outputs couldn’t be changed (i.e. the parties in a transaction and the amounts of Bitcoin being sent), the signature used to unlock the inputs could be modified slightly (such that it was still a valid signature) but would completely change the TXID when hashed. With the signature no longer a part of the TXID in Segwit, transaction malleability is no longer a problem.

3.3.1.1.2 Increased Block Capacity

By changing the block size limit from a byte’s limit to a new 4,000,000 weight limit, the number of transactions allowed in each block can be increased while maintaining backward compatibility with the existing cap of 1MB per block. How? Simple math. Our equation for Segwit nodes is as follows:

$$4 * normal\ bytes + (1 * witness\ byte) = 4{,}000{,}000$$

Non-Segwit nodes in the network will not be able to see the witness data, making their equation:

$$4 * normal\ bytes = 4{,}000{,}000$$

$$normal\ bytes = 1{,}000{,}000$$

So, with Segwit, we’ll never go over the 1MB block size limit on older nodes, making this backward compatible. Only Segwit nodes will be able to see the signature data, but existing nodes will still have access to all the transactions.

Segwit won’t bring about nodes with a block size of 4MB though as blocks aren’t comprised 100% of witness bytes. The actual size of the blocks will depend on the adoption rate of

Segwit, although the expected average block size will be around 1.7–2MB based on tests showing around 60% of a transaction to be witness data.

3.3.1.2 Sharding Design Choices

Now, we will discuss the various design choices for implementing sharding that has been proposed by the blockchain community:

3.3.1.2.1 Scaling by Thousand Altcoins

The co-founder of the Ethereum platform, Vitalik Buterin, introduced the concept of “Scaling by a thousand Altcoins” in his presentation (Buterin, 2017). This design proposes the use of multiple blockchains instead of a single blockchain. Each blockchain in this multi-blockchain environment consists of its own set of validators and is known as a shard. For the rest of this discussion, we use a generic term “validator” to refer to participants or entities that validate transactions and produce new blocks, with the help of a suitable consensus mechanism such as mining with Proof of Work, or via a voting-based mechanism. For now, let’s assume that the shards never communicate with each other. Although this design is simple, it is sufficient to highlight some of the major challenges in sharding.

3.3.1.2.1.1 Validator partitioning and Beacon chains

The first challenge is the weakening of security of each shard as having their own validator makes them several magnitudes insecure than the entire chain. So, if a non-sharded chain with X validators decides to hard-fork into a sharded blockchain, and splits X validators across 10 shards, each shard now only has X/10 validators, and corrupting one shard only requires corrupting (51/10) % or 5.1% of the total number of validators as can be seen from Figure 6.

This paves the way for the second challenge: Who selects the validators for each shard? Controlling 5.1% of validators is only damaging if all 5.1% of the validators are in the same shard. If validators can’t choose which shard they get to validate in, a participant controlling 5.1% of the validators is highly unlikely to get all their validators in the same shard, heavily reducing their ability to compromise the system (Skidanov, 2018b).

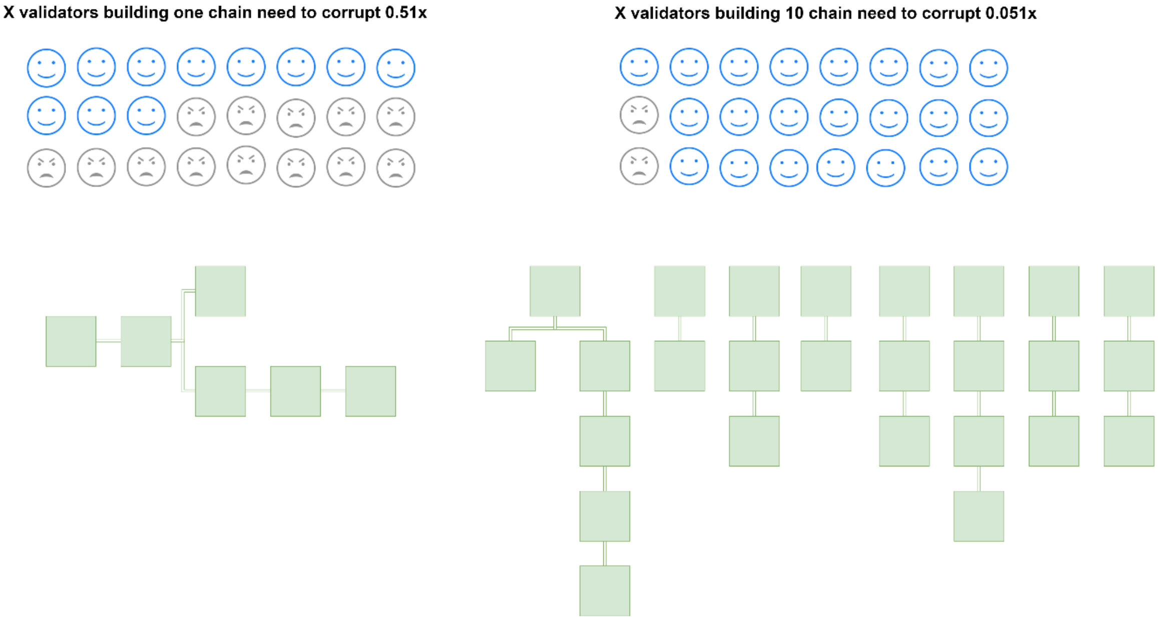

X validators building one chain need to corrupt 0.51x
X validators building 10 chain need to corrupt 0.051x

Figure 6: Validator ability to corrupt a shard

Almost all sharding designs today rely on some source of randomness to assign validators to shards. Randomness on the blockchain is a challenging topic in itself and requires more research in the future, but for now, let’s assume there’s some source of randomness we can use.

Both the randomness and the validators assignment require computation that is not specific to any particular shard. For that computation, practically all existing designs have a separate blockchain that is tasked with performing operations necessary for the maintenance of the entire network. Besides generating random numbers and assigning validators to the shards, these operations often also include receiving updates from shards and taking snapshots of them, processing stakes and slashing in Proof-of-Stake systems, and rebalancing shards when that feature is supported. Such chain is called a Beacon chain in Ethereum.

3.3.1.2.2 Quadratic Sharding

Sharding is often advertised as a solution that scales infinitely with the number of nodes participating in the network operation. While it is in theory possible to design such a sharding solution, any solution that has the concept of a Beacon chain doesn't have infinite scalability. To understand why, note that the Beacon chain has to do some bookkeeping computation, such as assigning validators to shards, or snapshotting shard chain blocks, that is proportional to the number of shards in the system. Since the Beacon chain is itself a single blockchain, with computation bounded by the computational capabilities of nodes operating it, the number of shards is naturally limited.

However, the structure of a sharded network does bestow a multiplicative effect on any improvements to its nodes. Consider the case in which an arbitrary improvement is made to the efficiency of nodes in the network which will allow them faster transaction processing times.

If the nodes operating the network, including the nodes in the Beacon chain, become four times faster, then each shard will be able to process four times more transactions, and the Beacon chain will be able to maintain 4 times more shards. The throughput across the system will increase by the factor of 4 x 4 = 16, thus, the name *quadratic sharding*.

It is hard to provide an accurate measurement for how many shards are viable today, but it is unlikely that in any foreseeable future the throughput needs of blockchain users will outgrow the limitations of quadratic sharding. The sheer number of nodes necessary to operate such a volume of shards securely is orders of magnitude higher than the number of nodes operating all the blockchains combined today.

3.3.1.2.3 State Sharding

Up until now, we haven't defined very well what exactly is and is not separated when a network is divided into shards. Specifically, nodes in the blockchain perform three important tasks: not only do they 1) process transactions, but they also 2) relay validated transactions and completed blocks to other nodes and 3) store the state and the history of the entire network ledger. Each of these three tasks imposes a growing requirement on the nodes operating the network:

- The necessity to process transactions requires more compute power with the increased number of transactions being processed;
- The necessity to relay transactions and blocks requires more network bandwidth with the increased number of transactions being relayed;
- The necessity to store data requires more storage as the state grows. Importantly, unlike the processing power and network, the storage requirement grows even if the transaction rate (number of transactions processed per second) remains constant.

From the above list, it might appear that the storage requirement would be the most pressing since it is the only one that is being increased over time even if the number of transactions per second doesn't change, but in practice, the most pressing requirement today is the compute power. The entire state of Ethereum as of this writing is 100GB, easily manageable by most of the nodes. But the number of transactions Ethereum can process is around 20, orders of magnitude less than what is needed for many practical use cases.

Practically, under State sharding, the nodes in each shard build their own blockchain that contains transactions that affect only the local part of the global state that is assigned to that shard. Therefore, the validators in the shard only need to store their local part of the global state and only execute, and as such only relay, transactions that affect their part of the state. This partition linearly reduces the requirement on all compute power, storage, and network bandwidth, but introduces new problems, which will be discussed in RQ2.

3.3.2 RQ2: What are the limitations of these designs?

We will now discuss some of the limitations and challenges that arise based on the designs of segregated witness and sharding:

3.3.2.1 Segregated Witness Limitations

Some of the risks associated with Segregated witness are as follows:

- Miners would get lower transaction fees for each transaction.
- Segwit implementation is complex and it requires that all the wallets implement segwit themselves.
- Segwit would significantly increase the amount of resources required to maintain the network since, the network capacity, transactions, bandwidth would increase.
- It might result in a hard fork of the Bitcoin network which may ultimately, decrease the financial value of both the networks.
- Finally, Segwit would be difficult to maintain. The sidechain containing the signature data will need to be maintained by miners as well. However, unlike the main blockchain, the miners have no financial benefits on doing so. Hence, some sort of reward protocol needs to be implemented to incentivize the miners to maintain the signatures on the sidechain.

3.3.2.2 Sharding Limitations

There are three main issues that arise with the proposed sharding solutions. We will assess these limitations in detail in this section:

3.3.2.2.1 Cross-shard transactions

“Scaling by a thousand Altcoin” as a model is not a very useful approach to sharding, because if individual shards cannot communicate with each other, they are no better than multiple independent blockchains. Even today, when sharding is not available, there’s a huge demand for interoperability between various blockchains.

Let’s for now only consider simple payment transactions, where each participant has an account on exactly one shard. If one wishes to transfer money from one account to another within the same shard, the transaction can be processed entirely by the validators in that shard. If, however, Alice that resides on shard 1 wants to send money to Bob who resides on shard 2, neither validators on shard 1(they won’t be able to credit Bob’s account) nor the validators on shard 2 (they won’t be able to debit Alice’s account) can process the entire transaction. There are two families of approaches to cross-shard transactions:

- **Synchronous:** whenever a cross-shard transaction needs to be executed, the blocks in multiple shards that contain state transition related to the transaction get all produced at the same time, and the validators of multiple shards collaborate on executing such transactions.
- **Asynchronous:** a cross-shard transaction that affects multiple shards is executed in those shards asynchronously, the “Credit” shard executing its half once it has sufficient evidence

that the “Debit” shard has executed its portion. This system is today proposed in Cosmos[13], Ethereum Serenity[14], Near[15], Kadena[16], and others. A problem with this approach lies in that if blocks are produced independently, there’s a non-zero chance that one of the multiple blocks will be orphaned, thus making the transaction only partially applied. Consider Figure 7 below that depicts two shards both of which encountered a fork, and a cross-shard transaction that was recorded in blocks A and X’ correspondingly. If the chains A-B and V’-X’-Y’-Z’ end up being canonical in the corresponding shards, the transaction is fully finalized. If A’-B’-C’-D’ and V-X become canonical, then the transaction is fully abandoned, which is acceptable. But if, for example, A-B and V-X become canonical, then one part of the transaction is finalized, and one is abandoned, creating an atomicity failure.

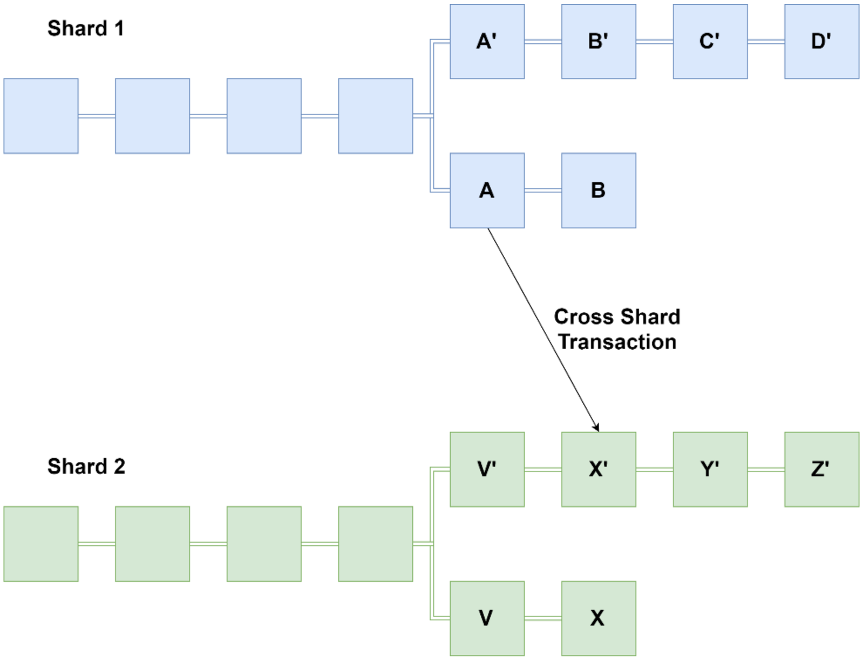


Figure 7: Cross-shard transactions

[13] https://cosmos.network/
[14] https://medium.com/utopiapress/what-is-ethereum-serenity-f433d824c974
[15] https://nearprotocol.com/
[16] https://kadena.io/en/

Note that communication between chains is useful outside of sharded blockchains too. Interoperability between chains is a complex problem that many projects are trying to solve. In sharded blockchains, the problem is somewhat easier since the block structure and consensus are the same across shards, and there's a beacon chain that can be used for coordination. In a sharded blockchain, however, all the shard chains are the same, while in the global blockchains ecosystem there are lots of different blockchains, with different target use cases, decentralization and privacy guarantees.

Building a system in which a set of chains have different properties but use sufficiently similar consensus and block structure and have a common beacon chain could enable an ecosystem of heterogeneous blockchains that have a working interoperability subsystem. Such a system is unlikely to feature validator rotation, so some extra measures need to be taken to ensure security.

3.3.2.2.2 Malicious Forks

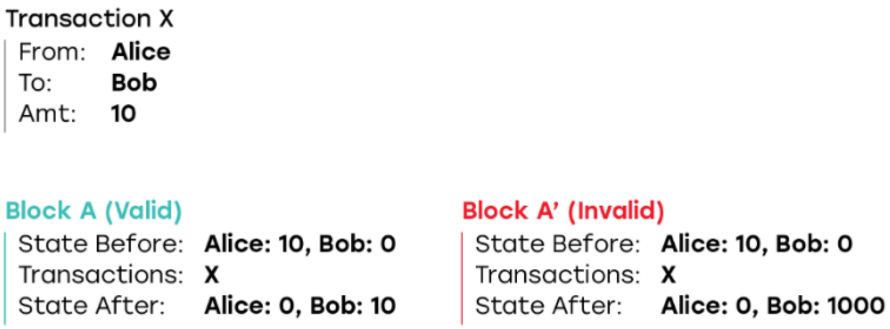
A set of malicious validators might attempt to create a fork. Note that it doesn't matter if the underlying consensus is BFT or not, corrupting a sufficient number of validators will always make it possible to create a fork.

