DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2017

Decentralized Transactions in a Centralized Environment

A Blockchain Study Within the Transport Industry

MARCUS LÖF

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION

Decentralized Transactions in a Centralized Environment

A Blockchain Study Within the Transport Industry

MARCUS LÖF

Degree Programme in Computer Science and Engineering & Master in Computer Science
Date: May 26, 2017
Supervisor: Sonja Buchegger
Examiner: Mads Dam
Swedish title: Decentraliserade transaktioner i en centraliserad omgivning - En blockchainstudie inom transportindustrin
School of Computer Science and Communication

Abstract

The blockchain technology constitutes a domain where significant research is done. The technology revolutionized the world through the cryptocurrency *Bitcoin*, and since then new applications of the technology have emerged. One of the applications is to represent real assets as digital assets on a blockchain, so called *smart-property*. In this thesis a smart-property solution is utilized to address creditworthiness issues within the transport industry. A *Proof-of-Concept* (PoC) is implemented using smart-property through *colored coins* on *Bitcoin's* blockchain. To conclude the usefulness of the solution, two alternative solutions are proposed for comparisons. Requirements for a solution to the problem are specified, and the solutions are evaluated against them. Thus the thesis investigates and compares the solutions' abilities to address the creditworthiness problem motivating the thesis. The evaluation aspects constitute of: confidentiality, integrity, availability, consistency, immutability, response time, cost, customer usefulness, trust and environmental issues. It is concluded that a smart-property solution is adequate for the problem. The solution however got inadequacies, mainly regarding confidentiality, but that is concluded not to affect the problem domain.

Sammanfattning

Blockchainteknologin utgör ett område där mycket forskning utförs. Teknologin revolutionerade världen genom kryptovalutan *Bitcoin*, och sedan dess har nya applikationer av teknologin växt fram. En av applikationerna är att representera verkliga tillgångar som digitala tillgångar på en blockchain, så kallad *smart-property*. I denna uppsats används smart-property för att lösa kreditvärdighetsproblem som finns inom transportindustrin. En implementation för att påvisa konceptets ändamålsenlighet utförs där smart-property används genom *colored coins* på Bitcoins blockchain. För att kunna bedöma användbarheten hos lösningen, föreslås även två alternativa lösningar för jämförelse. Krav för en lösning på problemet specificeras i uppsatsen och lösningarna evalueras mot dessa. Alltså undersöker och jämför denna uppsats lösningarnas förmåga att lösa kreditvärdighetsproblemet som motiverar denna uppsats. Aspekterna för evaluering utgörs av: konfidentialitet, integritet, tillgänglighet, konsistens, oförändlighet, responstid, kundnytta, pålitlighet och miljöpåverkan. Slutsatsen som dras är att en lösning baserad på smart-property är adekvat för problemet. Lösningen har dock brister, främst vad gäller konfidentialitet, som däremot inte påverkar problemdomänen.

Preface

The thesis is my last work in order to take my degrees *Master of Computer Science* and *Degree Programme in Computer Science and Engineering* (Civilingenjörsutbildning i datateknik) from *KTH*. The *blockchain technology* has been an extremely interesting field to work within, and I have exclusively been inspired by the work.

I believe the technology is beneficial when parties that do not trust each other want to keep a common database without a trusted central party. It is particularly useful when ownership is to be transacted between the untrusted entities. I think blockchain is favorable if the data benefits from a chronological timeline and is allowed to be public to all participants of the blockchain. I also believe the technology is overhyped, it is important to understand the limited domain where the technology adds value. I would like to claim that the blockchain technology is not beneficial for many of the usage areas proposed in the blockchain sphere. My feeling is that the general audience is amazed with what the blockchain technology provided through Bitcoin, and consequently tries to throw the technology at non-existing problems to become equally successful. This may be useful for research purposes, but the industry should be careful and ask themselves if blockchain technology really is what they need for their use case.

A key aspect for public blockchains is being able to fully secure them without the waste of power required for *Proof-of-Work* and to improve the throughput scalability significantly which is hoped to be done through *Lightning Network*. To work in a field where new material was constantly released during the time of the thesis work was inspiring, and I am thankful this is the area I did my thesis in.

Moreover I would like to thank *Scania* for giving me the chance to carry out the study, and in particular my industry supervisor *Sussi Miller-Tiedemann* and team leader *Karin Avatare*.

Lastly I would like to point out that another master thesis student, Jim Lindberg from Uppsala University, worked regarding the blockchain technology for Scania as well during the time of my thesis. I am grateful I had a competent person to discuss blockchain related ideas with.

Contents

Chapter 1

Introduction

This chapter introduces the topic of the thesis and research question along with the motivation for its relevance. The objectives and delimitations of this work are moreover presented.

Blockchain is a new technology mostly associated with Bitcoin [1], and is the underlying technology of the famous cryptocurrency. The creator of Bitcoin under the pseudonym *Satoshi Nakamoto* released a paper explaining the cryptocurrency in 2008, however Bitcoin was in practice deployed in 2009. The mechanisms in Bitcoin has since been examined and concepts have been identified, in particular the blockchain which is not explicitly mentioned in the original paper [2]. Hence Nakamoto may have unintentionally invented a technology that can be applied in various other areas. The unique invention was to handle and transact value among a network of untrusted entities without a trusted intermediary [3], however several other use cases for the technology have emerged and some predict a paradigm shift in computer science.

One of the arisen use cases is storage of arbitrary real assets as digital resources on a blockchain. A central party issues that digital resource X on the blockchain is representing the real asset Y, however the asset storage and the transactions occur solely on the blockchain. This is referred to as *smart-property* [4, 5]. Hence the scenario differs from the initial completely decentralized system that Bitcoin provides since one relies on the central party to maintain the value of the coupling between the real and the digital asset. Particularly the system is centralized utilizing the decentralized transactions and storage of the blockchain.

In the transport industry creditworthiness is an existing problem. Vehicles may suddenly need reparation due to various causes. The workshops repairing the vehicles desire real-time payments while the vehicles and their drivers may not be able to provide it. This is particularly problematic for foreign hauliers, where one easily cannot conclude the creditworthiness of the counterparty [6, 7, 8, 9, 10]. A bank transaction is time consuming, especially when transacting abroad since the time is dependant on the participating countries and banks in a transaction. For instance *ASB* [11] estimate their international transaction time to two bank days while *Lloyds Bank* [12] claim four days at most for their international transactions, however neither can ensure that a transaction will not require vastly longer time. Moreover transaction fees may be high in case of international

transactions. Furthermore it is problematic to transact money during non-business hours, for instance at night which is a problem since vehicles may need service at any time [8]. The creditworthiness problem may also result in that towed vehicles and their goods are kept in custody as a deposit until the payments are actually confirmed on the receivers' accounts. Out-of-hours such as weekends it may imply that a vehicle may be kept until the next bank day or even longer, which is problematic for the hauliers [9, 10].

To solve the above problem regarding payments, payments are currently done through invoice [8, 9] which however raises creditworthiness issues. Credit cards may occasionally be utilized, however generally the hauliers do not trust their drivers with such a card implying the haulier desires an invoice instead [8, 9]. This entails the transaction time problem as highlighted above.

For e.g. a workshop to be sure that they will get paid by invoice if they do reparations on a vehicle, a third party, for instance *Scania Assistance* may give the workshop a *Guarantee of Payment* (GoP). Thus if the payment is not performed by the haulier, the third party guarantees the payment. However the third party only guarantees it given that a GoP is in turn issued by the *home detailer* corresponding to the vehicle [8]. The home detailer uses its customer relations and credit checkups to conclude if the party is creditworthy so that a GoP can be issued, however a GoP can be denied [10]. The parties issuing the GoPs are however not legally entitled to pay, there are on the contrary mutual business relationships that are damaged [8]. Thus there are creditworthiness questions. Moreover it is prolix to make all the required credit checkups on international hauliers and home detailers. Parties are thus at risk and a more secure system is desired while still enforcing instant payments in order to be useful to the parties.

If a vehicle is loaded with digital assets, it facilitates direct payment using those assets. Thus the vehicle may ensure creditworthiness by holding assets in a digital wallet. Particularly it may be useful when having autonomous vehicles without drivers that need to pay for services or goods. Moreover it is useful if the driver does not have any payment means, which is common, when an incident occur [8, 9]. The vehicle is however present and may hold assets used for payment. The proposal is similar to a stored-value card which is a card that can save monetary value [13], however the difference being it can be loaded and unloaded in real-time without the use of a bank which requires transaction days. Even transactions done within the same bank cannot be done in real-time, and may moreover only be performed on bank days [14] implying a problem when one desires real-time transactions out-of-hours.

Cryptocurrencies without a blockchain existed before the invention of blockchain, however the blockchain technology was the first technology that allowed a decentralized transaction system while still avoiding *double-spending* [15]. Double-spending implies spending the same asset twice, which is examined in more detail in *2.5.6*.

Bank transactions further consist of central entities, the banks, that control and validate the transactions and the balances. However this implies a risk if the security of the central parties is compromised, and a secure decentralized system ought to be created. The above points make blockchain technology adequate for the use case.

1.1 Objective

In this thesis a *Partially Decentralized* (PD) system, a smart-property solution, is proposed for the use case above. A PD system is in this thesis defined as a system where a central party, *Scania*, is a publisher of the digital resource *ScaniaCoin* on a blockchain. The digital asset ScaniaCoin represents the real asset money. Accordingly it is partially centralized since Scania is responsible for issuance and redemption of the ScaniaCoins. On the contrary the ScaniaCoins' storage and transactions occur decentralized on a blockchain. Consequently it is a partially decentralized (PD) solution.

The digital resources represent real assets and can be used to perform real-time trades, and in real-time load vehicles with digital assets. The asset storage and the transactions occur decentralized without the knowledge of Scania, however Scania is responsible for issuance and redemption of the digital assets. A *proof-of-concept* (PoC) of such a system is implemented in this thesis. To ensure the usefulness of the system, alternative solutions must be used for comparison. In this thesis the alternative solutions consist of a *completely centralized* (CC) system and a *completely decentralized* (CD) system. A CC system in this thesis is defined as a central party controlling the entire system, the underlying database system may distributed. No public blockchain is used for the CC system. A CD system in this thesis is defined as directly using the underlying cryptocurrency of a public cryptocurrency blockchain.

1.2 Problem Statement

A PD system needs to be fully examined and analysed from a security and a customer usefulness perspective. In this thesis such a system is suggested and analysed. Actors are expected to only put in a limited amount of real assets into the system as a buffer for emergency cases, consequently the system is holding temporary assets. Except for being secure the system also has to enable efficient transactions. To ensure usability and completely analyse the security, comparisons to alternative solutions of the system is required. In particular a CC and CD system are used for comparison in this thesis.

The problem statement hence becomes: *To what extent does a centralized system with decentralized temporary asset storage and transactions on a blockchain fulfill security, reliability and customer usefulness compared to alternative solutions?*

1.3 Delimitations

The alternative solutions used for comparison to the PD system are limited to a CC system and a CD system. The aspects analysed are limited to the aspects of the CIA triad, consistency, immutability, response time, cost, customer usefulness, trust and environmental issues. The CIA triad concerns confidentiality, integrity and availability of the data and is described in depth in *2.1 CIA Triad*.

Public blockchains currently suffer from throughput issues and real-time transactions are problematic. In this thesis the possibility of real-time transactions and significantly in-

creased throughput is assumed due to *Lightning Network* explained in *2.10*. Moreover no legal aspects of a commercialization of the system proposed in the thesis is taken into account.

For the PoC the PD payment system will be implemented as an application. However the service for exchanging real money for ScaniaCoins is not implemented, see *9 Future Work* for more details.

Services out-of-hours as e.g. reparations are complex to estimate the price for in advance. During work days there exist tools to make approximate estimations, however the mechanics working out-of-hours do not have access to them. This introduces another problem, even if real-time payments can be performed one has to know the amount to transact [8]. The cost estimation problem out-of-hours is beyond the scope for this thesis.

Chapter 2

Background

This chapter presents the essential background material to understand the blockchain and its tools for the latter chapters. Moreover the CIA triad is presented as a foundation to the analysis and aspects provided in chapters 4,5 and 6 of the thesis.

2.1 CIA Triad

Confidentiality, Integrity and Availability (CIA) is the core of information security. Attacks toward these properties are attacks on the information security of the system. Confidentiality implies that only authorized users must be able to access specific information. Integrity entails that information is not altered by unauthorized actors. Availability regards ensuring information accessibility for authorized users [16].

2.2 Blockchain Tools

A blockchain is at its core a database structured as a linked list with elements, called blocks, chained together. Each block consists of a batch of transactions, which are instructions modifying the state of the database. The blockchain technology consists of several important elements that ought to be examined in order to fully define a blockchain. The theory of it is given in this section.

2.2.1 Digital Signature

Asymmetric cryptography utilizes a public/private key-pair and a set of associated operations [17]. The key-pair can be generated, where the public key is distributed to the all participants while the private key is kept secret for each participant. It is infeasible to forge the private key given the public key. To ensure a message originated from a specific participant, a message is signed by the participant with a private key. The signature operation can be defined as in *Formula 2.1*.

$$Signature = Sign(PrivateKey, Message) \tag{2.1}$$

To verify that the message originated from the indented sender, the public key earlier published by the intended sender is used. If the public key corresponds to the private key used to sign the message, one concludes the message must be from the intended sender [18]. The verification operation can be defined as in *Formula 2.2*.

$$Verification = Verify(PublicKey, Message, Signature) \tag{2.2}$$

A few properties follow from the above. Reliability of the authentication is implied since the signature is bound to a signer. Consequently a sender cannot deny sending a specific message providing non-repudiation. The message itself may not be altered in transit since it would invalidate the verification, hence integrity is provided. Moreover the system ensures it is not possible to forge a signature given the knowledge of previously signed messages, thus digital signatures ensures a message origin [19].

2.2.2 Cryptographic Hash Function

A hash function is a function that given arbitrary data produces a string of fixed size, which is called the hash [15]. For a hash function to be secure and be suitable in cryptography a few properties are needed:

1. *One-way computation*: The hash $H(x)$ should be easily computed given x while it should be virtually impossible to extract x given $H(x)$.
2. *Collision free*: It is infeasible to find x and y where $x \neq y$ and $H(x) = H(y)$.
3. *Deterministic*: Given x and a hash function H there is a deterministic hash value $H(x)$ that is unique for the corresponding x.

