Master’s Thesis

Using Blockchain Technology and Smart Contracts to Create a Distributed Securities Depository

Eric Wall
Gustaf Malm

Department of Electrical and Information Technology,
Faculty of Engineering, LTH, Lund University, 2016.

Using Blockchain Technology and Smart Contracts to Create a Distributed Securities Depository

Eric Wall, eric.wall.770@gmail.com
Gustaf Malm, malm.gustaf@gmail.com

Department of Electrical and Information Technology
Lund University

Supervisors

Christopher Jämthagen, LTH
Stefan Wahlberg, Cinnober Financial Technology
Oskar Janson, Cinnober Financial Technology

June 29, 2016

Printed in Sweden
E-huset, Lund, 2016

Abstract

In the existing securities market structure, a securities trade between two parties requires the involvement of several financial intermediaries ensuring the safety of the transaction. However, the complexity of today's market structure in conjunction with the lack of interoperability between financial data infrastructures and the disalignment of business practices, are causing costs, risks and friction—resulting in settlement often taking several days.

Blockchain technology is the innovation powering the cryptocurrency Bitcoin, which is a network in which digital tokens can be traded peer-to-peer by the means of cryptography and decentralized consensus. The lack of intermediaries and short settlement period of cryptocurrencies make blockchain technology an inspiring database structure for the securities market. In this paper, we examine the potential of using blockchain technology to create a distributed securities depository. The decentralized consensus algorithm of blockchain technologies allows several entities to maintain a shared record of information without having to trust each other individually, since consensus is formed on a per-network basis. Such a technology could nurture the realignment of the securities market—or, reinvent it altogether. Furthermore, the possibility of leveraging consensus-oriented execution of computer code creates larger opportunities than that of a mere depository; it allows for the creation of new, trustless markets where securities and their contractual clauses are no longer merely legal obligations, rather, they are self-enforcing, autonomous programs.

Here, we propose the overarching design choices suitable for a second-generation blockchain platform for securities trading, devised to pursue interoperability within the larger context of the effervescently evolving distributed ledger ecosystem, while attempting to pay the necessary regard to the demands of regulatory compliance within the securities industry.

Acknowledgment

We would like to thank Satoshi Nakamoto, for allowing us to re-imagine the future of finance.

Eric Wall & Gustaf Malm

Table of Contents

Glossary

There is not yet a clear consensus on the exact definitions of several of the terms commonly used in the context of distributed ledger technologies. In this paper, we strive to conform to the usage that is most prevalent and logical in existing work within both the technological and the financial industry.

Decentralization is the process of dispersing power or control away from a central point.

A **distributed** resource is a resource that has been allocated to multiple parties.

A **ledger** is a computer file which stores records, e.g. documentation of transactions.

Consensus is the process of multiple parties coming to agreement on a piece of information.

Distributed ledger technology is any type of consensus-oriented distributed database that records information on a shared ledger.

A **blockchain** is a type of distributed ledger in which new appendages to the ledger are added in the form of blocks, where the blocks are hash chained to each other.

A **node** is a computer that runs a distributed ledger client software, which validates or rejects new incoming data.

A **miner** is a node that contributes to the creation of the distributed ledger by choosing which transactions to include in the next ledger update.

A **cryptocurrency** is a decentralized digital currency that is minted through cryptographic means and operates using distributed ledger technology.

The **Bitcoin protocol** is the protocol that defines the rules for the Bitcoin cryptocurrency.

The **Bitcoin network** is the network of nodes that runs the Bitcoin protocol.

A **bitcoin** is the unit of account and the native token of the Bitcoin blockchain.

A **second-generation blockchain** is a blockchain that is substantially different from the Bitcoin blockchain in terms of capabilities and complexity.

A **smart contract** is a self-enforced computer program that executes the terms of a contract.

An **oracle** is a third party which provides a smart contract with specific data from the outside world.

A **security** is a tradable financial instrument, most commonly a stock, bond or derivative.

A **central securities depository (CSD)** is an entity which maintains the definitive record of ownership of securities in a country or region.

Delivery versus payment (DvP) is the settlement of a transaction which delivers a security in exchange for payment.

Chapter 1
Introduction

1.1 Background

Present-day securities market infrastructures are unnecessarily complex, overly fragmented, subjected to settlement latency and lacking standardization. These systems were not designed with a clear concept in mind, rather, they've been developed and extended with new functionalities over the course of decades while retaining many legacy practices. Securities, while originally in paper form, have been dematerialized in many parts of the world to exist solely as book entries in central securities depositories (CSDs). While efforts have been made to improve the efficiency of these systems, such as the TARGET2-Securities initiative in the European Union, the securities post-trade landscape is still fraught with costly and tardy back-office procedures, sometimes even requiring manual intendance. Estimates of total annual costs for clearing, settlement and post-trade servicing are in the range of US$65-80 billion globally [1].

If the systems were to be rebuilt from the ground up today, it is very likely that completely different design choices would have been made. A technology that has been touted as the innovation carrying the potential of reshaping the financial industry is blockchain technology. Blockchain technology—a subclass of distributed ledger technology—was originally devised in a white paper published in 2008 by an unknown author under the pseudonym Satoshi Nakamoto as a means of enabling the cryptocurrency Bitcoin; a digital currency that can be transacted peer-to-peer without being processed through a trusted intermediary. Blockchain technology, which is a type of database technology which allows multiple entities of conflicting interests to collaborate on maintaining a shared ledger of records, has in recent years gained substantial attention from financial institutions and technology companies, and has even been cited by MIT as a technology as revolutionary as the Internet.

Meanwhile the securities market is a heavily regulated industry stewarded by oftentimes unforgiving policymakers, the technology does introduce novel ways of managing data and trust, which enables new areas of innovation and research. However, blockchain technology is still in its experimental stages and its merits and capabilities are still being investigated.

1.2 Thesis goals

Financial companies have in recent years begun showing great interest in blockchain technology as a means of reinventing themselves in the next chapter of the information age. However, due to the novelty of the technology, many aspects of the innovation has generally remained poorly understood by many policymakers, industry leaders and technologists. It is the purpose of this thesis to explore the possible role of blockchain innovations in the securities market and to contribute to the academic understanding of the technology and its different areas of application.

Specifically, we examine the potential of blockchain technology in re-engineering the securities market infrastructure and attempt to make well-measured and motivated design choices suitable within the scope of a regulated industry. A resulting consequence of blockchain technology principally being invented to circumvent authoritative control, the nature of the invention is such that its direct incorporation into traditional business practices would remove many of the technology's game-changing advantages. As such, this paper will aim to find the balance between preserving such advantages to its possible extent while proposing design approaches that remain within regulatory compliance.

Furthermore, this paper strives to pay the necessary considerations to the plausible roadmaps of industry adoption of blockchain technologies from a global perspective, in the context of properly addressing potential issues regarding interoperability and scalability.

Summarized, this paper aims to:

- Provide an analysis of blockchain technology at its current state of development
- Address the possibilities of leveraging blockchain innovations in the securities market infrastructure
- Address the issues of balancing decentralized blockchain database structures within the financial industry to conform to regulatory policies
- Outline possible routes of adoption of blockchain technologies in the securities industry while addressing the challenges ahead
- Propose overarching blockchain design choices suitable for the purpose of designing a securities depository

Additionally, this paper aims to describe designs for delivery versus payment in different blockchain paradigms.

1.3 Outline

Chapter 2: We explain how blockchain technologies work. We explain Bitcoin and then propose an abstract description of the anatomy of blockchains—their key characteristics and the different possible design choices.

Chapter 3: We introduce how securities can be represented as an asset class on a blockchain as well as describe how they can be traded in different blockchain paradigms.

Chapter 4: We propose blockchain design choices suitable for the securities market and outline the possible roadmaps for adoption of blockchain technologies in the financial industry.

Chapter 5: We summarize our proposal in chapter 4.

Chapter 6: We conclude the paper and deliver our final thoughts.

Chapter 7: We suggest areas for future work.

Chapter 2

Blockchain technology

2.1 Bitcoin: A Peer-to-Peer Electronic Cash System

Blockchain technology is inextricably linked to the digital currency Bitcoin, which was invented in 2008 with the publication of the white paper *Bitcoin: A Peer-to-Peer Electronic Cash System* by an unknown author under the pseudonym Satoshi Nakamoto. In the paper, Nakamoto combines several cryptographic components to outline a peer-to-peer payment system today known as cryptocurrency, which allows for the transfer of digital assets without the need of a trusted intermediary. It is the amalgamation of these cryptographic components which has since then become known as blockchain technology.

The Bitcoin network consists of a network of nodes that handles communications and verifies transactions, in which the nodes compete for profit by creating “valid” blocks through a process of packeting incoming transactions and solving a resource-intensive task, which will be explained further later in this segment. There are no accounts in Bitcoin, rather, users hold private keys required to sign transactions. The protocol uses public-key cryptography in which bitcoins are linked to public keys through "unspent transaction outputs" (UTXOs), meaning, previous transactions in which bitcoins were transferred to the user that has not yet been spent. To know a private key is in this sense analogous to owning bitcoins, where a user’s bitcoin balance is defined by the cumulative amount of bitcoins in UTXOs associated with their corresponding public key. In this scheme, bitcoins can informally be said to be sent to public keys, and as such, a Bitcoin address is equivalent to a Bitcoin public key.[1] Since Bitcoin users are identified only by their public keys, Bitcoin is in this sense a pseudonymous protocol.

A user sends bitcoins by creating a Bitcoin transaction and submitting it to the Bitcoin network. This is mainly done through Bitcoin wallets—a type of software application which holds a users private keys for signing purposes. A transaction contains one or more inputs and one or more outputs (Fig. 2.1). The inputs reference a selected set of UTXOs and the outputs contain the Bitcoin addresses the user wishes to transfer bitcoins to and the amounts the user wishes to transfer. A small fee is attached to the transaction to incentivize miners to include it in the

[1] To be technically correct, a Bitcoin address is a Base58Check-encoded RIPEMD160-hashed SHA256-hash of the public portion of an ECDSA key pair, however, this is not necessary information in order to understand the principles of the protocol [2].

blocks they create. A valid transaction must have a greater or equal amount of bitcoins in the input as in the output, with the difference constituting the fee. The user then signs the transaction with the private keys associated with the UTXOs. The transaction is then broadcasted to the Bitcoin network. In theory, the fees are voluntary, but in practice, fees are enforced by virtually every wallet software to ensure swift processing of transactions.

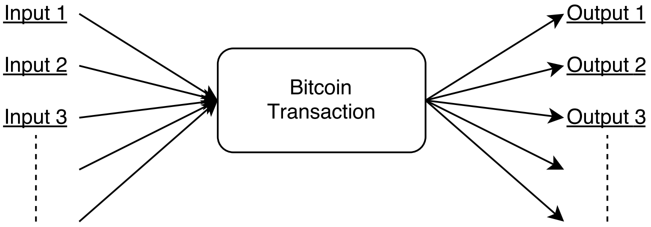


Figure 2.1: A single Bitcoin transaction, allowing for multiple inputs and outputs.

The transactions are sent over TCP using a simple broadcast network and are propagated to the memory pool of all nodes. The nodes creating blocks (called miners) receive the cumulative amount of fees from transactions as well as a *block reward* they can credit to their own wallets. The block reward is how new bitcoins are created in the system. In order to claim this reward, the miners must compete to solve a difficult hash computation of the block header information of the blocks they create. By adjusting a nonce value in the header, the miner can generate new hash values for a block. This hash value—a double-SHA256-hash—is what is known as the Proof-of-Work. Defined by the requirements on this hash value, these hashes can be arbitrarily resource-intensive to compute. The requirement for a proof to be valid is called *target* and is defined by a hexadecimal value the proof needs to meet.

```
target: 00000000ffff0000000000000000000000000000000000000000000000000000
hash:   000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
```

The SHA256 Proof-of-Work of the Genesis block and the corresponding target, mined 2009-01-03 18:15:05 UTC.

The hash of the block thus needs to be below the integer defined in the target. The Bitcoin network automatically adjusts this target depending on the hash rate of the network (the total hashes per second the network of miners can compute) to ensure that Bitcoin blocks are mined at a limited pace (every ∼10 minutes). The Proof-of-Work of Bitcoin accomplishes several tasks:

- It allows anyone with a processing unit to participate in the competition of creating new blocks
- It introduces bitcoins into the system at a steady pace

- It distributes bitcoins among participants using an arguably fair distribution mechanism
- It makes blocks tamper-resistant since any change to a block would require a new proof
- It allows for a consensus mechanism that is resistant to Sybil attacks[2]

Since the protocol allows anyone with a processing unit to participate in the creation of blocks, no single entity can censor certain transactions from being included into the blockchain—thus, Bitcoin is censorship-resistant. Additionally, since transaction processing is divided among multiple entities, the system is decentralized. Moreover, since processing units are a publicly available commodity and no restrictions on identity exist in the protocol, the Bitcoin network is an open network.

As a means of increasing the security to the record of transactions, the Proof-of-Work hash of the last mined block is always included as input in the next block, creating a hash chain that grows incrementally with every block, hence the name, "blockchain". This way, an attacker attempting to alter the contents of a block in the blockchain will not only need to recompute the Proof-of-Work of that block, rather, they would need to recompute all proofs for all the subsequent blocks in the chain, since all subsequent hashes depend on the previous. Since all the miners in the Bitcoin network are continuously completing new Proof-of-Works, utilizing their collective hash rate, such an attack becomes computationally infeasible—an attacker would need to control >50% of the hash rate of the network to maintain control over the blockchain.

Once a Bitcoin transaction is included from the memory pool[3] into a valid block, the transactions is said to have received one confirmation. For every subsequent block added to the chain, the transactions receives another confirmation. Thus, the recipient of a transaction can determine a transaction to be sufficiently irreversible once the cumulative Proof-of-Work—i.e. the number of confirmations—reaches a satisfactory level. Since blocks are ordered chronologically in which bitcoins are spent from UTXOs, bitcoins are protected from the double-spend problem—a failure case of digital cash schemes where a unit of account can be spent in several transactions.

However, a double-spend can exist temporarily if two miners solve the Proof-of-Work at the same time, each creating a block spending the same UTXO as the other. This causes a network split where miners will start building on separate blockchains. The event of a split in the Bitcoin network is an unavoidable effect of the CAP theorem, which states that distributed systems can fulfill no more than two out of three of the characteristics below [3]:

- Consistency—All nodes see the same state of the system at all times.

[2] In a system where consensus is reached through a type of voting process, a Sybil attack is an attack where an attacker exploits the pseudonymity of the network by assuming multiple identities in order to manipulate the voting result.

[3] The memory pool is the network memory containing the set of unconfirmed transactions that have been broadcasted to the network but have not yet been included in the blockchain.

- Availability—Every request to the system receives a response.
- Partition tolerance—The system remains operational even if some nodes fail.

For the Bitcoin network, sacrificing availability would mean that the system would need to go offline during synchronization periods (CP systems). Sacrificing partition tolerance would mean that Bitcoin would be vulnerable to node failures (CA systems). Since both these weaknesses are unacceptable to a cryptocurrency system, Bitcoin—like many other distributed systems—occasionally sacrifices consistency, but offers superior consistency compared to most other known AP systems [4].

In the Bitcoin protocol, the *longest chain* is the principle that suggests that the chain that is longest required the most effort to produce, and is consequently considered the preferred one to build upon. The probability that the divided miners on the separate chains continue to solve blocks simultaneously diminishes with each mined block, to a point where one chain eventually overtakes the other and one of the splits is dropped. Since a miner who chooses to work on a chain that is shorter has less likelihood to claim the rewards of solving a block in the chain that is going to be considered valid, this creates an incentive for miners to reach consensus and to work on a mutual version of the blockchain.

At its core, Bitcoin is both an invention in technology and economy. It works through means of game theory to incentivize users of conflicting interests to collaborate on maintaining a single version of a shared database of transactions. The Bitcoin network cannot effectively be shut down by any authority in the world, and has since its inception come to be accepted as currency in several jurisdictions [5]. Bitcoin has introduced and demonstrated how blockchain technology functions as a database structure which can fundamentally change the concept of trust in distributed systems. Bitcoin users do not need to trust transaction processors to act honestly by law—they simply rely on the expectation that the processing entities act in their own best self-interest.

2.2 Blockchain as a database structure

A blockchain can be thought of as a database structure. Although the technology was conceptualized with the advent of Bitcoin, it has since then been abstracted to refer to any type of distributed database technology that records data in continuously hash chained blocks. The true utility of the blockchain database structure is materialized through the use of a decentralized network. When the blockchain is distributed over a network of nodes, the nodes have the ability to verify the actions of the other nodes in the network, as well as the ability to create, authenticate and verify the new data to be recorded onto the blockchain. This networked model produces a system with the advantages of censorship and tamper resistance as well as having no single point of failure.

Blockchains have no single point of failure since the network does not rely on a central entity. The nodes run the same software and manage the same data—thus they are dispensable by design. For an ideal effect of resilience, the networking nodes should be spread across different jurisdictions and geographical locations. This way the blockchain becomes resilient to network outages caused by natural disasters, physical attacks and confiscations from authorities. This relates to the benefit of censorship resistance. We know from history that governments and corporations do not always act in a way that is in the public's best interest or in a way that is always ethically just. This creates the need of database technologies that can record data that cannot be removed, modified or controlled by a centralized authority.

Blockchain are typically “append-only” database structures. The principle is that the data that is recorded onto the blockchain can't be manipulated afterwards—in other words, the intention is *immutable* data storage. The hash chain is what differs blockchain technologies from other distributed ledger technologies while the attribute of being consensus-oriented is what unites them. Indeed, it is the hashing that "chains" the blocks together, and without this characteristic a blockchain would simply be a ledger. When a node verifies that the referenced previous hash[4] in the header of a newly created block (created by another node) is the same value as the value the verifying node itself recognizes to belong to the valid chain, it also verifies that both nodes agree on the entire history of the blockchain.

Another advantage of using blockchain technologies compared to other types of database structures becomes apparent when managing tradable assets. Blockchains are designed to be used with a UTXO model as originally designed in the Bitcoin protocol, which means that every new transaction input requires a referral to the output of another transaction. That means that the asset referenced in a transaction is traceable through the blockchain up until its inception, which can be of considerable importance in industries where transparency as well as auditability and traceability are desirable features.

It is important to note that while transparency in a distributed ledger network is a natural consequence of the bilateral verification processes of pseudonymous nodes, it is not impossible for distributed ledgers to achieve confidentiality of transaction information even in open networks. Indeed, even though every transaction must be verified by every node in the network, there are new technologies being built which aim to address this issue. One such example is MIT's Enigma, which is a decentralized computation platform intended to act a supporting privacy-engine to a blockchain [6]. The platform distributes computations in a scheme of secret-sharing which preserves the confidentiality of information. Another example is techniques such as *confidential transactions* for the Bitcoin protocol which through the means of homomorphic encryption ensures that the contents of transactions not necessarily need be disclosed in the verification processes in order for the system to function [7].

The applicability of such extensions indicates a level of sophistication in the blockchain design pattern in the sense that they can better tap the potential of powerful computer network-supported database systems when compared to existing relational database management systems in terms of cryptographic security and general dynamicity. This dynamicity is being further explored in the development

[4] In the Bitcoin protocol, the hash of the previous block is referenced by a 256-bit hash in the `hashPrevBlock`-field in the block header.

of second-generation blockchain technologies (e.g. smart contract capabilities, see section 3.4).

As mentioned previously in this paper, the most significant innovation of distributed ledger technologies is that they allow several entities of unaligned interests to maintain a unified version of the same ledger through the use of consensus mechanisms. In Bitcoin, this means solving the double-spend problem of digital currencies. In distributed database technology, consensus mechanisms constitutes the solution to the multi-master replication problem. Principally, all consensus mechanisms in permissionless systems aim solve the epistemological Byzantine Generals problem—an agreement problem in anonymous consensus [8][9].

2.3 Scalability

In blockchain networks, every (valid) transaction is propagated through the entire network. They are only included into a block after the consensus process, after which the new block is propagated through the network, until every node has updated their version of the blockchain. The larger the network becomes, the more nodes a block has to propagate through. A blockchain network is a type of network which does not become faster the more processing nodes that are added to it, since every node has to process every transaction.