It is significantly more likely for more than 50% of a single shard to be corrupted than for more than 50% of the entire network to be corrupted. As discussed above, cross-shard transactions involve certain state changes in multiple shards, and the corresponding blocks in such shards that apply such state changes must either be all finalized (i.e. appear in the selected chains on their corresponding shards), or all be orphaned (i.e. not appear in the selected chains on their corresponding shards). Since generally the probability of shards being corrupted is not negligible, we can't assume that the forks won't happen even if a Byzantine consensus was reached among the shard validators, or many blocks were produced on top of the block with the state change.

This problem has multiple solutions, the most common one being occasional cross-linking of the latest shard chain block to the beacon chain. The fork choice rule in the shard chains is then changed to always prefer the chain that is cross-linked and only apply the shard-specific fork-choice rule for blocks that were published since the last cross-link.

3.3.2.2.3 Approving invalid blocks

A set of validators might attempt to create a block that applies the state transition function incorrectly. For example, starting with a state in which Alice has 10 tokens and Bob has 0 tokens (see Figure 8), the block might contain a transaction that sends 10 tokens from Alice to Bob, but ends up with a state in which Alice has 0 tokens and Bob has 1000 tokens.



Transaction X
From: **Alice**
To: **Bob**
Amt: **10**

Block A (Valid)
State Before: **Alice: 10, Bob: 0**
Transactions: **X**
State After: **Alice: 0, Bob: 10**

Block A' (Invalid)
State Before: **Alice: 10, Bob: 0**
Transactions: **X**
State After: **Alice: 0, Bob: 1000**

Figure 8: Approving invalid blocks

In a classic non-sharded blockchain such an attack is not possible since all the participant in the network validates all the blocks, and the block with such an invalid state transition will be rejected by both other block producers and the participants of the network that do not create blocks. Even if the malicious validators continue creating blocks on top of such an invalid block faster than honest validators build the correct chain, thus having the chain with the invalid block being longer, it doesn't matter, since every participant that is using the blockchain for any purpose validates all the blocks and discards all the blocks built on top of the invalid block.

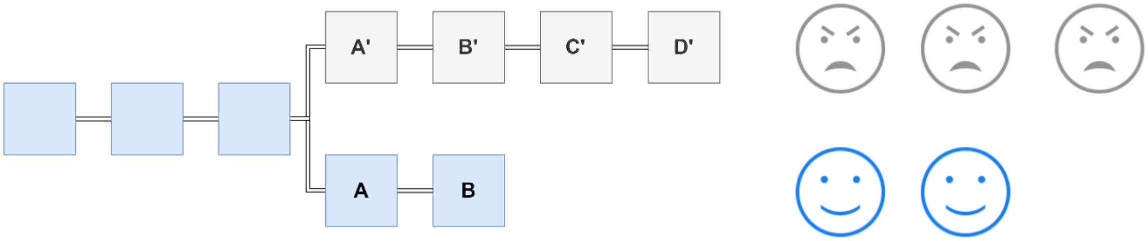


Figure 9: Malicious and honest validator behavior

Figure 9 shows five validators, three of whom are malicious. They created an invalid block A', and then continued building new blocks on top of it. Two honest validators discarded A' as invalid and were building on top of the last valid block known to them, creating a fork. Since there are fewer validators in the honest fork, their chain is shorter. However, in the classic non-sharded blockchain, every participant that uses blockchain for any purpose is responsible for validating all the blocks they receive and recomputing the state. Thus, any person who has any interest in the blockchain would observe that A' is invalid, and thus also immediately discard B', C' and D', as such taking the chain A-B as the current longest valid chain.

In a sharded blockchain, however, no participant can validate all the transactions on all the shards, so they need to have some way to confirm that at no point in the history of any shard of the blockchain no invalid block was included.

Note that unlike with forks, cross-linking to the Beacon chain is not a sufficient solution, since the Beacon chain doesn't have the capacity to validate the blocks. It can only validate that a sufficient number of validators in that shard signed the block (and as such attested to its correctness).

3.3.3 RQ3: What are the problems associated with implementing First Layer scalability solutions in public blockchains?

Since first-layer scaling solutions require a change in the codebase of an existing blockchain, these changes are incredibly difficult to implement in public permissionless blockchains such as Ethereum and Bitcoin. This problem arises because of the political decentralization nature of these blockchains. In order for such protocol changes, all the nodes on the blockchain network must agree on the change in protocol otherwise this change may create a hard-fork of the network which ultimately decreases its financial value. For instance, both Ethereum and Bitcoin suffered from hard-forks of their mainchain which led to the creation of Bitcoin Cash and Ethereum Classic. On the bright side, this difficulty in protocol change implementation on a public blockchain has given birth to in other innovative approaches of targeting blockchain scalability without changing the original codebase of such blockchains, but by implementing a second layer of blockchain on top of the mainchain. These solutions are known as second-layer scalability solutions and the most prominent of such proposals are the concept of sidechains, which are discussed in further detail in the next chapter.

Chapter 4. Second Layer Scaling Solutions

As discussed in Chapter 3, the proposed first-layer scaling solutions (Bala & Manoharan, 2018), (Ehmke, Wessling, & Friedrich, 2018) require a change in the protocol of blockchains which is extremely difficult to implement, especially in public blockchains due to their decentralized nature. A change in protocol needs to be agreed upon by all the peers on the blockchain network otherwise it may result in a hard-fork which may ultimately, reduce its value. This makes it extremely difficult to test changes to a pre-existing blockchain protocol or to add new functionality to it. Additionally, there has been a huge surge in blockchain-based systems in the recent years, for instance, Bitcoin is primarily supports peer-to-peer payment network, Ethereum is used for the deployment of decentralized applications and Hyperledger Fabric (Androulaki et al., 2018) is used for the enhancement of supply-chains[17]. Thus, it is hard to envision a single blockchain 'to rule them all' for the future. It would be more worthwhile instead to make these disparate blockchains interoperable so, that they can communicate and interact with one another.

In 2014, realizing this hindrance in the growth and further adoption of blockchains for building advanced, complicated and scalable software systems, Back et al. (Back et al., 2014) proposed a new and innovative method for improving the versatility and interoperability of traditional blockchains. In their paper, they proposed the idea of "sidechains" for the Bitcoin blockchain.

[17] https://cointelegraph.com/news/walmart-ibm-blockchain-initiative-aims-to-track-global-food-supply-chain

4.1 Sidechains

Sidechains are secondary blockchains which are connected to other blockchains by means of a two-way peg. A two-way peg is a mechanism that allows the bidirectional transfer of assets between the mainchain and the sidechain at a fixed or pre-deterministic exchange rate. Sidechains may have their own protocol and implementation which can be completely different from the main blockchain. Such adjustability provides the users flexibility to access various other functionalities and features offered on a sidechain by using the assets they already own on the main blockchain. Furthermore, sidechains are isolated from the main blockchain in such a way that in the case of a cryptographic break (or a maliciously designed sidechain), the damage is entirely confined to the sidechain itself.

Although promising, the sidechain technology is still relatively new and immature. There is a lack of comparative and empirical studies both in academic and industrial environments to analyze such multi-blockchain systems in a comprehensive manner. Hence, the motivation behind this chapter is to provide the first comprehensive review of the state-of-the-art sidechain platforms (which represent the most commonly used implementations of sidechain technology - hereafter referred to as sidechains) to understand the design choices, advancements, use cases, consensus mechanisms, asset transfer protocols and limitations of sidechains.

4.2 Answers to Research Questions (RQs)

In this section, we answer the research questions raised for second layer scaling solutions in Chapter 2. We start the discussion by explaining what and how a two-way peg works, what are the available design choices, their advantages, and limitations. We will then look at four state-of-the-art sidechain platforms namely Loom, POA, Liquid and RSK while discussing their use cases, consensus mechanisms, asset transfer protocols, and their limitations.

4.2.1 RQ1: What are the available design choices for implementing two-way pegs?

To understand the fundamentals and design choices for implementing a two-way peg enabled sidechain, we will discuss a trivial example in this section. Let us assume a sidechain is attached to a public and permissionless primary blockchain with a two-way peg. The primary blockchain: 1) operates a cryptocurrency called *MainCoin* and 2) cannot execute non-trivial smart contracts due to the absence of a Turing complete Virtual Machine. The sidechain: 1) operates its own cryptocurrency of named *SideCoin*, 2) has the capability of executing non-trivial smart contracts and 3) offers significantly higher transaction rate (i.e. higher transactions per second) than the mainchain. For the sake of simplicity in such multi-blockchain environment, the primary blockchain is called the *parent blockchain (or mainchain)* and the sidechain attached to it is called a *secondary chain* (the terms sidechain and secondary chains will be used interchangeably throughout the rest of this paper). In our example, a two-way peg allows the transfer of *MainCoins* from the mainchain to the sidechain and vice versa at a fixed rate of 1 *MainCoin* = 1 *SideCoin*. Suppose a user wishes to transfer 5 *MainCoins* from the mainchain to the sidechain to play a rock, paper and scissor game with another random user based on a smart contract (where winner takes all and a draw results in no exchange of coins) implemented on the sidechain, then this system could work in the following abstract manner:

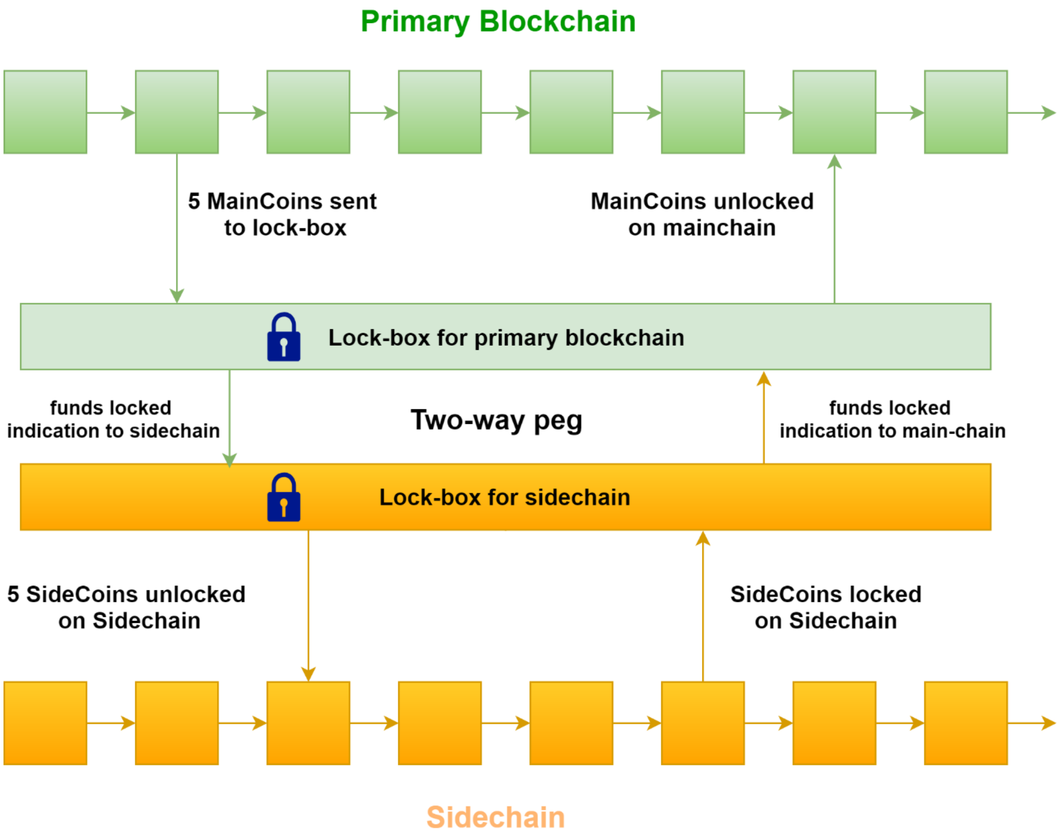

Primary Blockchain
5 MainCoins sent to lock-box
MainCoins unlocked on mainchain
Lock-box for primary blockchain
funds locked indication to sidechain
Two-way peg
funds locked indication to main-chain
Lock-box for sidechain
5 SideCoins unlocked on Sidechain
SideCoins locked on Sidechain
Sidechain

Figure 10: Transfer of funds between mainchain and sidechain with a two-way peg

1. The user sends 5 *MainCoins* to a special address (also known as a lock-box) where the coins are locked and can only be unlocked once funds on sidechain are locked and transferred back to the mainchain.
2. Once the funds locked on the mainchain, 5 *SideCoins* are created on the sidechain.
3. The user can now use these *SideCoins* to play the game of rock, paper, and scissors with another random user who is willing to bet the same amount of *SideCoins*.
4. Depending on the outcome of the game, 10 *SideCoins* are transferred to the winner or 5 *SideCoins* are transferred back to their respective owners (in case of a draw).
5. The user(s) can then transfer their funds back to the mainchain, which essentially means that the *SideCoins* will be locked/destroyed on the sidechain and an equivalent number of

MainCoins will be unlocked on the mainchain from the lock-box (in step 1) after *SideCoins* are destroyed on the sidechain.

The above steps are summarized in Figure 10 and can vary depending on the way in which a two-way peg has been implemented for the sidechain (sub-subsection 4.2.1.1). With this model, the total number of *MainCoins* in the mainchain ecosystem remains conserved whilst adding new functionality to it, i.e. execution of non-trivial smart contracts and faster transaction rates. Moreover, the implementation of these new features with sidechains do not require any major change in the core features or consensus protocol of the mainchain itself.

Based on our analysis, there are currently three major design choices for implementing a two-way peg for transferring assets from the mainchain to the sidechain and vice versa. These design choices are discussed below.

4.2.1.1 Centralized two-way pegs

The simplest way to implement a two-way peg is to have a trusted third entity hold custody of the locked funds. In this design, the trusted entity is solely responsible for locking and unlocking of funds on both the mainchain and its sidechain. Figure 11 shows the relevant steps in which the entire process of fund transfer takes place, both from the mainchain to the sidechain and vice versa.