Regarding property $2)$ collisions exists, however with a good scheme they are extremely rare [20].

2.2.3 Hash Pointer

A hash pointer is both a pointer to the storage location and the cryptographic hash of some data. Given the pointer the data can be retrieved and it is assured that the data is not tampered with [21]. The integrity property is derived from the cryptographic hash function which ensures that $x = y \Rightarrow H(x) = H(y)$. Hence if the hash value of a data element is changed, the content must have changed. A depiction of a hash pointer may be viewed in *Figure 2.1*.

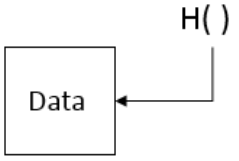

Figure 2.1: *Hash pointer illustration.*

2.2.4 Hash Chain

Linking together several data elements using hash pointers is called a *hash chain* [15]. The head of the chain is a hash pointer. A one-dimensional hash chain is a linked list using hash pointers originating from a genesis element.

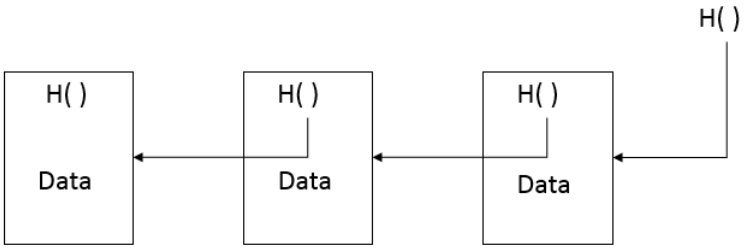

Figure 2.2: *1D Hash chain illustration.*

Since the hash pointer to the previous element is stored within the current element, a change in one element invalidates all elements afterwards in the chain. The hash function is collision free, hence the hash of the tampered element and the value of the hash pointer in the next element will not be equal. Thus one may determine if a chain has been tampered with. The implication is that in order to tamper with an element and keep the chain in a consistent state all subsequent elements must also be modified. Consequently the head of the chain also changes, thus entities with knowledge of the previous head may conclude that the integrity of the chain was violated [21].

2.2.5 Merkle Tree

A binary hash tree is called a *Merkle tree*. It is a tree of hashes, where a non-leaf node in the tree corresponds to the hash of its two children resulting in one root hash of the tree, the *Merkle root*. For a set of data one can create a Merkle tree by partitioning the data. The leaf nodes consist of the corresponding hashes to the data partitions.

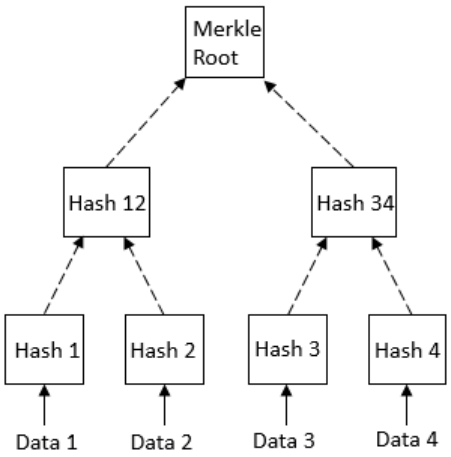

Figure 2.3: *Merkle tree.*

Consequently, in the same manner as with the one-dimensional hash chain it suffices to compare the roots of two valid Merkle trees to conclude if the data differ. The implication of it is that if one stores the corresponding data for each element in *Figure 2.2* beneath a Merkle tree, it suffices to hash the hash pointer to the previous element and the Merkle root for each element. Hence assuming knowledge of the hash chain header in the valid chain, one can download the minified hash chain consisting of elements with only a previous hash and *Merkle root* and verify the validity of the chain [22]. Moreover this strategy allows removal of superfluous data from an element in the chain without breaking it by pruning the actual data but keeping the Merkle tree root intact as in *Figure 2.4*. This saves memory [2] as it allows actors to be *light clients* and only store the relevant data to the actor and not all data on the chain [23], hence a *simplified protocol version* (SPV) [24].

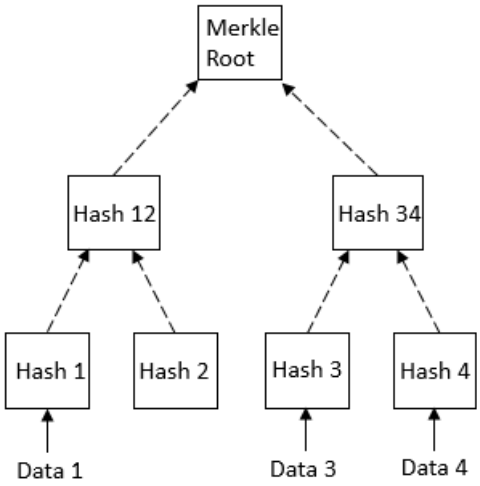

Figure 2.4: *Pruned Merkle tree.*

2.2.6 Transaction

As mentioned briefly a blockchain is a database and a transaction is an instruction modifying the state of the database. Transactions regarding ownership, e.g. *A* sends *2 XYZ* to B, can favorably be modelled on a blockchain. An *unspent transaction output* (UTXO) model may be utilized where each transaction specifies transaction inputs which are unspent transaction outputs [25]. A transaction creates new unspent transaction outputs. An output can only be used as input to a new transaction once and the whole output must be spent. If not the full value of the unspent output is transacted to the counterparty, the rest may be sent back to oneself. Accordingly an asset may only be transacted once for a given unspent output, and an asset may be tracked till its origin. At any point an actor represented by a public key may digitally own the asset [26].

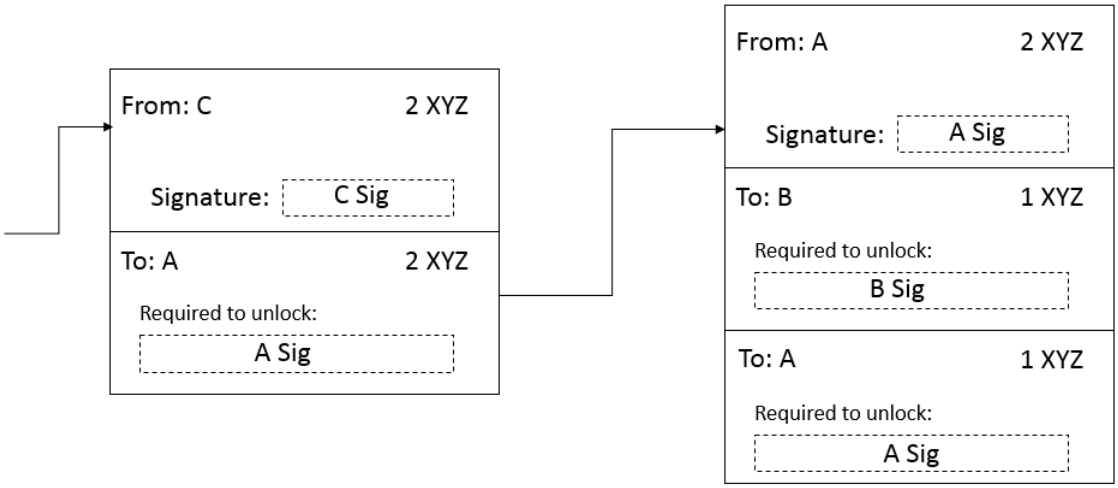

Figure 2.5: *Depiction of a transaction of ownership. The figure depicts how 2 XYZ is sent from C to A followed by A sending 1 of these XYZ to B and 1 XYZ back to A.*

A transaction can be signed using a digital signature and accordingly the one issuing the transaction can be identified in order to ensure validity [2], for instance only being able to spend funds belonging to the spender. To claim the transacted assets, the receiver needs to sign the transaction to prove ownership as can be seen in *Figure 2.5*. The general definition of a transaction can be found in *Appendix A.1*.

2.2.7 Block

A block is a unit which encapsulates a batch of transactions. Blocks may be chained similar to elements of a hash chain, implying a *block chain* (blockchain). Creating a chain of transactions would result in a long chain, to instead make blocks the unit of the hash chain improves efficiency. If many actors are interested in the transactions, one has to share the transactions with all the actors. It is more efficient to announce a block containing several transactions than publishing one transaction at the time [27].

The block may be divided into a *block header* and a *payload*. The block header holds the metadata of the block such as e.g. the hash pointer to the previous block, the Merkle root derived from the payload and a timestamp [22]. The payload is the part storing the actual transaction data, and it may be done beneath a Merkle tree as in *2.2.5*. *Figure 2.6* illustrates a block possible to integrate as a unit in a hash chain.

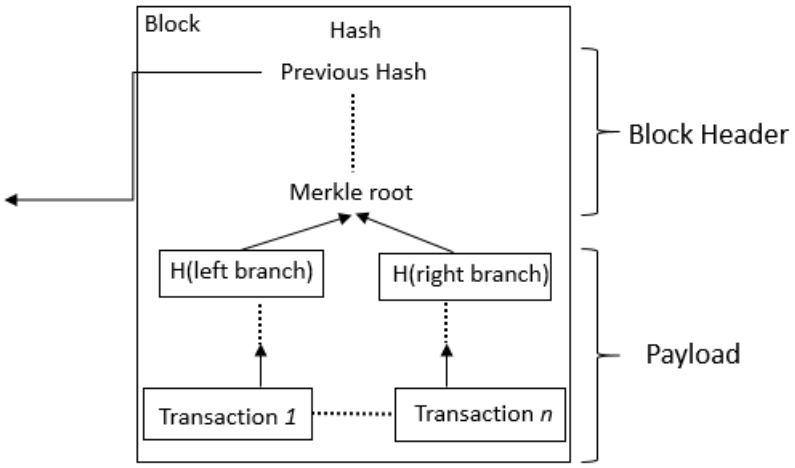

Figure 2.6: *Illustration of a block with Merkle tree structure.*

To increase the efficiency and still verifying the integrity of the chain one may, as mentioned in *2.2.5*, not store all the actual data. Hence the hash of a block is solely the hash of the block header [28]. In case a transaction is tampered with, the Merkle root changes meaning the block hash changes and thus one may determine whether the integrity of the chain has been violated.

2.3 Peer-to-Peer Network

A *peer-to-peer network* is a collection of loosely coupled interacting autonomous nodes. It is decentralized and nodes may join and leave the network unimpeded. If all participating nodes have the same privileges it is called a *pure* peer-to-peer network [29]. Usually resources are shared between the nodes. To join the network only one node has to be known, this node is called a seed node in a blockchain context [30, 31].

2.4 Blockchain Definition

Consensus of the blockchain definition is yet to be established [32], however in this thesis the following definition is proposed:

A blockchain is a database structured as a one-dimensional hash chain of blocks originating from a genesis block. A blockchain may be distributed and maintained by a set of participants of a peer-to-peer network that do not trust each other. A consensus mechanism among the participants is required for them to agree on the state of the database. A blockchain may improve storage efficiency by introducing a data structure to fingerprint the data. A blockchain may utilize digital signatures to ensure changes to data are issued by the adequate identity.

At its core a blockchain is a one-dimensional hash chain. A professor from Princeton University strengthens this by claiming a blockchain is a linked list implemented with hash pointers [21], hence a one-dimensional hash chain.

Some claim a blockchain is a database distributed in a peer-to-peer network, for instance Abeyratne and Monfared [33]. The distributed property should however be considered a neat application of the blockchain database, not a requirement for a blockchain. Thus a blockchain can be applied in a distributed context, this property is what makes blockchain powerful and consequently blockchain is often referred to as a distributed database. Specifically a blockchain may be a decentralized distributed database, that can be managed without the participants trusting each other [34]. With a distributed database a consensus mechanism is required to ensure the same version on all sites, this accordingly applies to a blockchain as well.

The database relies on a structure of a chain, meaning a vast usage of it results in a long chain consuming memory. A hash pointer is stored in each block pointing to the previous block, meaning one cannot change data in the previous block without invalidating the pointer in the next block as stated in *2.2.4*. Thus removal of unnecessary data invalidates the blockchain. To let participants of the blockchain store a valid copy of the blockchain but only data relevant to the participant a data structure to fingerprint the transactions is required, which may be solved by a Merkle tree [2]. Consequently storage efficiency for participants of the blockchain using a data structure to fingerprint the data is not a requirement for a blockchain, but is a tool that increases the usefulness of the blockchain.

A blockchain may utilize digital signatures to ensure the origin of issued database transactions, accordingly a blockchain may couple data to an owner. This is however also only a valuable tool for a blockchain, not a requirement. A valid transaction must preserve consistency, which may require that transactions must be issued by the adequate identity corresponding to the altered data and thus digital signatures may be utilized. In a monetary system this would correspond to only being able to spend one's own money [35].

2.5 Distributed Consensus

A blockchain as a distributed database on a peer-to-peer network introduces the problem of having consensus between the nodes without a central authority. Nodes may have different versions of the same blockchain database, hence it must be agreed among the nodes which version is the *true* one.

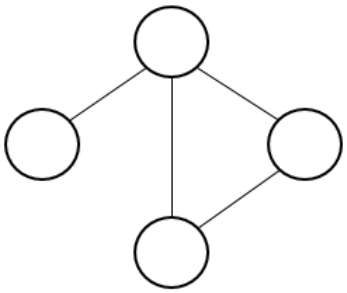

Figure 2.7: *Peer-to-peer network with nodes connectected via arbitrary constellations of edges.*

2.5.1 Transaction and Block Propagation

When a transaction is issued on a public blockchain network the nodes propagate the transaction over the network using the *gossip protocol*, meaning a node propagates the transaction to all its connected peers [23, 29]. Not all peers in the network are connected to each other as illustrated in *Figure 2.7*, meaning a transaction might have to be broadcasted in several steps. Each node maintains a list of all transactions it has been notified of but not yet in the blockchain, a *memory pool* [36]. All nodes may examine every transaction. Knowledge of previously issued transactions prevents transaction propagation running in infinite loops by nodes not propagating transactions previously notified of. The nodes compare the hashes of the transactions, since the hash function is collision free equal hash value implies the same transaction. A node may further avoid propagating certain transactions if the transactions conflict with the consistency of the database. For instance two transactions contradictory to each other may be issued simultaneously, when a node is notified of one of the transactions it later rejects the other one [31]. Consequently all nodes must be aware of everything on the blockchain, implying transparency [37]. However the actors on the network are only identified by their, hashed [38], public keys [39].