Furthermore, while traditional databases keep an updated record of all the accounts and balances in the system, blockchains keep an updated record of all the transactions that ever occurred in the history of the network. Consequently, although by design, blockchain database structures scale worse than centralized databases and traditional distributed database systems.

The load a blockchain network puts on a node can be divided into four categories:

- CPU load—the required processing speed a node needs in order to process a certain amount of transactions per second
- Memory usage—the required amount of RAM a node needs to process a certain amount of transactions per second
- Network bandwidth—the required bandwidth a node must have access to in order to allow propagation of data at a certain bit rate
- Storage capacity–the required disk space a node needs to store the blockchain data

The hardware requirements act as a barrier to entry to a blockchain network. If a blockchain network aims to facilitate more transactions per second, the minimum requirement on each node will increase. Thus, there is a trade-off between scale and decentralization [10]. This has given reason for alternative types of Bitcoin nodes to exist. The nodes that have been described so far are referred to as full nodes, as they are nodes which perform the full array of functions, e.g. storing the entire blockchain and verifying all blocks and transactions. Contrariwise, Simple Payment Verification (SPV) nodes are a lightweight type of node which do not

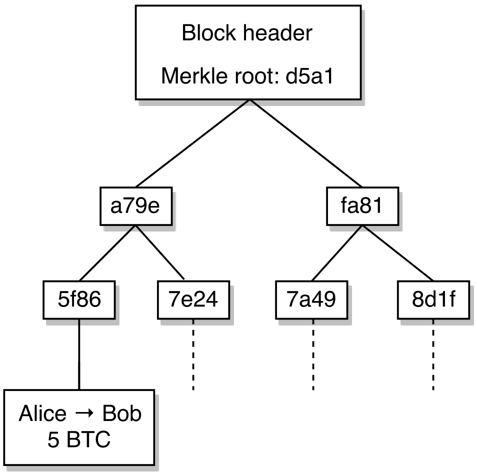


Figure 2.2: A depiction of a Merkle tree, in which the hashes of transactions are recursively paired up and rehashed to create a Merkle root. The Merkle root for the transactions contained in each block can be found in the block header.

store the entire blockchain, rather, they utilize Merkle trees to efficiently confirm a transaction's existence in a block.

Firstly, in order to create a Merkle tree, all transactions in a block are hashed and used to form a row. Secondly, these transaction hashes are paired up and hashed again, forming a new row with half of the hashes of the initial one. This step is repeated until a row which only contains a single hash is reached. This single hash is called the Merkle root and it is saved in its respective block header to be utilized by SPV nodes (Fig. 2.2) [11][12].

Since an SPV node does not store the entire blockchain, it is reliant upon full nodes to relay information regarding transactions. However, while it does not store the entire blockchain, it does store the block header of each block. The confirmation of the existence of a transaction in a block is done by reproducing the calculation of the Merkle root, using the hash of the transaction and by requesting the missing hashes in the Merkle tree from a full node. By using Merkle trees, SPV nodes can validate the existence of transactions without being subjected to the load of operating a full node.

Another lightweight alternative are so called *pruned* full nodes. Similarly to SPV nodes, a pruned node does not store the entire blockchain, rather, it only stores the last few blocks[5] together with a set containing all unspent transactions (UTXOs). These type of nodes still download and verify all transactions, however, pruning does allow the disk space usage to be reduced. For instance, in Bitcoin, a full node requires ~70 GB disk space, compared to the more modest 2-3 GB required by a pruned node [13]. As mentioned, an SPV node, as opposed to a pruned full node, has not downloaded and verified all transactions in the blockchain, which

[5] A minimum of 288 blocks are maintained in local storage (~2 days worth).

means it has to fetch any transactions it is interested in. This introduces privacy issues for SPV nodes because requesting information regarding transactions might reveal the SPV node's intentions.

New optimizations to Bitcoin are constantly underway. One such development is the Lightning Network, a protocol under development which allows a large portion of transactions of a blockchain to take place off-chain using multi-hop payment channels. A payment channel is a method for two parties to keep a balance between themselves by exchanging valid signed transactions. No party in the transaction is subjected to any risk if the channel is broken since they can always settle the balance of the payment channel to the blockchain via the last signed transaction. This action could be viewed as the blockchain equivalent to netting trades. Since only the final balance of a payment channel will need to be processed by the network, the Lightning Network could allow blockchain networks to process many more transactions per second than what currently possible [14].

2.4 Consensus models

2.4.1 Proof-of-Work discussion

Proof-of-Work is a type of Byzantine fault tolerant consensus model. Byzantine fault tolerant systems are systems impervious to Byzantine failures—a type of failure in which a faulty node is not restricted to simplistic faulty behaviors (e.g. network loss, in which case a node either responds correctly or becomes non-responsive). Rather, Byzantine fault tolerant systems are (to some extent) tolerant to failures which include any type of malicious node behavior [9].

The longest chain containing the most Proof-of-Work is used as one of the principles nodes follow to reach consensus. Since Proof-of-Work is achieved through computational efforts, it has a measurable cost associated to it. As the miners working to solve the Proof-of-Work are economically incentivized by the value of the block reward and the transaction fees, their profit is derived from their hash rate versus the cost of their equipment and electricity. This has resulted in competition among miners to develop more cost-effective hardware, moving from CPU to GPU to FPGA to ASIC mining.

The total hash rate of the Bitcoin network is at the time of writing 1,350,692,770 GH/s [15]. If we assume only the most efficient mining hardware on the market is used—the Antminer S7—which has a power efficiency of 0.25 J/GH, the Bitcoin network consumes approximately 2.96 TWh a year.[6] This is roughly the same power consumption as of the island of Jamaica ($\approx$ 3.01 TWh/year) which has a population of 2.95 million [16]. It should be noted that the S7 miner is a recent addition to the market which is considerably more efficient than its older counterparts. It is therefore not unlikely that the actual energy consumption of the Bitcoin network is significantly larger.

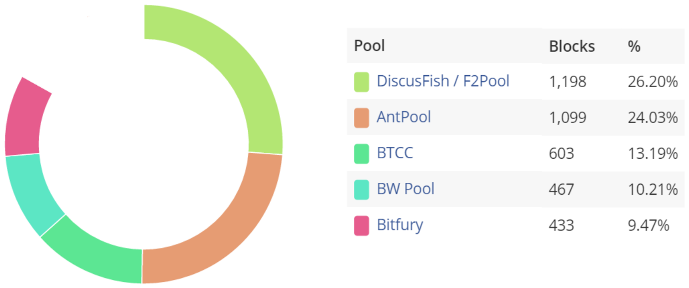
In light of these costs, opponents argue that Proof-of-Work is an excessively wasteful and environmentally hostile consensus mechanism. In reaction to such

[6] 1,350,692,770 GH/s × 0.25 J/GH = 337673192.5 J/s = 337673192.5 W
337673192.5 W × 24 hours × 365 days $\approx 2.96 \times 10^{12}$ Wh/year = 2.96 TWh/year

stances, there have been proposals in the cryptocurrency community which attempts to use the computational power of Proof-of-Work to complete tasks that has some tangible value to society, such as finding prime numbers [17]. However, it follows the logic of economic theory that if the Proof-of-Work produced a byproduct of tangible value, the revenue from such a byproduct would be deducted from the cost of the mining operation in the miners revenue model. Thus, by increasing the marginal revenue of miners, it would consequently allow them to spend more on electricity costs until the profits are once again at equilibrium [18].

The cost of Proof-of-Work is what secures the hash chain of the Bitcoin network. From adding the average amounts of hashes per block up until the current block height of 416000, we can estimate that the Bitcoin network has calculated roughly $2^{84.8}$ hashes during its existence—an achievement equivalent to breaking the security of 80-bit cryptographic primitives. Indeed, it is by design that the cost of the proof securing the network constitutes the cost for an attacker to break that security. Seemingly, Proof-of-Work is a good consensus model for decentralized networks with many participants; they split the costs of establishing the security which makes it very expensive for a single attacker to break it. From the example above, an attacker would need to amass a mining operation with the power consumption of Jamaica only to get an even chance of creating the next Bitcoin block—a block that would still be rejected by the other nodes in the network if the it includes malicious transactions attempting to steal coins (e.g. a double-spend). An alternative attack that circumvents the costs of purchasing the necessary hardware would be to subvert existing hash rate. This would simultaneously reduce the requirement since it inadvertently reduces the remaining honest hash rate.

Since mining is highly competitive, the probability is extremely low for a miner running a modest mining operation of solving the Proof-of-Work before the rest of the network. Therefore, most miners join mining pools where they get paid by the pool administrators proportionally to the added hash rate they contribute. Since a large pool has a higher probability of solving the Proof-of-Work by distributing the computation, the pool can win the block reward more regularly and thus distribute earnings at a steady pace to individual miners. The success of mining pools has caused the regrettable effect of centralization of the Bitcoin network, as 83.1% of the total hash rate is comprised of only five mining pools (Fig. 2.3).



Pool	Blocks	%
🟩 DiscusFish / F2Pool	1,198	26.20%
🟧 AntPool	1,099	24.03%
🟩 BTCC	603	13.19%
🟦 BW Pool	467	10.21%
🟥 Bitfury	433	9.47%

Figure 2.3: Mining centralization. The five largest mining pools control 83.1% of the hash rate of the Bitcoin network. *Source: Blocktrail (time frame 2016-05-04 → 2016-06-04).*

Mining pool network effect could prove to be problematic to the decentralized aspirations of Bitcoin, since it is the pool administrator who chooses which restrictive rules of the protocol to follow and which transactions to include into the blockchain. While hashers still have the ability to change pools if they don't support the parameters chosen by the pool administrator, such endeavors are often restrained by inertia. Thus, opponents argue that the censorship resistance of the Bitcoin network is threatened by mining centralization.

The debate concerning the merits of Proof-of-Work has given cause to development of a diversity of other consensus models, each with different characteristics and varying performance depending on the type of system it is applied to. It is important to note that the Proof-of-Work consensus model was designed to suit the needs specific to Bitcoin. While this consensus model has notable advantages in cryptocurrency applications, it does not necessarily encompass the needs of all blockchain technology systems.

2.4.2 Proof-of-Stake

In blockchain technologies, the principal rule of decentralizing control is for the network to distribute the right to create a block fairly among the nodes. Proof-of-Work achieves this because it randomly selects a miner with a probability proportional to the miner's share of processing power through the laws of mathematics. Proof-of-Stake is a consensus model that aims to provide the same type of fairness in the distribution of consensus without requiring miners to burn external resources. In this scheme, miners compete with their amount of *stake* in the network rather than with processing power.

The stake in this example typically refers to the possession of native tokens in the system (e.g. cryptocurrency). The proposed way of measuring stake involves cryptographically proving ownership of tokens, while some implementations also involve the escrowing of tokens (locking tokens to the ledger) for a period of time. Proof-of-Stake is in this sense used to prove commitment and exposure to the network.

In variations where the stake is escrowed, creating a block containing malicious transactions which is consequently rejected by the network results in the forfeiture of the escrowed stake. Following the same principles of Proof-of-Work-based consensus, for variations where the stake is not immediately forfeited, attackers still need to amass more than half the stake of the network in order to be able to maintain the longest chain in order to successfully manipulate the blockchain. This way, Proof-of-Stake protocols are also protected by a cost of attack similarly to Proof-of-Work.

Seemingly, through this scheme, a blockchain network can reach consensus without the costly energy expenditure associated with Proof-of-Work. More so, this design avoids the confirmation time latency we've come to know in Proof-of-Work-based systems, since miners need not perform any cumbersome mathematical calculations to provide the proof [19]. However, since Proof-of-Stake incentivizes miners to hold their tokens to compete for the block reward, opponents argue that Proof-of-Stake still introduces a cost, although obscured by complexity. The arguments concern market liquidity and the fact that miners take on both risk and opportunity costs in locking down capital, which—while more difficult to measure—still constitutes a cost.

Furthermore, economic theory infers that miners will keep locking more capital as long as it is profitable to do so. If the reward is the same, any opaque costs of Proof-of-Stake or any other type of consensus model will always match the costs of Proof-of-Work eventually in accordance with the of profit-incentives and revenue models of miners [18].

Consensus models in cryptocurrency applications comprise an intersection of computer science and game theory which is not immediately understood. While recent work argues that Proof-of-Stake may be an unworkable consensus algorithm if applied alone since there is no associated external cost to forking the network, the subject remains one of the most controversial topics in the cryptocurrency community [20][21]. Current implementations have subsequently leaned towards combining Proof-of-Stake and Proof-of-Work, constituting a new consensus model called Proof-of-Activity [22].

2.4.3 Consensus ledgers

Consensus ledgers are a type of distributed ledger technology that does not store transaction histories, rather, a network of nodes reach consensus on only the current state of the network. Ripple is a payment protocol developed by the company with the same name and is the most prominent consensus ledger, which uses its own cryptocurrency XRP as native token. This protocol, in contrast to blockchains, does maintain the notion of accounts rather than the UTXO model of Bitcoin.

Similarly to the previous consensus models discussed which are strictly block-oriented, the Ripple ledger which holds the information of all Ripple accounts is updated through a network consensus on the next batch of transactions. However, in contrast to cryptocurrency networks, the Ripple network is a network where each node itself decides which nodes it want to use to derive consensus from on the current ledger. This set of trusted nodes is called a "Unique Node List" (UNL) and Ripple uses this set of nodes in a voting process where the trusted nodes vote over multiple rounds to decide on which transactions to include in the last closure of the ledger [23]. This set of transactions might differ between nodes so in order to validate that all nodes have the same resulting ledger, a signed hash of the set will be broadcasted, and a supermajority will need to be formed by all nodes in the network [24]. Each round of voting works as follows:

1. Each node in the network collects all valid transactions that has not yet been recorded in the ledger and compiles them into a "candidate set". This candidate set is then propagated through the network.

2. Since each node only considers the opinion of the nodes on its UNL it disregards any candidate set it receives from nodes not on the list.

3. One or several rounds of voting follows from the nodes in the UNL in order to surpass a certain agreement threshold on the transactions in the candidate sets. The current threshold is set at 80%, at which point all transactions meeting this requirement are included in the ledger and the rest are added to the candidate set of the next round.

4. In order to guarantee that all nodes derive the same ledger a signed hash of the ledger is relayed from each node in the network and it will not consider the last round closed until a supermajority is reached.

While only a predefined set of nodes is used in the process of generating each ledger update, it is not the same set used for the entire network, rather, each individual node decides whom it wants to trust. As long as the network is sufficiently interconnected it is very likely that a supermajority will be reached. Occurrences of consensus failures in sufficiently interconnected consensus ledgers is most likely caused by latency issues and/or transaction inconsistency, and in such instances the consensus process is simply restarted [23].

As previously mentioned, the Ripple consensus algorithm does not use a blockchain in order to record all previous transactions. Instead, it utilizes a more classic database structure which only records the most recent transactions and current accounts and balances. While this does provide for more efficient storage it also means that transactions are not traceable back to their inception.

It is important to note that consensus in distributed databases are not the invention of distributed ledger technologies. Indeed, the concepts have existed for decades through protocols such as Raft and Paxos [25][26]. While building on the same principles, a comprehensive comparison between traditional consensus protocols and modern distributed ledger consensus algorithms has yet to materialize.

2.4.4 Threshold signature scheme

Another way a distributed ledger system can reach consensus on new data to be added to its ledger is through the means of a threshold signature scheme. In a threshold signature scheme, a network of nodes reach Byzantine agreement[7] through the use of a multi-signature scheme. Consensus on an update is considered to be reached if it is signed by a multi-signature meeting the signature threshold, meaning k signatures out of l signing nodes can generate a valid signature where k is a subset of l. Byzantine agreement that is tolerant to a collusion comprising 50% of the network can thus be reached through the trivial requirement that $k > \frac{l}{2}$. The safety of such a system is dependent on the likelihood that the subset k nodes are not corrupted [27][28].

A blockchain using a threshold signature scheme to verify blocks can thus reach consensus on a chain by constructing a valid multi-signature (consisting of k signatures) on the latest block while using the same hash chain linkage as previously discussed. The parameter k can be modified to suit the needs of a particular system, of which the upper limit can vary with how prone that particular system is to network failures and latency issues, while the lower limit can vary with the desired tolerance to Byzantine failures. However, a system where $k = l$ gives the ability of censorship to individual nodes, since a node can thwart consensus simply by refusing to sign a block. Thus, in order to disable censorship resistance from individual nodes as well as relaxing the requirements against network failures,

[7] As previously explained in the context of Byzantine fault tolerance, in order for an agreement to be resistant to Byzantine failures, it requires the agreement to be able to tolerate the collusion of adversarial participants to a certain extent.

$k < l$ should at least be fulfilled. As a general rule, we can assume that the two-out-of-three threshold applied by many consensus system can be used a starting point when selecting the value of this parameter [29].

In this scheme, nodes would take turns creating the next block in a round-robin fashion within a predetermined time frame to allow for the communications creating the multi-signature to take place. These rounds are the threshold signature scheme analogy to Proof-of-Work confirmation times, and can be performed near-instantly (milliseconds). However, this scheme assumes that the nodes in the network are supported by a public key infrastructure with known identities in order to prevent Sybil attacks. While considerably more efficient than the other consensus models discussed in this paper and thus suitable for certain applications, it reintroduces the requirement of trust back into blockchain technology.

An example of a signature algorithm which meets the requirements of this scheme is the Schnorr signature algortihm Ed25519 [30][31].

As previously mentioned in the example of Proof-of-Activity, consensus models can be combined. Similarly, threshold signature schemes can be combined with Proof-of-Work to enable multi-tiered security. This can be done either by the signing nodes utilizing their processing power in parallel with the signing routine to generate Proof-of-Work—or, more interestingly—by "piggybacking" on other Proof-of-Work networks through *timestamping*.

Since the Bitcoin network is open to anyone, one can inject arbitrary data into the Bitcoin blockchain by transmitting a transaction carrying user-specified information. For instance, this could be used to include the last hash of a threshold signature scheme block header into a Bitcoin block at regular intervals. If an attacker manages to manipulate the transaction history of the threshold signature scheme blockchain, the timestamped Bitcoin blockchain record could be used to disprove the altered version of the chain. While not adding perfect security, it does prevent the attacker from successfully altering the history of the blockchain prior to the attack's success. Although this adds the cost of Bitcoin transaction fees to the scheme, this may be considered a small price in comparison to the advantages of securing the blocks of another blockchain with the hash rate of Bitcoin network. Blockchain timestamping has become a practice employed by different distributed ledger technologies in the industry such as Factom and tØ [32][33].

2.5 Permissioned and permissionless blockchains

Permissionless blockhains such as the Bitcoin blockchain have also been referred to as *public* blockchains, in the sense that the networks are open to the general public to join as users or serve in as nodes, but also in the sense that the blockchain data is publicly transparent. Development efforts by companies, banks and other financial institutions to leverage this database structure in examples in which the data is not transparent have thus contrariwise been referred to as *private* blockchains, and the merits of such efforts have been the subject of much skepticism. Some opponents of private blockchains argue that private blockchains do not offer any real benefits over traditional database solutions as they too are subjected to centralized control. However, there are nuances to permissioned blockchains in the sense that they need not be entirely decentralized nor centralized. Indeed, it has been argued that while decidedly less decentralized than permissionless blockchains such as Bitcoin, permissioned blockchains such as ones consisting of a consortium of sorts, e.g. a collaboration of financial institutions, are still considerably more decentralized than master-access databases [34].

Another nuance to the permissioned model concept, while more overlooked in public discussions, is the relatively broad spectrum on which permissions can be configured. Although there is yet no official taxonomy defined for many of the aspects of distributed ledger technologies, we will for the purpose of properly addressing this nuance hereafter use the following classification of blockchain permissions, as proposed by the BitFury Group [35]:

1. A *public* blockchain is a blockchain, in which there are no restrictions on reading blockchain data and submitting transactions for inclusion into the blockchain

2. A *private* blockchain is a blockchain, in which direct access to blockchain data and submitting transactions is limited to a predefined list of entities

3. A *permissionless* blockchain is a blockchain, in which there are no restrictions on identities of transaction processors

4. A *permissioned* blockchain is a blockchain, in which transaction processing is performed by a predefined list of subjects with known identities

While the terms *private* and *permissioned* blockchains have been used interchangeably, it is from these definitions apparent that a permissioned blockchain does not necessarily need to be private. Indeed, a permissioned blockchain could be either private or public—however, this too can be seen as a non-binary distinction. While the acts of reading blockchain data and submitting transactions are both restricted to a predefined list in private blockchain implementations, that list could be restricted to contain only institutions—or, in a more permissive design—regular citizens who gain access through the means of e.g. an electronic citizen identification system.