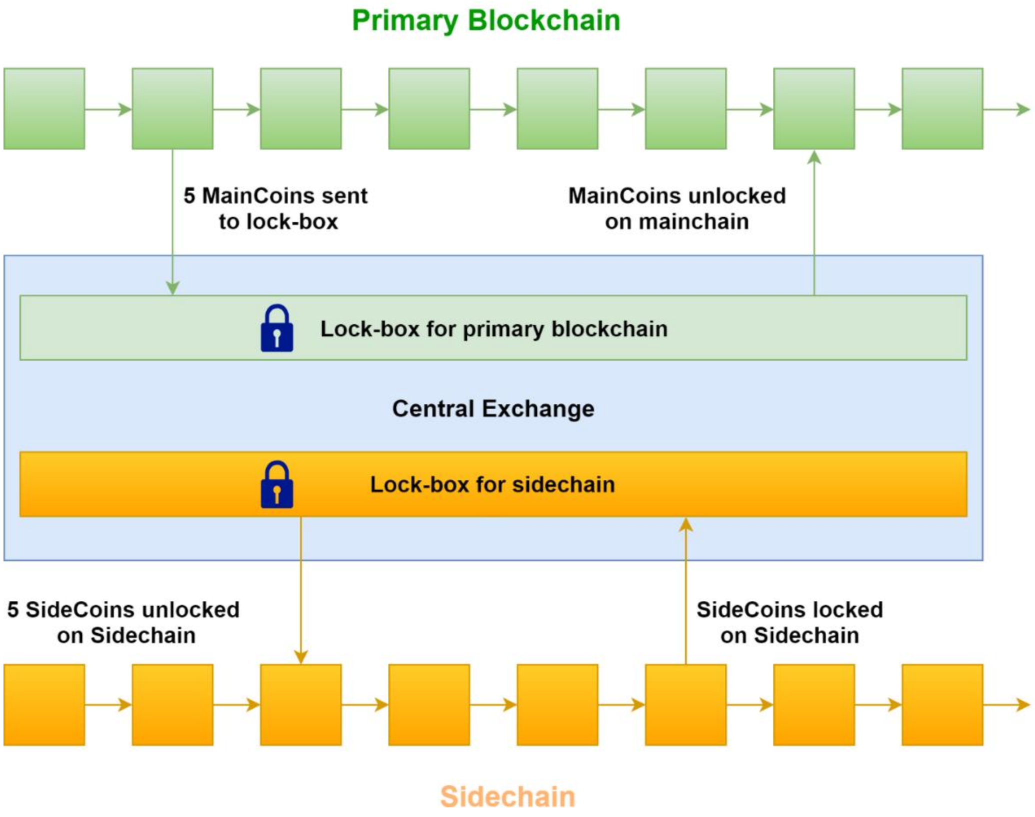

Primary Blockchain
5 MainCoins sent to lock-box
MainCoins unlocked on mainchain
Lock-box for primary blockchain
Central Exchange
Lock-box for sidechain
5 SideCoins unlocked on Sidechain
SideCoins locked on Sidechain
Sidechain

Figure 11: Centralized two-way peg implementation

Based on this two-way peg design, the steps for fund transfer (based on our example above) are modified in the following manner:

1. The user sends 5 *MainCoins* to a lock-box address maintained by a *trusted centralized entity* meant for regulating fund transfer between the two blockchains.
2. The trusted entity then generates 5 *SideCoins* on the sidechain and sends these funds to the user's requested address.
3. The user can now use these *SideCoins* to play the game of rock, paper, and scissors with another random user who is willing to bet the same amount of *SideCoins*.
4. Depending on the outcome of the game, 10 *SideCoins* are transferred to the winner or 5 *SideCoins* are transferred back to their respective owners (in case of a draw).

5. The user(s) can then transfer their funds back to the mainchain, by sending their *SideCoins* to the lock-box address on the sidechain which is also maintained by the same *trusted central entity*. The user(s) also specify the address where the funds need to be sent on the mainchain.
6. The trusted central entity destroys the *SideCoins* on the sidechain and sends the equivalent number of *MainCoins* to address specified by the user(s).

4.2.1.2 Multi-Signature or Federated two-way pegs

An improvement over centralized two-way pegs are the federated two-way pegs (Back et al., 2014), (Dilley et al., 2016). In such a design, a group of entities or notaries control the lock-box rather than just one central entity. Consequently, the entire federation or group collectively holds custody of the locked funds and regulates fund transfer between the primary blockchain and its sidechain. The fund transfer takes place only when the majority of the entities i.e. ‘n’ out of ‘m’ entities (where ‘n’ is the majority and ‘m’ is the total number of entities in the federation) within the Federation sign the transaction (Deng, Chen, Zeng, & Zhang, 2018). Figure 12 demonstrates the sequential steps with which fund transfer takes place between the two blockchains using a federated two-way peg.

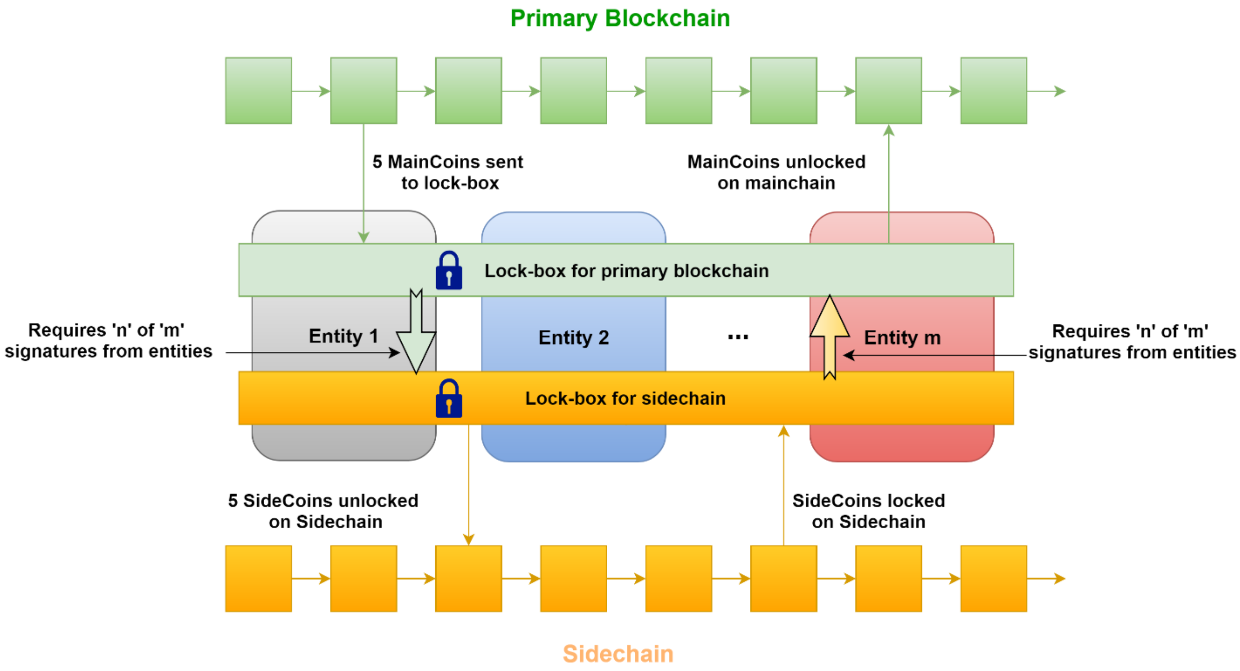

Primary Blockchain
5 MainCoins sent to lock-box
MainCoins unlocked on mainchain
Lock-box for primary blockchain
Requires 'n' of 'm' signatures from entities
Entity 1
Entity 2
...
Entity m
Requires 'n' of 'm' signatures from entities
Lock-box for sidechain
5 SideCoins unlocked on Sidechain
SideCoins locked on Sidechain
Sidechain

Figure 12: Federated two-way peg implementation

Based on a federated two-way peg design, the steps for fund transfer (based on our example above) are modified as follows:

1. The user sends 5 *MainCoins* to a lock-box address maintained by a *federation of entities* meant for regulating fund transfer between the two blockchains. The entities of the federation then sign this transaction after verifying that the funds have been received in the lock-box.
2. If the majority of the entities within the Federation sign the transaction, then the federation generates 5 *SideCoins* on the sidechain and sends these funds to the user's requested address.
3. The user can now use these *SideCoins* to play the game of rock, paper, and scissors with another random user who is willing to bet the same amount of *SideCoins*.
4. Depending on the outcome of the game, 10 *SideCoins* are transferred to the winner or 5 *SideCoins* are transferred back to their respective owners (in case of a draw).

5. The user(s) can then transfer their funds back to the mainchain, by sending their *SideCoins* to the lock-box address on the sidechain which is also maintained by the same *federation of entities*. The user(s) also specify the address where the funds need to be sent on the mainchain.
6. The entities of the federation again sign the transaction after verifying that the funds have been received in the lock-box on the sidechain.
7. If the majority of the entities sign the transaction, then the federation destroys the *SideCoins* on the sidechain and sends the equivalent number of *MainCoins* to address specified by the user(s).
8. In the case when the majority of the entities within the federation do not reach an agreement regarding a transaction, then the funds are sent back to their respective owners on either chain.

4.2.1.3 Simplified Payment Verification (SPV)

Simplified Payment Verification (SPV) allows a lightweight[18] client to prove that a given transaction was included in a legitimate block of the longest Proof-of-Work (PoW) blockchain, without having to download the entire chain from the genesis block itself. These lightweight (or SPV) clients are only required to download the block headers of the entire blockchain, which are much smaller in size than the actual block itself. To verify if a given transaction was included in a legitimate block, an SPV client requests a proof of inclusion, in the form of a Merkle branch of that transaction. Figure 13 demonstrates the entire process of transfer of funds from the mainchain to the sidechain and vice versa based on two-way peg implemented with SPV proofs.

[18] https://www.mycryptopedia.com/full-node-lightweight-node/

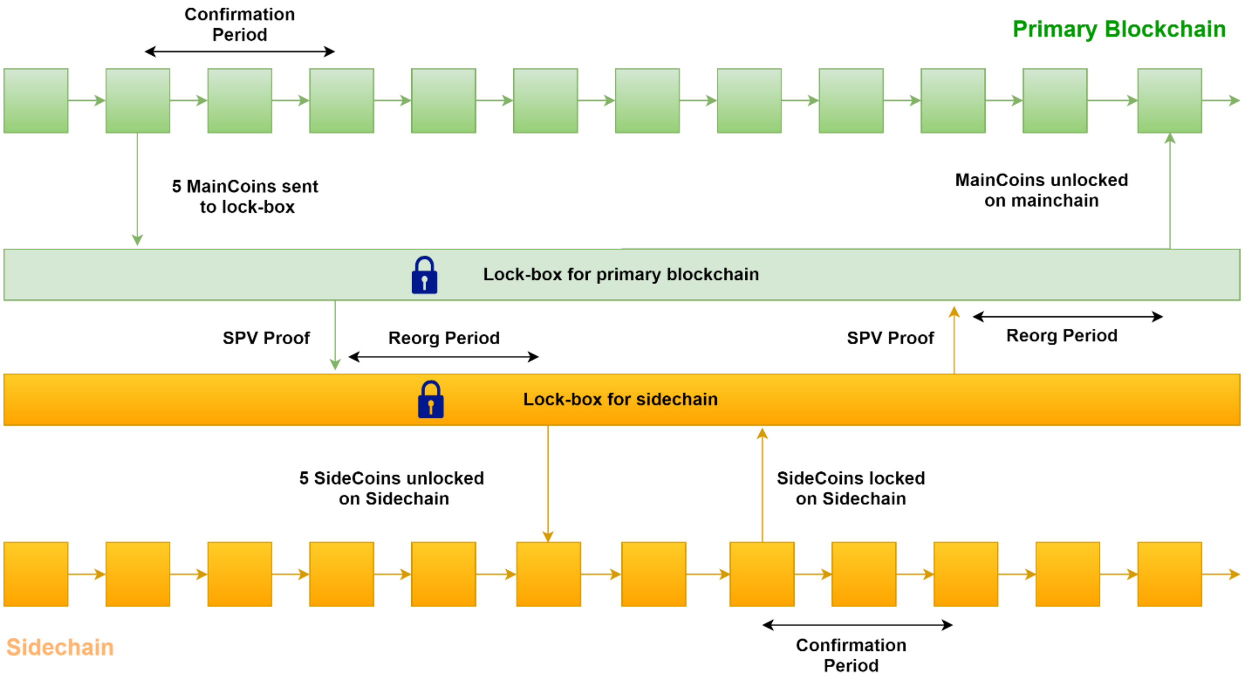

Confirmation Period
Primary Blockchain
5 MainCoins sent to lock-box
MainCoins unlocked on mainchain
Lock-box for primary blockchain
SPV Proof
Reorg Period
SPV Proof
Reorg Period
Lock-box for sidechain
5 SideCoins unlocked on Sidechain
SideCoins locked on Sidechain
Sidechain
Confirmation Period

Figure 13: Two-way peg based on SPV proofs

SPV proofs indirect proofs in the sense that a given transaction is not proven to be consistent with the entire blockchain from the genesis block itself. Instead, it is shown to be a part of valid block upon which miners have mined newer blocks, subsequently forming the longest chain. The way in which this is done is as follows:

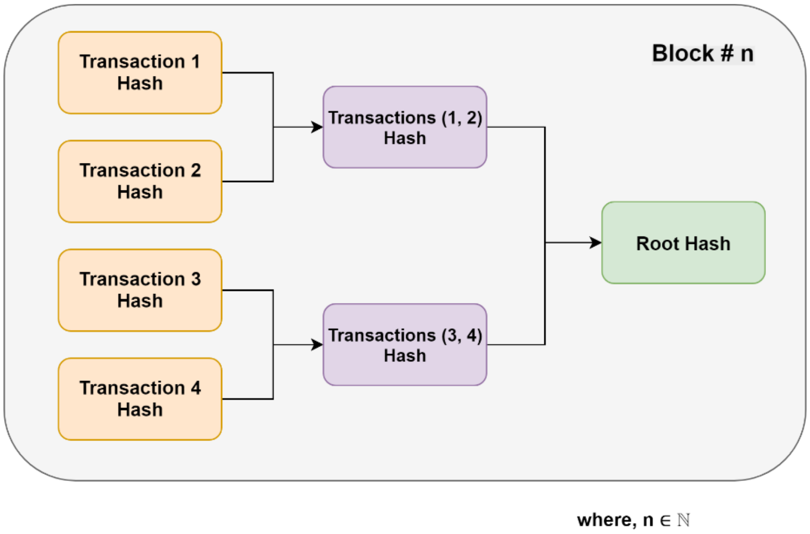


Figure 14: A block of transaction hash Merkle tree

1. After a transaction is submitted for the transfer of funds from the mainchain to the sidechain or vice versa (i.e. the funds are locked in the lockbox), there is a confirmation period, which is strategically in place to allow miners to mine on top of the last block which consequently, allows the generation and submission of SPV proof.
2. The SPV proof is then submitted by the user and the block in which his/her transaction is recorded is located.
3. The user then provides the hashes along the Merkle tree branch on which his/her transaction lies. This is done in the following manner:
    a. Suppose a user is looking to validate Transaction 2 (Figure 14), he/she can obtain the hash of Transaction 1 and a combined hash of Transaction 3 and 4 i.e. Transaction (3, 4) from a number of other full nodes.
    b. With this information, the user can compute the root hash of the Merkle tree in the block.

4. If these hashes all collectively hash to the original Merkle root of the transaction hash tree in that block, then the transaction is valid.

After an SPV proof is submitted there is a reorganization or reorg period in which other users may submit their own SPV proofs to contradict the user’s transaction. The SPV proof in which more blocks have been mined is considered to be the correct proof and decides the fate of the transaction.

Given an SPV based two-way peg design, the steps for fund transfer (based on our example above) are modified as follows:

1. The user sends 5 *MainCoins* to a lock-box address which is usually maintained by the miners of the network. Once the coins are locked on the mainchain, the user has to wait for a predetermined confirmation period to allow the mines to create new blocks to create SPV proofs.
2. Once sufficient blocks are created by the miners, the user can submit an SPV proof verifying that the coins were locked on the mainchain.
3. After the SPV proof is submitted, the user has to wait for the reorg-period where other users can submit their SPV proofs to nullify fraudulent transactions, in case one has taken place.
4. After the SPV proof is verified 5 SideCoins are unlocked on the sidechain.
5. The user can now use these SideCoins to play the game of rock, paper, and scissors with another random user who is willing to bet the same amount of *SideCoins*.
6. Depending on the outcome of the game, 10 *SideCoins* are transferred to the winner or 5 *SideCoins* are transferred back to their respective owners (in case of a draw).
7. The user(s) can then transfer their funds back to the mainchain, by sending their *SideCoins* to the lock-box address on the sidechain and repeating the same process mentioned in steps 1 – 4 on the sidechain side.