In case of contradictory transactions, there is a race condition. What a node regards as the correct transaction depends on its network position as illustrated in *Figure 2.8*.

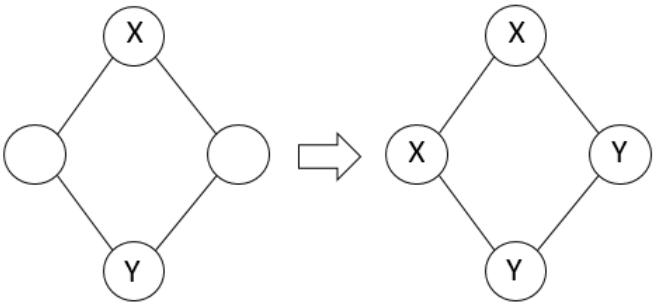

Figure 2.8: *Illustration of the race condition for the transactions. A transaction X is broadcasted by one node while the contradictory transaction Y is broadcasted by another node resulting in a split opinion. To the left: Initial state with two contradictory transactions. To the right: Depiction of transaction propagation of contradictory transactions.*

Which of the transactions makes it into the blockchain is decided by the opinion of the node creating the next block. Blocks are created, *mined*, by block creating nodes, *miners*, out of batches of transactions from the miners' *memory pools*. Thus when a block is created which includes one of the two transactions, the contradictory nodes drop their transaction since it will not finish on the blockchain [31].

A miner may decide which block in the blockchain known by the miner to build the new block onto, even if the precept is to choose the chain with the most work behind. Often this is roughly equivalent to the longest chain known to the miner [31]. The created blocks are propagated over the peer-to-peer network. Other honest nodes forward a block if all transactions are valid and the block is building on the chain with the most

work behind, from their perspective [31]. For a transaction to be valid it may not violate the consistency of the database, e.g. it is not contradictory to another transaction. Moreover a blockchain may enforce that in order for the transaction to be valid, it has to be digitally signed by the identity issuing the transaction and concern data the identity is allowed to alter. The nodes validating the transactions and blocks are called *verification nodes*.

The latency of the network may cause a race condition for block propagation similar to the transaction race condition for transactions shown in *Figure 2.8*. If two different blocks are published simultaneously a fork in the distributed consensus is created, the nodes do not agree on one blockchain [40]. All nodes keep their own version of the blockchain as depicted in *Figure 2.9* meaning the general state of the network could look like as in *Figure 2.10*.

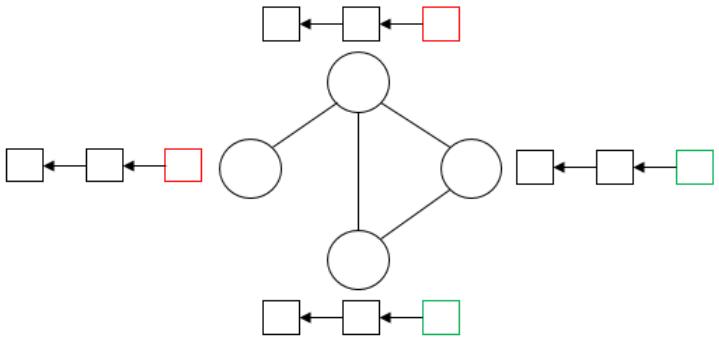

Figure 2.9: *Illustration of the network and their respective versions of the blockchain.*

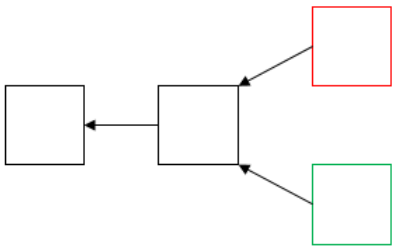

Figure 2.10: *Illustration of a fork in the distributed blockchain.*

When the next block is mined one of the chains may be extended, as depicted in *Figure 2.11*, implying the work behind that chain might exceed the work behind the other one. Hence the network can reach a consensus on which blockchain to continue on.

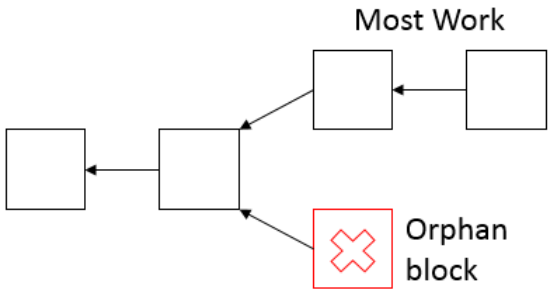

Figure 2.11: *Illustration of a settled fork in the distributed blockchain.*

The blocks on the discarded branch are called *orphan blocks* [41], and their transactions are eventually batched into new blocks put onto the prevalent branch [40]. Accordingly the deeper a block is buried in the chain, the higher probability the block will remain in the long term consensus chain. Each block appended to the chain building on a block X gives further confirmation of the block being in the final chain [42]. If no block is building on block X, the block has zero confirmations. With one block building on block X it has one confirmation *ad infinitum* [41]. Note that it is probabilistic, in theory no transaction can be proven to last in the long term chain forever [42].

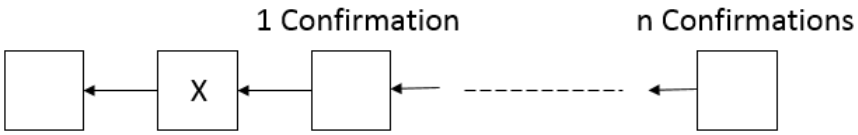

Figure 2.12: *0-n confirmations for a block X.*

2.5.2 Proof-of-Work

The consensus mechanism prescribed extends the branch with the most work behind it as stated in *2.5.1*. The most common type of work is currently *Proof-of-Work* (PoW). The concept implies that each block created proves a significant work was invested in its creation [43, 44]. As mentioned in *2.2.4* modifying one element in the chain requires modification of all subsequent elements to still be a valid chain. Hence it is ensured that dishonest nodes desiring to modify previous blocks are required to work more intensive than honest nodes aiming to extend the current chain with most work behind [43]. The mechanism thus increases the security of the blockchain.

A mathematical puzzle can be used as PoW, where the first miner to find the solution to it may publish the block. The puzzle may be computational heavy implying many computations are required to solve the puzzle and miners with more computational power accordingly have increased chances. The probability of a miner first solving the puzzle ought to be proportional to the miner's proportion of work and contribution, a property called *progress-free*. Consequently blocks are created by miners in proportion to their contribution toward solving the puzzle [45].

The consensus mechanism is accordingly to extend the branch with the most computa-

tions behind it. If blocks are mined with the same puzzle difficulty with the same interval, the branch with the most work behind is equivalent to the longest branch. This is the consensus mechanism which results in a long term consensus chain.

2.5.3 Proof-of-Stake

Proof-of-Stake (PoS) is a consensus mechanism relying on a proof of ownership that may be put at stake [44, 46]. No heavy computational work is required, this is referred to as *virtual mining* [42, 47]. Using Proof-of-Stake the miners mine blocks in proportion to their stake [48] and several algorithms exist. It is roughly equivalent to PoW since instead of using money to usurp hardware in proportion to the miner's wealth to solve the puzzles, the miners mine blocks in proportion to their wealth in the system directly. A miner may mine a block with the probability in proportion to its stake, with the next miner being chosen similarly. If the selected miner does not propose a block in time, the next miner is selected likewise [49]. The consensus mechanism relies on the blockchain with the most work behind it, accordingly the most stake. A comparison between Proof-of-Work and Proof-of-Stake can be found in *Appendix A.5*.

2.5.4 Mining Incentives

The security of the blockchain relies on incentives for the miners to act according to protocol. Consequently to create and validate blocks with appropriate transactions issued by the network and to build onto the branch with the most work behind it. A monetary incentive is possible, where the miners are rewarded for mining blocks ending up in the long term consensus chain [2], while being penalized for mining on blocks that do not. In Proof-of-Work mining on a block not finalizing in the true blockchain results in a cost of the power required for the computations [49], in Proof-of-Stake staking on a block not participating in the final blockchain ought to be penalized through the stake [44, 48]. If such behaviour is not penalized miners may mine on different chains simultaneously and collecting rewards, meaning it might be profitable to also mine on malicious blocks. The problem is called *Nothing-at-Stake*, since the miner does not lose anything by mining on several chains [48, 50]. Consequently it must be penalized to mine on blocks not participating in the final blockchain.

Thus it is profitable to mine on the blocks expected to belong to the true blockchain, and given that the majority is honest it is profitable for a node to also be honest [2, 42]. Moreover for PoS the rationale is that stakeholders are keen of their stake and accordingly got incentives to secure the system [48]. Consequently a monetary incentive exists for miners to follow the prescriptions. Incentives may also be of social or business nature of importance to stakeholders. For instance having a consortium with known participants running a blockchain, incentives exist to be honest in order to keep the social and business relations for successful collaborations.

2.5.5 51% attack

When an actor controls the majority of the mining resources a *51% attack* can be launched. Such an actor with dishonest intentions may outpace the rest of the honest nodes and consequently partly control the blockchain [2]. The attacker may control which blocks and hence transactions to put onto the blockchain [44], since the attacker provides the chain with the most work behind. Consequently the attacker can make it unprofitable for other miners to mine [42]. Furthermore the attacker can suppress specific transactions from joining the blockchain. However if digital signatures for identification is used, the attacker may not issue transactions from other actors in the system since transactions then must be signed by the issuer. The attacker may build an independent chain of blocks with forged transactions longer than the honest nodes' chain, but the honest nodes will not pursue the forged chain since blocks are invalid [51]. A 51% attack requires the majority of the mining resources, however it is possible to partly control the blockchain with less than the majority as presented in detail in *Appendix A.4*.

An actor partly controlling the blockchain, may perform a *Double Spending Attack* [36, 51]. In a blockchain where the transactions are transfers of resources, the attack is to spend the same resource twice. The higher proportion of mining power the higher chance to succeed, a 51% attacker is bound to triumph. A transaction $A \Rightarrow B$ may be issued by the attacker A in order for B to send some goods X to A.

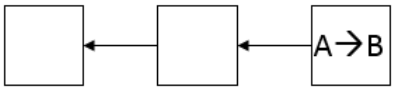

Figure 2.13: *Double Spending Step 1.*

The transaction is put into the blockchain, B verifies the transaction and sends X to A. Because of A controlling the majority of mining power, A can issue $A \Rightarrow A$ instead and create a fork prior to the transaction $A \Rightarrow B$ and yet create the branch with the most work behind. Consequently the final consensus blockchain includes $A \Rightarrow A$, while B is scammed [52] as illustrated in *Figure 2.14* and *Figure 2.15*.

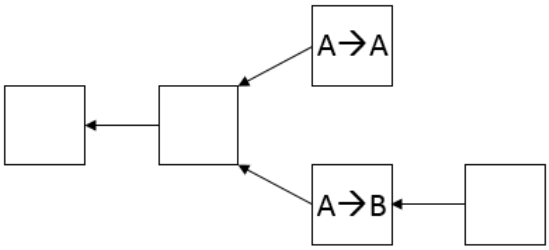

Figure 2.14: *Double Spending Step 2.*

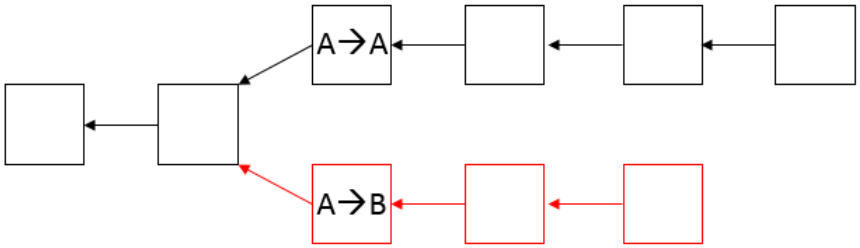

Figure 2.15: *Double Spending Step 3.*

Obtaining such high percent of mining power is however unreasonable in a public decentralized blockchain as may be read in *Appendix A.5*. However lately vulnerabilities to routing attacks have been identified, using the Internet infrastructure to attack Internet based blockchains. Through these attacks the availability may be compromised and chances for double spending attacks are increased. This is done by e.g. managing to partition the network in two and spending the same asset on both partitions, making it problematic when the partitions later are reunited. Thus mining power to perform the attack is not required [53]. Splitting the Internet however ought to be difficult.

2.5.6 Smart Contract

A smart contract is an executable piece of code that may reside on a blockchain, consequently all participants may inspect the script. A smart contract is roughly equivalent to stored procedures in conventional relational databases, however a smart contract resides on the blockchain [54] and is thus not bypassable, while a stored procedure is not necessarily enforced [55, 56]. The smart contract is executed on all the nodes, meaning every node is running a virtual machine. Hence the blockchain may be a distributed virtual machine. Since the code is executed on every node, the contract ought to be well specified and deterministic to avoid inconsistency. The smart contract hence is an autonomous actor with transparent and predictable behaviour [3, 56].

2.5.7 Wallet

A *wallet* in the blockchain context is a set of key-pairs [39, 57]. To transact on a blockchain which enforces identification through digital signatures, one has sign the transactions. Since the network is auditable by everyone participating in the blockchain, as mentioned in *2.5.1*, it makes it problematic in case the blockchain concerns transaction of ownership. As stated in the same section the identities are identified by their public keys corresponding to the private keys used for signing transactions. Thus one may conclude the balance for each public key on the network [37], meaning if the real corresponding identities are found one can conclude their balances. To obstruct the problem, one can generate a new asymmetric key-pair for each issued transaction sending the remaining rest of the transaction input, as in *2.2.6*, to the newly generated address. Moreover the receiver may generate a new key-pair each time receiving [58, 59]. Accordingly an identity owns a set of asymmetric key-pairs and the corresponding balances to the public keys. The set

of keys constitutes a *wallet* [39, 57]. Consequently it complicates inference between real identities and balances [39, 58], implying a semi-transparency.