In the context of permissioned blockchains, certain design choices that were made for the Bitcoin protocol become nonessential. The Proof-of-Work consensus model is well suited for cryptocurrency protocols because it among other things solves the problem of Sybil attacks. Thus, in a permissioned design which is supported by known identities, a Sybil attack-resistant consensus model becomes superfluous. Instead, permissioned designs are considerably better favored by threshold signature schemes, which are not disadvantaged by the costs associated with pseudonymous consensus. In contrast to permissionless networks, the nodes in permissioned designs can be incentivized to act honestly through other means than economic rewards—namely, they are exposed to the risk of legal prosecution if engaged in fraudulent activity.

With regard to Proof-of-Work discussed earlier, we can determine that the consensus process in a decentralized permissionless blockchain is relatively costly and slow. While Proof-of-Stake-based systems are ridded of the resource-intensive computation process, all permissionless decentralized consensus-oriented networks are still impeded by the underlying characteristic that verification processes need to be performed by nodes of varying bandwidth and latency from all over the world.[8] Permissioned blockchains using threshold signature schemes, in which the nodes can be restricted to a predefined list that includes only high-bandwidth low-latency participants, and additionally, do not need to prove either stake, work or other quantifiable efforts, can therefore scale superiorly to permissionless blockchains in terms of both efficiency and throughput. As a comparison, while the 7 transactions per second maximum limit of the Bitcoin network is by no means to be regarded as an upper limit to the throughput of permissionless blockchains in general, permissioned implementations such as the Domus Tower boast throughputs in the range of a million transactions per second [37][38]. While higher throughput permissionless blockchains do exist, none can effectively compare to permissioned implementations.

When moving from a permissionless blockchain to a permissioned one it is important to remember that not all the praised characteristics that have come to be associated with blockchains are preserved. A cursory summary of the key advantages of blockchains compared to traditional database technologies can be seen below:

- Decentralization
- Transparency
- Immutability
- Censorship resistance

We will now analyze each of these characteristics in terms of the extent to which they materialize in permissioned implementations.

Decentralization: The context in which a blockchain solution is a beneficial database structure is that in which multiple parties of conflicting interests stand to gain something by sharing a mutual ledger of information. If these parties are part of the same organization and controlled by the same authority, as is often the case within many private companies and institutions, the purpose of decentralizing the control of the ledger perishes. In order for the benefits of a permissioned blockchain over a traditional database solution to remain non-negligible, it is imperative that the nodes are sufficiently sovereign from each other. Even so, restricting nodes to a predefined list will always have a centralizing effect on the network.

Transparency: One of the key features of a blockchain database structure is the transparent record of ownership of assets. Contrariwise, in financial markets, confidentiality of transactions is oftentimes of paramount importance, e.g. banks wishing to shield information from competitors. While this makes the notion seemingly ill-fitted, it is at the time of writing mainly banks which most strongly consider using permissioned blockchains to improve their infrastructures. However, there

[8] Ethereum, a permissionless second-generation blockchain platform, has introduced a *sharding* technique. While still merely conceptual, it involves sub-dividing the system state so that not all nodes need to process every transaction, which may mitigate this issue [36].

are important distinctions to the concept of transparency; while certain transactions in the financial system require confidentiality, there are situations where transparency can be very beneficial, such as in the case of tracking government spending or foreign aid.

It should also be noted that transparency in a permissioned model does not necessarily entail transparency to the general public, but can also imply facilitating auditability to certain permissioned entities, e.g. regulators or auditors. Such a feature could be useful in order for regulators to be able to assess the trustworthiness of a financial system, e.g. to avoid situations where financial instruments become exceedingly complex as in the case of collateralized debt obligations in the financial crisis of 2008. In essence, permissioned blockchains do not necessarily thwart the transparency characteristic, rather, they provide a more configurable approach in assigning different levels of read access among different entities.

Immutability: Blockchains are considered to be effective in minimizing counterparty risk owing to the very high settlement consistency in blockchain transactions. However, immutability is not an inherent characteristic of blockchains—it is a characteristic of Proof-of-Work. If all the keys of a threshold signature scheme was compromised, an attacker could rewrite an entire blockchain history within an instant (hence timestamping is recommended, see section 2.4.4). Contrastingly, if the total hash rate of the Bitcoin network was compromised, an attacker would need to power it for ~10 minutes to rewrite just the latest block. In a general sense, settlement can only be considered final in blockchain transactions because of the resources it would cost an attacker to alter the state of the system, and there is no inherent cost for a permissioned blockchain using a threshold signature scheme to alter previous transactions. This does not mean that permissioned blockchain transactions cannot be immutable, it only means that the degree of immutability in permissioned blockchains will be supported by the trust in the imperviousness of the permissioned entities rather than cost of external resources.

Censorship resistance: One of the main features of Bitcoin is censorship resistance. When Visa and MasterCard blocked donations to WikiLeaks in 2010, the legality and ethical justification of the act was disputed by various members of the public [39]. More importantly, it demonstrated that two corporations and the government of the U.S. were able to stymie the cash flow of an international organization at their sole discretion without a clear legal mandate to do so. Censorship of financial transactions exerted by an authority is only possible when few entities are in control the system and can deliberately choose which data to include and which to ignore. In contrast, no corporation or government can stop Bitcoin transactions from reaching WikiLeaks. This is why many proponents of cryptocurrencies argue against the use of permissioned blockchains in the financial industry, because ultimately, a predefined list of entities would still be in control of the financial system, which exposes it to corruption.

The arguably most powerful characteristic of the blockchain database structure is that it removes the need of trusted third parties to truthfully maintain a record of information. Using the Bitcoin blockchain, users don't need to trust banks to maintain honest records of their bank accounts, they only need to trust that the nodes are acting in their own self-interest and that the network is sufficiently difficult to attack. Comparatively, in consortium-based permissioned blockchains, users need to trust not one, but a collective of financial institutions to not collude, and that the network is sufficiently difficult to attack.

2.6 Sidechains

Sidechains are the concept of parallel blockchains that allow assets from one blockchain to be transferred into another. The concept was formally announced in 2014 by Blockstream, which is a private company consisting of many of the Bitcoin Core development team members. While the concept originated from Bitcoin, the broader theory behind sidechains is applicable to any blockchain design. The intention of sidechains was to enable the transfer of bitcoins to other blockchains, thus enabling sidechains to act as alternative cryptocurrency systems without requiring the minting of new coins. However, the initial permissionless sidechain design was discovered to contain non-trivial security flaws. Meanwhile, newer designs are currently under development, but have at the time of writing not reached maturity.

Nevertheless, transferring assets from one blockchain to a permissioned blockchain is in fact already workable by the sidechains design. The idea is fairly simple—by introducing a function called a two-way peg, assets will be allowed to be transferred from one blockchain to another and back [40]. One of the key principles of the two-way peg is that it is impossible to return more assets to the "parent chain" than what originated from it, thus, the total number of assets in the parent chain can not be compromised by the implementation of the peg. Thus, any new rules can be implemented in the sidechain without posing a risk to the parent chain.

Let's take a closer look at the two-way peg; assets are locked on the parent chain through the means of multi-party escrow—meaning that they are sent to an address which requires a multi-signature to unlock. This multi-signature is formed by the permissioned entities on the sidechain. When the assets on the parent chain are in escrow, the sidechain can allow the creation of these assets on its own chain. In order to reintroduce the assets from the sidechain on the parent chain, the owners of the assets on the sidechain must prove that they have destroyed the coins on the sidechain by sending them to an unspendable address. When this proof is provided, the permissioned entities on the parent chain release the same amount of assets from the parent chain escrow.

Sidechains can extend the functionality of a parent blockchain by introducing new features on the sidechain. For example, since a permissioned sidechain can leverage a threshold signature scheme consensus model, this allows for near-instant confirmation times of an originally permissionless token such as the bitcoin. One example of such a sidechain is the sidechain Liquid, maintained by Blockstream for various Bitcoin exchanges [41]. Other examples of features sidechains can potentially bring to permissionless networks are things such as supporting multiple asset types from different blockchains to exist on a mutual sidechain, smart contracts and prediction markets (see section 3.4).

2.7 Colored coins

Colored coins is a concept that leverages an existing cryptocurrency system, e.g. the Bitcoin network, to track ownership of another asset. The basic idea is that by assigning—or, "coloring"—a specific unit of bitcoin on the Bitcoin blockchain to represent an asset, users can generate their own key pairs and trade these assets with the support of the Bitcoin infrastructure. The colored coin meta-layer protocol tracks the ownership of the colored coins and embeds colored coins transaction into the Bitcoin network, which in this sense functions as an engine performing secure transactions and settlement. The tie between the colored coin representation of ownership and the ownership of the actual asset needs to be backed either by the issuing agent or by a public agreement [42].

Piggybacking on the infrastructure of an open blockchain like the Bitcoin blockchain to track the ownership of another asset naturally imposes both the beneficial and the limiting properties of that blockchain onto the piggybacking system. The main benefits of using Bitcoin as the underlying engine for colored coins is that the meta-layer derives its security from the massive computational power of the Bitcoin network. This allows for relatively easy deployment of blockchain-backed assets without the need of developing a new blockchain client, minting a new token or attracting miners to secure the chain. One of the main drawbacks of this method is that the throughput of the meta-layer is confined to the throughput of the underlying layer. Colored coin approaches to tracking the ownership of an asset using the Bitcoin network as the underlying engine is thus confined to the 7 transactions per second limit as well as the ~10 minute block confirmation times.

One implementation of the colored coins concept is the open-source protocol Open Assets Protocol. On May 11, 2015, Nasdaq announced that they were using the Open Assets Protocol for their private market platform Linq [43][44]. Using this platform, companies can issue and manage private securities while the Bitcoin network facilitates the transfers of ownership as well as auditability of the transaction history, thus enabling near real-time settlement and round-the-clock uptime. Since the securities are completely dematerialized, in the sense that the cryptographic proof of ownership of the colored coin representation on the Bitcoin blockchain is the only representation of the ownership of the security anywhere, users of the platform are assured that their security is not double-spent through a different channel.

The colored coins approach borrows the decentralized control from the underlying network. Indeed, in contrast to the case if Nasdaq deployed their own private blockchain, Nasdaq cannot in this case manipulate the records of ownership on their own volition. However, the tie between the representation of the ownership of the security and the actual rights associated with ownership of that asset is still only upheld by Nasdaq. Meanwhile the colored coins approach allowed Nasdaq with relatively low effort to deploy a blockchain for internal use, it did thus not provide trustless security to end customers (section 3.3 covers this subject more thoroughly).

2.8 Scripting

Transactions in Bitcoin are not defined in the traditional sense as "transfer X assets from address A to address B". Instead, each transaction output is defined by a script. This is a powerful approach because it allows for both innovation and complexity when defining how transactions can be spent. As previously mentioned, this transaction design pattern is called an UTXO model, which means that inputs into a transaction will refer to previous outputs and that there are no Bitcoin accounts, just unspent outputs associated with certain public keys. The most common Bitcoin transaction output type is called `Pay-to-PubkeyHash` and is defined by the following script [45]:

```
<sig> <pubKey> OP_DUP OP_HASH160 OP_DATA_20 OP_EQUALVERIFY OP_CHECKSIG
```

In order to spend this output the spender will have to provide a public key `<pubKey>`, which when hashed should match the data included in the `OP_DATA_20` operation code (opcode). In addition, the spender will need to provide a signature made by the corresponding private key `<sig>`, with the signed data being a portion of the new transaction, most notably the inputs and outputs. As mentioned, `OP_DATA_20` as well as the other opcodes in these scripts are used to facilitate the stack-oriented programming features which all nodes will be required to verify as part of verifying transactions. Furthermore, a Bitcoin transaction allows for several inputs and outputs in a single transaction, which might be necessary if you only have access to smaller UTXOs but want to create a single larger transaction, or alternatively, if you only have access to a larger UTXO and want to pay a smaller amount and return the change to yourself via an additional output.

One of the more interesting alternatives to this basic script is the multi-signature (multi-sig) script. By using a multi-sig script we create a scenario where multiple key pairs are required to spend the transaction output. However, all keys that can be used to unlock the output does not necessarily have to be included, but instead only a certain threshold needs to be reached. An example of such a scenario would be an output requiring at least two out of three keys in order to be spent, with the opcode used to accomplished this being `OP_CHECKMULTISIG` [46].

However, there are some intentional limitations to the Bitcoin scripting language; it does not allow access to any data outside the block it is contained within, and data access inside the block is very restricted and generally limited to the actual script and its inputs. Another restriction is that the scripts are not allowed to contain any iterative loops. This limitation is implemented because all scripts will be verified by all nodes in the network and it would be possible to include very resource-intense scripts if iterative loops were allowed. Bitcoin was also designed to be deterministic, meaning that all verifying nodes must produce the same result when verifying a transaction, hence queries for dynamic data (e.g. API-calls) are not allowed. The Bitcoin scripting language is non-Turing-complete. However, second-generation blockchain implementations allowing for the inclusion of a Turing complete scripting language exist—the most prominent one being Ethereum (however, Ethereum too is deterministic).

Chapter 3

Securities and smart contracts

3.1 Securities and central securities depositories (CSDs)

The industry of safekeeping and storing securities is called the custody industry. Today's multi-tiered custody industry is not something that was developed intentionally with a clear concept in mind; instead it has gradually evolved over time with ever-increasing complexity. It originally concerned the safekeeping of the physical representations of securities, however, one of the main difficulties with this was that when the ownership of a security changed, the certificate had to be moved from the seller's custodian vault to the buyer's.

To resolve this, the approach of most countries was to introduce a so called central securities depository (CSD) which would hold the assets for all the custodians in the country. This would allow the transfer of assets to be conducted simply by changing the owner in the CSDs books while the custodians remained to provide information regarding the customers' transactions and in general being the link between the investors and issuers of securities. With the assets rendered, immobile the dematerialization of the assets were the logical step forward, which allowed the ownership of assets to be represented by only the entries in the books without any underlying physical certificates needed to be stored [47].

A security can take the form of several different financial instruments, for instance, common stocks raise capital for a company by selling equity ownership, while bonds can be a way for governments to borrow capital through a promise of interest payments. As such, different securities are issued to the market in different ways depending on the asset class and composition of the financial industry in that region. The common denominator for securities, however, is the fact that they are tradable assets, and many CSDs (e.g. Euroclear) can actually manage the servicing of all security asset classes. Likewise, the common goal for all CSDs is to provide a definitive record of ownership of securities as well as facilitating the centralized settlement.

Although having extended the scope of their services since their conception, CSDs and custodians (whose role is now significantly marginalized and largely invisible to investors) still only make up a part of the puzzle that is the securities market. While securities do not actually move around the market anymore, since they are now immobilized by the CSDs, there are other important operations involved in the trade of securities, such as clearing and netting (performed at clearing houses), price discovery and matching counterparties (performed at trading venues, e.g. stock exchanges). The scope of the CSDs role extends beyond that of settling the trades of securities and has also come to include delivery versus payment (DvP) and asset servicing, such as administering the corporate actions subjected to different securities, e.g. dividends, stock splits, et cetera [48].

Besides the custodians, CSDs, stock issuers and investors there are a few other participating entities when involved when a security is traded. Firstly, both the seller and buyer of an asset will be required to use a stockbroker which has the required knowledge and provides access to a stock exchange. In addition, this middleman will introduce a fee in the form of commission. Secondly, the trade of an asset for cash is something that introduces a risk that one of the two parties might default after one part of the transaction has taken place. In order to alleviate this risk for the buyer and seller, an additional institution called a Central Counterparty Clearing House (CCP) exists which takes on this risk (in exchange for an additional fee). This institution takes control of both the asset and the funds before they are relayed to the buyer and the seller. A simplified description of the process of a trade is outlined below:

1. The buyer and the seller inform their brokers of a trade they want to make.
2. The brokers' trades are matched at a trading venue.
3. Trade details are sent to the CCP performing reconciliation and netting while concentrating credit risk.
4. The CCP receives the securities from the seller's broker through the seller's custodian.
5. The CCP receives the funds from the buyer's broker through the buyer's custodian.
6. The CCP instructs the CSD to perform delivery versus payment (DvP), crediting the buyer's custodian with the assets and the seller's custodian with the funds.

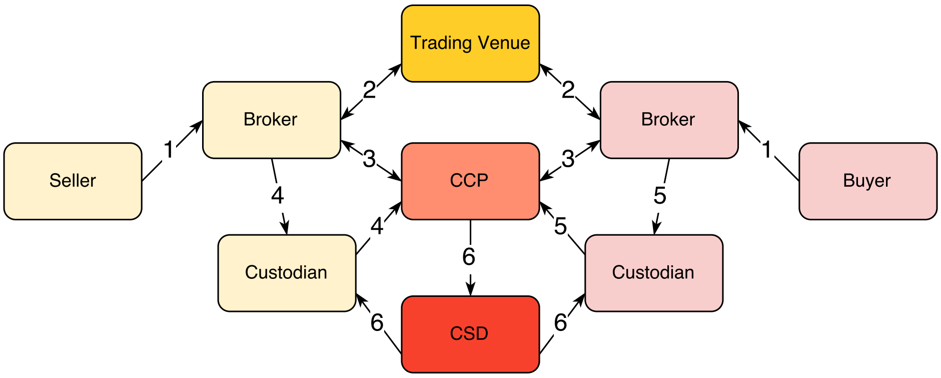


Figure 3.1: A depiction of the financial entities involved in a securities trade. The enumerated steps involved are described above.

The clearing and settlement using these traditional processes are subjected to latency, risk and large operational costs. Current conventions maintained by the Depository Trust & Clearing Corporation (DTCC) in the U.S. and the European Commission of the EU are characterized by settlement periods of a maximum of three days (T+3) and two days (T+2) respectively, reflecting the industry standards. The EU recently made the move from T+3 to T+2 settlement through new systems called TARGET2 and TARGET2-Securities, a development which has taken eight years to achieve [49]. The time and cost required for this change reflects how difficult it is to improve and advance the currently patched and multi-tiered financial system.

The problems in the post-trade settlement industry has been described by several industry players and can be summarized by the following list [29][50][51]:

- Lack of interoperability between siloed database systems
- Lack of standards
- Unnecessary complexity
- Expensive back-office procedures
- Long settlement cycles
- Settlement failures
- Lack of automation, requiring manual processes with risk of human errors
- Limited collateral fluidity
- Limited uptime

Distributed ledger technologies have demonstrated completely new approaches to managing transactions which are not subjected to these problems in the same way. Distributed ledger technologies have round-the-clock uptime, near-instant settlement and require no trusted intermediaries. Therefore, industry interest has been sparked in reforming these traditional systems with the hope of addressing the issues above.

Recent analysis indicates that the industry infrastructural costs could be reduced by US$15-20 billion annually in the banking sector alone when leveraging distributed ledger technology [1]. Furthermore, we have seen that distributed ledger technologies can support complex scripting features, which creates opportunities both in automating back-office procedures and in defining financial instruments in programmatic code. If a deployment of a grand scale open-source blockchain for the securities market were to be successful, this could create an entirely new landscape for the financial industry and financial instruments.

3.2 Peer-to-peer delivery versus payment (DvP)

In the previous chapter, we illustrated the architecture for delivery versus payment in traditional systems which requires several intermediaries and institutions. This convoluted architecture is the reason why settlement of securities is a costly process often taking several days. In this chapter, we will look at a way of achieving delivery versus payment directly using blockchain technology, without requiring any intermediaries.

Delivery versus payment in blockchain database structures will take different forms depending on whether or not the two assets changing hands exist on the same ledger or not. For example, if both cash and securities are tracked by the same blockchain, then delivery versus payment can be completed in a single transaction through a technique called *partial transactions*. However, multi-asset ledgers are a relatively rare occurrence. We will therefore begin by providing a description of how to approach the problem of achieving delivery versus payment in a scenario involving two different blockchains. Below follows the technical explanation of one such solution which can be described as a *two-phase commitment scheme*. This scheme was theorized by Noel Tiernan and further explained by Blockstream [40][52].