4.2.2 RQ2: What are the advantages and limitations of these design choices?

The advantages and limitations of each of the design choices discussed in RQ1 are discussed below:

4.2.2.1 Advantages of centralized two-way pegs

There are two major advantages of using a centralized two-way peg design: 1) Centralized two-way pegs are easy to visualize and implement due to their simplistic design which involves just one entity to oversee the transfer of assets between blockchains. 2) the design could provide extremely fast transfer of funds from the parent blockchain to its sidechain and vice versa as the central entity generally requires a simple proof of locked funds in the lockbox, which they can verify themselves at any given time.

4.2.2.2 Disadvantages of centralized two-way pegs

Using a trusted central entity comes with its own drawbacks, such as: 1) Public blockchains such as Bitcoin and Ethereum are designed to improve political decentralization and using such a two-way peg design introduces a degree of political centralization as one has to trust a single entity to manage fund transfer from a primary blockchain to a sidechain and vice versa. 2) Using a centralized two-way peg design introduces a single point of failure in such multi-blockchain ecosystems as unforeseen circumstances such as power failures, hardware failures or natural disasters would temporarily or permanently cease asset transfers between the blockchains, which would cripple the sidechain network and 3) If the centralized entity is rogue or malicious, it can steal all the funds stored in the lock-box.

4.2.2.3 Advantages of federated two-way pegs

The advantages of using a federated two-way peg design are: 1) It improves upon centralized two-way peg design by improving the political decentralization of such multi-blockchain systems to some extent and 2) These designs could be implemented with specialized federation protocols for fast transfer of funds between the blockchains. Some of these protocols are *Strong Federations* (Dilley et al., 2016) (which is discussed further in RQ3c).

4.2.2.4 Disadvantages of federated two-way pegs

Federated two-way pegs can have drawbacks such as 1) Such design does not entirely eliminate the political centralization problem as this design still relies on a small group of entities to regulate and manage fund transfer between blockchains and 2) Funds in the lock-box could be stolen if the majority of the entities of a federation lose their private keys due to a malicious internet attack or social engineering.

4.2.2.5 Advantages of SPV based two-way pegs

The main advantage of an SPV based two-way peg is that it eliminates the third party required for fund transfer between two blockchains as in case of Centralized and Federated two-way pegs.

4.2.2.6 Disadvantages of SPV based two-way pegs

A disadvantage of an SPV based design is that these designs tend to be slow as a user needs to wait for confirmation and reorg periods before having access to his/her funds on either mainchain or sidechain.

Table 4: Summary of advantages and disadvantages of two-way peg designs

Two-way peg Design	**Advantages**	**Disadvantages**
Centralized	• Asset transfer between blockchains can be fast • Simple design and implementation	• Politically centralized • Introduces a single point of failure • assets can be stolen by a malicious central entity
Federated	• Better political decentralization than centralized two-way pegs • Asset transfer between blockchains can be fast • Can work well with the right number and type of entities that form the federation (see Chapter 5)	• Not politically decentralized • Assets can be stolen if private keys of the majority of entities are stolen
SPV	• Politically decentralized	• Slow transfer of assets between blockchains

Table 4 summarizes the Centralized, Federated and SPV based two-way peg designs based on their advantages and disadvantages.

4.2.3 RQ3: Which state-of-the-art platforms are implementing sidechains?

We will now discuss four major state-of-the-art sidechain platforms namely Loom (Loom, n.d.-b), (Loom, n.d.-a), Proof-of-Authority (POA) Network (Arasev, 2018), (POA, n.d.-b), Liquid (Dilley et al., 2016), (Blockstream, n.d.) and RootStock (RSK) (S. D. Lerner, 2015), (RSK, n.d.) that improve scalability and facilitate interoperability in the multi-blockchain ecosystem. We chose these platforms based on the following reasons:

- Popularity in the community: The popularity of a platform was determined by either one or both of the following criteria: 1) the number of users that are registered on the platform (e.g. CryptoZombies[19] a DApp on Loom has accumulated over 240,000 users since going live (Bentley, 2018)) and 2) partnership of the sidechain platform with prominent or well-known blockchain companies or organizations (e.g. BitPay[20] and BITMAIN[21]) (POA, n.d.-c), (RootStock, n.d.), (O`KeeffeDaniel, 2018).

- Availability of documentations, white papers, forums and technical support (Loom, n.d.-b), (Dilley et al., 2016), (S. D. Lerner, 2015), (POA, n.d.-a), (Arasev, 2018).

4.2.3.1 Loom

Loom (Loom, n.d.-b), (Loom, n.d.-a) is a platform for running Decentralized Applications (DApps) and games on sidechains connected to the Ethereum Blockchain. It utilizes the Delegated Proof-of-Stake (DPoS) protocol to reach consensus. Each DApp runs on its own sidechain (called a DAppChain) pegged to the Ethereum main-net. This allows the users and developers to run multiple nodes for an application on the sidechain. Along with the Delegated Proof-of-Stake consensus, Loom runs on a Byzantine-fault-tolerant state machine replication as a backend P2P layer called Tendermint[22]. In the Loom architecture, a transaction on the Loom network is not immediately settled on the Ethereum mainchain but instead, they are settled in bulk in order to increase scalability.

4.2.3.2 POA Network

The POA network (Arasev, 2018), (POA, n.d.-b) is an open-source public Ethereum sidechain for developing smart contracts. It uses Proof of Authority (POA, 2017) as its consensus protocol. The platform provides the users and developers of smart contracts and decentralized applications with

[19] https://cryptozombies.io/
[20] https://bitpay.com/
[21] https://www.bitmain.com/
[22] https://tendermint.com/

the flexibility to develop on Ethereum standards with more scalability and interoperability between other blockchain networks.

POA network supports native Solidity (“Solidity,” n.d.) smart contracts, which allows effortless portability of smart contracts and decentralized applications from the Ethereum environment to the POA network. The platform charges minimal transaction fees which combined with about four magnitudes in transaction speed over Ethereum encourages and promotes the development of scalable games and applications. Additionally, POA provides bridging (especially for ERC-721 tokens) capabilities which allows users to transfer their non-fungible tokens from one blockchain to another easily.

4.2.3.3 Liquid

Liquid (Dilley et al., 2016), (Blockstream, n.d.) is a commercial sidechain by Blockstream. It enables instantaneous movement of funds between exchanges, without waiting for the delay of confirmation in the Bitcoin blockchain. The transactions on the Liquid platform are completed in an average of two minutes. Liquid supports private transactions which allow traders and exchanges to trade/transact in private, preventing front-running of large orders.

Liquid also supports Issued assets where an organization or a company that serves as the custodian of assets (physical or cryptocurrency), can issue a tokenized version of the asset using the platform. Once the assets are tokenized on the Liquid platform, they can be traded freely within the network, taking advantage of Liquid’s speed and private trading features. The Liquid Network consists of a ‘*Strong Federation*’ (Dilley et al., 2016) (discussed in RQ3c) which consists of several financial institutions and cryptocurrency exchanges who all run high-performance computing hardware to secure the network.

4.2.3.4 RootStock (RSK)

RSK (S. D. Lerner, 2015), (RSK, n.d.) is an open-source sidechain pegged to the Bitcoin main-net for the execution of smart contracts, it is an evolution of QixCoin (“Qixcoin,” n.d.), a Turing-complete cryptocurrency developed in 2013. RSK implements the concept of merged mining (S. Lerner, 2016) which provides incentives to the miners of the Bitcoin blockchain to be actively involved by mining on RSK platform.

RSK incorporates a Turing complete, resource-accounted, and deterministic virtual machine (called the RootStock Virtual Machine or RVM) for the parallel execution of smart contracts in the Bitcoin ecosystem by several nodes. The execution of smart contracts can result in the processing of messages between multiple other smart contracts, creation of new transactions or change of a state of smart contract’s persistent memory. RVM is compatible with Ethereum’s Virtual Machine (EVM) at op-code level which allows the execution of Solidity (“Solidity,” n.d.) smart contracts on RSK.

4.2.4 RQ3a: What are the use cases of these platforms?

The use cases of each platform are now discussed to answer this research question:

4.2.4.1 Loom Use Cases

The Loom network has mainly been used for the following use cases:

- **Digital Social Interaction:** The original use case of the Loom Network is DelegateCall[23] which is a forum where questions can be asked and each answer that a user provides and upvotes earns them ‘Karma’. Karma can be traded on the Ethereum chain for ERC-20

[23] https://delegatecall.com/

tokens. ERC-20 tokens are fungible tokens, or coins, on the Ethereum Network which are not unique and can be divided into smaller portions.

- **Game development:** The second use case for the Loom Network is for running games such as games built on Unity[24]. Games require quick transaction times and the performance of the mainchain is much faster than it would be if the games were run on the Ethereum blockchain itself. The gas fees required for the transactions on the Loom sidechain are much less than on the Ethereum chain making it more practical for game development.

4.2.4.2 POA Use cases

The purpose of the POA network is to prove the possibility of cross-chain transfers between an Ethereum chain and a sidechain. Interoperability is a major goal of the POA Network along with an increase in scalability and the connectivity of Ethereum. POA aims to have a solution to communicating between two stand-alone blockchains. Some of the major projects that have used the POA network as of November 2018 are as follows:

- Swarm City[25], a decentralized commerce platform, has used the ERC20 to ERC20 bridge to transfer tokens from the Ethereum chain to Kovan test-net[26]
- Sentinel Chain (Lai, 2018) is transferring ERC20 tokens from the Sentinel Chain to other EVM-based blockchains.
- Virtue Poker[27] has used the POA bridge along with their own sidechain to eliminate expensive transactions.
- Colu Network[28] has partnered with the POA network to connect their own sidechain.

[24] https://unity3d.com/
[25] https://swarm.city/
[26] https://kovan-testnet.github.io/website/
[27] https://virtue.poker/
[28] https://cln.network/

- POA network is additionally working with more projects to help deal with the scalability and high gas cost of the Ethereum network.

4.2.4.3 Liquid Use cases

Since, strong federations were designed to provide solutions to problems related to transaction latency, commercial privacy, reliability and fungibility, the most prominent use case of the Liquid platform is in international exchange:

- **International Exchange:** Bitcoin can facilitate cross border payments and remittance, but it is limited by its own design choice that hampers its performance (Karame, Androulaki, & Capkun, 2012). It also suffers the wrath of market-dynamics like most if not all cryptocurrencies at this time. Consequently, the high latency of the Bitcoin network requires Bitcoin to be tied up in multiple exchanges and brokerage environments. The lack of privacy also adds to its cost of operation. Additionally, local currency trade with Bitcoin can be a subject to illiquidity due to market fragmentation because of which many organizations and commercial entities choose to operate or design their own high-frequency methods of exchange (Moore & Christin, 2013). These solutions and workarounds have often introduced in centralized systems and other issues (Karame et al., 2012). Thus, with strong federations, Liquid, introduces improved security and privacy, with lower latency than the Bitcoin network.

4.2.4.4 RSK Use cases

The compatibility of RVM with EVM opens the door up for the implementation of several innovative smart contracts and use cases as it allows the developers working on the Ethereum platform to take advantage of Bitcoin’s robustness. Some of the most important use cases are discussed below:

- **Retail Payment Systems:** With the implementation of RSK, Bitcoin could be adopted globally for day-to-day retail transactions. In its current state, it is not feasible to use Bitcoins in retail due to its slow confirmation time (~ 10 minutes – 1 hour to ensure irreversibility). RSK can allow consumers to have the security of Bitcoin with faster transaction times (~ 10 seconds). This would allow merchants to accept payments faster without having to rely on third-party gateways. Additionally, the RSK platform can handle a high volume of transactions per second (~ 300 - 1000 transactions per second) which is yet another necessity for a payment processing platform to succeed in the retail industry.
- **Supply Chain Traceability:** With RSK smart contracts could be implemented to track and trace the physical location and condition of a product. Such contracts could be particularly useful in food, retail, healthcare, and transportation industries. Once again, such contracts would be backed by the security and robustness of the Bitcoin protocol.
- **Digital Identity:** Developing countries struggle with the lack of documentation and identification for the poor, which can prevent them from voting, accessing healthcare and financial aid, reporting criminal activities. Hence, with RSK digital global registries could be implemented at extremely low costs, this could be a major step in the improvement of the overall infrastructure of such countries.

4.2.5 RQ3b: How does asset transfer take place on these platforms?

We now discuss the asset transfer protocol in each of the platform discussed in RQ3 to answer this research question:

4.2.5.1 Asset transfers on the Loom Platform

The Loom network has plans to allow for ERC721 and ERC20 tokens to be transferred from the Ethereum blockchain to the DAppChain and vice-versa using Plasma-based relays[29]. At the moment, Loom only allows for ERC721 tokens to be traded on the network. ERC721 tokens are non-fungible tokens meaning that they can be collected, and each individual token is unique and irreplaceable. Currently, Loom uses a Transfer Gateway to support the transfer of these tokens. When the tokens are being deposited to the DAppChain, the tokens are sent to a gateway contract before being sent to the Gateway Oracle where the transfer is forwarded to the Gateway Contract on the DAppChain. Figure 15 shows the asset transfer from the Ethereum blockchain to the DAppChain.

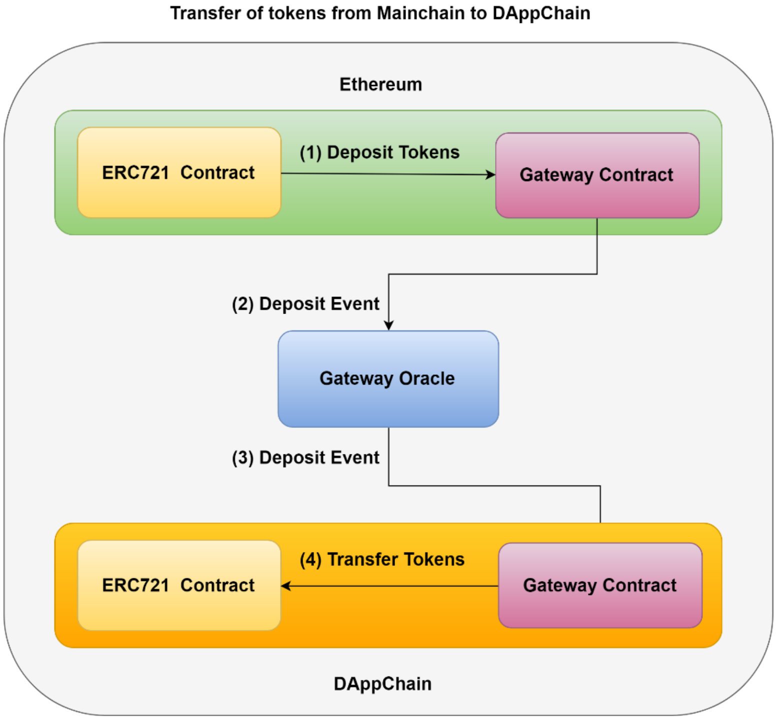


Figure 15: Asset transfer from Ethereum to DAppChain

[29] https://blog.gridplus.io/introducing-trusted-relay-networks-6c168f72a6f6

The Gateway Oracle typically runs on nodes that are serving as delegators for the Delegated Proof of Stake consensus algorithms although a gateway oracle can run on nodes that are standalone. If the tokens are being withdrawn from the DAppChain back to the Ethereum mainchain, the tokens are sent back to the Transfer Gateway Oracle where the user submits a Merkle proof of the user's transaction history and the withdrawal awaits a signature of approval. With this signed withdrawal record, the user may withdraw tokens back to the Ethereum mainchain. Figure 16 shows the asset transfer from the DAppChain to the Ethereum blockchain.