2.6 Distributed Database

A distributed database is a database system residing on different physical locations communicating over a network. The database may be distributed by either *replicating* or *partitioning* the database. Replicating the database implies storing the entire database at each physical location, while partitioning the database means splitting the logical database into fragments residing on the different physical locations [29, 30]. In a traditional database the data is stored in relations or tables. When having a replicated distributed database, changes in one of the physical databases need to be reflected onto the others. A replicated database is *mutually consistent* if there is a distributed consensus among the databases. A *strong mutual consistency* ensures every replica is identical after each committed transaction, while *weak mutual consistency* implies the replicas eventually will converge to identical values [29]. It is often sufficient to ensure the latter [60].

A transaction originates from a site. In a *master-slave* structure there is one primary authority copy of the database which the rest inherit, transactions are applied to this authority first. The master controls the database and keeps it in a consistent state, which allows a serialized behaviour and hence is a bottle-neck. A *multi-master* structure allows any replica to issue updates. The transaction manager at the originating site of the transaction is the coordinator of the transaction while the rest are the cohorts. Thus the coordinator is responsible for coordinating the replicas to achieve mutual consistency [29, 61].

2.6.1 Commit Protocol

To achieve consistency with a distributed database, a commit protocol is required. An *eager* commit protocol requires all replicas to be updated before the coordinator completes the transaction. The performance is restricted to the network speed and the performance of the slowest node, however it can give atomic commitment in the distributed environment and accordingly mutual consistency [29, 62].

A *lazy* commit protocol only requires one other node to be updated before the coordinator commits and the transaction is further propagated. The number of messages required is thus less than for an eager protocol. The lazy protocol does not ensure mutual consistency, different replicas may be updated with different values contradictory to each other requiring elaborate settlement protocols. However far too many reconciliations implies a system delusion where the inconsistency is infeasible to settle [62]. The reconciliation protocols could depend on timestamps of the transactions, having the transaction with the most recent timestamp deciding. For this to work the clocks need to be synchronized [29, 62].

Paxos protocol family are protocols for consensus in distributed systems. For a distributed database *Two-Phase-Commit*, being a Paxos protocol [63], is a standard commit protocol achieving distributed consensus [64]. The protocol operates in two phases. In the first

phase the coordinator is polling the cohorts on their opinion regarding a transaction: commit or abort. In the second phase the coordinator has collected the cohorts responses and depending on it issues whether to commit or abort the transaction, the participants ought to follow the coordinator. The coordinator may abort the transaction if even one cohort wants to abort. The outcome may also be decided by the majority vote and thus a *voting-based* protocol. It is possible to express the latter with *quorums* having the total number of votes V abort quorum Va and commit quorum Vc as $Va + Vc > V$, where $0 \leq Va, Vc \leq V$. To commit one must obtain at least Vc commit votes, to abort one must obtain at least Va abort votes. That the quorums' sum exceeds the number of votes implies it cannot be decided to commit and abort the same transaction. A local log is held by all participants to continue adequately in case of arbitrary failure [29, 30].

2.6.2 Security Issues

Security complexity of a distributed system is significantly higher than in a conventional. Concurrency control, enforced by commit protocols as in *2.6.1*, is required. Having data on several sites and imposing concurrency control are security issues. Replicated data on various sites may compromise confidentiality since additional sites and their communication need to be protected. Moreover the replicas need to achieve consensus using the communication. Same information might be altered on different sites concurrently which may leave the databases in an inconsistent state. Moreover updates might be applied in the wrong order. Hence security might be affected. Often the transactions include timestamps which enables reaching consensus, as mentioned in *2.6.1*. However sites may run malicious code which tampers with the timestamps and hence compromises security. To address this issue protocols for time consensus is required. Locks may be used to lock resources while altering them, which might result in deadlocks. The security complexity thus arises with a distributed system [60].

2.7 Distributed Database versus Blockchain

A blockchain is a database that may be distributed [65]. In particular a blockchain may be a replicated database, however as mentioned in *2.2.5* a node is not required to store all the data and thus it is possible for nodes to store partitions of the data. A conventional public blockchain is a *multi-master* distributed database, however other variants of blockchains as in *Appendix A.2* are other types of databases. The blockchain can thus be a kind of distributed database [66].

The data structure in which the data is stored is dissimilar to traditional databases. Data is stored as a chain of blocks, which makes it suitable to handle time dependant data [66]. It gives a chronological timeline where one can prove the order of events [37, 39]. It accordingly facilitates version handling and rollbacks to previous valid states by the nature of the data structure, while a conventional database would require to keep an extra record.

The data structure further have implications for the consensus mechanism. Comparing the consensus mechanisms in *2.5* and *2.6.1* one may determine differences. The consen-

sus mechanism for blockchains relies on the fact that the branch with the most work behind is the real branch, which roughly corresponds to the longest chain. It gives a clear metric on what version is real. A conventional distributed database settles inconsistency by a settlement process comparing timestamps, and as mentioned in *2.6* there is a risk with the system ending up in an inconsistent state infeasible to bargain due to unmanageable amount of reconciliations. A blockchain ought to solve the multi-master replication problem persistent in other databases [66]. A blockchain never has to explicitly reconcile in case of inconsistency [67], the consensus protocol eventually automatically converges the chain [3]. Consequently a blockchain provides a synchronization mechanism for a distributed database [61]. Hence a blockchain got weak mutual consistency given the majority of nodes being honest, while not having to explicitly reconcile in case of inconsistency [68]. A public blockchain instead relies on its gossip protocol of transactions and blocks as stated in *2.5*, and this implies that the participants do not have to be known. Traditionally all nodes of a distributed database are known. Moreover a distributed database may use locks in order to ensure a serialized behaviour, however many locks becomes complex and may result in deadlocks [60]. This is not an issue for blockchains. A blockchain thus has a simpler consensus mechanism.

A blockchain may be costly to forge if there is a cost with mining blocks. Except for having to convince the rest of the participants of a transaction which is the case for the traditional distributed database, one has to also re-mine blocks which might be expensive depending on the used consensus mechanism. Proof-of-work is for instance costly. With high enough cost blocks buried in the chain are practically *immutable* [33]. Consequently a blockchain may be an immutable append-only log, while a traditional database allows: Create, Read, Update, Delete [69]. A traditional database may however use rules to enforce an append-only database as well [67].

Blockchain utilizes smart contracts, which do not exactly exist in conventional databases where instead stored procedures exist. Both may execute arbitrary code when called with specified parameters, however the difference is that the smart contract co-exists with other data as the smart contract resides on the blockchain and must thus be executed [68]. Stored procedures on the contrary are separated from the actual data, and the stored procedures may hence be bypassed [55]. A smart contract is rather an enforced stored procedure [56] as mentioned in *2.5.6*.

The blockchain enables a decentralized environment without a trusted third party, disintermediation [65]. A distributed database traditionally has not been used where each replica is controlled by a party not trusting the rest [34, 70], however blockchain facilitates it [33]. Illustrations of the differences of trust boundaries, inspired by [34], are presented in *Figure 2.16* and *Figure 2.17*.

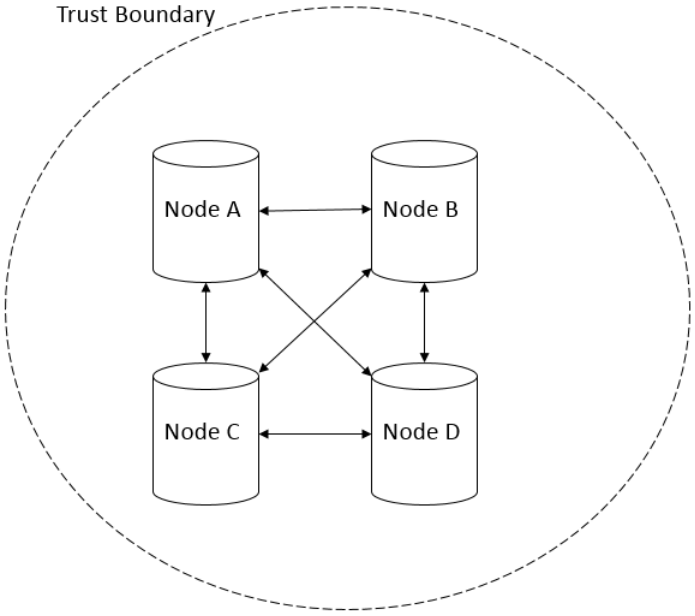

Figure 2.16: *Illustration of a traditional distributed database and its trust boundary.*

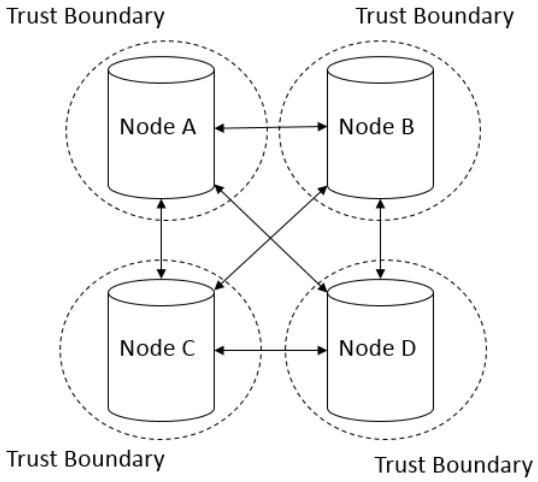

Figure 2.17: *Illustration of a blockchain and its trust boundaries.*

That the database can be managed among parties not trusting each other is partly due to utilization of asymmetric cryptography and digital signatures coupled with data, meaning an honest node only accepts transactions issued regarding data the issuer is allowed to change. Data is thus coupled to an owner [35, 70]. A conventional database would have to couple a public key to each data row to achieve the same. Moreover incentives, as written in *2.5.4*, may exist for a node to act honestly. Due to the consensus mechanism used a malicious node cannot by itself attack the system.

Furthermore a distributed database traditionally has been used for a rather limited amount of identified nodes, about up to 20. A blockchain however is built to handle significantly more nodes [71].

Consequently a blockchain is a database that may be distributed, with the potential of making it infeasible to change data due to heavy costs for instance using Proof-of-Work. The data structure is different from the conventional databases allowing a simpler consensus mechanism. The consensus mechanism may not require knowledge of all the nodes, differencing from the traditional distributed database protocols. A transaction log is built in automatically while a traditional database would have to keep a separate record, thus making a blockchain suitable for time series [37]. Smart contracts are enforced on the blockchain in contrary to stored procedures. Asymmetric cryptography and digital signatures may be utilized in a blockchain to couple data to an identity, and the consensus mechanism is adequate if majority of nodes are honest which makes it possible to reach consensus among untrusted entities. A blockchain may thus be a database that can be handled by a set of nodes not trusting each other. The amount of nodes may exceed the rather limited amount traditionally used for a distributed database. Lately hybrid versions between a traditional distributed database and a blockchain have been presented [70].

2.8 Colored Coins

Due to the blockchain's property of a chain, it is possible to tag an asset of the underlying ownership transacting blockchain and track all made transactions and hence the flow of ownership. This implies other assets may be represented on the blockchain, called *smart-property*. One implementation of it is *colored coins* [72]. Metadata is attached to the transactions indicating the color, practically a unique hash ID [72], of the asset transferred. The colored coin is hence an extra layer added upon the blockchain. A token may represent real world entities that specific persons may own at a given time. Hence ownership of real world assets are represented digitally on the blockchain [73]. The issuer of the tokens representing the real world assets is controlling the exchange between the digital token and the real world asset [24, 74]. Trust in the issuer is required since the actor is responsible for the coupling [4]. The security given by the underlying blockchain is inherited by colored coins since it builds on the infrastructure [73].

2.9 Internet of Things

Internet of things (IoT), refers to objects connected to the Internet. Interaction with embedded systems through Internet can be used to communicate various information. The object may hold assets which are communicable [75]. For the transport industry the vehicles are connected and communicate data with the servers. Using the data various data analysis are conducted which may be used to provide services for the users [76]. The vehicles will be autonomous in the future, which opens up for a combination between IoT and autonomous vehicles.

2.10 Lightning Network

The blockchain technology currently suffers from two shortcomings when being distributed. The scalability in terms of throughput is bad [77] and in theory one can never be completely sure an issued transaction ends up in the long term consensus chain [42]. As mentioned in *2.5.1* regarding the latter, the more block confirmations the higher probability for it to happen. In e.g. Bitcoin often six confirmation is required since the probability increases exponentially with each confirmation, this corresponds to an hour [41] which is problematic when real-time payments are required.

Moreover the throughput of transactions for Bitcoin is seven transactions a second [70], which may be compared to Visa which manages 2000 transactions a second [4]. In order to achieve comparable throughput in a completely distributed system, all the transactions cannot be propagated over the whole network to reach consensus. The size of the blocks batching the transactions would be significant, implying only a few nodes would be able to do block validation and hence the decentralization and thus security is compromised. Instead transactions ought to be conducted off the blockchain [77, 78], which is what *Lightning Network* does [79, 80].

Lightning Network, not yet publicly released but the most anticipated technology that will be deployed upon blockchains *modeling ownership* [81], ought to solve the problems. Instant transactions are enabled together with a scaled up blockchain through *micro-transaction channels* off the blockchain without compromising the security. The channels are in reality only transactions, and the parties elect when to submit the netting balance. The channel is opened with the parties depositing resources to be used in the channel and broadcasting it to the blockchain, this prevents double spending and is called the *opening transaction* [81]. A *2-of-2-multisignature* (2-multi-sig) is used between two parties, meaning both parties must digitally sign the committed funds and accordingly it is not possible for one party to take all funds without permission. Infinite number of transactions can be made on the channel. Any of the parties may broadcast the latest balance of the channel to the blockchain when convenient, however not all transactions must be submitted to the blockchain [77, 79]. Only the first opening transaction and the latest balance, the *closing transaction*, are broadcasted to the blockchain. Hence only two transactions per channel independent of the number of transactions on the chain.