3.2.1 Atomic cross-chain DvP

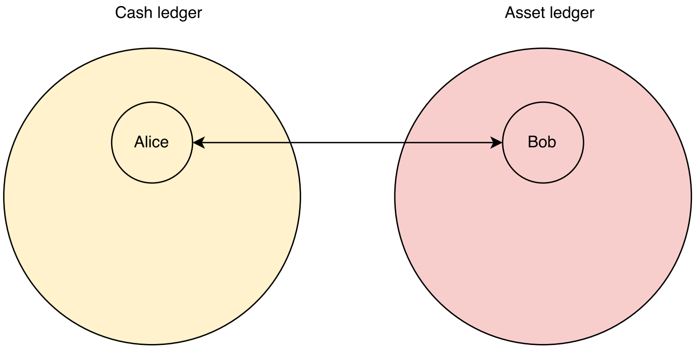


Figure 3.2: The problem at hand. Alice and Bob wants to trade cash for assets, but they are on separate ledger networks.

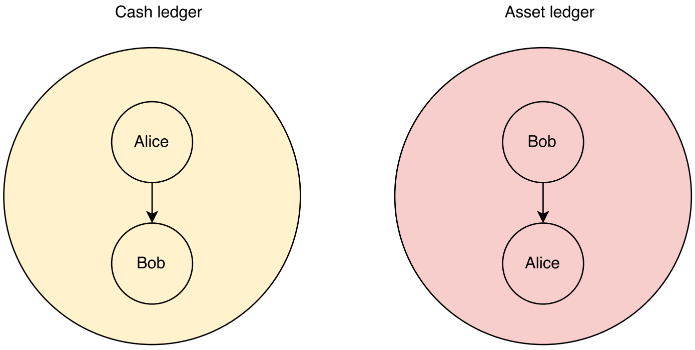


Figure 3.3: The solution. Since the assets and cash cannot leave their respective ledgers, Alice and Bob hold a key pair with both ledgers.

In this example, Alice owns 10 units on a cash ledger (e.g. a cryptocurrency) and would like to trade them for assets (e.g. a security) on an asset ledger, which is a ledger with tokenized (tradable) assets. Bob owns 5 units of an asset on the asset ledger and would like to trade them for cash. Alice and Bob both have a key pair on each ledger denoted pk_{A} for the asset ledger and pk_{C} for the cash ledger.

Alice chooses a secret α and creates transaction 1 (Fig. 3.4) which sends 10 units on the cash ledger to an output O_1, such that the output is only spendable with a combination of α and a signature from Bob's pk_{C}. This transaction is not transmitted to the network at once because if the transaction was transmitted, Alice would have no recourse if Bob drops out of the transaction scheme at this point, resulting in a situation where the funds would be stuck in O_1.

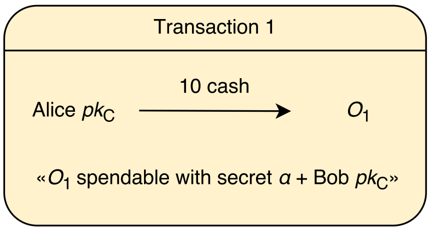


Figure 3.4: Transaction 1. Allows Alice to escrow her cash to Bob.

Instead, Alice creates a second transaction (transaction 2, Fig. 3.5) which allows her to return the funds from O_1 to herself if they have not been spent after 48 hours (locktime). Since the output O_1 is controlled by Bob's pk_{C}, this transaction requires a signature from him. Therefore she sends the unfinished transaction to Bob using a medium of her choice and asks Bob to sign it. Once this transaction has been signed and returned to Alice she can safely transmit transaction 1 to the cash ledger. At this stage, there is no risk to either Alice or Bob; since Bob does not know secret α he cannot spend O_1, and Alice has a recourse of returning the cash from O_1 after 48 hours if Bob drops out of the transaction.

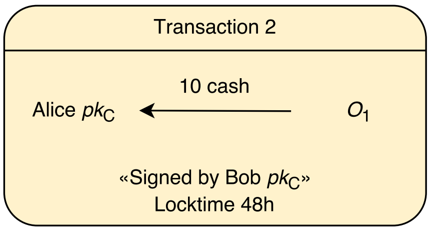


Figure 3.5: Transaction 2. Allows Alice to return her escrowed funds if Bob drops out of the transaction.

Bob creates transaction 3 (Fig. 3.6) sending 5 units on the asset ledger to an output O_2, which is spendable with Alice's pk_{A} in combination with the secret α which she knows. If Alice spends O_2 she will reveal α, which in turn will allow Bob to spend O_1.

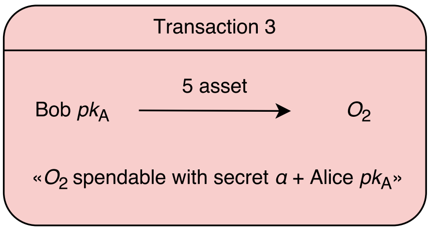


Figure 3.6: Transaction 3. Allows Bob to escrow his funds to Alice, knowing that she can only spend it if she reveals α

Similarly to transaction 1, this transaction is not transmitted to the network at once because if the transaction was transmitted, Bob would have no recourse if Alice drops out of the transaction scheme at this point, resulting in a situation where the funds would be stuck in O_2. Bob creates a second transaction (transaction 4, Fig. 3.7) returning the assets from output O_2 and asks Alice to sign it by the same reasoning as we made for transaction 2.

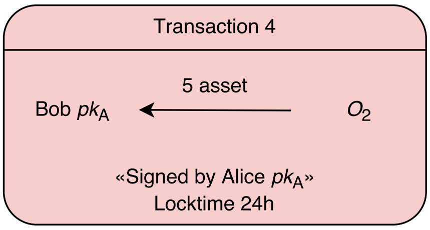


Figure 3.7: Transaction 4. Allows Bob to return his escrowed funds if Alice drops out of the transaction

This transaction will however only have a locktime of 24 hours—this is to prevent Alice from delaying the spending of O_2 until the locktime is about to expire. By introducing this time difference between the locked transactions, Alice can only subject herself to risk by waiting for the locktime to expire, since Bob will be able to return O_2 24 hours before Alice will be able to return O_1.

Now, Bob can safely transmit transaction 3 to the asset ledger since he can return O_2 after 24 hours if Alice does not spend it, and if she does spend it, she will reveal α, allowing Bob to spend O_1. The transmitting of transaction 3 concludes the *commitment phase* of the cross-chain DvP scheme (Fig. 3.8).

Alice and Bob can now enter the *execution phase*. Alice begins by sending Bob's assets to herself by spending O_2 using her pk_A and α. She sends this to a new output O_3 on the asset ledger which only she controls (transaction 5, Fig. 3.9). As explained, when Alice spends O_2 she reveals α on the asset ledger blockchain (publicly readable). Bob now has at least 24 hours (thanks to the time difference between the locktimes) to spend O_1. Just as Alice, he does this by creating a new output O_4 on the cash ledger which only he controls and sending Alice's cash to this output (transaction 6, Fig. 3.10). Transaction 6 concludes the execution phase and the DvP has now completed (Fig. 3.11).

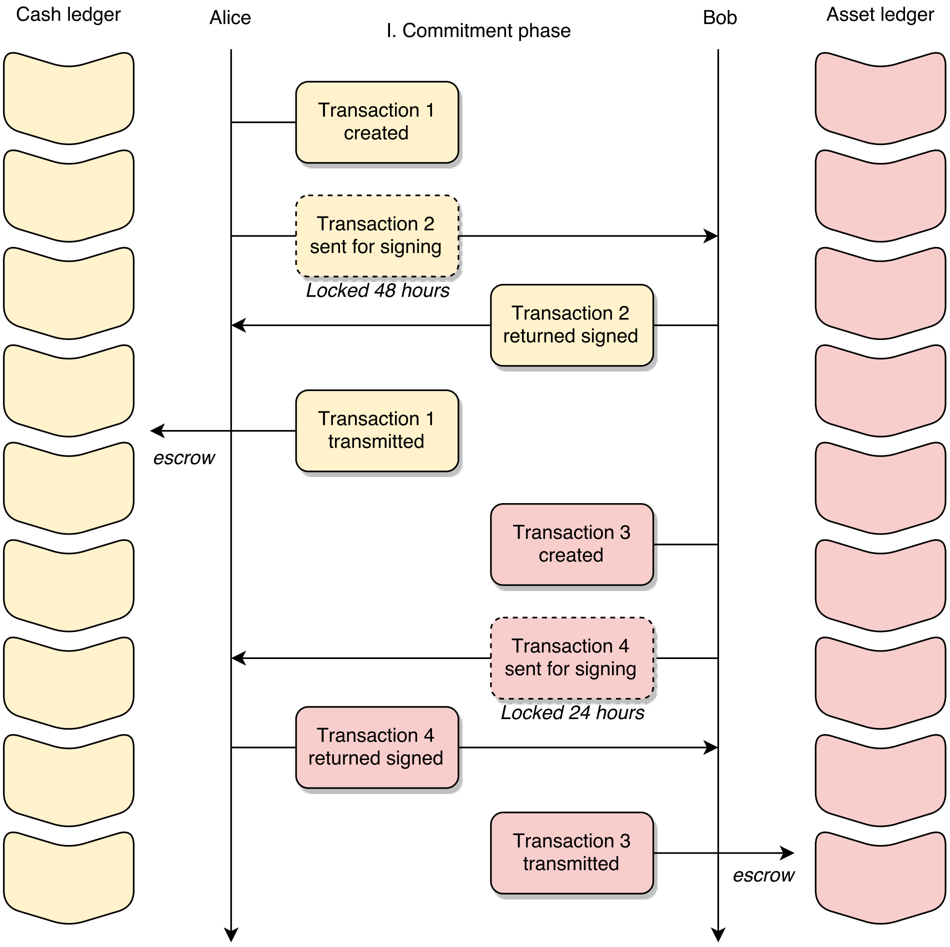

Cash ledger
Alice
I. Commitment phase
Bob
Asset ledger
Transaction 1 created
Transaction 2 sent for signing
Locked 48 hours
Transaction 2 returned signed
Transaction 1 transmitted
escrow
Transaction 3 created
Transaction 4 sent for signing
Locked 24 hours
Transaction 4 returned signed
Transaction 3 transmitted
escrow

Figure 3.8: Commitment phase. Alice and Bob escrow assets and cash on their respective ledgers.

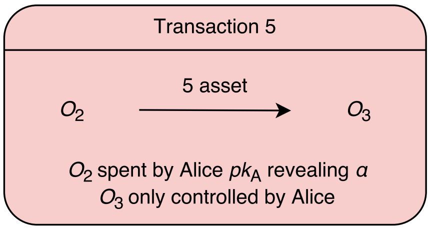

Transaction 5
5 asset
O_2
O_3
O_2 spent by Alice pk_A revealing α
O_3 only controlled by Alice

Figure 3.9: Transaction 5. Spends Bob's escrowed funds, but also reveals α.

This scheme is devised such that cross-chain DvP can be achieved atomically. Atomic trades implies that the trade either succeeds completely or fails completely,

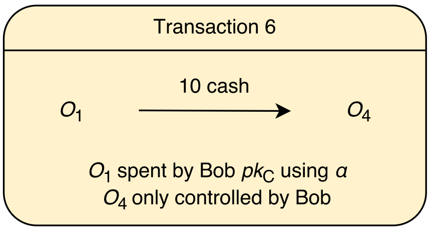

Transaction 6
10 cash
O_1
O_4
O_1 spent by Bob pk_C using α
O_4 only controlled by Bob

Figure 3.10: Transaction 6. Spends Alice's escrowed funds, but only if Bob knows α.

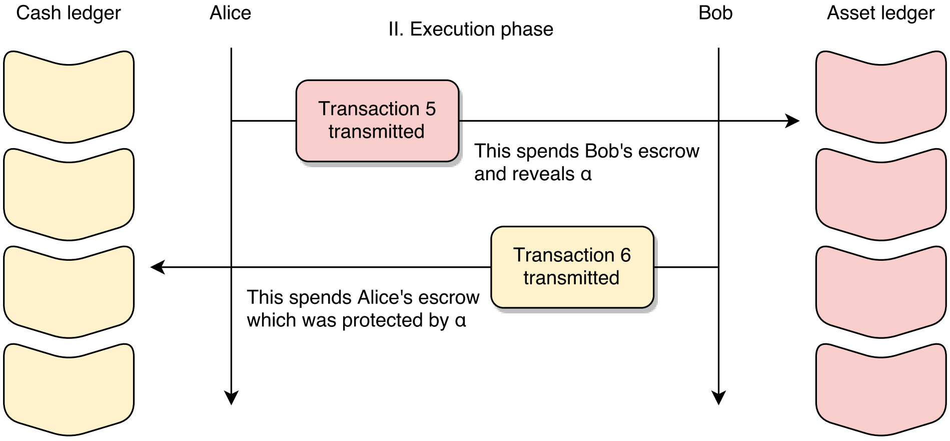

Cash ledger
Alice
II. Execution phase
Bob
Asset ledger
Transaction 5 transmitted
This spends Bob's escrow and reveals α
Transaction 6 transmitted
This spends Alice's escrow which was protected by α

Figure 3.11: Execution phase. The transaction is completed.

such that delivery does not happen without payment and vice versa. In contrast to traditional DvP, this is ensured by the cryptographic features of blockchain technology rather than by the grace of a trusted intermediary.

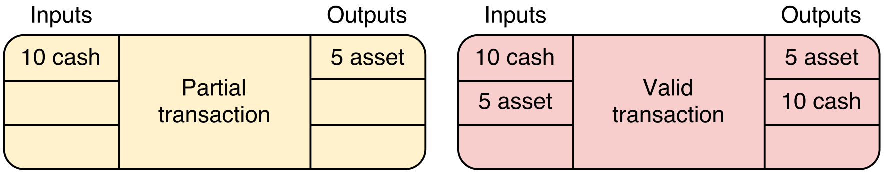
The description outlined in this section is relatively specific to the Bitcoin protocol, but it can in theory be expanded to be used between any blockchains as long they support similar cryptographic escrowing functions which allow its assets to be locked temporarily and provably on both sides of the transaction. We call this feature ledger-based escrow, since tokens are escrowed without intermediary support. While this shows that it is possible to achieve DvP without the assets ever leaving their respective ledger, it is rather complex and introduces latency (requires two settlement cycles on each ledger and a time-separated two-phase action). If we could transfer the assets to the same ledger it would create the opportunity of a more ideal method of DvP (see section 3.2.2 below).

3.2.2 Partial transactions

The most straightforward way to implement atomic DvP using blockchain technology is when both cash and assets are tracked by the same blockchain. The technique is called partial transactions, which is the principle of signing part of a transaction in a way that makes the signed portion fixed such that altering the portion makes the transaction invalid [53]. Partial transactions are already supported in the Bitcoin protocol, but since Bitcoin as of yet does not support other assets in its blockchain it is today more commonly used as a method for crowdfunding where all the outputs of a transaction are signed but anyone is free to add inputs. However, if we were to have a blockchain which supported multiple asset classes we could with relative ease utilize the technique to encompass a DvP use case. This technique is showcased in the example below.

Alice, wanting to trade her cash for an asset at a predetermined rate, begins by creating a transaction with her 10 units of cash as input and an output with e.g. 5 units of a desired asset to an address which she controls (Fig. 3.12.a). She proceeds to sign the input and the output and places the signature in the body of the input script. In Bitcoin, this is facilitated by choosing the hashtype `SIGHASH_SINGLE | SIGHASH_ANYONECANPAY` as a parameter in the signature. Compare this to the default case of a Bitcoin transaction which is to include a signature of all the transaction outputs, meaning that if one output is changed, then the input script will be invalidated. In this case on the other hand, we will allow additional inputs and outputs to be added, although the input of Alice's cash will only remain valid as long as the output with the asset to her address remains the same [54].[1]

Alice can now send this unfinished transaction to Bob (or anyone for that matter) through a medium of her choice to complete the transaction. Bob includes one or several inputs with a sum ≥ 5 units of the asset, as well as specifying an output for receiving Alice's cash which he controls (Fig. 3.12.b). Bob can now return the transaction to Alice who transmits the transaction to the network, or transmits it himself. In either case, since the entire DvP is included in a single transaction there is no way for one part of the transaction to be rejected without the entire transaction being rejected (since nodes either accept or reject transactions completely).



(a) Partial transaction with input and output added by Alice.

(b) Bob completes the partial transaction by adding an additional input and output.

Figure 3.12: Example of a partial transaction by Alice as well as the completed transaction by Bob.

[1] Do note that the transaction as a whole is invalid as long as the sum of the inputs of each individual asset is less than the sum of the output for that same asset (where cash is also a type of asset).

3.3 Representing securities on a blockchain

In the previous chapters we explained the complexity of trading securities in traditional systems and described how digital assets can be traded peer-to-peer using a blockchain. The relief from the requirement of trusted intermediaries motivates the use of a blockchain to track the ownership of securities. Platforms facilitating this type of securities trading would be able to provide advantages such as:

- Near real-time settlement (between milliseconds and minutes)
- Peer-to-peer delivery versus payment
- Round-the-clock uptime
- Auditability (transparency, UTXO model)

If the ownership of securities could be tracked as blockchain tokens and the required licensing from regulators was granted, these advantages would indeed become available to the securities market. Such a platform could be enabled either through the use of a meta-layer on top of an existing open blockchain network as described in section 2.7 on colored coins, or through the development of a new blockchain platform built for this purpose. One example of a very recent initiative in this area is the proprietary permissioned distributed ledger platform OpenCSD offered by SETL, which was launched on June 1, 2016 [55].

Using a permissionless token such as the bitcoin as base for the tracking of securities ownership offers plentiful benefits in the sense that there would be no barrier of entry to purchasing securities. Although, at the time of writing, such securities trading is prohibited in many jurisdictions. There have been some examples of companies circumventing these rules by offering equity in other ways, one example being the crowdsale launched by the Bitcoin wallet company Mycelium; by utilizing the colored coins platform Colu, Mycelium sold 5% of the stake of their future revenue through an offering of tokenized SARs (stock appreciation rights) [56]. However, under decentralized pseudonymous trading, an investor would need to trust the issuer entirely since these platforms generally are outside the reach of law enforcement. Indeed, a significant hurdle is the fact that securities such as ones constituting equity ownership do not hold value by themselves. They are merely claims or promises on returns (e.g. dividends) that need to be honored by some entity in order to hold value. This creates a stark contrast between securities traded on a distributed ledger and cryptocurrencies inherent within the ledger; bitcoins can be traded without trusting the counterparty, because the bitcoin token does not require the backing of an issuer. Nor does the trade require trusting the seller that the bitcoin is valid (the buyer merely needs to trust the ledger). In contrast, in the case of stocks and bonds, the guarantee of receiving dividend or interest payments is only upheld by the means of legal enforcement should an issuer betray their promise.

This separates securities trading from cryptocurrencies in a fundamental way because securities trading on a blockchain thus reintroduces the requirement of trust and thereby risk and associated expenses. The level of risk introduced highly depends on the amount of control and permissions given to authorities to enforce claims, as well as the availability of support from clearing institutions in this framework. In order to facilitate this type of control the blockchain must implement an identity system. Although known identities in blockchains are chiefly associated with permissioned blockchains, permissionless systems can facilitate an identity layer through different channels, e.g. through a TLS/SSL signing of blockchain public keys, although securities trading using such a scheme still would require the backing of a legal framework.

Companies could theoretically issue securities themselves without any backing authority or regulatory unit, however this would mean that there would be no safety net for an investor neither against fraud nor against companies neglecting the contractual clauses of the security (e.g. dividends, voting rights). Riskier investments generally demand larger potential returns to remain attractive, resulting in larger costs for the issuers on using such a platform.

There have been some initiatives of issuing securities using blockchain technology which have received regulatory support—for example, the U.S Securities and Exchange Commission (SEC) amended a filing by Overstock which requested permission to issue some of their own stock on their blockchain platform $t\theta$ [57]. Another notable example in the financial industry is the company R3 (see section 4.1), which has demonstrated experiments on their distributed ledger platform Corda, on which banks automated the creation of the contractual clauses of financial agreements and notarized the hashes of said agreements on the ledger [58].