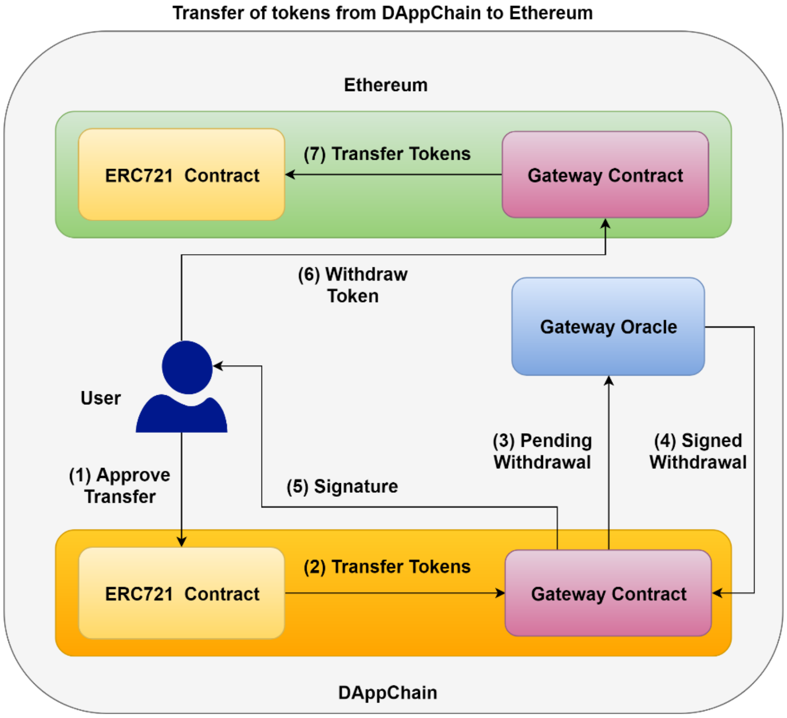


Figure 16: Asset transfer from Ethereum to DAppChain

The mainchain gateway contract needs to approve of the signature produced by the Gateway Oracle. When a user initially deposits tokens to the DAppChain from the mainchain, an address mapper contract creates a mapping of both the private key for Ethereum and the private key for the

DAppChain. The reason that a signature is not required for depositing to the sidechain but is required for withdrawing back to the mainchain is that the signature is used to decrease the dependence on the personalized consensus algorithm being used on the sidechain when in need of transferring. The Ethereum mainchain, on the other hand, has a more trusted consensus algorithm.

4.2.5.2 Asset transfers on the POA network

There are three different types of transfers that can take place.

- **Native to ERC20:** In this case, “Native” refers to the POA tokens. POA tokens are locked in a smart contract and POA20 tokens are then generated on the Ethereum blockchain. The POA20 tokens are the POA equivalent of the ERC20 tokens that are found on the Ethereum blockchain. These tokens are burned on the Ethereum blockchain before the smart contract is activated and the tokens are unlocked on the POA blockchain.
- **ERC20 to ERC20:** Token “X” from the first Ethereum network is locked on the first Ethereum Network. Token “Y” is generated on the second Ethereum network and then burned on the second network. The smart contract is activated, and the Token “X” is unlocked on the primary Ethereum network. The difference between this bridge and the first Native to ERC20 bridge is instead of the bridge only supporting the transfer of tokens to and from the POA network this bridge allows for the transfer of tokens between any two networks operating on the Ethereum chain.
- **ERC20 to Native:** The ERC-20 to Native bridge allows for the transfer of DAI tokens from the Ethereum Network to the xDAI chain. The DAI token is an ERC-20 token that maintains a 1:1 ratio with the United States Dollar (USD) meaning that each DAI token is always worth exactly one US dollar. The xDAI chain is an Ethereum based blockchain using the USD-stable XDAI token. DAI tokens differ from XDAI tokens by the fact that

DAI tokens live on the Ethereum mainchain whereas the XDAI tokens live on a separate xDAI sidechain. It also maintains a ratio of 1:1 with the US Dollar and is backed by Ethereum collateral. XDAI tokens are minted on the xDAI chain network and burned on the xDAI chain network. The Smart Contract is activated, and the DAI tokens are unlocked on the Ethereum network. A subset of the total number of validators function as validators for each set of bridge transactions. Also, each bridge is bilaterally allowing for a transfer back and forth between two blockchains. The transfers happen within one's own wallet by having representations of tokens on one network be minted on the other network.

4.2.5.3 Asset transfers on the Liquid Platform

Native Assets: The Liquid network supports accounting of other assets (including traditional currencies, real-world assets, and other cryptocurrencies) in addition to Bitcoin. These are known as native assets and are accounted separately from the base Bitcoin cryptocurrency. These assets can be issued by any participant by means of a special asset-generating transaction. They can also optionally set conditions by which additional issuance can take place in the future:

- A policy for an asset being generated is decided upon by the asset issuer, which includes conditions for asset redemption.
- The asset issuer creates a transaction with one or more special asset-generating inputs, whose value is the full issuance of the asset. This transaction uniquely identifies the asset.
- A member of the strong federation confirms the asset-generating transaction after which the assets become transactable.
- The asset issuer then distributes these assets to its user-base as per requirement. This is done by using standard strong federation transactions.

- When the users wish to redeem their asset tokens, they transfer their asset holdings back to the issuer in return for out-of-band goods or the provided service. The issuer then destroys these tokens.

Peg-out Authorization: When Bitcoins are frozen on the Bitcoin blockchain and pegged into the Liquid network they become Liquid Bitcoin (L-BTC). The L-BTCs can be then utilized on the Liquid network and can be transferred back to the Bitcoin blockchain at any given time. As discussed earlier, moving assets back to the Bitcoin blockchain is foreseen and mediated by a set of watchmen, who create the transactions on the Bitcoin side. These transactions take place with the help of peg-out authorization proofs which have the following design:

- **Setup:** Each participant *i* chooses two public-private keypairs: (P_i, p_i) and (Q_i , q_i), where p_i is an “online key” and q_i is an “offline key”. The participant then provides P_i and Q_i to the watchmen.
- **Authorization:** To authorize a key W (which will correspond to an individually-controlled Bitcoin address), a participant takes the following steps.
  - They compute $L_j = P_j + H(W + Q_j)(W + Q_j)$ for every other participant index *j*, where H is a random oracle hash that maps group elements to scalars.
  - The participant knows the discrete logarithm of L_i, and can, therefore, produce a ring signature over every L_i. They do so by signing the full list of online and offline keys as well as *W*.
  - The participant sends the resulting ring signature to the watchmen or embeds it in the sidechain.
- **Transfer:** When the watchmen produce a transaction to execute transfers from the sidechain to Bitcoin, they ensure that every output of the transaction either 1) is owned by them or 2) has an authorization proof associated to its address.

4.2.5.4 Asset transfers on the RSK Platform

Asset transfers on RSK take place with federated two-way pegs. When Bitcoins are transferred from the Bitcoin blockchain to the RSK sidechain they are referred to as "*SmartBitcoins*" (SBTC) (S. D. Lerner, 2015). Hence, SmartBitcoins are Bitcoins living natively on the RSK platform, they can be transferred back to the Bitcoin blockchain at any given time for a standard RSK transaction fee.

The federation that controls the asset transfers on RSK comprises of well-known and community respected members/entities. Each entity of the federation is identified by a public key for the checkpoint signature scheme. An entity can be added or removed from the federation by means of an embedded predefined voting system. The addition/removal of an entity from the federation requires a high majority of votes.

The RSK platform aims to maximize the incentives for merged-mining. However, RSK is not completely dependent on merged-mining as it is robust to merge-mining shortages. In case of such situations, the federation automatically takes charge of the RSK network to keep it secure.

4.2.6 RQ3c: What consensus mechanism do these platforms utilize?

We now discuss the consensus mechanisms utilized by the platforms discussed in RQ3 to answer this research question:

4.2.6.1 Loom Consensus Mechanism

Loom allows for any consensus mechanism to be implemented on a personalized DApp chain, although the Loom SDK provides support for DPoS on a shared sidechain. In Delegated Proof of Stake, witnesses are elected who propose blocks and verify transactions. These witnesses serve a fixed term before elections take place again. Each voter is required to register with the account's public address and the power of each vote is proportional to the number of tokens that account holds. Accounts are permitted in DPoS to proxy their votes to trusted third parties who vote with power proportional to proxy balance + sum (balance of principles).

4.2.6.2 POA Consensus Mechanism

The POA network takes advantage of Proof of Authority consensus mechanism. The validators make all of the governance decisions through exclusive Distributed Applications. US public notaries serve as the validators on the network. The validators must be publicly known individuals whose participation can be easily reviewed adding a layer known as an Identity at Stake model[30]. The POA network rewards validators based on the amount staked. Currently, there are a total of 23 validators throughout the United States.

4.2.6.3 Liquid Consensus Mechanism

Dilley et al. (Dilley et al., 2016) recognized both, the latency issues with using a Proof-of-work consensus mechanism and using a centralized system. Inspired by that, the authors decided to implement Liquid in a manner that would allow users to transfer assets between blockchains by providing explicit Proof-of-Possession (PoP) within transactions. Building up on the idea of federated two-way peg design introduced by Back et al (Back et al., 2014), the authors have introduced the concept of Strong Federations (Dilley et al., 2016). Strong Federations are made up of two types of independent entities, namely:

- **Block-signers:** maintain the blockchain consensus and to advance the sidechain. They sign transaction blocks on the sidechain.

[30] https://blockonomi.com/proof-of-authority/

- **Watchmen:** are responsible for transferring assets from the sidechain to the mainchain by signing transactions on the mainchain. Thus, they are only required to be online when assets are beings transferred between the blockchain.

In a Strong Federation, entities that form the federation cannot directly control a user's assets on the system other than their own. In such systems, just the knowledge of a private key is enough to practice the *right to spend* and hence, no intervention of a third party is required. Strong federations also have a mechanism that allows settlements to be transferred back to the mainchain in case of a federation failure.

Liquid replaces dynamic miner (such as in Bitcoin) with a fixed signer set for a federation to have low latency and eliminate the risk of reorganization from a given hostile minority. It implements a validation of a script (which can be static or can change subject to fixed rules) instead of a Proof-of-Work consensus protocol similar to private chains (Friedenbach & Timón, 2013). In federated two-way pegged chains, as discussed in RQ1, the script implements a 'n' of 'm' multi-signature scheme which requires each block to be signed by a predetermined number of signers/entities (for instance 'n' of 'm' signers/entities). As a result, this mechanism can achieve Bitcoin like Byzantine robustness as a minority of malicious entities would not be able to affect the system. Figure 8 depicts how the consensus is achieved on the Liquid platform.

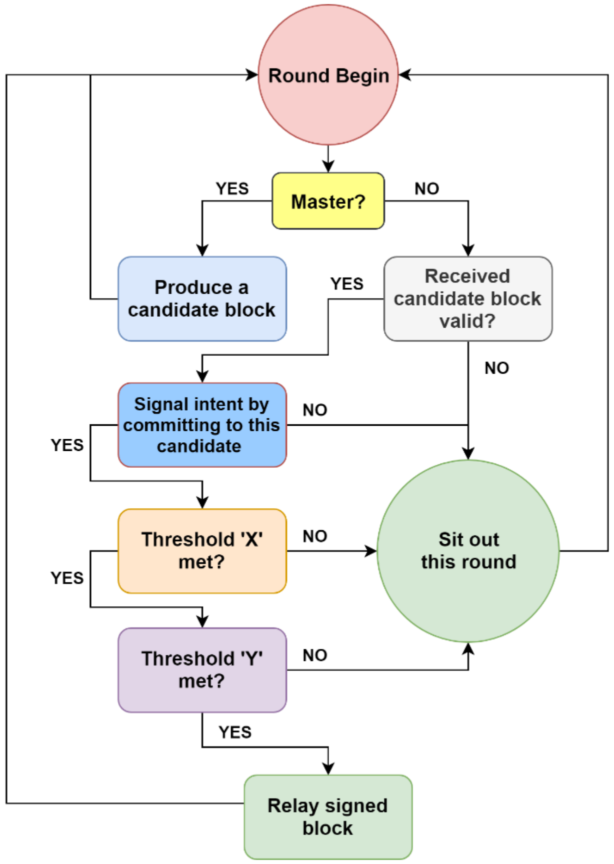


Figure 17: Block-signing by entities on the Liquid platform

Figure 17 can be summarized in the following steps:

- Entities propose candidate blocks in a round-robin fashion to all other signing participants.
- Each entity signals its intent by pre-committing to sign the given candidate block.
- If threshold X is met, each entity signs the block.
- If threshold Y (which may be different from X) is met, the block is accepted and sent to the network.
- The next block is then proposed by the next entity in the round-robin.

In Bitcoin, there is a tendency for chain reorganization in the newly added blocks due to the probabilistic generation of blocks (Eyal & Sirer, 2018). Since block generation in case of strong federations are based on a fixed set of block signers instead of being probabilistic, Liquid chain never reorganizes. This allows significantly faster transaction confirmation times than Bitcoin.

4.2.6.4 RSK Consensus Mechanism

While mining on the Bitcoin blockchain, conflicting situations may arise when multiple miners solve a block at the same chain height. In such situations, it becomes hard to decide which miner's block to select and add to the network. Additionally, miners are often required to stop mid-state and restart mining on new blocks each time a new block is solved and added to the network. These situations result in poor mining efficiency, greater network latencies, and mining time gaps.

To mitigate this RSK utilizes DECOR+ (S. D. Lerner, 2015) protocol, a reward sharing scheme which reduces competition while mining providing miners with the option to switch to the newest block later. With DECOR+ conflicts are resolved deterministically when all nodes have the same blockchain state information, the resolution is chosen in such a way that it maximizes the revenue for all miners involved whether they were involved in the conflict or not. The protocol has the following main features:

- If a miner switches each time a new block is accepted to the RSK network, they compete for a full block reward.
- If a miner switches late i.e. they keep mining older blocks, they create uncles[31] and earn a share of the block reward.

[31] https://www.investopedia.com/terms/u/uncle-block-cryptocurrency.asp

In neither of these situation blocks are fully orphaned[32], as the DECOR+ protocol pays a reward to uncles, which are counted as normal blocks (GHOST protocol (Sompolinsky & Zohar, 2016)). This greatly increases the efficiency of mining on RSK.