The current balance is stored in the channel, implying old balances are not valid to be submitted to the blockchain. An agreement is achieved where both parties must only broadcast the latest balance, else the contract is violated and all funds bound to the contract may be taken by the cheated party as a penalty. The contract is a *Hashed Time Locked Contract* (HTLC) which is setup between A and B in case A aims to transact to B. The contract consists of B having a secret R that is run through a one-way hash function to obtain H. During the contract negotiation H is given from B to A. If B can provide the secret R that generated H within N days, the *nLockTime* [79, 80], A is forced to transact to B. If B cannot provide the secret no transaction is performed, A may keep the asset corresponding to the transaction. Smart contracts are hence utilized together with 2-multi-sig. The HTLC is a method for forming agreements while still minimizing the required trust using one-way hash function and transaction nLockTime [82].

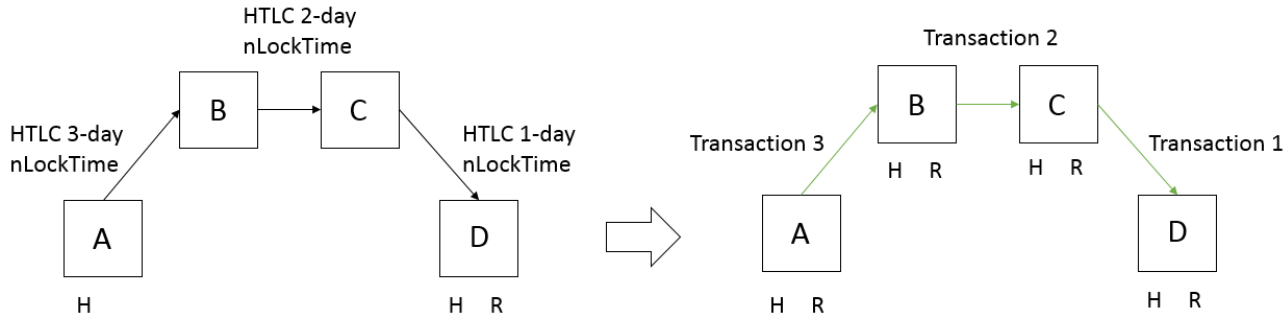

Figure 2.18: *Illustration of a Lightning Network transaction made off-chain through a multi-hop chain.*

HTCL makes it possible to create a *multi-hop chain*, as depicted in *Figure 2.18*, and utilize common channels to increase efficiency. If A wants to transact to D but does not have a direct channel to D, it can be done via common channel parties. To setup the transaction, D gives H to A and keeps R. For each hop from A to D a HTLC is instituted between the nodes of the edges. For each transaction and edge there is a lock time which implies that if party X can provide R to Y within the lock time, Y makes the transaction to X. If no R is provided, Y will not make the transaction. The lock time ensures one cannot be cheated. Node Z will not be forced to transact to node $Z + 1$ unless receiving the secret R which forces $Z - 1$ to transact to Z since Z now may provide R. The lock time is structured in a decrementing manner ensuring this. Hence the chain of contracts results in the transaction of ownership $A \Rightarrow D$ being made. If any party cheats a penalty is conducted as mentioned above, but will however only affect the cheating node [80].

Consequently the throughput scaling is achieved using a large network of micro-transaction channels *off-chain*. Not broadcasting every single transaction increases the throughput significantly. Consequently instant transactions can be performed and the scalability is greatly increased, however block sizes may have to increase slightly [79]. *Segregated Witness* is a tool that helps the throughput issue further in combination with Lightning Network [83], briefly described in *Appendix A.7*.

Lightning Network is currently in an alpha which allows publicly testing [84, 85]. It is believed to be released within a few years [39]. Lightning Network has not yet been deployed in production or fully analysed yet which raises questions of how it will work in a large scale blockchain system [86]. Moreover similar work is in progress regarding off-chain transactions [78], Andrew Miller et al. [77] claim to improve the transaction channels compared to Lightning Network. They call their channels *Sprites*.

Chapter 3

Related Work

This chapter puts this thesis work into the context of the current state of the domain. Work within the transport industry and blockchain is presented as well as similar smart-property applications.

Despite the technology being in its youth several executed projects have touched upon the blockchain technology, mostly in the financial sector. Within the transport industry much weight has been placed on using blockchain for *transparent transports*. The key idea of *transparent transports* is to enlighten various actors on how goods are transported, in order to be able to verify that the transport chain of the goods has been fair. A paper regarding the topic was published in the International Journal of Research in Engineering and Technology [33] where the blockchain technology was reviewed for the transparent transports context. Moreover a similar Swedish project and study has been conducted, where a mobile application for transparent transports was implemented [87, 88].

Another use of blockchains in the transportation domain is examined in a paper by Rowan et al. Communication between vehicles on the road is proposed utilizing the blockchain technology and side-channels for sets of closely driving vehicles to use the security of blockchain. The communication is in particular required for autonomous vehicles in the future [89]. Having Internet of Things (IoT) autonomous vehicles implies security issues, where a hacked vehicle might be used as a weapon. Dorri et al. propose blockchain technology to be appropriate for securing the vehicular eco-system [90].

Moreover Yuan and Wang [54] conducted a pre-study of a blockchain based intelligent transportation system. Data is currently collected by vehicles and sent to cloud-based platforms for analysis, however it is highlighted that it is not preferable with centralized solutions considering they are vulnerable to malicious attacks and might not scale well. Further intelligent transportation systems owned by different actors implies lack of mutual trust in case data is to be shared, thus intermediaries have to be introduced entailing an otiose complexity. Similarlily Sharma et al. claim that the blockchain technology is useful for communication of data between intelligent transport systems without the centralized cloud-based platform [91]. They further envision a scenario where the driver's phone synchronizes with the vehicle and through this manage smart payments for e.g. fuel using smart contracts over a blockchain.

Bosch and TÜV have demonstrated a blockchain use for storing tachograph data. It is a common problem that tachographs are being tampered with, using the blockchain technology the data sent to the blockchain will be immutable and thus prevent cheating [92].

Work regarding *Partially Decentralized* (PD) systems, as defined in this thesis, exists. An example is *Cuber*. Cuber utilizes smart-property and an implementation of colored coins. It is currently in an experimental beta, and facilitates storing tokens representing Euro on the Bitcoin blockchain. Hence the Euro-tokens are transacted on the blockchain in a decentralized manner with a central party, *LHV Bank* [93], issuing and redeeming the Euro-tokens for real Euro. Cuber is the first real world application trying to integrate the use of the Bitcoin blockchain with a bank and its services. One of their goals is to replace cash with Cuber for real-time payments [94].

Another example of such a system but regarding land registry was developed by the government of Honduras using the blockchain technology to reduce the risk of fraud [95]. The project evolved from claims that the blockchain technology and colored coins would revolutionize the land registry management.

ChromaWay, a significant actor within the blockchain business, published a white paper [96] describing a blockchain-based ownership recording system in the context of promoting their own colored coins protocol *EPOBC*. The paper discusses aspects of having real properties stored and traded as digital assets on a blockchain, smart-property. The properties proposed may be anything with a high value, for instance a car or a digital asset as for instance access rights. Having a central issuer of colored coins and the aspects of that is further touched upon in the paper [96].

A master thesis from Chalmers provided a protocol for exchanging smart-property and was compared in terms of security and scalability to for instance the original colored coins approach. It was concluded that their proposed protocol had better scalability but was more complex to use than colored coins [5].

Some work is thus done in nearby areas to the thesis scope. Cuber is a project similar to what the thesis suggests but for another domain. The domain has implications for the usefulness of such a solution as later examined in this thesis. To the best of our knowledge no academic paper is yet presented investigating pros and cons of systems like Cuber, in particular not in the transport industry. Consequently an exhaustive analysis of a PD system holding temporary assets is lacking in academic spheres and is hence aimed to be provided through this thesis.

Chapter 4

Methodology

This chapter presents the methodology including a design research methodology combined with a literature study and interviews.

A *design research methodology* [97] was used since the problem statement of the thesis relies on an architectural foundation. Thus to analyse how a PD solution, compared to alternative solutions, can address the problem motivating this thesis. Practically creating and implementing an artifact was thought to be valuable to gain insights, justifying a design research methodology. The methodology includes six *activities*.

1. *Problem identification and motivation*
2. *Define the objectives for a solution*
3. *Design and develop*
4. *Demonstration*
5. *Evaluation*
6. *Communication*

To identify a problem an examination of the transport industry and its challenges was conducted. The examination included a literature study and interviews with stakeholders of the industry. The interviews were initiated with a brief presentation to the interviewee about blockchain and potential areas of usage in order to create the foundation for further discussions. The rest of the interview had a brainstorming characteristic branching from a few topics as a framework. The above combined with reasoning resulted in an identified application which may be useful for the transport industry. The motivation for it is presented in *5.2 Motivation*.

Objectives for a solution was then stated, using the previously gathered knowledge of what is relevant to the stakeholders combined with new interviews with hauliers, workshops, home detailers and tow truck companies. For the interviews open questions regarding the topic were prepared, leaving room to freely discuss around them. The requirements for a solution are listed in *5.3 Solution Objectives*.

A solution was then designed and implemented as described in *5.4 Design and Development*. *A Proof-of-Concept* (PoC) artifact was created and iteratively evaluated and improved during the work. Stakeholders were at different stages asked to give feedback about the PoC, for instance opinions about functionality that is missing. However no formal evaluation of the usability was done regarding the user experience of the artifact considering it was a PoC.

First a low-fidelity prototype using *Balsamiq* [98] and a conceptual idea of the system was created. With iterative enhancements the final PoC solution was implemented. The final solution was decided to be implemented as an Android app, presented graphically in *Appendix B*. Ideally the software perhaps would be placed in a vehicle, however given the scope of the project and the current status of the possibility to practically do it, it was concluded infeasible. The Android application however ought to demonstrate the equivalent functionality, and was considered adequate. Moreover one may envision the use of a phone which synchronizes with the vehicle and thus might be used to manage the required functionality.

The created artifact was demonstrated through practical usage, in order to gain new insights. The demonstration included to test the required functionality as described in more detail in *5.6 Demonstration*.

The proposed solution and the alternative solutions were evaluated and compared toward the previously listed objectives of a solution. Conclusions were drawn from the demonstration when possible. Moreover interviews were conducted with stakeholders regarding their opinions about relevant aspects to the solution objectives as mentioned above. The interviewed stakeholders were: hauliers, tow truck companies, workshops and home detailers. Further insights from the implementation and the blockchain knowledge from the wide literature study were used for reasoning toward the solution objectives. The evaluation is presented in *6 Evaluation*.

Lastly *communication* of the study is performed through this thesis and oral presentations at Scania and KTH.

Chapter 5

Design Proposal

This chapter presents the design proposal together with its motivation and a list of solution objectives. Moreover an overview of the involved actors is provided.

5.1 Transport Industry Actors

This section gives an overview over the different stakeholders to the problem motivating this thesis. Their roles and interactions are further presented.

- *Haulier*: A company or person working in the transport industry and transporting goods. A haulier is moreover the customer of home detailers, workshops and towing companies.
- *Home Detailer*: A home detailer sells vehicles and may have various businesses with hauliers. For instance the home detailer may issue guarantees of payments on the behalf of their customers, the hauliers. This means that e.g. workshops may instantly work with a vehicle since they are guaranteed the payment indirectly by the home detailer if the customer itself does not pay. This entails a credit risk, however the home detailer uses credit checkup services and their customer relations to decide whether to issue a GoP. They are very restrictive.
- *Workshop*: A workshop repairs vehicles and may order towing for vehicles in need. Thus the workshop may pay the tow truck company for the towing, however the cost is included in the reparation invoice later sent to the customer.
- *Tow Truck Company*: A company for vehicle assistance if a vehicle needs reparations and cannot transport itself to the workshop.
- *Scania Assistance*: A service for managing the complex creditworthiness problem within the industry, a central point which may organize towing and reparations.

An overview of the typical interaction process between the actors in case a vehicle requires service is depicted in *Figure 5.1*.

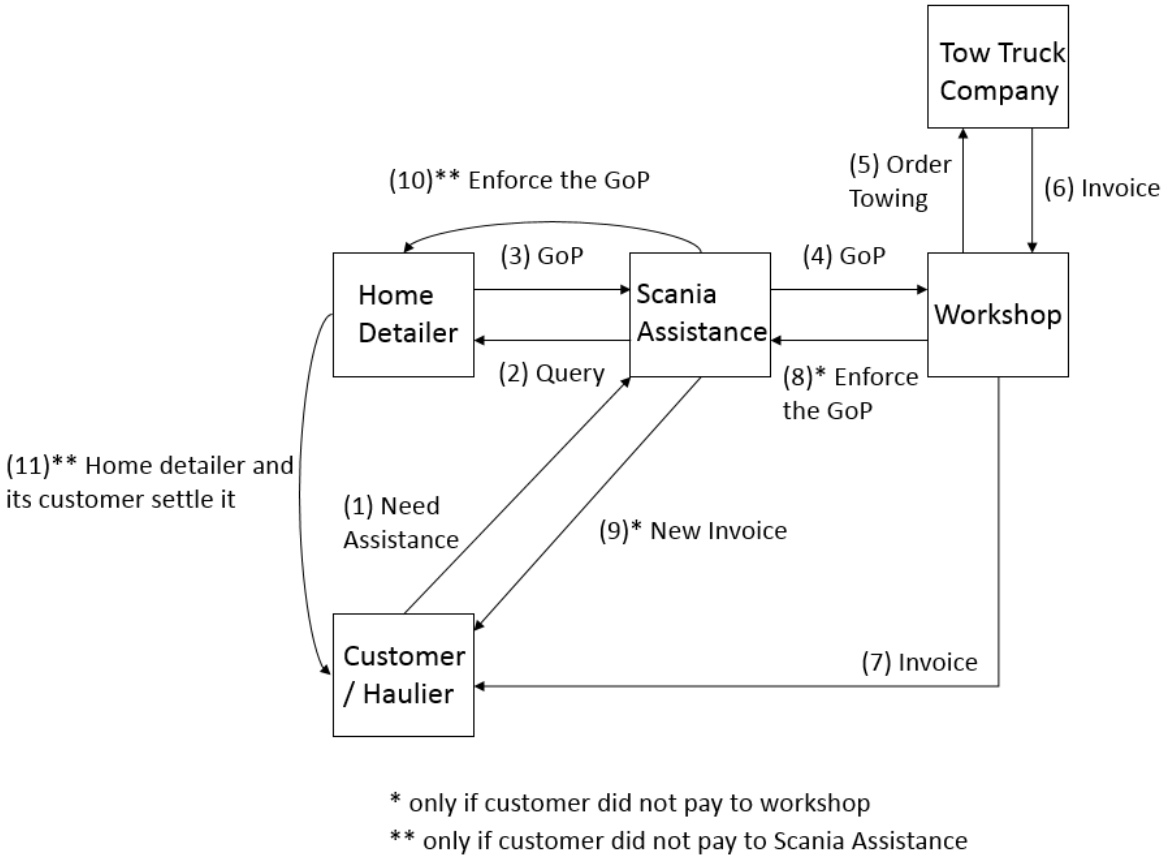

Figure 5.1: *Illustration of the prolix payment process.*

5.2 Motivation

The problem with creditworthiness explained in *1 Introduction* motivates the value of having vehicles store digital assets to pay for e.g. reparations. It is time consuming to transact through a bank, especially when making international transactions. Paying by invoice is not preferable since the workshops want direct payment in case they do not trust the counterparty with an invoice payment. This practically means a third party, e.g. Scania Assistance, has to ensure the workshop the payment for the reparation until the haulier pays. The guarantee of payment (GoP) also includes the payment for the towing, which the workshop orders and pays for temporarily until the the customer or the GoP issuing party pays. The GoP is however only issued if the home detailer to the corresponding vehicle issues a GoP, which is done depending on the relation between the haulier and its home detailer. Further credit checkups are utilized. An illustration of the process can be seen in *Figure 5.1*.