These are early experiments using distributed ledger technologies in the context of the securities market. Activities such as timestamping agreements and tracking ownership within proprietary ledgers do not tap the full potential of what blockchains are capable of. By better utilization of the scripting language of a blockchain, we could theoretically[2] achieve:

- Automation of dividends and interest payments (section 3.5)
- Trustless derivatives trading (section 3.6)
- Self-executing corporate actions
- Automated securities servicing
- Asset control

In section 3.4 on smart contracts, we outline how features of this character can be implemented.

Moreover, there is an important quality to the concept of representing assets on a blockchain that needs to be understood; just like bitcoins "live" entirely within the blockchain and does not need any third party to support its autonomy and sovereignty, a security which is exclusively represented on a blockchain as a smart contract could also function autonomously within a blockchain. That is the important distinction; a blockchain-security which merely notarizes a legal obligation is not autonomous, similarly to how a physical item of which the ownership is represented on a blockchain does not cement the real ownership of the item. If the concept of ownership is merely abstract, it relies on trust in third parties for it to hold practical value (this will be covered in more detail in section 3.5).

[2] By "theoretically" we mean according to the findings of this report with regard to the examination of distributed ledger technologies.

3.4 Smart contracts

The scripting features of blockchain technologies described earlier can be used to create what are known as "smart contracts". The concept of smart contracts was introduced by Nick Szabo in 1993 and describes a type of cryptographic contract in which the contractual obligations of the contract are executed through self-enforced computer code [59].

Such contracts need to be resolved by an unbiased mediator. This makes distributed consensus-oriented ledger networks ideal platforms for this purpose, since distributed ledger networks can apply game theoretical motives for the mediator network to act honestly. It is the decentralized verification processes that make distributed ledgers suitable for smart contracts. Smart contracts are verified in the same way as Bitcoin's regular script-based transactions, which are—as described previously—verified by every node in the network. This means that every node has to run every contract and thus execute every contracts code placed on the blockchain.

Ethereum is a second-generation blockchain technology which was designed specifically as a decentralized smart contracts platform. In addition to using a Turing complete scripting language to facilitate this, Ethereum has a built in virtual machine (EVM) which works like a decentralized computer. Smart contracts in Ethereum can be written in high-level languages such as Serpent and Solidity, which are compiled to stack-based byte code resembling a mix between LISP, Assembly and Bitcoin's Script.

As previously mentioned, the Bitcoin scripting language is not Turing complete because Turing complete contract code containing complex calculations such as iterative functions can be resource-intensive and even contain infinite loops. Therefore, a decentralized permissionless platform needs to avoid spam in order to be feasibly sustainable. Ethereum solves this issue by requiring a fee for contracts which is consumed by the nodes to pay for the costs of executing the contract. Depending on the complexity of the contract and how the execution of the program branches out into more computations, Ethereum contracts can run out of "gas" (Ethereum's native token ETH) and require additional deposits.

All transactions in blockchain technologies are essentially smart contracts—it is only the complexity which varies. An example of a simple smart contract is the 2-out-of-3 multi-signature output script described in section 2.8. The locked bitcoins of a 2-out-of-3 multi-signature transaction can be thought of as a contract between two parties which uses a third party mediator to release the funds to the right person. This third party could be any entity which both parties trust not to collude with either party in the transaction. For example, the third party could be a mediator which settles a bet between two parties.

Moreover, smart contracts can actually be used to facilitate delivery versus payment. While the examples described previously in this chapter may be ideal for trading simple assets, we will see later in this paper how smart contract DvP can be used in more complex situations. Such a smart contract holds an asset and releases that asset to another user if a payment of a predetermined amount in another specified asset class is made, simultaneously releasing the payment to the issuer of the smart contract. Here is an example in Solidity of such a contract:

```
contract DvPcontract {
    address owner;
    uint public askingPrice;

   /*
    * Alice creates a smart contract and sets an asking price.
    * This contract will have its own address in the Ethereum
    * network. When constructing the contract, Alice will also
    * deposit the asset she wishes to exchange.
    */
    function DvPcontract(uint price){
        owner = msg.sender;
        askingPrice = price;
    }

   /*
    * The variable "this.balance" allows access to all ETH
    * currently in the contract. If this was to be expanded
    * to allow additional assets, these could theoretically
    * be accessed by this.assetBalance.
    */

   /*
    * Bob calls the trade function in the contract, including
    * askingPrice amount of ETH in his call.
    */
    function trade(){
        if(this.balance > askingPrice){
            owner.send(this.balance);
            (msg.sender).send(this.assetBalance);
            suicide(owner); /* contract is spent, therefore removed */
        }
    }

   /*
    * Includes a cancel function in case Alice changes her mind
    */
    function cancel(){
        if (msg.sender != owner) return;
            owner.send(this.assetBalance);
            suicide(owner);
    }
}
```

A slightly more complex smart contract would be a crowd funding contract where contributors can contribute to a contract until it reaches a target goal. Here is an example of how such a contract might be implemented in Solidity.

```
contract crowdFund {
    address public creator;
    uint public goal;
    Contributor[] public contributeList;

    struct Contributor {
        address returnAddress;
        uint amount;
    }

   /*
    * Initializes a crowdfund; define a receiving address
    * and set a goal.
    */
    function crowdFund(uint newGoal){
      creator = msg.sender; /* creator == creator address */
      goal = newGoal;
    }

   /*
    * Contributes to this crowdfund. Records the contribution
    * address in case the crowdfund gets canceled.
    */
    function contribute(){
      contributeList.push(Contributor(msg.sender,msg.value));
      checkGoal();
    }

   /*
    * Tests whether the goal is met or not.
    */
    function checkGoal(){
        if(this.balance >= goal){
            suicide(creator);
        }
    }

   /*
    * Cancels the crowdfund, returning all contributions.
    */
    function cancel(){
        if(msg.sender != creator) return;
        for(uint i = 0; i < contributeList.length; i++){
            contributeList[i].
                returnAddress.send(contributeList[i].amount);
        }
   }
}
```

3.4.1 Oracles

The most interesting smart contracts require some data-feed input from an outside source, e.g. a derivative smart contract that needs access to the exchange rate between two currencies or the spot price of a share. It is, however, not possible for smart contracts in Bitcoin nor Ethereum to make API-calls to the Internet. This is because Bitcoin and Ethereum were designed as deterministic protocols, meaning that any transaction or smart contract needs to return the same value for each node that verifies it in order to avoid consensus failures. Since API-calls to the Internet are non-deterministic (dynamic) such sources would interfere with the consensus processes of these ledger systems. That is why Bitcoin and Ethereum smart contracts can only fetch data within their own internal ledgers.

There are methods of addressing these problems in blockchain technologies—by introducing "oracles". An oracle is a third party which provides a smart contract with specific data from the outside world. One example of such a service is Oraclize which is compatible with Ethereum. Oraclize works as a trusted link between the Internet and the Ethereum blockchain by pushing in data from URLs and other services [60].

However, this means that the smart contract is no longer decentralized since the data-feed is controlled by a single entity which could manipulate the responses. To address this concern, Oraclize uses a service called TLSNotary which uses TLS/SSL to provide cryptographic proof that the API-response was in fact retrieved from the correct source. However, this still requires some degree of trust towards Oraclize since they could attempt to query the server multiple times and cherry-pick a response [61]. Theoretically, oracles would not need to be trusted since cryptographical proofs could be issued by the main source. A price feed could be fetched from e.g. Nasdaq, if Nasdaq simply provided an API which responds to queries, containing the following information:

- Query information
- Query response
- Timestamp
- TLS/SSL signature

The user creating the smart contract that is dependent on this query would add the Nasdaq's public key to be used by the smart contract when verifying the response. The oracle would still be required to anchor the data in the blockchain to conform to the protocol's deterministic principle, which in would mean that nodes only need to read from the anchored data in the verification process, thus avoiding problems of dynamic API-calls as well as not inadvertently DDoS-attacking the outside data-source.

This method still requires the client to trust a single source of information when acquiring data from outside sources, in this case Nasdaq. This introduces a risk of Nasdaq becoming dishonest, however, this risk can be mitigated by e.g. querying multiple data-sources, filtering abnormal responses, and calculating a mean value of the remaining. The risk level in such a case may be considered to be tolerable in certain cases.

3.4.2 Prediction markets

The perhaps most trustless approach to receive truthful information from the outside world in a distributed ledger platform, is through the means of prediction markets. Prediction markets are a type of forecasting tool which utilizes a network of users to predict the outcome of a given event. Users place bets attempting to predict the outcome of an event, and are economically rewarded according to how close to the right answer they are when the bet settles in a zero-sum game. Thus, users will be incentivized to invest time and research into their predictions since their profits or losses depend on the proximity of their wagered value to the estimated value by the prediction market. We can observe prediction markets as a method of reaching a type of decentralized consensus on what the outcome of an event in the future will be. Thus, prediction markets allow a type of derivative which derives its value based on the outcome of virtually any event in the future or the performance of any underlying asset. It could in theory be possible for the resulting value of a prediction market bet to be used as input in a separate smart contract.

A prediction market can be compared to a sports betting exchange, where users can buy or sell odds (or, to use correct sports betting terminology, back or lay bets). Before a game has started, there will be an industry of profit-incentivized sports bettors researching the likelihood of win versus loss for the respective teams. This economic model thus produces a crowdsourced research initiative into the likelihood of an event that cannot be mathematically calculated, since the probabilities are derived from non-mathematical input variables. Due to the bettors being able to buy and sell odds themselves on the platform (versus only buying odds offered by a bookmaker), either by matching existing buy/sell order in the book or offering new odds, the bettors act like market makers/takers similarly to how it works on a stock exchange. Since these odds can be bought and sold in real time, even while the game is playing, the odds can track the reality of the game with very high accuracy. When one team scores a goal, the odds will quickly change in favor for that team, since of profit-seeking bettors will purchase any "cheap" bets in the book that remain from before the goal had been taken into account. Thus, the betting exchange tracks the probabilities of the outcome of a game in real-time through economic incentives. A general prediction market allows bettors to bet on any type of event, not just sports—thus creating a type of financial instrument called an "event derivative". One example of such a project is Hivemind by economist Paul Sztorc [62].

3.5 Modeling a security as a smart contract

The possibility of trading ownership of a security with round-the-clock uptime, near-instant settlement and complete auditability in a decentralized manner has already been demonstrated in examples using e.g. colored coins approaches. The really interesting aspects of blockchain technology and securities trading come in play when not only tracking the ownership of securities, but rather, expresses the business logic of securities in the form of autonomous smart contracts.

A security can be viewed as a contract between an issuer and the holder of the certificate. Smart contracts encode the contractual clauses into self-enforcing computer code. For instance, an equity share modeled as a smart contract should ideally give a holder the following:

- Right to dividends
- Options to purchase
- Voting rights

While interest and dividend payments could be automated by a smart contract framework in theory, the smart contracts would in these examples not be able to hold the funds within the contract as they need to remain liquid to the issuer of the security. In the example of a bond, the idea is that the holder of the bond lends money to the issuer in exchange for interest payments at regular intervals. If the issuer had to lock funds as proof of being able to pay the interest, then the bond would become pointless since it would provide no benefit to the issuer.

While the smart contract platforms cannot eliminate the risk of an issuer defaulting (and should not, since taking on risk is the reason interest is paid), they could still be used to facilitate efficient and smooth interaction between issuers and holders of securities by providing a single channel for the issuers to interact with all security holders without needing assistance from asset servicing agents. Below, we present an example of how dividend payments could be implemented in a smart contract. It takes advantage of the Turing completeness of Ethereum by including iteration over a set of stockholders, in a straightforward use case for the issuer.

```
contract EquityMain {
    address public creator;
    uint public totalEquityCount;
    address[] public equityList;

    function EquityMain(){
        creator = msg.sender;
        totalEquityCount = 0;
    }

   /*
    * Function usable by the issuer.
    * The funds distributed by this method are taken directly from
    * the smart contract and can be deposited at its address,
    * either beforehand or in the same transaction as the method
    * is called.
    */
    function payDividend(uint amount) {
        if(msg.sender != creator) return;
        if(amount < this.balance) return;
        uint DividendPerShare = amount / totalEquityCount;
        for(uint i = 0; i < equityList.length; i++){
            StockHolder holder = StockHolder(equityList[i]);
            holder.send((holder.equityCount())*DividendPerShare);
            StockHolder(equityList[i]).pay();
        }
    }

   /*
    * Method used by the underlying stockholder contracts. It
    * is called when they sell some but not all of the equities
    * they hold. Triggers the creation of additional smart
    * contracts.
    */
    function additionalStockholder(uint amount, address newOwner){
        bool contains = false;
        for(uint i = 0; i < equityList.length; i++){
            if(equityList[i] == msg.sender){
                contains = true;
            }
        }
        if(contains){
            equityList.push(address(new StockHolder(amount, newOwner)));
        }
    }

   /*
    * Method usable by the contract's creator, i.e. the issuer.
    * Issues new equities and sets the issuer as the initial owner.
    */
    function IssueEquities(uint amount){
        if(msg.sender != creator) return;
        equityList.push(address(new StockHolder(amount, msg.sender)));
        totalEquityCount = totalEquityCount + amount;
    }
}

contract StockHolder {
    address public owner;
    address public topLevelContract;
    uint public equityCount;

   /*
    * msg.sender is the messenger who created this contract,
    * i.e. the contract above.
    */
    function StockHolder(uint amount, address newOwner){
        owner = newOwner;
        equityCount = amount;
        topLevelContract = msg.sender;
    }

   /*
    * This method is used in case the owner
    * wants to sell some or all of his equities.
    */
    function changeOwner(address newOwner, uint amount){
        if(msg.sender != owner) return;
        if(amount > equityCount) return;
        if(amount <= 0) return;
        if(amount == equityCount){
            owner = newOwner;
        }else{
          equityCount = equityCount - amount;
          EquityMain(topLevelContract).additionalStockholder(amount,
              newOwner);
        }
    }

   /*
    * Method used by the issuer of the equity
    * to distribute dividends.
    */
    function pay(){
        if(msg.sender != topLevelContract) return;
        owner.send(this.balance);
    }
}
```

As mentioned previously, securities that essentially are claims or promises reintroduce the requirement of trust into blockchains. If it was possible for securities to execute the business logic of its contract without relying on third parties, such securities could be traded without the need for any intermediary involvement. The possibility of trading securities without needing intermediaries would provide substantial benefit to end users. In the next section, we will look at such a type of security that could potentially be issued without any requirement of trust and be traded as trustlessly as a bitcoin token.

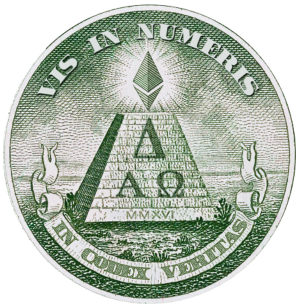


Figure 3.13: Proposed logo for TheDAO, a decentralized autonomous organization managing US$200m which exists as a smart contract within the Ethereum platform. The latin reads "vis in numeris - in codex veritas" which roughly translates to "strength in numbers - truth in code". The logo reflects the sentiment that blockchains allow its users to organize themselves through code rather than legal agreements. *Source: Hibryda.*

3.6 Decentralized derivatives market

In this section, we introduce the concept of a decentralized derivatives market. While this section only intends to serve as a brief introduction to this concept, its purpose is to help readers imagine a context in which the capabilities of the technologies described in this paper is realized to a greater extent than that of simply tracking ownership. As we have seen previously in this chapter, smart contracts can facilitate the functions of custody, brokering, clearing and even enforce the delivery versus payment settlement. However, as we have previously mentioned, in the case of stocks and bonds, a smart contract cannot guarantee dividend or interest payments since the contract would be required to hold the funds within itself from which the payments would be made, thus making the funds unavailable to the issuer of the stock or bond.

Derivative contracts are type of security that derives its value from the performance of an underlying asset. The underlying asset can be any commodity, but also be things such as foreign currency exchange rates and interest rates. While many types of derivatives exist, such as forwards, swaps and options, we will in this section look only at a type of derivative called a futures contract. In a futures contract, two parties enter an agreement to transact an asset for cash at a predetermined price on a specific date in the future. As an example, a manufacturer of electronic goods who needs a certain amount of silver for its conductive properties could enter a futures contract with a silver provider to purchase a certain amount of silver at a predetermined price at some point in the future. Let's imagine a futures contract where the silver provider agrees to sell 1 kilogram of silver six months in the future at the price of $500 per kilogram. This way the manufacturer's and the silver provider's expected profits are not affected by the market price fluctuations of silver.

In many cases of futures trading, contracts are cash-settled. In our example above, this would mean that if for instance the price for silver had increased to $600 per kilogram at the time of the contract's maturity date, the manufacturer and the silver provider would not conduct the trade, instead they would simply settle the cash difference where the silver provider would pay the manufacturer $100. In the reversed scenario where the price had decreased to $400 per kilogram, the manufacturer would pay the silver provider $100. This way, the manufacturer and the silver provider are still protected against the price fluctuations of silver, while not being restricted to each other to conduct their business—the manufacturer can simply purchase silver from another provider six months in the future and the total cost will still be $500 when netting the difference from the futures contract. Similarly, the silver provider can sell silver at any price six months in the future with ensured $500 revenue in total for that kilogram of silver.

In derivatives trading, cash settlement is the most common method of settlement since it is often the most practical method—in some cases the actual physical delivery is even impossible in such cases where the underlying asset is a stock market index or an interest rate. Cash-settled futures contracts allow traders to speculate on assets without physically owning them, and they also allow futures traders to enter leveraged positions. For instance, a trader could enter a position to sell 1,000 kilograms of silver at $500 per kilogram without owning any silver.

Respectively, another trader could accept that offer without owning the $500,000. In this scenario, both the buyer and the seller would maintain a margin balance with their broker. If the spot price of silver increases to $503 per kilogram, the seller would now be at a $3,000 loss. If the margin balance maintained with the broker is less than $3,000 dollars, the seller would be margin called. In the case that the seller has no more funds to add to his margin balance, the position would close and his margin balance would be forfeited to the buyer. In this type of trading, the respective margin balances of the traders are updated daily according to their respective profits/losses, which ensures that neither party will be unable to pay [63].

The functions of a cash-settled futures contract could be served entirely by a smart contract. As such, these smart contracts would not require trust between the buyer and the seller, which set them apart from stocks or bonds smart contracts. In its simplest form, a trader wishing to speculate on an underlying asset would create a smart contract which would require the following:

- Deposit of funds as margin balance from both the buyer and the seller
- Oracle service tracking the underlying asset spot price
- Future date on which to settle the contract
- Predetermined price at which to settle the contract
- Leverage multiplier parameter (e.g. 1x, 2x, 5x, 10x, 20x)
- Addresses for settlement payouts of the buyer and the seller (payout addresses)

A smart contract that monitors an underlying asset through an oracle service could manage the margin balances of two traders and re-divide the funds between them according to the parameters defined in the contract. As the contract is managed by a smart contract and not by any third party, the derivatives contract could be entered and settled peer-to-peer. This is different to other forms of blockchain-supported securities trading because it would not only provide the tracking of ownership—it would also enforce the business logic of the security. Assuming a neutral oracle (see section 3.4.1), this trading would be trustless. This means that the security would not only be represented on the blockchain, it would "live" within the blockchain, and it would thus similarly to Bitcoin need no authority backing, since the smart contract would in this sense contain the custody, clearing and settlement of the security within itself.

A decentralized smart contract cash-settled futures trading platform using blockchain technology would bear many advantages compared to traditional platforms. Firstly, it would naturally inherit the benefits associated to tracking securities ownership on a blockchain (we recall section 3.3):

- Near real-time settlement (between milliseconds and minutes)
- Peer-to-peer delivery versus payment[3]

[3] Peer-to-peer delivery versus payment of smart contracts will be explained in the next section, "Trading a smart contract".

- Round-the-clock uptime
- Auditability (transparency, UTXO model)

Secondly, it would further improve upon existing platforms in terms of the following:

- Automation
- Disintermediation
- Trustlessness
- Flexibility

Such a platform would indeed be flexible, Users could choose from any combination of underlying assets tracked by oracles and decide on any type of leverage. However, in order to maintain the disintermediative and trustless properties of the platform, there needs to be a requirement that profits of either party will always be capped by the margin balance deposited by the counterparty. This is due to the fact that decentralized markets need to guarantee solvency at all times.