When the RSK hashing power is below 50% of the total Bitcoin hashing power, the network could be vulnerable to 51% attacks and double spending problems. To prevent such situations, RSK utilizes federated checkpoints, which are signed by federation entities and can be used by a client to decide which is the best block with the help of multi-signature majority. Moreover, if the total RSK hashing power goes below 5% of the total Bitcoin hashing power, the federation would be able to create signed blocks. Finally, the clients stop using federated checkpoints by defaults if the total RSK hashing power is over 66% of the Bitcoin hashing power and the paid fees in a block is higher than or equal to the average reward of a Bitcoin block.

4.2.7 RQ3d: How do these platforms impact the scalability of their mainchain?

Table 5 provides a comparison of Ethereum, Loom network and the POA network based on average block confirmation time, transaction rate, smart contract execution capability, security guarantee and if the transactions are confidential.

As it can be seen form Table 5, the similarities between the mainchain i.e. Ethereum and its sidechain are that all the platform support smart contract execution and none of them support private transactions. The table also shows that the loom network has the fastest block confirmation times and supports a high rate of transactions.

[32] https://www.investopedia.com/terms/o/orphan-block-cryptocurrency.asp

Table 5: Comparison of Ethereum, Loom, and the POA network

Features	**Ethereum[33]** **(mainchain)**	**Loom[34]** **(sidechain)**	**POA network[35]** **(sidechain)**
Average block confirmation time	~ 15 seconds	~1 second	~ 5 seconds
Transactions rate	~ 15 transactions/sec	>> 1 transaction / sec	~ 60 transaction / sec
Turing complete Smart contract execution	Yes	Yes	Yes
Security guarantee	Staking	Validators + Voters	Validators
Confidential transactions	No	No	No

Table 6 summarizes the key differences between Bitcoin, Liquid and RSK based on average block confirmation time, transaction rate, smart contract execution capability, security guarantee and if the transactions are confidential.

Table 6: Comparison of Bitcoin, RSK and Liquid

Features	**Bitcoin** **(mainchain)**	**Liquid[36]** **(sidechain)**	**RSK[37]** **(sidechain)**
Average block confirmation time	~ 10 minutes	~ 1 minute	~ 30 seconds
Transactions rate	~ 7 transactions/second	>> 1 transaction/second	300 – 1000 transactions/second
Turing complete Smart contract execution	No	No	Yes
Security guarantee	SHA256D miners	Strong federation	SHA256D merger miners + federation
Confidential transactions	No	Yes	Planned for future

[33] https://etherscan.io/
[34] https://blockexplorer.loomx.io
[35] https://blockscout.com/poa/core/
[36] https://blockstream.com/liquid/
[37] https://stats.rsk.co/

It is clear from Table 6 that only RSK supports the execution of smart contracts. It also has the fastest block confirmation times and highest transaction rates compared to the Bitcoin (mainchain) blockchain or the Liquid network.

4.2.8 RQ3e: What are the limitations of these platforms?

To answer this research question and to conclude the set of research questions discussing second layer scaling solution, we now discuss some of the limitations of state-of-the-art platforms discussed in RQ3.

4.2.8.1 Limitations of Loom

Some of the limitations of the Loom network are as follows:

- The entire transaction history of the sidechain is stored on the Ethereum mainchain instead of the sidechain itself decreasing the data integrity of the sidechain. The Merkle roots of the entire transaction history of the sidechain is periodically updated on the mainchain leaving open opportunities for an attack in between updates of the sidechain's transaction history (Bharel, 2019).
- To further increase the reliability on the mainchain, the security guarantees of the Loom network hinge on the ability to transfer tokens back to the mainchain. If the tokens are not approved for transfer back to the mainchain, the tokens can be at risk of being compromised. Loom's security is based on the mainchain being the target of an attack and not the sidechain a game is running on. There is more incentive in putting forth the resources to take over the Ethereum mainchain then a DApp supporting a decentralized game. Loom uses Plasma to securely transfer tokens back to the mainchain without needing

to trust the consensus algorithm on the sidechain. In a plasma exit, this is where a Merkle proof needs to be presented and can be challenged and the exit can fail.

- Another limitation of the Loom network is being restricted to OS X and Linux operating systems. The closest support for Windows is the Windows subsystem for Linux. Also, Loom's transfer gateway functionality can hurt the performance of the transfer of tokens between the two blockchains. The transfer gateway depends on an active presence on the Loom network and if there is not one, the transfer of tokens will be delayed.
- Loom network is based on federated two-way pegs, which introduce centralization in its blockchain-sidechain ecosystem as discussed in RQ1.

4.2.8.2 Limitations of POA

Some of the limitations of the POA network are as follows:

- POA network suffers from the problem of centralization due to the power that the 23 validators hold. The governance of the network is entirely determined by these validators. These validators reside solely in the United States and are public notaries of the United States. They are chosen by individual qualities such as public reputation, personal knowledge, and experience. They also need to be diverse geographically within the United States, so validators come from different states. One of the restrictions on adding to the number of validators is finding potential validators that meet the needed qualifications.
- Since all the validators of the POA network are based in the United States, this introduces geographical centralization element in the network. This type of model is undesirable as the validators may choose to censor information from other regions or countries.

- The POA Network plans on an increase in validators but there is worry that an increase in the number of validators will impede the performance of the network as it would take a longer time for block-signatures and hence, transaction confirmations.

4.2.8.3 Limitations of Liquid

Some of the limitations of the Liquid platform are as follows:

- Currently, only members of the Liquid network can run full nodes. Although the developers plan to allow other users to run full nodes to validate the network, it is not feasible at its current state.
- Liquid nodes require more computing resources than Bitcoin as the platform requires a Bitcoin node alongside the Liquid node to be able to validate asset transfers.
- The liquid network uses federated two-way pegs which introduces political centralization in the sidechain ecosystem.

4.2.8.4 Limitations of RSK

Some of the limitations of the RSK platforms are as follows:

- Currently, the RSK main-net is not available to all developers. The platform currently employs a whitelisting process where a development team/company is required to have a fully functional/semi-functional project approved by RSK to gain access to the network for testing and deployment on the platform. The whitelisting process can take a minimum of 3 days for approval. The platform aims to open the network for all users once the first stage of the bounty hunting program is completed.
- The use of federated two-way pegs introduces political centralization in the sidechain ecosystem as discussed in RQ1.

4.2.8.5 Comparison of Sidechain Platforms

Table 7 provides a comparative summary of Loom, the POA Network, Liquid and RSK platforms based on possible use cases, consensus mechanism, two-way peg design and limitations. The table also highlights the advantages that these platforms provide over their parent chains.

Table 7: Comparison of sidechain platforms

Platform	Use Cases	Consensus Mechanism	Two-way peg design	Advantages over mainchain	Limitations
Loom	DelegateCall, Game development, scalable DApps	Delegated Proof-of-Stake (DPoS)/any consensus mechanism	Federated two-way peg	Scalability, Efficiency needed for games	1. limited Windows (OS) support 2. If the tokens are not approved for transfer back to the mainchain, the tokens can be at risk of being compromised 3. The Loom Network runs on the idea that it is not necessary to store every transaction on the sidechain 4. Centralization due to federated two-way peg
POA	Scalable smart contracts	Proof-of-Authority	Federated two-way peg	Interoperability between blockchains	1. Centralization due to federated two-way peg. 2. geographically centralized which may introduce censorship 3. Plans to increase the number of validators which could impede performance.
Liquid	International Exchange	Proof-of-Possession	Federated two-way peg	Faster transaction rates than Bitcoin	1. Currently not open to all users 2. Running Liquid full nodes requires more resources than running Bitcoin full nodes 3. Centralization due to federated two-way peg
RSK	Retail Payment Systems, Supply Chain Traceability, Digital Identity	Proof-of-work based merged-mining with Bitcoin, DECOR+	Federated two-way peg	Ability to execute smart contracts	1. Currently not open to all users/developers 2. Centralization due to federated two-way peg

An interesting observation from Table 7 suggests that all the four platforms discussed in this section use federated two-way pegs. This is because in its current state it is not possible to implement SPV based two-way pegs in Bitcoin, due to missing opcodes from its protocol (See Chapter 5 – sub-section 5.2.3). Whereas, when it comes to the Ethereum based sidechain platforms, the Loom network intends to implement a more robust, secure and decentralized two-way peg design in the future and finally, the POA network's decision to implement a federated two-way peg was based on the idea of preservation of a human element in a blockchain ecosystem.

4.3 Other projects and frameworks

There are other innovative sidechain projects and frameworks that slightly fell short of our criteria for selection. The reasons why these projects were not selected were because of incomplete and/or active development, technical difficulties and lack of thorough documentation and support.

Plasma is a framework proposed by Buterin and Poon (Buterin & Poon, n.d.), which may have the potential to provide highly scalable solutions for the blockchain-based decentralized financial industry as it incentivizes and enforces the execution of smart contracts. The platform is potentially aiming to achieve more than a billion state updates per second. The smart contracts running on the platform are incentivized to continue operation autonomously with the help of network transaction fees. This process ultimately relies on the underlying blockchain (for instance, Ethereum) to enforce transactional state transitions.

The Elements project (BlockStream, n.d.) was launched in June 2015. It is an open-source, blockchain platform which is also sidechain-capable. It provides features such as Issued assets and confidential transactions. Blockchains developed with the Elements platform can be configured and developed to either run as standalone blockchains or as pegged sidechains to other blockchains which allow assets to be transferred between disparate blockchains. It utilizes and extends the current Bitcoin codebase; hence, it allows developers to take advantage of the *bitcoind*[38] Application Programming Interface (API) to develop blockchains and test proof-of-concept projects. Since Elements is built upon the Bitcoin's codebase, it can also serve as a test-net for introducing changes to the Bitcoin protocol.

In the context of the Elements platform, a sidechain is an extension to an existing blockchain. Assets are transferable between chains allowing the main chain to benefit from the enhanced features of the sidechain, such as rapid transfer finality and confidential transactions. While a sidechain is aware of the main chain and its transaction history, the main chain has no awareness of the sidechain, and none is required for its operation. This enables sidechains to innovate without restriction or the delays associated with main chain protocol improvement proposals. Indeed, rather than trying to alter it directly, extending the main protocol with a sidechain allows the main chain itself to remain secure and specialized, underpinning the smooth operation of the sidechain.

[38] https://bitcoin.org/en/developer-reference#serialized-blocks

Chapter 5. Open Issues and Future Directions

Both first- and second-layer scaling solutions are innovative approaches for improving the scalability of public and permissionless blockchains. But based on our research, they require extensive further research for the advancement of the blockchain domain, this is because in their current state these solutions face multiple issues which need to be addressed. In this chapter, we will discuss some of the major issues surrounding the proposed first- and second-layer scaling solutions and then, we will propose initial steps that could be taken to address or mitigate these issues.

5.1 First-layer scaling solutions

First-layer scaling such as segregated witness and sharding face multiple implementation threats and challenges such as threats of a hard fork, miner incentivization, extreme physical resource requirement, etc. In this section, we will discuss some of the issues that the blockchain community faces when it comes to implementing first-layer scaling solutions.

5.1.1 Data Validation in Sharding

Consider Figure 18 on which Shard 1 is corrupted and a malicious actor produces invalid block *B*. Suppose in this block *B* 1000 tokens were minted out of thin air on Alice’s account. The malicious actor then produces valid block *C* (in a sense that the transactions in *C* are applied correctly) on top of *B*, obfuscating the invalid block B, and initiates a cross-shard transaction to Shard 2 that transfers those 1000 tokens to Bob’s account. From this moment the improperly created tokens reside on an otherwise completely valid blockchain in Shard 2. A few simple approaches to tackle this issue could be:

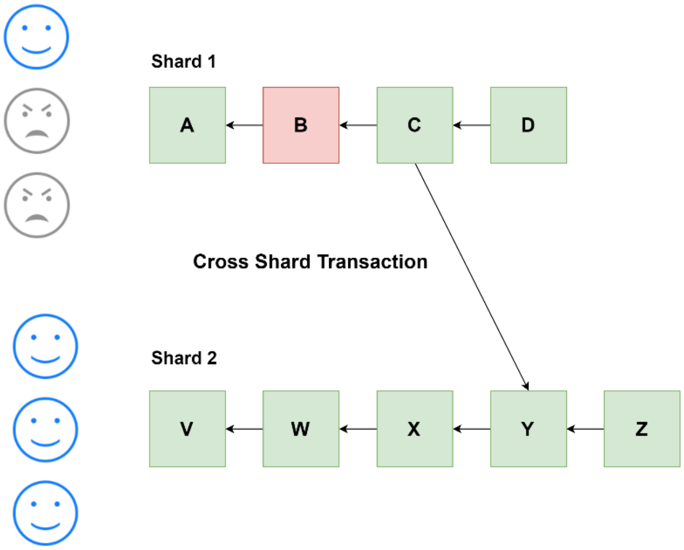


Figure 18: Data validation problem in sharding

- For validators of Shard 2 to validate the block from which the transaction is initiated. This won't work even in the example above since block C appears to be completely valid.
- For validators in Shard 2 to validate some large number of blocks preceding the block from which the transaction is initiated. Naturally, for any number of blocks N validated by the receiving shard, the malicious validators can create N+1 valid blocks on top of the invalid block they produced (Skidanov, 2018c).

A promising idea to resolve this issue would be to arrange shards into an undirected graph in which each shard is connected to several other shards and only allow cross-shard transactions between neighboring shards (Skidanov, 2018c). If a cross-shard transaction is needed between shards that are not neighbors, such transaction is routed through multiple shards (Skidanov, 2018a), (Martino, Quaintance, & Popejoy, n.d.). In this design, a validator in each shard is expected to validate both all the blocks in their shard as well as all the blocks in all the neighboring shards.

Consider Figure 19 below with 10 shards, each having four neighbors, and no two shards requiring more than two hops for a cross-shard communication:

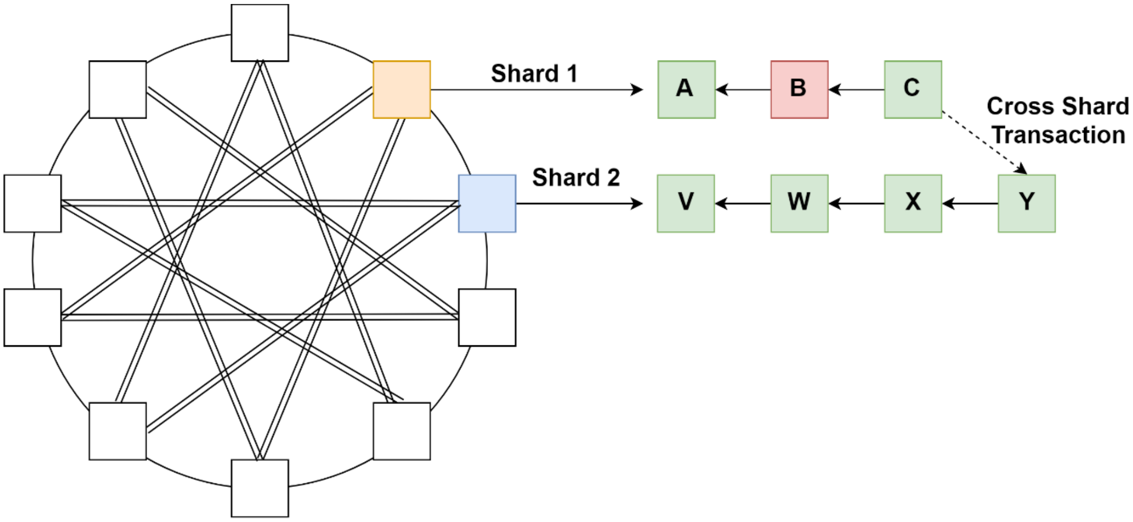


Figure 19: Cross-shard transactions amongst neighboring shards

Shard 2 is not only validating its own blockchain, but also blockchains of all the neighbors, including Shard 1. So, if a malicious actor on Shard 1 is attempting to create an invalid block *B*, then build block *C* on top of it and initiate a cross-shard transaction, such cross-shard transaction will not go through since Shard 2 will have validated the entire history of Shard 1 which will cause it to identify invalid block *B*.