To avoid the credit risk for the parties *ScaniaCoins* ($C) is proposed to be issued by Scania on a blockchain. This allows vehicles, as IoT devices, to have a digital wallet. Hauliers may exchange real assets for $C and keep them on a company account as a buffer. The digital assets may then be loaded onto the vehicles in real-time and used to make real-time trades for services and goods. The payee may later redeem real world assets from

Scania. Consequently the payee only needs to trust Scania to redeem the real world assets instead of trusting the actual payer as mentioned in *1 Introduction*.

5.3 Solution Objectives

Following requirements are specified for a solution:

1. *Confidentiality*: It should not be possible to access an arbitrary actor's resources and view its secret transaction counterparties.
2. *Integrity*: Assets should only be altered by the appropriate identity. Thus one should not be able to spend ScaniaCoins of others.
3. *Availability*: It must be possible to access one's assets, view balance, make transactions and view one's transaction history at any time.
4. *Consistency*: The database system must be kept in consistent states to avoid ambiguous interpretations of data.
5. *Immutability*: A legitimate transaction is binding and cannot be revoked after it is submitted.
6. *Response time*: The transactions ought to be in real-time since real-time payments are desired at any time.
7. *Cost*: Transactions should not be more expensive than traditional credit payments, for instance using business credit cards or invoice.
8. *Customer Usefulness*
   (a) It must be easy and quick to load and unload vehicles with assets.
   (b) The solution ought to be an improvement to the current approach for the third parties at risk.
9. *Trust*: Users must be able to trust the system. Users must also trust the underlying currency, else the system itself will never be trusted.
10. *Environmental friendly*: The system should not contribute significantly to the environmental issues.

5.4 Design and Development

5.4.1 Bitcoin and Coinprism

The proof-of-concept (PoC) solution was implemented on top of the Bitcoin blockchain for several reasons. Firstly it implies that the solution inherits the security properties of the underlying blockchain [39, 72, 99]. Since Bitcoin provides the most work behind its blockchain it accordingly makes it the most secure [96]. Moreover being the most popular blockchain, tools exist which enable a PoC implementation.

One of the tools is *Coinprism* which is an implementation of a colored coin wallet using the colored coins protocol *Open Assets Protocol* (OAP) [100, 101]. The protocol utilizes Bitcoin's possibility of adding metadata to a transaction. Bitcoin uses a scripting language and all data in a transaction after the operational code OP_RETURN is ignored and thus leaves the possibility to add metadata [102]. The metadata added by OAP contains an *asset ID*, identifying an unique asset, and an *asset quantity* for each transaction. Consequently the underlying value of the bitcoin that is colored does not affect the value of the colored coin itself [99, 101].

Because of Bitcoin's security combined with good opportunities for a PoC using existing APIs like Coinprism, it was decided to be an adequate approach for the created artifact. Moreover it should be noted that the underlying blockchain and the colored coins API are both parts that could be swapped and the concept does not rely on any of them, they were simply found appropriate for the PoC.

The Coinprism API [103] provides a colored coins layer between the artifact and the underlying blockchain. The API-calls utilized were:

- *Create an address*: Creates new key-pairs. A user requires several asymmetric key-pairs as mentioned in *2.5.7*. The public key and asset address to send assets to, and the associated private key required to make digital signatures and prove one's identity.
- *Issue colored coins*: Issues the colored coin ScaniaCoin. A unique *asset ID*, as the OAP suggests, was created for the ScaniaCoin asset.
- *Get balance of an address*: Supplies the data for obtaining the balance for an address.
- *Get recent transactions of an address*: Supplies the data for getting the transaction history for an address.
- *Send an asset*: Creates a raw transaction sending ScaniaCoins from X to Y.
- *Sign an unsigned raw transaction*: Signs an issued transaction using the adequate private key.
- *Push a signed raw transaction to the network*: Broadcasts the signed transaction to the blockchain.

The PoC was deployed upon the *testnet* (*https://testnet.api.coinprism.com/v1/*) instead of the *production net*. On the testnet the bitcoins have no real value, consequently one may obtain free testnet bitcoins to use in development from so called *faucets* [104, 105]. Thus bitcoins were received to a main address from a faucet and the main address later distributed the bitcoins with ScaniaCoins on top, to the user addresses used for testing.

A depiction of the architecture stack is provided through *Figure 5.2*.

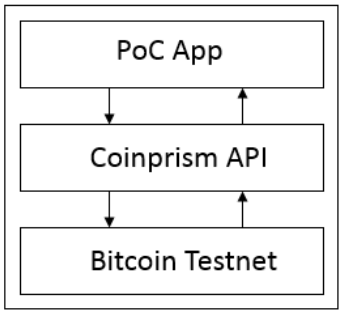

Figure 5.2: *Illustration of the layers in the PoC architecture.*

5.4.2 Assumptions

As mentioned in *2.10 Lightning Network* transactions are not currently instant, time is required before the blockchain network adds the transactions into blocks. However a working Lightning Network is assumed in this thesis, implying instant transactions and thus balances had to be simulated through a slight modification in the implementation. The balance for an address is the sum all unspent outputs belonging to that address. The Coinprism API provides two balance fields for each unspent transaction output which may be used while a transaction is unconfirmed: *balance* and *unconfirmed balance*. To simulate the transactions being in real-time the balance for each unspent output was considered the sum of the balance plus unconfirmed balance. The algorithm for obtaining the balance for an address can be seen in *Algorithm 1*.

Algorithm 1 Calculate Balance For an Address

```
1: procedure GETBALANCE(address)
2:     Asset[] unspent_assets = CoinprismAPI.getBalance(address).assets
3:     balance = 0
4:     for (i = 0; i < unspent_assets.length; i++) do
5:         Asset asset = unspent_assets[i]
6:         if asset.assetID == ScaniaAssetID then
7:             balance += asset.balance
8:             balance += asset.unconfirmed_balance
9:     return balance
```

Moreover the timestamp of a transaction in the application had to be simulated as well. The API provides a *block time* field for each transaction, meaning the time it was put into a block. This field is null while the transaction is unconfirmed, if this is the case the transaction time is regarded as the current time in the application to simulate real-time transactions.

5.4.3 Limitations

Since the PoC utilized an existing API to interact with a blockchain, the API and the corresponding service limited the PoC slightly. For instance Coinprism did not feature adding a message to a transaction, which is an essential functionality to the stakeholders

in the transport industry [8]. Moreover a Coinprism account was required per identity, thus for the PoC each key-pair was regarded as a separate user. In reality the set of key-pairs constitute a wallet as stated in *2.5.7*.

5.4.4 Storage of the Keys

Each identity handles two different types of keys and one type of address. One *public key* used as the sender address, one *private key* to sign transactions and one *asset address* which is the address to receive funds to. The public keys and the asset addresses may be publicly known, however the private keys must be kept secret. Having access to the private keys implies the possibility of issuing transaction from the corresponding public keys. When setting up a user, a public and private key and asset address were stored on files on the device. The public keys and the asset addresses were stored unencrypted since these are publicly known, however the private keys were password encrypted with a password chosen when setting up the user. The publicly known key and asset address were stored in one file while the private key was encrypted and stored as a byte array in another file on the device. The encryption algorithm used was a *Password Based Encryption* (PBE) with *Secret Hash Algorithm* (SHA) and 256-bit *Advanced Encryption Standard* (AES) with the *Block Chaining* (BC) *Cipher Block Chaining* (CBC). The algorithm was considered the most suitable one of the existing possible choices for built in Android, and accordingly adequate for the PoC. 256-bits instead of 128-bits for AES was used because it means a larger space of possible keys, however it should be noted that attacks on the 256-bit scheme is found making it possible to be faster than brute-forcing [106]. SHA is used in many security standards and by both governments and the industry [107]. CBC is one of the recommended modes of operation by NIST [108].

Later when making a transaction, the user provides the password to decrypt the encrypted private key. If the correct password is provided, the transaction may be issued. However if the provided password is wrong a few outcomes are possible.

- Most likely this results in a *bad padding* decryption error while trying to decrypt using the wrong password. A padding is added while encrypting to not be able to conclude the length and content of the unencrypted data. Moreover this results in a multiple of a block size. When the password is provided one should be able to recover the encrypted data unambiguously by separating the data from the padding. Thus one can conclude the password is wrong if this process fails [109].
- In rare cases using the wrong password might not give an error, however very few passwords will result in a valid padding. This implies two possible outcomes:
  - The wrongly decrypted private key now has a number of characters differencing from the expected key length, thus one can directly conclude the password must have been wrong.
  - The wrongly decrypted private key has the correct number of characters, which means the transaction will be issued but the network will not accept the transaction since the wrong private key is provided. This is the most rare outcome.

5.4.5 The PoC Application

Using Bitcoin and Coinprism in the way described in *5.4.1*, the PoC application was developed. Since it was infeasible to integrate the artifact into a vehicle in the time scope, it was decided to create an Android app to demonstrate the ease of ScaniaCoin-transactions through a blockchain.

The application provides the following features, which may be viewed graphically in *Appendix B*:

- *View one's own address*: One can get one's own asset address through the application in a *Wallet Import Format* (WIF) which is an encoded address into a format easier to copy [110]. To prove how easy this WIF encoded address is to share, the application provides a QR-code encoded WIF address which is a two-dimensional barcode [111]. The QR-code can be scanned and decoded instantly by another entity. A QR-code is depicted in *Figure 5.3*.

Figure 5.3: *QR-code example.*

- *View balance*: It is possible to view the current user's balance, it is calculated as described in *5.4.2*.
- *View transaction history*: The transaction history of a user can be viewed. All data about a transaction is in blockchains with an *unspent transaction output* (UTXO) model provided as a list of inputs and a list of outputs as described in *2.2.6*. Bitcoin utilizes this model, implying the Coinprism API consequently does. Using the application there is always one address providing the inputs and one address which ought to claim the output for a given transaction, hence the sender and receiver may be found and displayed in the application.
- *Make a transaction*: Given an address to the receiver and the amount to be sent, a user can confirm the transaction with the appropriate password followed by broadcasting it. The address can either be typed, pasted or scanned using the built in QR-scanner of the app. The QR-scanner uses the Android Google Barcode-Scanner API [112] to decode the QR-code. Accordingly the receiver may provide its QR-code encoded address and the sender may instantly scan the code to retrieve the address. The *Figure 5.4* depicts the interaction and protocol between the PoC app and the Coinprism API when making a transaction.

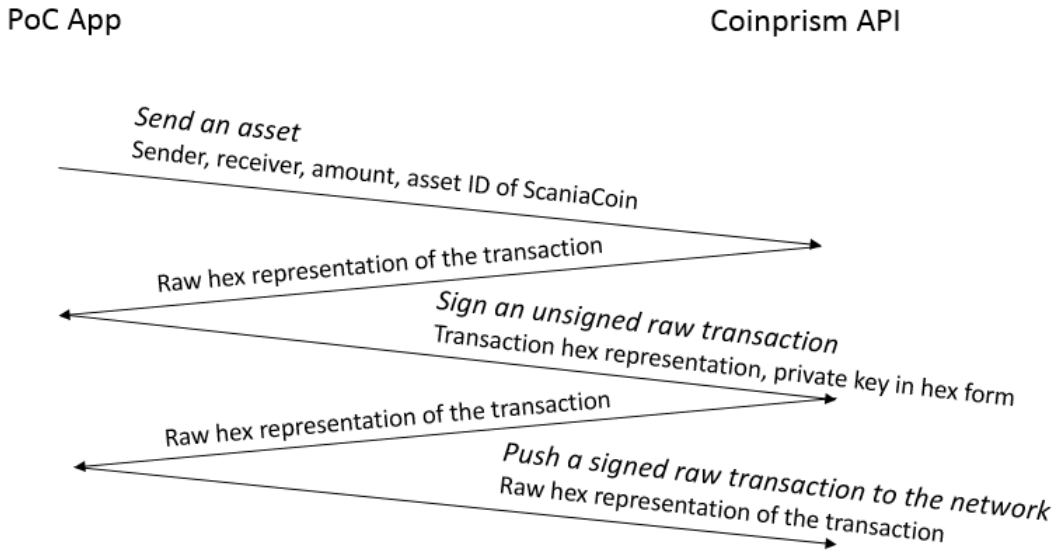

Figure 5.4: *Illustration of the interaction between the application and the Coinprism API when issuing a transaction.*

5.4.6 System Overview

The overview of the conceptual structure is depicted in *Figure 5.5*. The hauliers may buy ScaniaCoins and load their vehicles with the digital assets. The assets may be traded for services or goods peer-to-peer, the payee may exchange the ScaniaCoins for real assets. Scania acts as a centralized exchange point coupling real assets to the digital assets on the blockchain. Thus the transactions and the balances are decentralized while Scania is only involved regarding the issuance and redemption of the ScaniaCoins. The trust to issue and redeem is separated from the trust to maintain the balances and control the transactions in a correct manner. Instead the asset storage and transactions rely on an existing secure infrastructure using Bitcoin's blockchain. It is a *Partially Decentralized* (PD) architecture as defined in *1 Introduction*.