3.7 Trading a smart contract

While the concept of a smart cash-settled futures contract is fairly straightforward, this concept does introduce the need for a design that outlines how such securities could also be traded. As the notion of this design is only theoretical, this section aims only to describe how such a design could be implemented from an architectural perspective. As closing one's position in a derivative contract entails selling the derivative to someone else, trading a derivative that is embodied by a smart contract would imply trading one's payout address in the contract for someone else's liquid funds. The value of owning the payout address would depend on the margin balance that was initially deposited as well as the performance of the underlying asset relative to the negotiated price at which the contract will be settled. In the futures smart contract following a successful trade, the leverage and time to maturity remains the same while one of payout addresses is replaced with the payout address of the new counterparty.

To explain how this scheme works, we assume a scenario where Alice and Bob enters a smart futures contract. Later, Alice wishes to exit the contract by selling her position, an offer which a new buyer, Carol, accepts. As we have seen previously in section 3.4, smart contracts can be used to facilitate delivery versus payment. Ideally, we would like to incorporate such a feature into the cash-settled futures contract. While the payment portion of the delivery versus payment smart contract remains fairly similar in this design, the delivery would not involve delivering an asset, rather, the contract would update the payout address of the exiting counterparty. To allow this, the futures smart contract needs to facilitate a function that puts a position in the contract up for sale to a price selected by the exiting counterparty. The scheme works as follows:

1. Alice and Bob enters a cash-settled futures smart contract (section 3.6).

2. Alice calls a function in the smart contract in which she defines a price at which she wants to exit the contract at and confirms this with her signature.
3. Carol notices that Alice's position in the smart contract is up for sale through a blockchain monitoring interface.
4. Carol signs a transaction containing the funds matching the price set by Alice as well as specifying a payout address of Carol's choosing.
5. The smart contract receives the payment, replaces Alice's payout address with Carol's and forwards Carol's payment to Alice.

Chapter 4

Proposing a blockchain design for the securities market

4.1 Adoption of distributed ledger technology in the financial industry

The purpose of this chapter is to propose well-measured overarching design choices for a blockchain infrastructure suitable for the securities market with the intention of re-engineering the role of CSDs. However, as we've seen, CSDs are by no means isolated entities—they live in the center of the web of institutions involved in the clearing and settlement cycle. As such, any blockchain design intended to rehaul the securities depository infrastructure needs to pay respects to the entirety of the securities market infrastructure when its design choices are formulated—especially since the purpose of the endeavor is to alleviate the interoperability issues and tedious reconciliation processes associated with existing systems. In fact, in order to avoid blockchain infrastructures being developed in parallel and thereby recreating the very issues they were intended to solve, the undertaking requires the alignment of the entire industry. Therefore, the undertaking is not merely a technological challenge, but a massive challenge in coordination and collaboration.

As previously mentioned in this paper, blockchain technology has seen an overwhelming avalanche of the interest from the financial industry in recent years. It has been suggested that 2015 was the year that financial institutions realized that they needed to adapt in order to avoid having their positions in the industry challenged by the new technologies. A selection of papers published by financial institutions in the recent year can be seen below:

- *CSDs, virtual currency investments and "blockchain" technology*, published by the European Central Securities Depositories Association (ECSDA), 2015
- *Virtual Currencies and Beyond: Initial Considerations*, published by the International Monetary Fund (IMF), 2016
- *Embracing Disruption — Tapping the Potential of Distributed Ledgers to Improve the Post-Trade Landscape*, published by the The Depository Trust & Clearing Corporation (DTCC), 2016

- *Blockchain in Capital Markets: The Prize and the Journey*, published by Euroclear, 2016
- *The Impact and Potential of Blockchain on the Securities Transaction Lifecycle*, published by the SWIFT Institute, 2016
- *The Distributed Ledger Technology Applied to Securities Markets*, published by the European Securities and Markets Authority (ESMA), 2016
- *Distributed ledger technologies in securities post-trading*, published by the European Central Bank, 2016

While these institutions see some differences in the potentials in blockchain technology, one sentiment is consistently clear: long settlement cycles, limited business hours and high remittance fees are all symptoms of an outdated unnecessarily complex infrastructure. The DTCC and Euroclear represent the two largest CSD organizations in the world, and have both argued that they are the candidate best positioned to coordinate the development of blockchain technology for the industry, recognizing the fact that the undertaking needs to be a collaborative effort. It also sheds light on the competitive dimension to the road to adoption. In a paper titled *Blockchain & Financial Services: The Fifth Horizon of Networked Innovation*, published by the Massachusetts Institute of Technology in April, 2016, the authors outlined the five "horizons" in networked communication. The first being the Internet, the second being the World Wide Web, the third being the cloud and the fourth being the ubiquity of Internet-connected smart phones. This brings us to the fifth horizon—blockchain technology [64]. And so, if blockchain technology does ignite a financial technological revolution of sorts, as with any revolution, there is most likely going to be winners and losers.

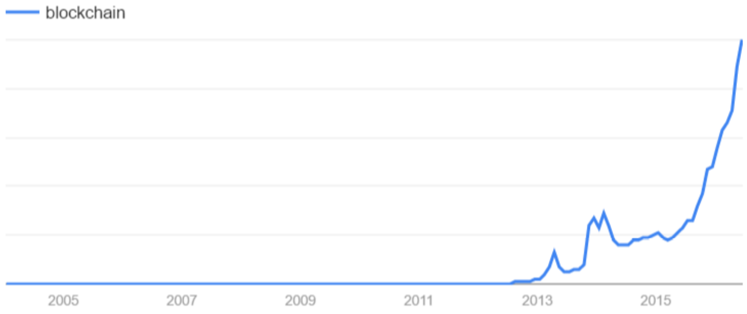


Figure 4.1: Blockchain search term interest over time. *Source: Google Trends.*

The "battle of the blockchain" has already begun, and the battlefield is not only the securities market, rather, the entirety of the financial landscape is embroiled. On September 15, 2015, nine of the worlds largest banks united in a consortium led by the startup R3, with the goal of tapping the potential of blockchain technology for financial transactions [65]. As of April 5, 2016, the consortium consisted of 43 global banks and have trialed several blockchain platforms for this purpose [66], partnering with Microsoft for their Blockchain-as-a-Service offering, as well as launching their own platform named R3 Corda, described as a "distributed ledger for recording and managing financial agreements". On April 18, 2016, the bank Barclays announced that they were experimenting with "smart contract templates" to trade interest rate swap derivatives using the R3 Corda platform. Microsoft was not the only tech giant competing for the R3 partnership; the company also performed tests on distributed ledger platforms provided by companies such as IBM, Intel and Amazon [67].

Meanwhile, the DTCC has entered a partnership with startup Digital Asset Holdings, led by former JPMorgan Chase executive Blythe Masters, to adopt distributed ledger solutions for the financial markets, starting by targeting the the repurchase agreement market [68]. Simultaneously, both the DTCC and Digital Asset Holdings, as well as R3, IBM, Intel, SWIFT, JPMorgan Chase, Wells Fargo, Deutsche Börse Group, CME Group, Hitatchi, Fujitsi, Blockstream, Red Hat, Cisco, and many others, are all members of the collaborative open-source blockchain development project Hyperledger, a project maintained by the Linux Foundation.

The blockchain technology initiatives launched in the recent year are too many to be covered in this paper, but some other notable mentions are:

- On May 11, 2015, Nasdaq announced that they were using the Open Assets Protocol for their private market platform Linq [43][44]
- On December 9, 2015, the U.S Securities and Exchange Commission (SEC) amended a filing by Overstock which requested permission to issue some of their own stock on their blockchain platform $t\theta$ [57]
- On January 21, 2016, Digital Asset Holdings announced a partnership with Australian Securities Exchange to develop a distributed ledger solution for the Australian equity market [69]
- As of May 3, 2016, the Post-Trade Distributed Ledger Working Group (PTDL) consisted of 37 financial institutions, among them banks, clearing houses and exchanges [70]
- On May 26, 2016, Ripple announced that the consumer bank Santander Bank had begun piloting international remittances using the Ripple payment protocol [71]
- On June 1, 2016, the Australian stock transfer company Computershare and SETL announced the proprietary permissioned distributed ledger platform OpenCSD, a subscription-based settlement system for securities [55]

4.2 Ledger ecosystem

It is likely that many different ledgers will exist in parallel to each other. While some will exist in parallel due to market competition, others will need to exist in parallel due to necessity. As previously explained, blockchains can be thought of as a new database structure for recording information. As such, they will be used to track a wide variety of assets not only restricted to digital currencies and financial instruments. Rather, blockchains will evolve to encompass other things such as supply chains, health records, insurance information, art works, royalties, diamonds, et cetera, with many such projects already underway. Since different records of information need to be managed by a different sets of rules, they will naturally exist on different platforms.

Similarly, different financial assets also benefit from being managed by different rules. It is unlikely that there will be a one-size-fits-all distributed ledger solution to hold all possible financial assets ranging from anonymous second-generation cryptocurrencies to government-backed currencies to stocks, bonds, derivatives and all other financial agreements. Even in the cases where the assets are of the same class, e.g. stocks, it is unlikely that all the stocks in the world will exist on a single distributed ledger platform, because of the same difficulties of coordination and differences in legislations which constitutes the basis for the reasons as to why stocks do not all exist in the same CSD today.

When designing a distributed securities ledger platform, one first needs to consider which asset classes the ledger should manage. Before we can orient ourselves in answering such a question, we will need to consider the possible advantages and drawbacks of designing a distributed ledger which manages many asset classes compared to a distributed ledger specialized in tracking a single asset class—in other words, comparing a multi-asset ledger versus a multi-ledger ecosystem.

The biggest advantage of a multi-asset ledger is that different asset types could be traded using the ideal DvP scheme—partial transactions (described in section 3.2.2). In order for a cross-chain DvP scheme to work in a multi-ledger ecosystem, the users must hold an account with both ledgers and as a consequence, the users will be subjected to the systemic risk of both ledgers, as well as the added latency of the process. In this sense, multi-asset ledger settlement is a less convoluted process.

However, designing a multi-asset ledger does not necessarily entail that more assets will find their way onto that ledger. Financial instruments such as stocks are a finite resource, and a multi-asset ledger will need to compete for the right to represent companies stocks just the same as a single-asset ledger will. Additionally, as a consequence of distributed ledgers using a singular consensus model for all the asset types tracked on that ledger (atleast as they exist today), multi-asset ledgers would need to compete against several specialized single-asset ledgers with uniquely-purposed consensus models for each asset class. It is not difficult to see how this would become problematic. A multi-asset ledger would for example need to struggle very hard to simultaneously compete versus a single-asset ledger tracking an anonymous cryptocurrency and a single-asset ledger using threshold signature scheme with known identities. Single-asset ledgers will always be better at solving the problems specific to the asset it is tracking, while multi-asset ledgers will need to balance themselves between being either useful to a few, very similar assets, or useless to many, very different assets.

For a system tracking only the ownership of assets, these problems are less severe. However, as we've seen in the instance of securities, merely tracking ownership gravely undermines the potential of blockchain technologies. A system that wishes to facilitate things such as corporate actions, voting, automation of dividend and interest payments, smart contract derivatives or only just balancing permissioned granular access control with decentralization and trustlessness, a multi-asset ledger design becomes extremely complex or even impossible to devise. It is possible that multi-asset ledgers may be able to service a function in the distributed ledger ecosystem, but as long as they use the same consensus model for all assets they manage, they are most likely only going to be prevalent in the form of two-way pegged platforms linked to several single-asset ledgers where assets are only held temporarily for multi-asset DvP settlement.

4.3 Designing for interoperability

One of the main reasons as to why settlement in traditional securities infrastructures takes several days is owing to the fact that the securities are processed through many different market participants which have minimal transparency into each others siloed systems. As a result of this, there is a need for the process of verifying that two sets of records are in agreement with each other when settling a transaction, which is called reconciliation. Since reconciliation has to be performed by several entities in the financial system whenever they transact with one another, the process causes friction and adds costs to the system.

It is easy to imagine how a multi-ledger ecosystem could quickly see an increase in complexity of a similar character. For instance, let's suppose that a customer of e.g. OpenCSD wants to buy a certain security, but the security in question does not exist on the OpenCSD platform, but it does exist on the R3 Corda platform. A user would then need to hold an account with both platforms, or accept that they cannot buy the security. What may be considered an "optimal" solution for the customer, would be if some administrators of the OpenCSD platform would hold an account with R3 Corda to purchase the security there and create a representation of the security on its own platform. One of the problems this introduces, is that it imposes a risk to the OpenCSD platform administrators, since they would now be exposed to the systemic risk of the R3 Corda platform, as well as adding on any latency from the R3 Corda settlement cycle to its own settlement process. The more convoluted a system is, the more risk is introduced. The complexity and risk is subsequently translated into cost and latency for end users.

The evolution of the multi-ledger ecosystem has already begun and appears to be an unavoidable destination for the distributed ledger technology adoption roadmap in the securities market. The way forward for distributed ledger technologies in order to effectively address these looming problems, is by designing for interoperability through standardization. Let us try to understand what interoperability means in this context. In order to conceptually comprehend the notion, we will use the analogy of the Internet to approach this matter from a more abstract perspective. In this analogy, payment networks like Bitcoin and permissioned settlement systems like OpenCSD represent the equivalency of Local Area Networks (LANs). LANs can be open (public WiFis) or require permission from the access point, just as distributed ledgers can be open to the public or require permission from the network. Interaction between users connected to the same LAN is typically problem-free, however, when two users of two separate LANs want to interact with each other, that is when problems arise. What is needed in this context, is a way of connecting LANs by using a public standard for data exchange between the networks—e.g. an Internet Protocol (TCP/IP).

Luckily, it appears that there already exists a method of connecting ledgers to each other. The method described in the section 3.2.1 which outlines a scheme for cross-chain DvP can be expanded upon to facilitate any type of cross-ledger transaction. The requirement for the cross-chain DvP scheme to work atomically, meaning that either delivery and payments both happens or neither happens, is that the ledger supports the feature of *escrowing* the funds. In transaction 5 (Fig. 3.9, p. 31), when Alice spends the assets Bob escrowed to her, she reveals the secret which allows Bob to spend the cash escrowed to him. However, this scheme is incomplete in the sense that it requires Alice and Bob to hold accounts[1] with each other's ledgers, which is exactly what we want to avoid.

One project aiming to solve this issue is the Interledger project. The Interledger project is an open-source project started by developers of Ripple and is now managed by the W3C Interledger Community Group, which aims to formulate a standardized protocol called the Interledger Protocol (ILP) for how payments can be issued across ledgers. While an exact description of how the Interledger Protocol works is outside the scope of this paper, we offer a basic explanation below.

Continuing from the cross-chain DvP scheme in which Alice wants to pay Bob on a different ledger, imagine that there is a third party, which we will call the connector. The connector holds an account on both ledgers. In this scheme, Bob chooses the secret and provides a hash of the secret to Alice. When Alice wants to pay Bob she escrows funds to the connector, which the connector can only spend if he knows the hash generated from Bob's secret. In the Interledger Protocol, this is known as the *condition.* The connector now escrows his funds in a transaction to Bob, which can only be spent by Bob by revealing the secret. Thus, when Bob claims the fund from escrow, the connector can claim the escrowed funds from Alice. If Bob does not claim the funds from escrow, Alice can redeem her escrowed funds using the same locktime principle as in the original cross-chain DvP scheme. Similarly, the connector can redeem his escrowed funds on Bob's ledger.

While the connector technically is an intermediary, neither Alice nor Bob need to trust him owing to the trustless cryptographic escrow feature provided by the ledgers. As such, the no-trust relationship between the three parties means that no risk is added by involving the connector in the transaction. Continuing from the Internet analogy, the connector in this protocol has a similar role to a gateway. The connector will charge a fee for providing the service, but since the connector does not need to take on any credit risk from Alice or Bob and therefore only needs to take on the systemic risk of the ledger systems, the connector can offer this service at a low price. Additionally, in the Interledger Protocol, the role of connector is open to anyone, which opens up for fees being offered competitively and are thus minimized.

In the Interledger Protocol, there can be any number of connectors, and the protocol defines an algorithm for finding the shortest path through the web of connectors for Alice and Bob. When Alice wants to send a payment to Bob

[1] The term *account* is used metaphorically here to refer to any means of owning assets on a ledger.

and thus escrows funds with the first connector, this causes a cascading wave of escrowing to occur up until it reaches Bob through the chain of connectors. Reversively, when Bob claims the last connector's escrowed funds and thereby executes a transaction, this causes a cascading wave of executing transactions traveling back until it reaches Alice.

While this description thus far only describes how Alice can pay Bob, the Interledger Protocol can enable DvP between Alice and Bob in a similar manner by instantiating a loop back to Alice where Bob would perform the function of a connector; suppose Alice wants to buy a stock from Bob, she starts an escrow chain, using a condition that she herself generates. The chain starts with her and goes through some number of ledgers until it reaches Bob. Bob sees the escrowed payment to himself and puts the stock transfer in escrow just like a connector would. The stock transfer then goes through multiple ledgers until it reaches Alice. Once Alice sees the prepared incoming stock transfer, she fulfills the condition and all the transfers execute in reverse order just like regular Interledger payments.

The Interledger Protocol will potentially be able to deliver any asset into any account and will trade assets as needed to construct the necessary paths. We believe that the Interledger Protocol is the best designed solution to the looming interoperability problems of the evolving multi-ledger ecosystem.

4.4 Choosing a permissioned or permissionless model

Due to the regulated nature of the financial industry, opponents argue that infrastructural designs which promote decentralized control such as the Bitcoin network are unfit for the securities market since financial institutions and regulators would not be able to impose laws that market participants must follow. According to European Union law, institutions that deal with the clearing and settlement of securities such as central securities depositories need to adhere to a defined regulatory framework such as the *Regulation (EU) No 909/2014 of the of the European Parliament and of the Council of the European Union on improving securities settlement in the European Union and on Central Securities Depositories and amending Directives 98/26/EC and 2014/65/EU and Regulation (EU) No 236/2012*, to protect customers and facilitate policymaker oversight [72]. In a similar narrative, the U.S. Securities and Exchange Commission requires the offering or selling of securities to be registered under federal securities law.

Meanwhile blockchains do impose rules on what a user can and cannot do, such as preventing users from double-spending bitcoins or spending bitcoins they do not own, an exhaustive study of the extent to which regulatory rules can be defined in technical code has yet to materialize. However, the European Central Bank recently published a paper on *Distributed ledger technologies in securities post-trading* which concluded that "irrespective of the technology used and the market players involved, certain processes that feature in the post-trade market for securities will still need to be performed by institutions" [29]. This argues the relevance of *permissioned* distributed ledger technologies when designing a viable infrastructure for the securities market where transaction processing nodes are regulated entities, i.e. a distributed ledger technology adjoined with a public key infrastructure (PKI).

The permissioned aspect of blockchains opens up opportunities in some areas while hampering its potential in others. Where some regulators are alarmed by the dangers of blockchain payment networks in terms of enabling criminal activities, others emphasize the new opportunities arising in approaching economic regulatory efforts through the application of software controls. The principle of using proprietary blockchain technologies to enforce regulations through technical code rather than through traditional legal enforcement has been dubbed "RegTech" by United Kingdom financial incumbents [73]. In the light of such possibilities, some regulators have expressed a wait-and-see inclination to the evolution of the technology [74].

One benefit of using a permissioned model where identities are known is that of granular access control; permissions and roles concerning access and control can be defined at different levels and be distributed to suit the needs of the system, thus creating a more adaptable framework for defining privacy, confidentiality and control in the system. For instance, in a financial market, the ability to create blocks and the responsibility to process them could be delegated to a predefined list of institutions, while the ability to inspect and monitor them could be provided to regulators, courts and auditors. While providing legal oversight may seem counterproductive to the ambitions of blockchain technology as blockchains were incipiently conceptualized as a means of circumventing authoritative control and avoiding censorship, blockchains do not have to be confined to this use. Blockchains are simply a novel type of database structure and its design pattern can be used in a wide assortment of applications.