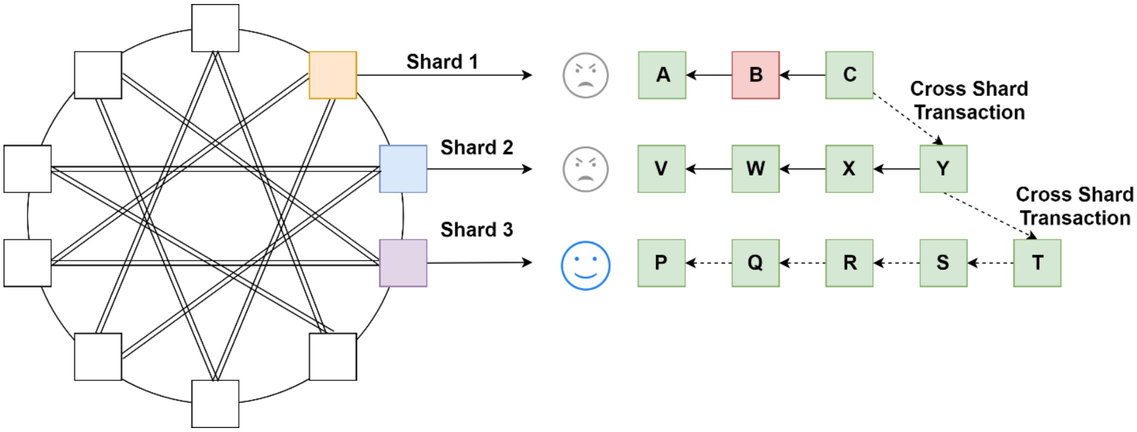


Figure 20: An adversary executing cross-shard transaction

While corrupting a single shard is no longer a viable attack, corrupting a few shards remains a problem. Figure 20 shows an adversary corrupting both Shard 1 and Shard 2 successfully executes a cross-shard transaction to Shard 3 with funds from an invalid block *B*: Shard 3 validates all the blocks in Shard 2, but not in Shard 1, and has no way to detect the malicious block.

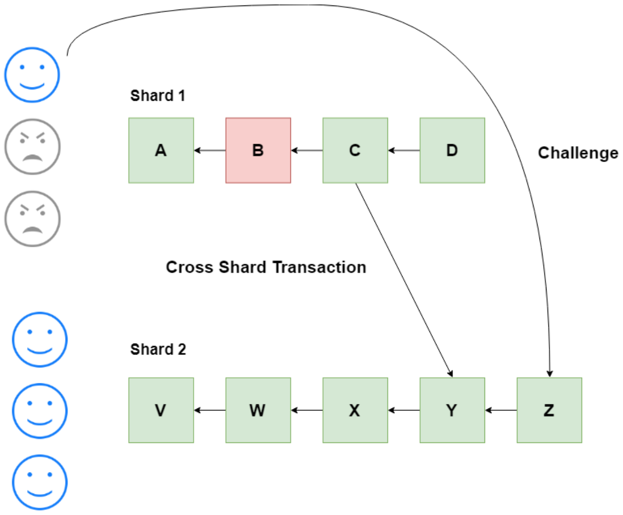


Figure 21: A possible approach for data validation

The idea behind solving the data validation problem in blockchain sharding is shown in Figure 21: whenever a block header is communicating between chains for any purpose (such as cross-linking to the beacon chain, or a cross-shard transaction), there'd a period during which any honest validator can provide a proof that the block is invalid.

There are various constructions that enable very succinct proofs that the blocks are invalid, so the communication overhead for the receiving nodes is much smaller than that of receiving a full block. With this approach for as long as there's at least one honest validator in the shard, the system would be secure.

5.1.2 A threat of hard forks

We have emphasized a major problem with the implementation of first-layer scaling solutions on public and permissionless blockchains (Chapter 3 - RQ3) i.e. the agreement within the miners regrading a specific change in the protocol of the blockchain. There will often be times when a proposed change in protocol for a blockchain will not be agreed upon by the majority of the miners of the blockchain network. This leads to a difficult situation in which forcing a change in the protocol of the blockchain is likely to cause a hard fork of the network as can be seen in the case of Bitcoin forking into Bitcoin Cash. Thus, this makes public and permissionless blockchains such as Bitcoin and Ethereum extremely stringent and deterrent to change, even when the change is for the overall improvement of the blockchain network. This aspect of a public blockchain is a major limitation and a huge stumbling block in the implementation of first-layer scaling solutions.

Often times lack of proper communication amongst the miners of a blockchain regarding a proposed change in protocol might also result in change resistance. Although there are protocols available for proposing changes to the Ethereum and Bitcoin protocol such as, Ethereum Improvement Proposals (EIP) (Ethereum, n.d.) and Bitcoin Improvement Proposals (BIP) (Bitcoin, 2011), these proposals are hardly ever advertised to the blockchain community especially the miners of these networks (Khatwani, 2017). Hence, we suggest a lightweight broadcasting and voting system for clearly describing and advertising the proposed changes, their advantages and limitations to the miners of a blockchain network. The system can then be used for conducting polls within the miners on whether a change in the protocol should be implemented or not. This would be an effective way of conveying or advertising a change in protocol for a public blockchain amongst its miners instead of directly forcing a change which may result in a hard-fork or just waiting and hoping that the miners will reach an agreement regarding the change at some point in the future.

5.2 Second-layer scaling solutions

Sidechains are still relatively new proposals and are by no means mature enough to change the blockchain world at this time, but they sure are promising for the future of the blockchain industry. In this section, we discuss the most important open issues in the infant sidechain domain and suggest future measures and recommendations for the mitigation or elimination of these issues.

5.2.1 Centralization in federated two-way pegs

Political decentralization is an important characteristic of a blockchain network, as discussed in Chapter 4 – RQ1, federated two-way pegs introduce a level of political centralization in the sidechain ecosystem. Hence, it is important to identify and select honest and trusted entities to form a federation for the security and integrity of a network. It is extremely critical that entities have their economic interests well aligned with the proper functioning of a federation. It would obviously be a mistake to rely on a random assortment of volunteers to support a commercial sidechain holding significant value. Beyond the incentive (R M Parizi & Dehghantanha, 2018) to attempt to extract any value contained on the sidechain, these volunteers would also have little incentive to ensure the reliability of the network. To mitigate these concerns, we propose a federation should at least have the following attributes which may potentially lead to good results:

- Federations are most secure when each entity has a similar amount of value held by the federation. Incentives can be aligned using escrow, entity allocation, or external legal constructs such as insurance policies and surety bonds
- The total number of entities that form a federation should lie in the range - [15, 30]. This is to maintain political decentralization and still provide the users with the ability to verify the authenticity of each entity within the federation in a relatively short period of time.
- The identity and authenticity of each entity should be verifiable. Some ways to achieve this could be providing proof of identity with government issued ID's or licenses, proof of physical address, etc.
- Entities should be distributed geographically to prevent down-time in case of power failure, natural disasters, etc.
- Entities should be disparate from one another and should not engage in business with one another, this would eliminate conflict of interest and censorship.

5.2.2 Security of Federated two-way pegs

Federated two-way pegs introduce a security risk in the sidechain ecosystem. For instance, if the private keys of the majority of the network are compromised, then the assets locked in the lockbox (or on the sidechain) are vulnerable to theft. This is because as discussed in Chapter 4 – RQ1, a transaction in a federated two-way peg design requires 'n' of 'm' signatures to be approved (where 'n' is the majority in a total of 'm' entities). One way to mitigate this threat would be to migrate to

SPV based two-way peg design where the lockbox is usually controlled by the miners of the network and the only way to unlock the funds from the lockbox is to provide a valid SPV proof.

5.2.3 SPV based two-way pegs on Bitcoin

SPV proofs can provide a solution to the political centralization issue with federated two-way pegs. As discussed in Chapter 4 – RQ1, SPV proofs require no single entity or a group of entities (Federation), for transferring assets from the mainchain to the sidechain and vice versa. Unfortunately, SPV based two-pegs cannot be implemented on a sidechain pegged to the Bitcoin blockchain at this time. This is because in its current state Bitcoin is missing a few opcodes from its protocol such as:

- **OP_WITHDRAWPROOFVERIFY:** *OP_WITHDRAWPROOFVERIFY* would unlock 'reserve' coins on a sidechain. A user would need to provide inputs to an output such that the output would evaluate to true- which would unlock the reserve coins. The user would then be credited on the sidechain with the amount of coins they locked up on the Bitcoin blockchain. The change on the sidechain would also be sent back to the federation's reserve address (Stewert, 2017).
- **OP_REORGPROOFVERIFY:** *OP_REORGPROOFVERIFY* would allow users to submit SPV proofs in the reorg-period (Chapter 4 – RQ1). This opcode would correct invalid states of two types: 1) double spends (Bala & Manoharan, 2018), (Bae & Lim, 2018) of a parent chain lock and 2) parent chain reorganizations (Elements, 2016).

We propose the addition of these opcodes to the Bitcoin protocol in the future. This would allow the community to implement sidechain technology with SPV based two-way pegs instead of relying on federated two-way pegs.

5.2.4 Lack of research support on current sidechain platforms

The sidechain domain is still relatively new and hence, most state-of-the-art sidechain platforms are still in development or bounty hunting phases. Based on the authors’ experimental experiences, registering or submitting DApps on some the platforms (e.g. Liquid and RSK) discussed in the previous chapter is extremely difficult and selective as the developers do not provide access to all users on their platforms at this time. To make matters worse some of these platforms are not integrated to the Bitcoin or Ethereum test-nets at this time (e.g. Liquid). This makes performing empirical studies by researchers or practitioners on these platforms extremely difficult and expensive due to the market value of Bitcoin and Ether cryptocurrencies. Empirical research is an important tool in software engineering (Malhotra, 2015) which can reveal hidden trends, patterns, anomalies and limitations of a software system (Reza M Parizi, Dehghantanha, Choo, & Singh, 2018). Hence, we strongly advocate the integration of these platforms to their parent chain’s test-nets. This would allow the researchers in the community to analyze and evaluate these platforms based on several attributes such as performance, security, and privacy which would help in speeding-up development process and the overall advancement of sidechain technology.

Chapter 6. Conclusions

In the last decade, blockchain technology has grown exponentially with seemingly new use cases being discovered almost every day. Consequently, research in the domain has picked up pace in recent years both to discover issues and vulnerabilities in blockchains and to provide solutions to these problems and challenges. Scalability and limited functionality have shackled blockchains ever since its proposal and implementation in 2008. In response to this, the community has proposed first- and second-layer scaling solutions.

First layer scaling solutions require changes in the codebase of existing blockchains. We have discussed two of the most common first layer scaling solutions i.e. Segregated witness and sharding. Second layer scaling solutions do not require changes to existing blockchain codebase, instead, these solutions propose an implementation of a second layer on top of existing blockchains, for instance, sidechains.

Although these solutions are promising, a comprehensive study is still lacking in the literature to study the impact of scaling solutions on the scalability of public blockchains such as Bitcoin and Ethereum. Moreover, there has been a lack of studies discussion on how and where these solutions can be effectively integrated into blockchains to remedy current issues in a clear context. Hence, the motivation of our study was to take the first step and provide a comprehensive review of 1) the available design choices for the first layer scaling solutions for public blockchains, and 2) state-of-the-art sidechain platforms based on their use cases, consensus mechanisms, asset transfer protocols, and limitations. This thesis also identifies current advancements, analyzes their impact from various viewpoints and proposes directions for the future of research and development, Moreover, we have discussed general open issues that need well-deserved attention from the community for the advancement of the overall blockchain domain.

References

Abdellatif, T., & Brousmiche, K.-L. (2018). Formal Verification of Smart Contracts Based on Users and Blockchain Behaviors Models. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)* (pp. 1–5). IEEE. https://doi.org/10.1109/NTMS.2018.8328737

Amani, S., Bégel, M., Bortin, M., & Staples, M. (2018). Towards Verifying Ethereum Smart Contract Bytecode in Isabelle/HOL. In *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs* (pp. 66–77). New York, NY, USA: ACM. https://doi.org/10.1145/3167084

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., … Yellick, J. (2018). Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In *Proceedings of the Thirteenth EuroSys Conference* (p. 30:1--30:15). New York, NY, USA: ACM. https://doi.org/10.1145/3190508.3190538

Aniello, L., Baldoni, R., Gaetani, E., Lombardi, F., Margheri, A., & Sassone, V. (2017). A Prototype Evaluation of a Tamper-Resistant High Performance Blockchain-Based Transaction Log for a Distributed Database. In *2017 13th European Dependable Computing Conference (EDCC)* (pp. 151–154). https://doi.org/10.1109/EDCC.2017.31

Arasev, V. (2018). POA Network Whitepaper. Retrieved January 30, 2019, from https://github.com/poanetwork/wiki/wiki/POA-Network-Whitepaper

Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A Survey of Attacks on Ethereum Smart Contracts (SoK). In *International Conference on Principles of Security and Trust* (pp. 1–24). Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-662-54455-6

Back, A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A., … Wuille, P. (2014).

Enabling Blockchain Innovations with Pegged Sidechains. Retrieved January 25, 2019, from http://www.blockstream.com/sidechains.pdf

Bae, J., & Lim, H. (2018). Random Mining Group Selection to Prevent 51% Attacks on Bitcoin. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)* (pp. 81–82). https://doi.org/10.1109/DSN-W.2018.00040

Bala, R., & Manoharan, R. (2018). Security Enhancement In Bitcoin Protocol. In *2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 1–4). https://doi.org/10.1109/WiSPNET.2018.8538578

Bentley, D. (2018). Loom's Popular dApps Enable Ethereum Development at Scale. Retrieved January 30, 2019, from https://www.the-blockchain.com/2018/07/11/looms-dapps-gain-traction-as-go-to-resource-for-ethereum-development/

Bharel, D. (2019). Plasma Cash Developers' Guide: Everything You Need to Know (+ How to Use Loom's Plasma CLI). Retrieved January 24, 2019, from https://medium.com/loom-network/plasma-cash-developers-guide-everything-you-need-to-know-how-to-use-looms-plasma-cli-6f7b7a3c78d1

Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., … Zanella-Béguelin, S. (2016). Formal Verification of Smart Contracts: Short Paper. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security* (pp. 91–96). New York, NY, USA: ACM. https://doi.org/10.1145/2993600.2993611

Bitcoin. (2011). Bitcoin Improvement Proposals (BIP). Retrieved July 4, 2019, from https://github.com/bitcoin/bips

Blockstream. (n.d.). Liquid. Retrieved January 30, 2019, from https://blockstream.com/liquid/

BlockStream. (n.d.). Elements. Retrieved January 11, 2019, from https://elementsproject.org/

Buterin, V. (2017). *Sharding: Making Blockchains Scalable, Decentralized and Secure*. Retrieved from https://vitalik.ca/files/Ithaca201807_Sharding.pdf

Buterin, V., & Poon, J. (n.d.). Plasma: Scalable Autonomous Smart Contracts. Retrieved January 3, 2019, from https://plasma.io/plasma.pdf

Cai, S., Yang, N., & Ming, Z. (2018). A Decentralized Sharding Service Network Framework with Scalability. In H. Jin, Q. Wang, & L.-J. Zhang (Eds.), *Web Services -- ICWS 2018* (pp. 151–165). Cham: Springer International Publishing.