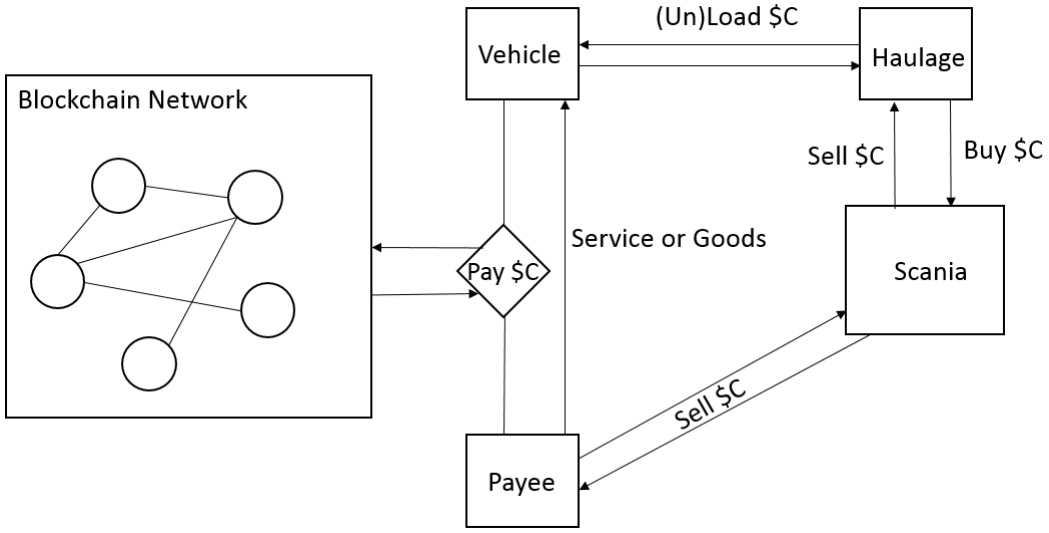

Figure 5.5: *Illustration of the partially decentralized system with a centralized exchange but decentralized storage and transactions.*

5.4.7 Alternative Solutions

The main system to be analysed is a centralized system storing temporary assets decentralized and using decentralized transactions. To fully conclude the usefulness of such a system, comparisons were done to two alternative systems possible to deploy in the same environment.

Completely Centralized

A *Completely Centralized* (CC) system implies no public blockchain is used. hauliers trade real assets for ScaniaCoins, Scania keeps track of all actors' balances in a database. The database is presumed to be distributed among various sites controlled by Scania as a central party. Except for keeping the balances of the users, the system is also required to handle all the transactions made in a non-faulty manner. The system is depicted in *Figure 5.6*.

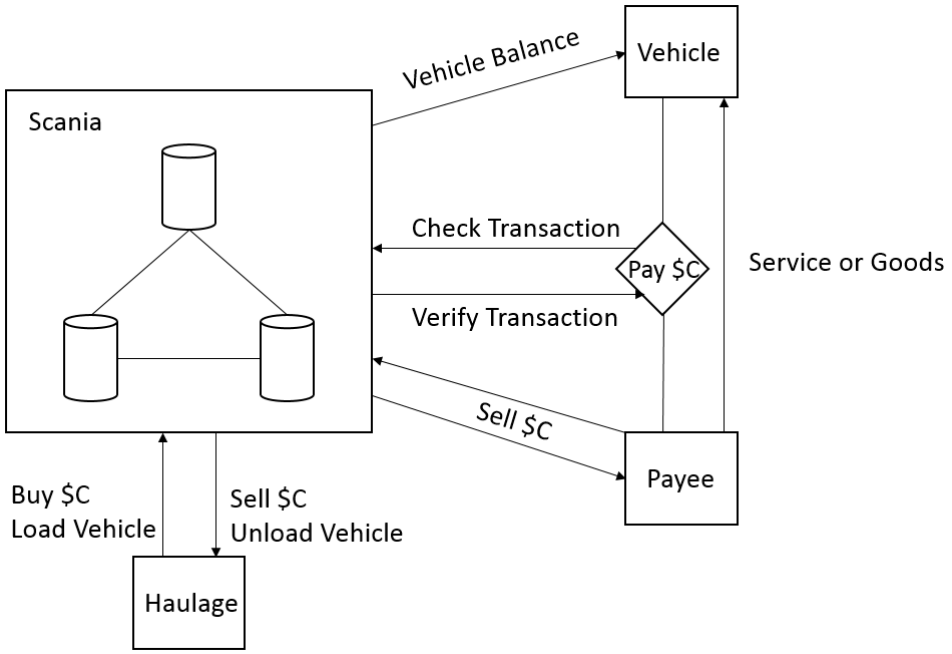

Figure 5.6: *Illustration of a completely centralized approach. Scania issues ScaniaCoins to be coupled to a vehicle that are used to buy services and goods. Transactions and balances are maintained by Scania.*

Completely Decentralized

A *Completely Decentralized* (CD) system involves no ScaniaCoins, instead the vehicles are loaded directly with the underlying cryptocurrency of the cryptocurrency based blockchain as defined in *1 Introduction*. In this study the underlying blockchain is Bitcoin's. The balances are stored decentralized and the transactions are conducted decentralized peer-to-peer. Moreover issuance and redemption of the bitcoins is decentralized. Accordingly Scania is not involved as any type of central party, but is still enabling the vehicles to be loaded with the cryptocurrency. The system is depicted in *Figure 5.7*.

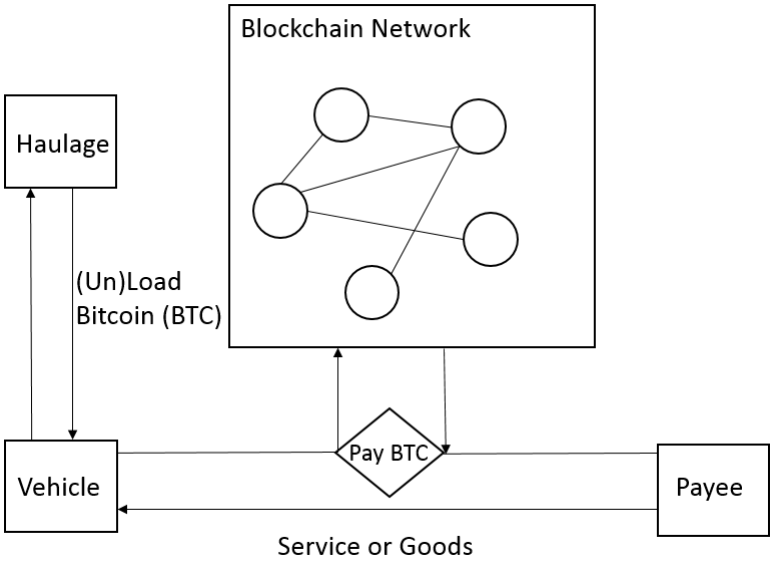

Figure 5.7: *Illustration of a completely decentralized payment system where the vehicles are loaded with the underlying currency of the blockchain.*

5.4.8 Solution Comparison

To get an overview of the three proposed solutions, *Table 5.1* is presented.

Table 5.1: *Overview over the three solutions.*

Solution	Issuance	Redemption	Transaction	Storage
PD	Centralized	Centralized	Decentralized	Decentralized
CC	Centralized	Centralized	Centralized	Centralized
CD	Decentralized	Decentralized	Decentralized	Decentralized

5.5 Currency Exchange

The PD and CC solutions both utilize ScaniaCoins, consequently Scania may set the exchange rate themselves. One may institute a legal contract containing the exchange rate between Scania and the customers. For instance 1 $C may correspond to 1 SEK. It is important to adjust the ScaniaCoin after the real world currencies. An approach is to let 1 $C = 1 SEK and let all foreign customers translate their currencies into SEK when buying and selling ScaniaCoins. Instituting one exchange rate for each currency is not feasible. With such a system the buyers and sellers may decide to buy and sell the ScaniaCoins using the most profitable currency. One may scam Scania using this approach, buying ScaniaCoins using the weakest currency and selling them back to Scania using the strongest currency.

For the CD solution Bitcoin is utilized, and thus also its exchange rate. The exchange rate for Bitcoin is driven by supply and demand of the cryptocurrency.

5.6 Demonstration

To *"demonstrate the use of the artifact to solve one or more instances of the problem"* [97], the testable functionality was tested. The PoC was developed as presented in *5.4 Design and Development*, and the functionality that was tested and proven to work in real-time was:

1. *Issue transactions*: Transactions could be issued from an address X to another address Y with the right amount of $C being transacted.
2. *Check balance*: The balance was reflected in an adequate manner affected by the issued and received transactions.
3. *Latest transactions*: The previous transactions, as a transaction history, were presented appropriately.
4. *Check password*: With the password encrypted private key transactions could be issued. A wrongly provided password resulted in a *bad padding error* as mentioned in *5.4.4*. Thus an error message explaining that the password was wrong could be presented. In rare cases a wrongly provided password may generate a private key of correct length as described in *5.4.4*, this did not happen when testing. It was however tested to sign a transaction with a key with the correct length but differing from the correct key, the network then did not accept the transaction.
5. *Address Sharing*: The PoC further provided a proof of the possibility of making transactions of $C easy to perform. The payment through the use of QR-codes demonstrates one way of performing easy payments. The receiver views its own QR-code encoded address in the app while the sender using its own instance of the app scans the barcode to decode the QR-code to a sendable address. Having the address and simply adding the amount to be sent and then supply the correct password was sufficient to issue a transaction.

Chapter 6

Evaluation

This chapter evaluates the solution proposed in 5.4 and the alternative solutions against the objectives in 5.3.

The requirements presented in *5.3* were when possible compared to the observed outcome from the demonstration. Since the solution depends on architectural choices, quantifiable measures are elusive. Using objective qualitative measures from the empirical evidence, as a possible evaluation approach [97], was instead done. The observations combined with objective theoretical investigations and gathered knowledge from the implementation and interviews, was used to reason toward the solution objectives. Moreover since the solution is building on colored coins and Bitcoin, it inherits the security properties of that blockchain. This was further used for reasoning. Worth noting is that a public blockchain solution such as Bitcoin seems to work in practice but is not rigorously proven in theory, consequently the area is lacking formal proofs [42, 71]. A public blockchain solution like Bitcoin relies on probability. To evaluate the usefulness of the system in comparison to alternative solutions the above approach in respect to the presented objectives was used as well. Regarding customer usefulness the interviews with stakeholders such as workshops, home detailers, hauliers and tow truck companies were used as a foundation.

The partially decentralized (PD) solution and the completely decentralized (CD) solution are both utilizing a public blockchain. For the rest of the thesis aspects examined when referring to the properties of the blockchain accordingly apply to both solutions.

6.1 Confidentiality

The confidentiality of the PD system is inherited by the underlying blockchain [96]. Everything on the public ledger is viewable by anyone, thus everyone can see all transactions and all balances corresponding to the public keys [33, 37]. Although on the blockchain only public keys can be viewed, from which one may not be able to conclude which real identities correspond to them. There is a chance one can infer which real identity ought to correspond to a public key. However since an identity is corresponding to a set

of public keys that can be newly generated at any time, it makes it significantly more difficult to track actions of an identity as stated in *2.5.7*. It is although still possible to infer, even if it is difficult. There exist cryptographic methods aiming to solve the issue and making it more grievous, however some information always have to be provided to the network [67]. No attack to the system is accordingly required to partly violate confidentiality [37, 113]. The same as above applies for the CD approach. However the PD system has a central party for exchanging $C. Thus it is possible for the third party to couple addresses used when exchanging to real identities. If paper trails are stored, this data may be exposed in case of a successful attack. The attacker may then access the couplings between real identities and public addresses used when exchanging [39].

Similarly for the completely centralized (CC) approach the confidentiality is bound to anyone with the adequate access rights to the central party's servers and databases [114], as earlier illustrated in *Figure 2.16*. To compromise the confidentiality an attacker has to intrude on the central party's system. Confidentiality is ensured until the system is successfully attacked.

The data essential to the problem motivation of this thesis is not controversial. Where a haulier's vehicles are e.g. towed, repaired or fueled is publicly known information, and the majority of hauliers are using the same services. Thus transaction counterparties for such activities is no problem exposing. However some counterparties may be controversial to expose, for instance goods suppliers [6, 7].

6.2 Integrity

Inheriting the properties of the underlying blockchain, a PD approach ensures integrity because of the usage of a hash chain as pointed out in *2.2.4*. If the integrity of a blockchain has been violated it can be concluded directly, and in practice it is infeasible to forge the Bitcoin blockchain because of the mining power of the decentralized network [33, 39]. Moreover due to the usage of digital signatures one cannot spend assets of others nor claim transaction outputs belonging to others as stated in *2.5.6*. The rationale behind it is that no honest node accepts transactions being signed with the wrong private key. A 51% attacker may forge a chain and its signatures but since the honest nodes never will accept it, the assets of the forged branch is bound to the attacker implying it loses its value in practice [51]. Moreover since the Bitcoin blockchain is the blockchain with the most invested money in, it also creates significant incentives to keep the blockchain fair and working. This thus results in a more secure system regarding the integrity [96]. The Bitcoin blockchain provides immense security [115].