For the juridical system, the conjunction of pseudonymity and settlement finality (the inability to reverse transactions) in the Bitcoin network is an unappealing combination since this creates a situation where authorities are less able to combat theft, fraud, money laundering, illegal drug trade and ransomware [75]. In distributed ledger technologies, any update can be made to the ledger as long as there is a consensus among the nodes to accept the update. With a set of permissioned entities managing such a ledger, entities could thus accept updates reversing unlawful transactions despite the reversing transaction not having been signed with its owner's private key.

In section 3.3 we discussed how securities are inherently different to digital currencies in the sense that securities essentially merely are promises. While in section 3.6, we outlined a scheme for trustless derivatives trading, it only covers a portion of securities trading as a whole. This suggest that there is grounds for separating trustless and non-trustless securities trading platforms. Securities trading platforms managing instruments such as stocks and bonds which require trust will benefit more from a permissioned model than a platform which controls sufficient collateral to enforce the clauses of a contract itself, since the permissioned model allows for the seizure of user's assets.

However, while offering and selling securities doe require the explicit blessing from regulators on beforehand, it is possible to imagine a permissionless securities trading platform in which the holders of stocks and bonds take on the full risk of the issuer defaulting on their promises. Such a system could be based off of reputation, such that statistics on the performance of a securities issuer allow traders to decide themselves whether the risk of the issuer defaulting is high enough to deter them from purchasing the security. While a recognized identity is not required to join a permissionless network, it is easy for a participant to let their identity on the network be known if they so choose; a well-known company could simply let their identity on the blockchain be known through a public announcement. In such a scenario in which the blockchain identity can be linked to a real-world identity, securities issuers are no more protected against prosecution if they commit fraudulent activities than they would be if a permissioned platform was used. The important difference here is that a permissioned platform offers a recourse of controlling the assets of a fraudulent issuer, while a permissionless platform does not.

In order to decide whether to use a permissioned or a permissionless model for a blockchain securities depository, one needs to have a clear idea on what the securities depository is aimed to achieve. A securities depository which is only to be used for a smaller market with a selected user base, permissioned models offer the greatest advantages. However, for a securities depository that aims to become the de facto record of ownership for a national or even global user base, some considerations must be made. We envision a securities blockchain platform which other entities such as stock exchanges would plug themselves directly into.

To understand how a blockchain can maximize its network effect, we will once again approach the challenge by drawing comparisons to the network which has had the greatest network effect of all time—the Internet. So the question becomes whether or not the Internet is permissioned or permissionless. Contrary to public belief, the Internet is *not* a permissionless network. In contrast to the Bitcoin network where any person can generate a valid ECSDA key pair, a user wishing to access the Internet must be allocated a valid IP address which stems from the Internet Corporation for Assigned Names and Numbers (ICANN). While ICANN employs a decentralized nonprofit governance structure, ICANN is a private American corporation. Seemingly, permissioned networks can achieve global spanning network effects as long as they are sufficiently *permissive.*

Another relevant observation with regard to the Internet is how identities are managed. The public key infrastructure (PKI) of the web, for instance, is managed at the application layer of the OSI model. Another order of identity, the host identity (IP address), is managed at the network layer. Similarly, a permissioned blockchain can manage the transaction processing node identities at the network layer of the blockchain network, and stock issuer identities and trader identities at the application layer.

Continuing from the taxonomy outlined in section 3.3, an example of a permissive blockchain is a public permissioned blockchain, which has relaxed[2] restrictions on identities accessing the network but puts severe restrictions on the identities of nodes processing transactions. We believe that such a model is the most logical starting point for a securities blockchain—one that would stand a chance of attaining regulator approval while simultaneously being open enough to allow the network effect of a global user base, as well as facilitating the necessary transparency and interoperability characteristics to avoid reintroducing the frictions of

[2] E.g with a relatively low barrier to entry such as if an electronic citizen ID was the only requirement for access.

siloed systems.

Interestingly, the verifying nodes in a permissioned blockchain consortium would not have to consist of only financial institutions. In theory, they could be managed by governmental entities and regulators, or by transparent, audited international nonprofit organizations under the scrutiny of a consortia of the previous. The amount of control each entity in the consortium should have can be to chosen to reflect any desired power balance; the entities do not need to be confined to hold to only one key each, rather, each entity can hold multiple keys—constituting a percentage of the total keys in the threshold signature scheme. It is not clear that a stock trader wouldn't rather rely on the signature threshold scheme consensus of a semi-decentralized institutional consortium rather than the pseudonymous consensus of a permissionless decentralized network.

4.5 A second-generation blockchain platform for securities

Now let's look at what permissioned platforms can do in the context of smart contracts. We know that a difficulty for smart contracts in permissionless platforms is the action of fetching data from the outside world. The solution thus far is to use oracle services which pushes the data into the blockchain. The difficult part of this design is to make sure that the oracle services act honestly and do not attempt to collude with any of the parties in the contract. The proposed solution in permissionless designs is to use multiple oracles who will compete with each other based on their reputation of being honest. As such, it appears that a trusted predefined consortium is required to settle securities contracts that require data from the outside world. Since a permissioned model already utilizes a predefined list of trusted entities for processing transactions, the same list conveniently constitutes the basis for an oracle consortium. This creates the possibility of merging the role of the oracle and the role of the transaction processing node into one in permissioned model, without undermining the decentralized control of the network. Where this would be an unscalable approach for large permissionless networks, smaller permissioned consortia could leverage threshold signature schemes to determine valid oracle responses for every contract processed by the network.

From this design we could continue to build an extremely powerful, scalable, high-throughput (see section 2.3) second-generation blockchain platform. In section 3.5, we presented an example of how a smart contract could enable stock issuers to automate dividend payments. In order to facilitate these more complex and expressive features, our ledger must support a Turing complete programming language. For this purpose, we propose a freemium securities blockchain platform in which regular transactions and non-iterative smart contracting functions are offered for free to traders, while access to the full-scale framework for corporate actions and asset servicing functionality would be provided in a pay-per-use model in which the payments are collected by the processing nodes. As with Ethereum, this cost would be necessary to ensure that the Turing completeness of the platform is not exploited to overload the network. This would be a similar business model to how ICANN earns revenue by overseeing the allocation of domain names and IP addresses, which ICANN subsequently uses, among other things, to run root DNS servers which service the Internet. This design does however introduce the need for a currency token in the platform. We address this further in section 4.7.

The existence of a currency token would indeed be a necessary utility for a second-generation securities blockchain platform, e.g. in order to allow smart contracts to manage both cash and assets. It is not too unlikely that we will see blockchain-based national currency systems supporting the financial industry in the future (see section 4.7.5). In order to facilitate automated payments in a system which does not allow a smart contract to lock down capital (see section 3.5), we must build smart contracts which can *pull* payments from addresses. If the securities blockchain had access to an interoperable blockchain-powered regulated currency system, we can imagine how this could be made possible. Securities issuers such as companies and governments, which pay interest or dividends to the holder of the security, would maintain a currency account for outgoing payments. These currency accounts would be under high scrutiny from regulators through the means of both legal and technical code to ensure that the stock issuers honor the obligations. This way, stock traders would be protected from the exit scam risk permissionless ledgers would impose.

A difficulty with such a scheme where the currency blockchain and the securities blockchain are separate ledgers is that the smart contracts would need to execute on both ledgers, while smart contracts as we know them are programs that operate within a specific ledger. A potential way to address this issue is discussed in the Codius project white paper [31].

Although many financial agreements require confidentiality, software such as block explorers could be built on top of the platform to retrieve real-time statistics and data. However, a discussion on which parts of such a platform that could viably allow transparency, or which restrictions should be placed on the identities of the entities that should be able to access that data is a matter outside the scope of this paper. Such decisions need to be made in collaboration with a decision-making board consisting of law-makers, regulators and policymakers which have the mandate to formulate a schematic for the distribution of the granular access control permissions, whereas technologists can only explain the possibilities, dynamics and dimensions to the different intricacies of the task.

4.6 Governance

In order to decentralize the control over a ledger, there is not only a need for decentralized consensus on the new appendages of data to be added to the ledger. There also needs to be a a decentralized consensus through which the software development process that defines the rules of the network is controlled. The Bitcoin network is made up of nodes that run the Bitcoin client software. The most popular client at this time of writing is the client "Satoshi" maintained by software development team Bitcoin Core [76]. While this is an open-source project which is open to the public to fork or suggest changes to, commit-access to the Satoshi client is restricted to a small, meritocratic group of individuals.

In the Bitcoin protocol, the term *block size limit* refers to the maximum size blocks must not exceed in order to be considered valid and accepted into the blockchain by the other nodes. Since blocks can only be added every 10 minutes and the size of blocks limits the amount of transactions that can be contained in a block, the block size parameter limits the throughput of the Bitcoin network. The current block size limit is 1 MB and limits the Bitcoin network throughput to a theoretical maximum of 7 transactions per second. Since the Bitcoin user base is growing, this limit is considered a bottleneck for user adoption. As such, members of the Bitcoin community has voiced concerns over this limit and are advocating for its increase. Seeing as Bitcoin Core have been unwilling to implement the increase within the time frame requested by members of the community, this has caused alternative Bitcoin client development projects to form.

One example of such an alternative Bitcoin client development effort is Bitcoin Classic, which is a fork of the Satoshi client, updating the block size limit to 2 MB. Bitcoin miners can switch to the Classic client to signal their support the limit increase. If this alternative client accumulates 75% of the hash rate, the clients will enable the increase and a split of the network will ensue when the first block which is invalid under the previous rules is created.

This type of change to the rules of the network is called a hard fork, due to its incompatibility with the previous rules. On the other hand, if a change merely restricts functionality instead of adding to it, updating the client will not be necessary for a majority of nodes since the new type of blocks were already valid using the old clients. In this type of update (called a soft fork) only the miners will be required to uphold the new rules. It should be noted that while multiple soft forks have occurred in Bitcoin's history, no hard forks have yet occurred. The requirement of consensus on block parameters is a principle that provides decentralized governance of the Bitcoin network.

Any blockchain project that wishes to decentralize control of the ledger also needs to adopt a decentralized model for governance. Blockchain projects are therefore arguably ideal open-source projects, which nurture wide-ranging collaboration in software development efforts and create the postulation for ensuring further maintenance. The Hyperledger Project, maintained by the Linux Foundation, is a collaborative open-source software project (see section 4.1 for more information regarding members). The project is governed by a board of directors representing the different organizations participating in the project so that a consensus can be reached on the characteristics and rules that defines the Hyperledger software. This process is similar to the governance model employed by ICANN, as well as most other governing bodies overseeing large projects spanning whole industries.

Furthermore, open-source collaborative blockchain projects have the benefits of fostering innovation and bringing in new ideas. In Bitcoin and the effervescent cryptocurrency community, new developments keep progressing and the concepts are constantly innovated upon. It has attracted countless entrepreneurs, enthusiasts and academics over the years, with Bitcoin companies attracting millions in venture funding. It is the open-source open-network properties of these protocols that allow these projects to harness that energy. The progress that has been made in blockchain technology is owing much to the fact that Satoshi Nakamoto did not release the concept as a patented proprietary software. Also, exhibiting these properties is what has made the Bitcoin network so robust. The Bitcoin network has developed over time, but it has been running since 2009 and thus been a target for numerous cyberattacks. Yet, the Bitcoin network is still operating, more resilient now than ever.

4.7 Currency token

An important piece of the puzzle when designing a blockchain platform for the securities market is not only the choice of currency, but also whether to use separate cash ledgers in conjunction with the securities ledger, or to design a blockchain that tracks the ownership of both cash and securities. We will look at each alternative individually.

4.7.1 Alternative 0: No currency token

Remember that a securities blockchain can be used to track only the ownership of securities and not support any blockchain currency at all. Such solutions would rely on traditional payment solutions to handle the management of cash. However, as traditional payment systems do not support trustless escrow features (see section 4.3 on Interledger) there is no way for them to support trustless DvP. A simple explanation why trustless DvP does not work in this scenario follows below:

In this example, Alice wants to buy an asset from Bob. Alice will use her traditional bank for the transaction, while Bob will transfer the security to Alice's account on the securities blockchain. Now, either Alice has to send the funds directly to Bob's bank account and *hope* that Bob releases the asset, or vice versa. It is apparent that this scheme would hold no guarantees, rather both Alice and Bob would deposit their assets and cash with a trusted third party who would facilitate the DvP service.

4.7.2 Alternative 1: Native cryptocurrency

Most existing blockchains are cryptocurrency systems. Cryptocurrency systems typically have their own native token, which is minted through mining (see section 2.1) and introduced into the system over time. The value of the token is derived from the utility of the platform—cryptocurrency systems providing unique, useful features to their user base attract investments and the native token appreciates in value. These systems are sensitive to network effect—as the user base of a system grows, so does the utility of the token since it can now be used to transact with more parties. Subsequently, the demand thus the appreciated value of the token increases.

The minting process of a new cryptocurency tokens is a delicate process. The supply must be controlled, and the tokens must be introduced at a steady pace using a fair distribution mechanism. This distribution mechanism can only be regarded as fair if the minting process is open to the public to participate in. In a blockchain, the minting process is intertwined with the processing of transactions. Thus, in a securities blockchain platform where transactions are processed by a predefined list of entities, the minting process would be unfair.

4.7.3 Alternative 2: Cryptocurrencies via cross-chain DvP

From section 3.2.1 we learned that Alice and Bob can transact cash for assets as long as they both hold accounts on both ledgers. Using the atomic cross-chain DvP scheme described, a securities blockchain could support peer-to-peer trading with other cryptocurrency systems as long as the ledgers provide ledger-based escrow. Moreover, interoperability solutions such as the Interledger Protocol described in section 4.3, if successful, may allow such atomic cross-chain DvP schemes to function without Alice and Bob needing to have accounts on multiple ledgers themselves, rather, this responsibility would be delegated to connectors.

4.7.4 Alternative 3: Cryptocurrencies via sidechains

Cryptocurrencies can easily be introduced on a permissioned securities blockchain by using a two-way federated peg as described in section 2.6. Any permissioned blockchain supporting a two-way federated peg would be regarded as as a sidechain to that cryptocurrency. The permissioned blockchain could in theory support a two-way federated peg to an unlimited number of cryptocurrencies. This way, the securities blockchain can utilize any cryptocurrency tokens without requiring its own minting process. When both securities and currencies exist on the same blockchain, the blockchain allows for peer-to-peer instant DvP via partial transactions (see section 3.2.2). It is also possible to create a separate intermediary "settlement blockchain", which would be both a sidechain to the securities blockchain and the cryptocurrency blockchain.

4.7.5 Alternative 4: A token backed by a traditional currency

Most people in the world do not use cryptocurrencies, and will want to purchase securities using their traditional national currencies. From alternative 0 we can gather that a securities blockchain would benefit from translating traditional currencies into a blockchain token representation of that currency. This is similar to what many digital payment systems already do today. When a user transfers money to e.g. PayPal, PayPal credits that user with the corresponding amount in its internal systems. When the user withdraws money from PayPal, PayPal issues a payment to the user using traditional payments systems (e.g. bank wires). Thus, an amount of e.g. dollars in PayPal is represented by an equivalent amount of "PayPal-dollars". The PayPal-dollar is backed by PayPal. The incorporation of such a feature would mean that customer funds would need be stored by the securities consortium for the duration they exist within the securities blockchain ecosystem.

More attractively, one can imagine a blockchain currency backed by a financial institution, e.g. a central bank. One example of such an initiative is RSCoin, a cryptocurrency experiment by The Bank of England [77]. The structure of RSCoin is based on Bitcoin in some ways and supports the same type of cryptographic features which allows us to escrow funds trustlessly. Thus, RSCoins can interact with our securities blockchain either by transferring the RSCoins via a two-way federated peg sidechains mechanism to the securities blockchain, or, through an atomic cross-chain DvP scheme (via Interleger connector or directly).

WAVES is a second generation blockchain initiative attempting to create easy solutions for financial institutions (central banks and other institutions alike) to issue currency directly on a ledger [78]. This currency would naturally be backed by the issuing entity. It is of utmost importance that the currency token can be trusted to retain its value, which is why currency tokens issued by a central bank would be very valuable addition to this ecosystem. This would be especially important because it eliminates any possible currency risk users would be exposed to by holding alternative currencies.

4.8 Technical feasibility

The DTC (the CSD subsidiary of DTCC) performs approximately 1.4 million settlement-related transactions per day (≈16.2 transactions per second) [79]. The Nasdaq stock exchange totals roughly 10 million trades per day (≈115.7 transactions per second). As previously mentioned. the Bitcoin network has an ideal maximum limit at 7 transactions per second, averaging ∼2.5 transactions per second [80]. However, the Bitcoin network consists of roughly 6,000 nodes from all over the world and keeps the throughput artificially low (1 MB block size) in order to sustain decentralization (see section 4.6) [81]. The recommended hardware specifications for a node running the Bitcoin Core client are [82]:

- 2 gigabytes of memory (RAM)
- 80 gigabytes disk space
- 400 kb/s data transfer rate
- A 900MHz quad-core ARM Cortex-A7 CPU[3]

A blockchain network consisting of a consortium using enterprise grade hardware could run nodes several hundred times more powerful than such installations. Modern data centers support terabyte data transfer rates and petabyte data storages and would not need an artificial block size limit [85][86]. As previously mentioned, a blockchain using a threshold signature consensus model can process millions of transactions per second. For a more comprehensive examination of this subject, see section 2.4.4.

[3] This is not an official requirement, but we can assume this to be satisfactory since the Raspberry Pi 2 Model B can run a full Bitcoin node [83][84].

Chapter 5

Summary

A blockchain is a new approach of managing and structuring data, specialized for tracking the ownership of tradable assets. A blockchain database structure makes for a well-measured solution for the purpose tracking the ownership of securities as well as any other asset class. In this sense, it is indeed possible to create a securities depository using blockchain technology. However, to allow next-generation financial activities in the securities market, such as peer-to-peer delivery versus payment and smart contract securities, the blockchain platform needs to incorporate a currency token.

In order to facilitate trading without a trusted intermediary, the currency token and the securities blockchain platform must be made interoperable with each other by employing ledger-based escrow or through sidechain two-way federated peg inclusion. Hence, cryptocurrencies make ideal candidates in the enabling of a currency token, whether the cryptocurrency is Bitcoin or a national cryptocurrency backed by a financial institution. We have summarized the characteristics we propose for a blockchain-based distributed securities depository below. The blockchain should employ:

- A decentralized governance model
- An open-source development model
- A permissioned threshold signature scheme consensus model
- A permissive model for platform access
- A granular access control model for regulatory oversight
- A cryptocurrency token
- A Turing complete scripting language accessible to paying users
- A freemium business model

Further, the blockchain must position itself correctly within the financial industry and distributed ledger ecosystem. As such, the blockchain must conform to industry standards currently being established. Particularly, the blockchain must be designed for interoperability with other ledger systems.

Chapter 6
Conclusion

There is a great opportunity in finance. In the securities market, the securities depositories represent the most fundamental layer on which all other systems are dependent. Over the course of several generations, this layer has been subject to effectivization of the settlement cycle through a process of ever-increasing centralization. A decentralization of the depository layer of the securities infrastructure is a vast undertaking which requires broad industry collaboration and should be be implemented gradually. The change constitutes a fundamental shift in finance with potentially revolutionary impact.

If the development of interoperable open-source permissive distributed ledger technologies is coordinated and made to follow mutual standards, this could potentially open up opportunities for all market participants to imagine, innovate and build, not only new financial instruments, but entirely new systems with a redefined concept of trust. It is the start of a replumbing process of the financial system which may reinvent it altogether. That is the prize and the journey—all made possible through the Nakamoto invention of the blockchain.