Chauhan, A., Malviya, O. P., Verma, M., & Mor, T. S. (2018). Blockchain and Scalability. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 122–128). IEEE. https://doi.org/10.1109/QRS-C.2018.00034

Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., … others. (2016). On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security* (pp. 106–125).

Deng, L., Chen, H., Zeng, J., & Zhang, L.-J. (2018). Research on Cross-Chain Technology Based on Sidechain and Hash-Locking. In S. Liu, B. Tekinerdogan, M. Aoyama, & L.-J. Zhang (Eds.), *Edge Computing -- EDGE 2018* (pp. 144–151). Cham: Springer International Publishing.

Dennis, R., Owenson, G., & Aziz, B. (2016). A temporal blockchain: A formal analysis. *Proceedings - 2016 International Conference on Collaboration Technologies and Systems, CTS 2016*, 430–437. https://doi.org/10.1109/CTS.2016.80

Dilley, J., Poelstra, A., Wilkins, J., Piekarska, M., Gorlick, B., & Friedenbach, M. (2016). Strong Federations: An Interoperable Blockchain Solution to Centralized Third-Party Risks. *ArXiv*

Preprint ArXiv:1612.05491.

Dolce, A. (2018). Blockchain Scalability Solutions: Overview of Crypto Scaling Solutions. Retrieved January 4, 2019, from https://masterthecrypto.com/blockchain-scalability-solutions-crypto-scaling-solutions/

Douceur, J. R. (2002). The Sybil Attack. In P. Druschel, F. Kaashoek, & A. Rowstron (Eds.), *Peer-to-Peer Systems* (pp. 251–260). Berlin, Heidelberg: Springer Berlin Heidelberg.

Duboc, L., Rosenblum, D. S., & Wicks, T. (2006). A Framework for Modelling and Analysis of Software Systems Scalability. In *Proceedings of the 28th International Conference on Software Engineering* (pp. 949–952). New York, NY, USA: ACM. https://doi.org/10.1145/1134285.1134460

Ehmke, C., Wessling, F., & Friedrich, C. M. (2018). Proof-of-Property - A Lightweight and Scalable Blockchain Protocol. In *2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)* (pp. 48–51).

Elements. (2016). The Federated Peg in Elements Alpha. Retrieved January 14, 2019, from https://github.com/ElementsProject/elementsproject.org/blob/master/source/_posts/the-federated-peg-in-elements-alpha.md

Ethereum. (n.d.). Ethereum Improvement Proposals (EIP). Retrieved from http://eips.ethereum.org/

Eyal, I., & Sirer, E. G. (2018). Majority is Not Enough: Bitcoin Mining is Vulnerable. *Commun. ACM*, *61*(7), 95–102. https://doi.org/10.1145/3212998

Feinstein, L., Schnackenberg, D., Balupari, R., & Kindred, D. (2003). Statistical approaches to DDoS attack detection and response. In *Proceedings DARPA Information Survivability Conference and Exposition* (Vol. 1, pp. 303–314 vol.1).

https://doi.org/10.1109/DISCEX.2003.1194894

Feng, X., Ma, J., Miao, Y., Meng, Q., Liu, X., Jiang, Q., & Li, H. (2018). Pruneable sharding-based blockchain protocol. *Peer-to-Peer Networking and Applications*. https://doi.org/10.1007/s12083-018-0685-6

Fiaidhi, J., Mohammed, S., & Mohammed, S. (2018). EDI with Blockchain as an Enabler for Extreme Automation. *IT Professional*, *20*(4), 66–72. https://doi.org/10.1109/MITP.2018.043141671

Frey, D., Makkes, M. X., Roman, P.-L., Ta\"\iani, F., & Voulgaris, S. (2016). Bringing Secure Bitcoin Transactions to Your Smartphone. In *Proceedings of the 15th International Workshop on Adaptive and Reflective Middleware* (p. 3:1--3:6). New York, NY, USA: ACM. https://doi.org/10.1145/3008167.3008170

Friedenbach, M., & Timón, J. (2013). Freimarkets: extending bitcoin protocol ´ with user-specified bearer instruments, peer-to-peer exchange, off-chain accounting, auctions, derivatives and transitive transactions. Retrieved January 14, 2019, from http://freico.in/docs/freimarkets-v0.0.1.pdf

Fynn, E., & Pedone, F. (2018). Challenges and Pitfalls of Partitioning Blockchains. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)* (pp. 128–133). https://doi.org/10.1109/DSN-W.2018.00051

Gencer, A. E., van Renesse, R., & Sirer, E. G. (2017). Short Paper: Service-Oriented Sharding for Blockchains. In A. Kiayias (Ed.), *Financial Cryptography and Data Security* (pp. 393–401). Cham: Springer International Publishing.

Giaglis, G., Kosba, A., Miller, A., Shi, E., Wen, Z., Papamanthou, C., … Deogun, J. S. (2017). Under-optimized smart contracts devour your money. In *2017 26th International*

Conference on Computer Communication and Networks (ICCCN) (Vol. 55, pp. 1–5). https://doi.org/10.1109/SANER.2017.7884650

He, G., Su, W., & Gao, S. (2018). Chameleon: A Scalable and Adaptive Permissioned Blockchain Architecture. In *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)* (pp. 87–93). https://doi.org/10.1109/HOTICN.2018.8606007

Herlihy, M. (2018). Atomic Cross-Chain Swaps. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing* (pp. 245–254). New York, NY, USA: ACM. https://doi.org/10.1145/3212734.3212736

Kan, L., Wei, Y., Hafiz Muhammad, A., Siyuan, W., Linchao, G., & Kai, H. (2018). A Multiple Blockchains Architecture on Inter-Blockchain Communication. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 139–145). IEEE. https://doi.org/10.1109/QRS-C.2018.00037

Karame, G. O., Androulaki, E., & Capkun, S. (2012). Double-spending Fast Payments in Bitcoin. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (pp. 906–917). New York, NY, USA: ACM. https://doi.org/10.1145/2382196.2382292

Khatwani, S. (2017). What is a BIP (Bitcoin Improvement Proposal)? Why do you need to know about it? Retrieved from https://coinsutra.com/bip-bitcoin-improvement-proposa/

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, *33*(TR/SE-0401), 28. https://doi.org/10.1.1.122.3308

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. *Engineering*, *2*, 1051. https://doi.org/10.1145/1134285.1134500

Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., & Ford, B. (2018). OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding. In *2018 IEEE*

Symposium on Security and Privacy (SP) (pp. 583–598). https://doi.org/10.1109/SP.2018.000-5

Lai, R. (2018). Sentinel Chain, 1–50.

Lerner, S. (2016). Drivechains, Sidechains and Hybrid 2-way peg Designs. Retrieved January 7, 2019, from https://docs.rsk.co/Drivechains_Sidechains_and_Hybrid_2-way_peg_Designs_R9.pdf

Lerner, S. D. (2015). RSK: White Paper Overview. Retrieved January 30, 2019, from https://docs.rsk.co/RSK_White_Paper-Overview.pdf

Loom. (n.d.-a). Loom Netwrok. Retrieved January 25, 2019, from https://loomx.io/

Loom. (n.d.-b). Loom SDK Documentation. Retrieved January 30, 2019, from https://loomx.io/developers/docs/en/basic-install-all.html

Lou, J., Zhang, Q., Qi, Z., & Lei, K. (2018). A Blockchain-based key Management Scheme for Named Data Networking. In *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)* (pp. 141–146). IEEE. https://doi.org/10.1109/HOTICN.2018.8605993

Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., & Saxena, P. (2016). A Secure Sharding Protocol For Open Blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 17–30). New York, NY, USA: ACM. https://doi.org/10.1145/2976749.2978389

Malhotra, R. (2015). *Empirical Research in Software Engineering: Concepts, Analysis, and Applications*. Chapman & Hall/CRC.

Malik, S., Kanhere, S. S., & Jurdak, R. (2018). ProductChain: Scalable Blockchain Framework to Support Provenance in Supply Chains. In *2018 IEEE 17th International Symposium on*

Network Computing and Applications (NCA) (pp. 1–10). https://doi.org/10.1109/NCA.2018.8548322

Manshaei, M. H., Jadliwala, M., Maiti, A., & Fooladgar, M. (2018). A Game-Theoretic Analysis of Shard-Based Permissionless Blockchains. *IEEE Access*, *6*, 78100–78112. https://doi.org/10.1109/ACCESS.2018.2884764

Martino, W., Quaintance, M., & Popejoy, S. (n.d.). Chainweb: A Proof-of-Work Parallel-Chain Architecture for Massive Throughput. Retrieved from https://kadena.io/docs/chainweb-v15.pdf

McManus, B. (2017). Understanding Segwit and the Bitcoin Scaling Debate. Retrieved from https://medium.com/@brenmcma/understanding-segwit-and-the-bitcoin-scaling-debate-c9f7170e9e79

Miller, D. (2018). Blockchain and the Internet of Things in the Industrial Sector. *IT Professional*, *20*(3), 15–18. https://doi.org/10.1109/MITP.2018.032501742

Moore, T., & Christin, N. (2013). Beware the Middleman: Empirical Analysis of Bitcoin-Exchange Risk. In A.-R. Sadeghi (Ed.), *Financial Cryptography and Data Security* (pp. 25–33). Berlin, Heidelberg: Springer Berlin Heidelberg.

Mylrea, M., & Gourisetti, S. N. G. (2018). Blockchain for Supply Chain Cybersecurity, Optimization and Compliance. In *2018 Resilience Week (RWS)* (pp. 70–76). https://doi.org/10.1109/RWEEK.2018.8473517

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. https://doi.org/10.1007/s10838-008-9062-0

O`KeeffeDaniel. (2018). Bitcoin Liquid Network Launches To Complement Lightning. Retrieved January 24, 2019, from https://cryptodisrupt.com/bitcoin-liquid-network-launches-to-complement-lightning/

Parizi, R. M., Amritraj, & Dehghantanha, A. (2018). Smart Contract Programming Languages on Blockchains: An Empirical Evaluation of Usability and Security. In S. Chen, H. Wang, & L.-J. Zhang (Eds.), *Blockchain -- ICBC 2018* (pp. 75–91). Cham: Springer International Publishing.

Parizi, R. M., & Dehghantanha, A. (2018). On the Understanding of Gamification in Blockchain Systems. In *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)* (pp. 214–219). https://doi.org/10.1109/W-FiCloud.2018.00041

Parizi, R. M., Dehghantanha, A., Choo, K. K. R., & Singh, A. (2018). Empirical Vulnerability Analysis of Automated Smart Contracts Security Testing on Blockchains. In *28th Annual International Conference on Computer Science and Software Engineering (CASCON'18)*. ACM.

POA. (n.d.-a). POA Core. Retrieved January 30, 2019, from https://forum.poa.network/c/poa-core

POA. (n.d.-b). POA Network. Retrieved January 30, 2019, from https://poa.network/

POA. (n.d.-c). POA Partnerships. Retrieved January 30, 2019, from https://medium.com/poa-network/tagged/partnerships

POA. (2017). Proof of Authority: consensus model with Identity at Stake. Retrieved January 18, 2019, from https://medium.com/poa-network/proof-of-authority-consensus-model-with-identity-at-stake-d5bd15463256

Qixcoin. (n.d.). Retrieved from http://qixcoin.com/%0A

Ren, Z., Cong, K., Aerts, T., de Jonge, B., Morais, A., & Erkin, Z. (2018). A Scale-Out Blockchain for Value Transfer with Spontaneous Sharding. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)* (pp. 1–10).

https://doi.org/10.1109/CVCBT.2018.00006

Robinson, P. (2018). Requirements for Ethereum Private Sidechains. *ArXiv Preprint ArXiv:1806.09834*.

RootStock. (n.d.). RSK Partners. Retrieved January 30, 2019, from https://www.rsk.co/

RSK. (n.d.). RSK. Retrieved January 15, 2019, from https://www.rsk.co/

Sel, D., Zhang, K., & Jacobsen, H.-A. (2018). Towards Solving the Data Availability Problem for Sharded Ethereum. In *Proceedings of the 2Nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers* (pp. 25–30). New York, NY, USA: ACM. https://doi.org/10.1145/3284764.3284769

Skidanov, A. (2018a). So what exactly is Vlad's Sharding PoC doing? Retrieved March 4, 2019, from https://medium.com/nearprotocol/so-what-exactly-is-vlads-sharding-poc-doing-37e538177ed9

Skidanov, A. (2018b). The authoritative guide to Blockchain Sharding, part 1. Retrieved April 1, 2019, from https://medium.com/nearprotocol/the-authoritative-guide-to-blockchain-sharding-part-1-1b53ed31e060

Skidanov, A. (2018c). Unsolved Problems in Blockchain Sharding. Retrieved April 4, 2019, from https://medium.com/nearprotocol/unsolved-problems-in-blockchain-sharding-2327d6517f43

Solidity. (n.d.). Retrieved March 1, 2018, from https://solidity.readthedocs.io/en/develop/

Sompolinsky, Y., & Zohar, A. (2016). Bitcoin's security model revisited. *ArXiv Preprint ArXiv:1605.09193*.

Stewert, C. (2017). OP_WITHDRAWPROOFVERIFY — The op code that powers SPV sidechains. Retrieved January 15, 2019, from https://medium.com/@Chris_Stewart_5/op-withdrawproofverify-the-op-code-that-powers-spv-sidechains-cefce996a324

Wood, G. (2014). Ethereum: a secure decentralised generalised transaction ledger Yellow Paper. *Ethereum Project Yellow Paper*, 1–32. https://doi.org/10.1017/CBO9781107415324.004

Yoo, H., Yim, J., & Kim, S. (2018). The Blockchain for Domain Based Static Sharding. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)* (pp. 1689–1692). https://doi.org/10.1109/TrustCom/BigDataSE.2018.00252

Yu, S., Lv, K., Shao, Z., Guo, Y., Zou, J., & Zhang, B. (2018). A High Performance Blockchain Platform for Intelligent Devices. In *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)* (pp. 260–261). IEEE. https://doi.org/10.1109/HOTICN.2018.8606017

Yu, Y., Liang, R., & Xu, J. (2018). A Scalable and Extensible Blockchain Architecture. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 161–163). https://doi.org/10.1109/ICDMW.2018.00030

Zamani, M., Movahedi, M., & Raykova, M. (2018). RapidChain: Scaling Blockchain via Full Sharding. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (pp. 931–948). New York, NY, USA: ACM. https://doi.org/10.1145/3243734.3243853

Zhou, L., Wang, L., Sun, Y., & Lv, P. (2018). BeeKeeper: A Blockchain-based IoT System with Secure Storage and Homomorphic Computation. *IEEE Access*, 1. https://doi.org/10.1109/ACCESS.2018.2847632