Chapter 7

Future work

At the time of writing, blockchain technology is still in its experimental stages. As the technology and the industry matures, several concepts will need to be revisited. For future work in on these topics, we suggest the following areas to be explored in more detail:

- The legal feasibility of the design outlined in this report with regard to regulation policies
- A thorough analysis of which parts of the data in a securities blockchain platform need to be confidential and which parts can be allowed to be transparent and how this affects interoperability
- A schematic which defines how the different levels of granular access controls should be defined in a permissioned securities blockchain platform
- A study of how corporate actions such as voting and stock splits can be facilitated by a second-generation securities blockchain platform
- The applicability of smart contracts in more financial scenarios than what is covered by this paper

References

[1] O. Wyman, Santander InnoVentures, The Fintech 2.0 Paper: Rebooting Financial Services, (2015) [Accessed 2016-06-17]. [Online]. Available: http://santanderinnoventures.com/wp-content/uploads/2015/06/The-Fintech-2-0-Paper.pdf

[2] Bitcoin Wiki, Technical background of version 1 Bitcoin addresses, [Accessed 2016-06-17]. [Online]. Available: https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses

[3] S. Gilbert and N. Lynch, ACM SIGACT News, Vol. 33 (2), Brewer's conjecture and the feasibility of consistent, available, partition-tolerant, web services, (2002)

[4] J. A. Garay and A. Kiayias, The Bitcoin Backbone Protocol: Analysis and Applications, (2016) [Accessed 2016-06-14]. [Online]. Available: https://eprint.iacr.org/2014/765.pdf

[5] Library of Congress, Regulation of Bitcoin in Selected Jurisdictions, [Accessed 2016-06-17]. [Online]. Available: http://www.loc.gov/law/help/bitcoin-survey/

[6] G. Zyskind, O. Nathan, and A. Pentland, Massachusetts Institute of Technology, Enigma: Decentralized Computation Platform with Guaranteed Privacy, [Accessed 2016-06-17]. [Online]. Available: http://enigma.media.mit.edu/enigma_full.pdf

[7] G. Maxwell, Confidential Transactions, [Accessed 2016-06-17]. [Online]. Available: https://people.xiph.org/~greg/confidential_values.txt

[8] A. Miller and J. J. LaViola, Anonymous Byzantine Consensus from Moderately-Hard Puzzles: A Model for Bitcoin, (2014) [Accessed 2016-06-17]. [Online]. Available: https://socrates1024.s3.amazonaws.com/consensus.pdf

[9] L. Lamport, R. Shostak, and M. Pease, SRI International, The Byzantine Generals Problem, (1982) [Accessed 2016-06-17]. [Online]. Available: http://research.microsoft.com/en-us/um/people/lamport/pubs/byz.pdf

[10] G. Maxwell, [bitcoin-dev] Capacity increases for the Bitcoin system, [Accessed 2016-06-14]. [Online]. Available: https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2015-December/011865.html

[11] R. Merkle, Advances in Cryptology — CRYPTO '87, A Digital Signature Based on a Conventional Encryption Function, (1987)

[12] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, (2008) [Accessed 2016-06-17]. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[13] blockchain.info, Blockchain Size, [Accessed 2016-06-17]. [Online]. Available: https://blockchain.info/charts/blocks-size

[14] J. Poon and T. Dryja, The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments, (2016) [Accessed 2016-06-17]. [Online]. Available: http://lightning.network/lightning-network-paper.pdf

[15] blockchain.info, Hash Rate, [Accessed 2016-06-17]. [Online]. Available: https://blockchain.info/charts/hash-rate

[16] Central Intelligence Agency, The World Factbook, [Accessed 2016-06-17]. [Online]. Available: https://www.cia.gov/library/publications/the-world-factbook/geos/jm.html

[17] S. King, Primecoin: Cryptocurrency with Prime Number Proof-of-Work, (2013) [Accessed 2016-06-17]. [Online]. Available: http://primecoin.io/bin/primecoin-paper.pdf

[18] P. Sztorc, Nothing is Cheaper than Proof of Work, (2015) [Accessed 2016-06-17]. [Online]. Available: http://www.truthcoin.info/blog/pow-cheapest/

[19] BitFury Group, Proof of Stake versus Proof of Work, (2015) [Accessed 2016-06-17]. [Online]. Available: http://bitfury.com/content/5-white-papers-research/pos-vs-pow-1.0.2.pdf

[20] V. Buterin, Proof of Stake: How I Learned to Love Weak Subjectivity, (2014) [Accessed 2016-06-17]. [Online]. Available: https://blog.ethereum.org/2014/11/25/proof-stake-learned-love-weak-subjectivity/

[21] A. Poelstra, On Stake and Consensus, (2015) [Accessed 2016-06-17]. [Online]. Available: https://download.wpsoftware.net/bitcoin/pos.pdf

[22] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake, (2014) [Accessed 2016-06-17]. [Online]. Available: https://eprint.iacr.org/2014/452.pdf

[23] D. Schwartz, N. Youngs, and A. Britto, Ripple Labs Inc, The Ripple Protocol Consensus Algorithm, (2014) [Accessed 2016-06-17]. [Online]. Available: https://ripple.com/consensus-whitepaper/

[24] B. Gehring, The Ripple Ledger Consensus Process, (2015) [Accessed 2016-06-17]. [Online]. Available: https://ripple.com/knowledge_center/the-ripple-ledger-consensus-process/

[25] D. Ongaro and J. Ousterhout, Stanford University, In Search of an Understandable Consensus Algorithm, (2014) [Accessed 2016-06-17]. [Online]. Available: https://raft.github.io/raft.pdf

[26] C. Dwork, N. Lynch, and L. Stockmeyer, Consensus in the Presence of Partial Synchrony, (1988) [Accessed 2016-06-17]. [Online]. Available: http://groups.csail.mit.edu/tds/papers/Lynch/jacm88.pdf

[27] V. Shoup, IBM Zurich Research Lab, Practical Threshold Signatures, (2000) [Accessed 2016-06-17]. [Online]. Available: http://www.iacr.org/archive/eurocrypt2000/1807/18070209-new.pdf

[28] G. Bracha, Asynchronous Byzantine agreement protocols, (1987) [Accessed 2016-06-17]. [Online]. Available: http://www.iacr.org/archive/eurocrypt2000/1807/18070209-new.pdf

[29] A. Pinna and W. Ruttenberg, European Central Bank, Distributed ledger technologies in securities post-trading, (2016) [Accessed 2016-06-17]. [Online]. Available: https://www.ecb.europa.eu/pub/pdf/scpops/ecbop172.en.pdf

[30] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, Ed25519: high-speed high-security signatures, (2011) [Accessed 2016-06-17]. [Online]. Available: http://ed25519.cr.yp.to/

[31] S. Thomas and E. Schwartz, Smart Oracles: A Simple, Powerful Approach to Smart Contracts, (2014) [Accessed 2016-06-17]. [Online]. Available: https://github.com/codius/codius/wiki/Smart-Oracles:-A-Simple,-Powerful-Approach-to-Smart-Contracts

[32] C. Metz, SEC Approves Plan to Issue Stock Via Bitcoin's Blockchain, (2015) [Accessed 2016-06-17]. [Online]. Available: http://www.wired.com/2015/12/sec-approves-plan-to-issue-company-stock-via-the-bitcoin-blockchain/

[33] P. Snow, B. Deery, J. Lu, D. Johnston, and P. Kirby, Factom: Business Processes Secured by Immutable Audit Trails on the Blockchain, (2014) [Accessed 2016-06-17]. [Online]. Available: http://www.wired.com/2015/12/sec-approves-plan-to-issue-company-stock-via-the-bitcoin-blockchain/

[34] V. Buterin, On Public and Private Blockchains, (2015) [Accessed 2016-06-17]. [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/

[35] BitFury_Group and J. Garzik, Public versus Private Blockchains, (2015) [Accessed 2016-06-17]. [Online]. Available: http://bitfury.com/content/5-white-papers-research/public-vs-private-pt1-1.pdf

[36] V. Buterin, DEVCON1: Scalable Blockchains & Asynchronous Programming, (2015) [Accessed 2016-06-17]. [Online]. Available: https://www.youtube.com/watch?v=-QIt3mKLIYU

[37] R. Creighton, Domus Tower Blockchain, (2016) [Accessed 2016-06-17]. [Online]. Available: http://domustower.com/domus-tower-blockchain-latest.pdf

[38] BitFury Group, Block Size Increase, (2016) [Accessed 2016-06-17]. [Online]. Available: http://bitfury.com/content/5-white-papers-research/block-size-1.1.1.pdf

[39] A. Greenberg, Forbes, Visa, MasterCard Move To Choke WikiLeaks, (2010) [Accessed 2016-06-17]. [Online]. Available: http://www.forbes.com/sites/andygreenberg/2010/12/07/visa-mastercard-move-to-choke-wikileaks/#1588b9bb4bc2

[40] A. Back *et al.*, Blockstream, Enabling Blockchain Innovations with Pegged Sidechains, (2014) [Accessed 2016-06-17]. [Online]. Available: https://blockstream.com/sidechains.pdf

[41] A. Hill, Introducing Liquid: Bitcoin's First Production Sidechain, (2015) [Accessed 2016-06-17]. [Online]. Available: https://blockstream.com/2015/10/12/introducing-liquid/

[42] V. Buterin and M. Rosenfeld, Colored Coins whitepaper, [Accessed 2016-06-17]. [Online]. Available: https://github.com/Colored-Coins/Colored-Coins-Protocol-Specification

[43] L. Recupero and W. Briganti, Nasdaq, Nasdaq Launches Enterprise-Wide Blockchain Technology Initiative, (2015) [Accessed 2016-06-17]. [Online]. Available: http://www.nasdaqomx.com/newsroom/pressreleases/pressrelease?messageId=1361706&displayLanguage=en

[44] R. Wells and E. Ditmite, Nasdaq, Nasdaq and Chain to partner on blockchain technology initiative, (2015) [Accessed 2016-06-17]. [Online]. Available: http://ir.nasdaq.com/releasedetail.cfm?releaseid=919287

[45] QuantaBytes, A Survey of Bitcoin Transaction Types, (2014) [Accessed 2016-06-17]. [Online]. Available: http://www.quantabytes.com/articles/a-survey-of-bitcoin-transaction-types

[46] Bitcoin Wiki, Script, [Accessed 2016-06-17]. [Online]. Available: https://en.bitcoin.it/wiki/Script

[47] D. Chan, F. Fontan, S. Rosati, and D. Russo, European Central Bank, The Securities Custody Industry, (2007) [Accessed 2016-06-17]. [Online]. Available: https://www.ecb.europa.eu/pub/pdf/scpops/ecbocp68.pdf?5ff757225862fdd1894d8dab08815b19

[48] Bank for International Settlements, Principles for financial market infrastructures, (2012) [Accessed 2016-06-17]. [Online]. Available: http://www.bis.org/cpmi/publ/d101a.pdf

[49] European Central Bank, History of T2S, [Accessed 2016-06-17]. [Online]. Available: http://www.ecb.europa.eu/paym/t2s/progplan/history/html/index.en.html

[50] DTCC, Embracing Disruption: Tapping the Potential of Distributed Ledgers to Improve the Post-Trade Landscape, (2016) [Accessed 2016-06-17]. [Online]. Available: http://www.dtcc.com/~/media/Files/PDFs/DTCC-Embracing-Disruption.pdf

[51] Euroclear and Oliver Wyman, Blockchain in Capital Markets: The Prize and the Journey, (2016) [Accessed 2016-06-17]. [Online]. Available: https://www.euroclear.com/dam/Brochures/BlockchainInCapitalMarkets-ThePrizeAndTheJourney.pdf

[52] N. Tiernan, Re: Alt chains and atomic transfers, (2013) [Accessed 2016-06-17]. [Online]. Available: https://bitcointalk.org/index.php?topic=193281.msg2224949#msg2224949

[53] G. Greenspan, Delivery versus payment on a blockchain, (2015) [Accessed 2016-06-17]. [Online]. Available: http://www.multichain.com/blog/2015/09/delivery-versus-payment-blockchain/

[54] Bitcoin Wiki, OP_CHECKSIG, [Accessed 2016-06-17]. [Online]. Available: https://en.bitcoin.it/wiki/OP_CHECKSIG

[55] SETL, SETL Launches Blockchain-Powered OpenCSD Platform, [Accessed 2016-06-17]. [Online]. Available: https://setl.io/

[56] J. Southurst, Mycelium Wallet to Sell Shares and Release 'Radical Upgrade' This Year, (2016) [Accessed 2016-06-17]. [Online]. Available: https://news.bitcoin.com/mycelium-shares-release-radical-upgrade/

[57] U.S. Securities and Exchange Commission, SEC Approves Overstock.com S-3 Filing to Issue Shares Using Bitcoin Blockchain, (2015) [Accessed 2016-06-17]. [Online]. Available: https://www.sec.gov/Archives/edgar/data/1130713/000110465915084240/a15-9206_9fwp.htm

[58] R. G. Brown, R3, Introducing R3 Corda$^{\text{TM}}$: A Distributed Ledger Designed for Financial Services, (2016) [Accessed 2016-06-17]. [Online]. Available: http://r3cev.com/blog/2016/4/4/introducing-r3-corda-a-distributed-ledger-designed-for-financial-services

[59] N. Szabo, The Idea of Smart Contracts, [Accessed 2016-06-08]. [Online]. Available: http://szabo.best.vwh.net/smart_contracts_idea.html

[60] Oraclize, Oraclize documentation, [Accessed 2016-06-08]. [Online]. Available: http://docs.oraclize.it/

[61] D. Smith, A. Gibson, and Oakpacific, TLSnotary - a mechanism for independently audited https sessions, (2014) [Accessed 2016-06-17]. [Online]. Available: https://tlsnotary.org/TLSNotary.pdf

[62] P. Sztorc, Truthcoin, Peer-to-Peer Oracle System and Prediction Marketplace, (2015) [Accessed 2016-06-17]. [Online]. Available: http://bitcoinhivemind.com/papers/truthcoin-whitepaper.pdf

[63] T. Shyr, Cash settlement, [Accessed 2016-06-17]. [Online]. Available: http://www.wikinvest.com/wiki/Cash_settlement

[64] D. Shrier, D. Sharma, and A. Pentland, Massachusetts Institute of Technology, Blockchain & Financial Services: The Fifth Horizon of Networked Innovation, (2016) [Accessed 2016-06-17]. [Online]. Available: http://www.getsmarter.ac/uploads/gs_file/file_document/129/mit_fintech_white_paper_excerpt.pdf

[65] J. Kelly, Reuters, Nine of world's biggest banks join to form blockchain partnership, (2015) [Accessed 2016-06-14]. [Online]. Available: http://www.reuters.com/article/us-banks-blockchain-idUSKCN0RF24M20150915

[66] L. Shin, Microsoft Partners With Blockchain Consortium R3, (2016). [Online]. Available: http://www.forbes.com/sites/laurashin/2016/04/05/microsoft-partners-with-blockchain-consortium-r3/#17d6e0a2158c

[67] K. S. Nash, Wall Street Journal, Key Blockchain Vendors, Cloud Providers Square Off in Major Test, (2016) [Accessed 2016-06-14]. [Online]. Available: http://blogs.wsj.com/cio/2016/03/02/key-blockchain-vendors-cloud-providers-square-off-in-major-test/

[68] DTCC, DTCC and Digital Asset to Develop Distributed Ledger Solution to Drive Improvements in Repo Clearing, (2016) [Accessed 2016-06-14]. [Online]. Available: http://www.dtcc.com/news/2016/march/29/dtcc-and-digital-asset-to-develop-distributed-ledger-solution

[69] Digital Assets Holdings, ASX Selects Digital Asset to Develop Distributed Ledger Solutions for the Australian Equity Market, (2016) [Accessed 2016-06-14]. [Online]. Available: https://digitalasset.com/press/asx-selects-digital-asset.html

[70] P. Rizzo, Coindesk, Post-Trade Distributed Ledger Group Grows to 37 Members, (2016) [Accessed 2016-06-14]. [Online]. Available: http://www.coindesk.com/ptdl-group-37-members-post-trade-ledgers/

[71] M. Long, Santander Becomes the First U.K. Bank to Use Ripple for Cross-Border Payments, (2016). [Online]. Available: https://ripple.com/insights/santander-becomes-first-uk-bank-use-ripple-cross-border-payments/

[72] Official Journal of the European Union, Regulation (EU) No 909/2014 of the of the European Parliament and of the Council of the European Union on improving securities settlement in the European Union and on Central Securities Depositories and amending Directives 98/26/EC and 2014/65/EU and Regulation (EU) No 236/2012, (2014) [Accessed 2016-06-14]. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32014R0909&from=EN

[73] UK Government Chief Scientific Adviser, Distributed Ledger Technology: beyond block chain, (2016) [Accessed 2016-06-17]. [Online]. Available: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf

[74] D. Palmer, Coindesk, UK Financial Regulator Vows to Give Blockchain 'Space' to Grow, (2016) [Accessed 2016-06-17]. [Online]. Available: http://www.coindesk.com/uk-financial-regulator-blockchain-space-grow/

[75] D. Godin, You're infected—if you want to see your data again, pay us $300 in Bitcoins, (2013) [Accessed 2016-06-17]. [Online]. Available: http://arstechnica.com/security/2013/10/youre-infected-if-you-want-to-see-your-data-again-pay-us-300-in-bitcoins/

[76] GitHub, Bitcoin Core, [Accessed 2016-06-17]. [Online]. Available: https://github.com/bitcoin/bitcoin

[77] G. Danezis and S. Meiklejohn, University College London, Centrally Banked Cryptocurrencies, (2016) [Accessed 2016-06-17]. [Online]. Available: http://www0.cs.ucl.ac.uk/staff/G.Danezis/papers/ndss16currencies.pdf

[78] S. Ivanov, Waves whitepaper, [Accessed 2016-06-17]. [Online]. Available: https://wavesplatform.com/whitepaper_v0.pdf

[79] DTCC, DTC's Settlement Service for equity, corporate debt and municipal debt securities transactions consolidates and facilitates end-of-day net funds settlement of a participant's net debits and credits resulting from various intraday activities, including institutional trading activity, stock loans, etc., [Accessed 2016-06-14]. [Online]. Available: http://www.dtcc.com/matching-settlement-and-asset-services/settlement/equity-corporate-debt

[80] Smartbit, Transactions per second, [Accessed 2016-06-17]. [Online]. Available: https://www.smartbit.com.au/charts/transactions-per-second

[81] Bitnodes, Snapshot of reachable nodes as of Tue Jun 14 2016 15:02:38 GMT+0200 (W. Europe Daylight Time), [Accessed 2016-06-14]. [Online]. Available: https://bitnodes.21.co/nodes/

[82] Bitcoin Core, Running A Full Node, [Accessed 2016-06-14]. [Online]. Available: https://bitcoin.org/en/full-node#costs-and-warnings

[83] Raspberry Pi Foundation, Raspberry Pi 2 Model B, [Accessed 2016-06-14]. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/

[84] R. Mccone, Build Your Own Raspberry Pi Bitcoin Full Node [Bitcoin Core], (2015) [Accessed 2016-06-14]. [Online]. Available: http://raspnode.com/diyBitcoin.html#networking

[85] P. Vagata and K. Wilfong, Facebook, Scaling the Facebook data warehouse to 300 PB, [Accessed 2016-06-14]. [Online]. Available: https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/

[86] A. Andreyev, Introducing data center fabric, the next-generation Facebook data center network. [Online]. Available: https://code.facebook.com/posts/360346274145943/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/

[87] E. Panayi and G. W. Peters, Understanding Modern Banking Ledgers through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money, (2015) [Accessed 2016-06-17]. [Online]. Available: http://arxiv.org/pdf/1511.05740v1.pdf

[88] GitHub, Sidechain Elements, [Accessed 2016-06-17]. [Online]. Available: https://github.com/ElementsProject/elements

[89] A. Taaki, Bitcoin Improvement Proposal, [Accessed 2016-06-17]. [Online]. Available: https://en.bitcoin.it/wiki/BIP_0001

[90] M. del Castillo, SWIFT: Blockchain Won't Remove All Third Parties in Securities Trade, (2016) [Accessed 2016-06-17]. [Online]. Available: http://www.coindesk.com/swift-research-blockchain-third-parties/

[91] J. Peterson and J. Krug, Augur: a Decentralized, Open-Source Platform for Prediction Markets, (2015) [Accessed 2016-06-17]. [Online]. Available: http://augur.link/augur.pdf

[92] E. G. Sirer, Bitcoin Guarantees Strong, not Eventual, Consistency, (2016) [Accessed 2016-06-14]. [Online]. Available: http://hackingdistributed.com/2016/03/01/bitcoin-guarantees-strong-not-eventual-consistency/

Series of Master’s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2016-530

http://www.eit.lth.se