If an attacker manages to access one of an identity's private keys, the attacker may violate the integrity of that identity since the attacker then can issue transactions from the private key's corresponding public key address. Consequently violating the confidentiality of a private key in turn violates the integrity of the corresponding identity. Randomly guessing a private key is due to the underlying blockchain done in Bitcoin with the probability $1/2^{160}$. With the computing power of the whole Bitcoin network used for finding the key, it is probable to take above 100 000 years to find the private key [116]. Considering a wallet consists of a set of key-pairs as stated in *2.5.7*, the probability of vi-

olating the integrity of all pairs by randomly guessing is negligible.

However the private keys are stored on the identity's device which also makes it vulnerable to offline attacks. Having the device with the password encrypted private keys, one may brute-force the password. If the password is badly chosen, even a dictionary attack is possible. However exponential backoff can be used to drastically slow down such attacks [39], this is done through introducing a lock time between each password guessing attempt and this lock time is exponentially increased for each bad guess [117]. The strength of the password is in this case thus correlated with the strength of the encryption, a strong password ought to secure the private keys for a sufficient time in combination with the exponential backoff. Further security can be compromised if e.g. a key logger is installed on the device, capturing the password [118]. Moreover a haulier may want to store the key-pairs of the vehicles on another location than solely on the artifacts since losing a private key implies a loss of all assets bound to the corresponding key-pair. If the confidentiality of the other key storage is violated, the integrity is. The above also applies to a CD approach.

With a CC approach it is possible to violate integrity in case even one of the sites is compromised. The attacked site may issue a transaction, since data is not bound to an identity as mentioned in *2.7*, which may later be transacted at all the sites. Similarly if an admin account handling the databases is hacked, the whole database may be compromised [34, 70]. Impossibility results have been proven that if even one node is faulty distributed consensus cannot be obtained in an asynchronous system [119, 120], however blockchain solves it using probability [120, 121].

6.3 Availability

As the Bitcoin blockchain is decentralized, attacks to the system such as e.g. a denial-of-service (DoS) attack are infeasible due to the robustness the decentralization provides [61, 74]. Nodes may be targeted with a DoS attack [53], but no central party as a single point of failure exists [33, 74]. The Bitcoin blockchain is however not completely distributed. The mining power is often bound to miners residing on approximately the same location meaning it is possible to isolate them by hijacking a few *border gateway protocol* (BGP) prefixes with a routing attack using the Internet infrastructure [53]. Internet is however comprehensive, thus the Bitcoin blockchain network ought to be reachable at all time [39]. A centralized solution with distributed databases is also vulnerable to routing attacks since communication with and between the physical databases may be hindered.

The blockchain is resilient and can handle damaged and malicious nodes on the network [37, 122]. However as mentioned in *2.5.5* a 51% attacker can compromise availability by suppressing transactions. Owning 51% of the mining power is however unreasonable as examined in *Appendix A.5*.

Availability is for a blockchain solution compromised in case an identity loses its keys. The identity is supposed to be authorized but due to the lost private keys, the identity cannot alter any information. Prevention of the scenario may as mentioned in *6.2* be done through storing the keys not only on the device. The above applies to both blockchain solutions: PD and CD.

The CC system is dependent on the up time and throughput of the central party's servers and databases. Considering it is distributed it provides robustness, even if it is far from the robustness provided by a public blockchain solution like Bitcoin. Partly because the blockchain is distributed over more nodes, but also since the ownership of the database is decentralized instead of centralized. However the CC approach should be considered sufficient regarding availability since it is distributed over several sites.

Moreover since a private key coupled to an identity is not enforced with the CC system, availability cannot be denied permanently because of a lost key. The central party may solve the problem.

For both scenarios availability is compromised in case of software or hardware failure.

6.4 Consistency

Consistency is a security issue as stated in *2.6.2*. Using the Bitcoin blockchain, the consensus mechanism is built in with Proof-of-Work and explicit reconciliation processes are not required as written in *2.7*. The mechanism relies on that the branch with the most Proof-of-Work behind is the real branch. The blockchain is kept in consistent states since each block accepted by a node must keep the consistency of the local replica of the database. Nodes may temporarily disagree on which of the consistent truths is real, but the fork is later resolved automatically due to Proof-of-Work. Honest nodes will never adapt to inconsistent chains. The deeper buried blocks in the chain are consistent among the network [123]. The above applies to both blockchain solutions.

Having a CC system the databases distributed among entities controlled by Scania require a commit protocol. With an eager commit protocol, throughput is affected negatively compared to a blockchain solution using Lightning Network since nodes may become bottle-necks. With a lazy commit protocol the database may obtain an inconsistent state and then needs to reconcile by communication. As mentioned in *2.6.1* these reconciliation protocols may rely on timestamps, with the latest timestamp of two contradictory transactions deciding. Consequently issued transactions can be overwritten. Moreover a sufficient amount of reconciliations may imply an inconsistent database infeasible to bargain [62].

Consequently the consistency of the database is simpler to keep in a blockchain than a traditional distributed database because of a chain being easier to manage than a set of data. However keeping consistency ought to be no problem for a traditional distributed database either. A traditional distributed database has existed for decades and is a mature product that is well tested to solve such issues in practise [67].

6.5 Immutability

A blockchain with enough work behind, as Bitcoin's blockchain, is practically immutable [33]. This means transactions issued on the blockchain are irrevocable. As mentioned in *2.5.6* a 51% attack can however revoke transactions by building on a branch outpacing the rest of the network. Theoretically any party may perform the above however the

probability is proportional to the fraction of mining power of the party. This implies a low probability of such an attack since a potential attacker is improbable to have too high fraction of mining power as detailedly examined in *Appendix A.5*. The solutions building on the public blockchain may thus be concluded to perform immutable transactions, and consequently the blockchain is described as an append-only log [37]. Once data is written it remains forever [124]. With a traditional public blockchain no central party can control nor change the actions on the blockchain since the transactions and balances are completely decentralized [37, 125].

For a CC solution the third party controls everything by both keeping the balance ledger and handling the transactions. Consequently it is possible to mutate data through the third party, making it vulnerable in case of an intrusion [126].

For the usage domain immutable, and hence irrevocable, transactions are valuable. It occurs that hauliers manage to cancel or reverse payments done through the bank after the towing and reparation is performed [7, 9, 10]. A bank may in theory be able to withdraw a transaction, but it might not be legally able to if the receiver does not accept sending it back. A legal process can be triggered [127, 128]. However the time between the customer issued the transaction, and thus can show a receipt, and the money is actually confirmed in the payee's account is where it occurs that transactions are canceled. The time is depending on the involved banks, and vehicles may be kept in custody during this time [10]. As mentioned in *1 Introduction* it can take days for international transactions and even several hours while transacted within the same bank and country.

6.6 Response Time

The response time with blockchain and Lightning Network is dependant on the connection speed between the two involved peers. It is thus instant unless it is the first transaction made between two actors and they do not share any common channels as derived from *2.10*. Hence an on-chain opening transaction is required, meaning the speed is bound to the on-chain transaction speed.

Transaction speed in a CC system relies on the throughput of the central party's servers and databases. The transaction speed is consequently real-time as well. Thus all proposed solutions outperform bank transactions in terms of transaction time.

6.7 Cost

Both of the blockchain solutions, PD and CD, require several transactions. Bank transactions from and to the issuer for issuance and redemption of the currency, and transactions on the blockchain. A transaction fee while transacting on the blockchain is required. This applies to both the CD solution, where there are several different possible exchanges, and the PD solution where there is only one exchange point. With a CC solution only the bank transactions from and to the issuer are required. Thus the centralized solution will always be cheaper, however with zero transaction fees on the blockchain the cost will be equal.

The cost for a transaction on the Bitcoin blockchain is currently approximately 4 *Swedish Kronor* (SEK) due to the supply of too few mined transactions and demand of too many issued transactions. The cost however ought to drop using off-chain transactions [69]. With Lightning Network the implication would be near zero transaction fees [39]. A bank transaction using a Visa card instead uses a percentual quota of the transaction amount as transaction fee. In Europe a Visa credit business card costs 1-2% of the transaction amount to do transactions with [129]. With a direct Internet bank transaction the cost depends on the involved banks. Transactions through the banks *Nordea* and *Handelsbanken* costs 1.50 SEK to transact and 1.50 SEK to receive for a business within EU [130, 131].

Consequently paying by invoice within EU can be estimated to cost $1.5 + 1.5 = 3$ SEK in total. A transaction with both of the blockchain solutions within EU thus can be estimated to cost $2 \times (1.5 + 1.5) + 3 \times 4 = 18$ SEK in total. This applies given that the exact amount is directly bank transacted to the issuer and transacted on the blockchain to the payee followed by redeemed instantly. The cost is due to that two bank transactions within EU is required, one when issuing and one when redeeming. Each entails 1.5 SEK for sending and receiving respectively. Furthermore three transactions over the blockchain (Scania ⇒ Payer ⇒ Payee ⇒ Scania) with the estimated transaction fee 4 SEK each is required. With a CC solution the blockchain transaction fees can be ignored, thus 6 SEK per pure transaction. Comparing the above numbers to the business credit card transaction fees at 1% and 2% of the transaction amount one obtains *Figure 6.1*.

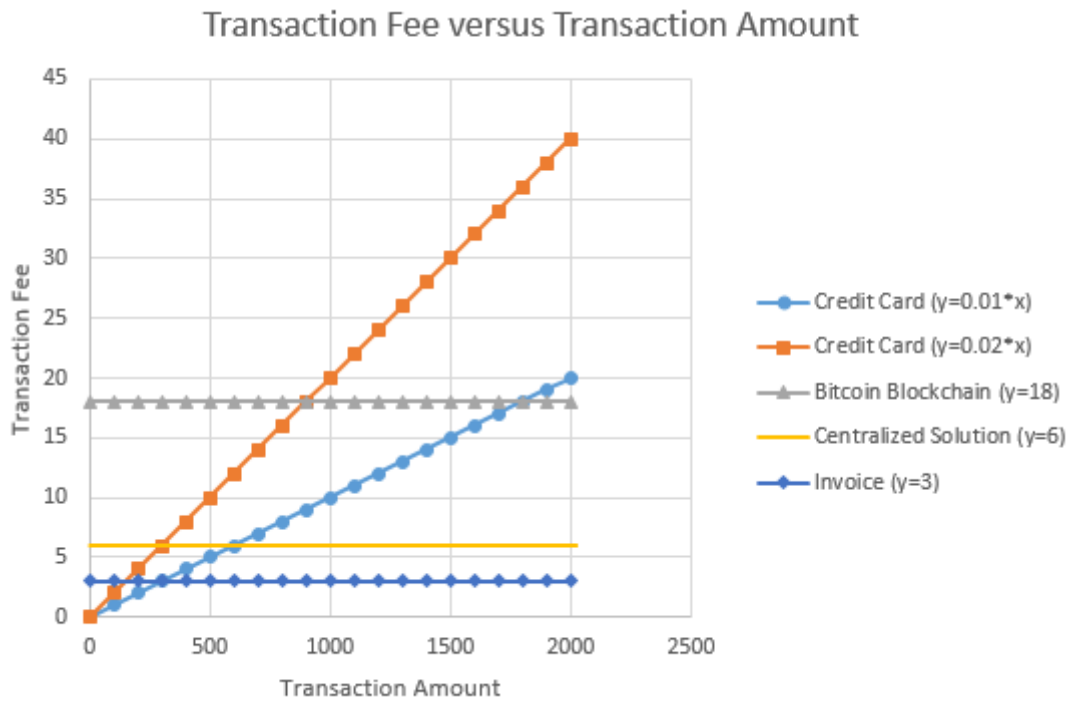

Figure 6.1: *Transaction fee versus transaction amount graph.*

Accordingly if *one single* transaction is made via the blockchain, as described above, it is cheaper to use a blockchain solution than a business credit card if one transacts above

900-1800 SEK with the current transaction fees. This is dependant on the percentual value between 1-2%. Similarly a CC solution is cheaper if one transacts above 300-600 SEK. With Lightning Network and near zero transaction fees the blockchain solutions will be much closer the latter numbers. The above amounts can be compared to the price of fueling a Scania truck, a cost which exceeds them [6]. Reparation and towing costs are likely to exceed the amount as well [8]. However paying by invoice instead of the proposed solutions is cheaper since it only requires one bank transaction rather than two.

It is thus profitable to use all the proposed solutions compared to a business credit card even making one single transaction. Additionally it has to be accounted for that one may batch several payments before redeeming and one is expected to buy more digital assets at once, implying more transaction fees are saved compared to the usage of a Visa business credit card. Similarly, with the current system hauliers may get one invoice a month per transaction counterparty [6, 7] and thus many transactions are batched. A pure invoice is cheaper, and the fee for the extra bank transaction is the cost for trust since any of the three proposed solutions can achieve real-time payments, minimizing credit risks.

6.8 Customer Usefulness

The opinion of the stakeholders is that particularly foreign hauliers are problematic for actors such as for instance workshops. One might not risk performing e.g. reparations without getting paid. Real-time payments would be appreciated by the tow truck companies, workshops and home detailers since it ensures payment. However it is also useful for the hauliers in case they are in reality creditworthy but are still temporarily denied service because the counterparty cannot verify it. It is costly for the hauliers to have vehicles standing without transporting the goods that is ought to be transported [6, 7, 9, 10]. Hence the market thinks it is an existing problem that would be appreciated if it was solved. Real-time payments are valuable to the market [8, 9, 10]. All solutions give customer usefulness since they solve the problem, to be able to transact in real-time within and across borders.

From the literature study and PoC it can be concluded that transactions on the Bitcoin blockchain may be handled in real-time, given that Lightning Network is deployed. The CC solution fulfills this as well. Consequently the system allows to quickly load and unload vehicles with assets, which is useful for the customers as stated above.

The actors on the market generally trust Scania [6, 7], this is useful for a Scania issued currency compared to Bitcoin. Through a study it was concluded that the volatility of Bitcoin is the most significant barrier toward adopting it as payment means [132].

The hauliers generally minimize the power given to their drivers regarding payments. Drivers usually only have a company card that only can be used for fuel on certain gas stations and only for certain vehicles, in order to avoid fraud [6, 7]. With a digital value on the vehicle a public blockchain cannot enforce such controls. On the contrary one may load and unload the vehicles with assets at any moment, thus the drivers may not scam their hauliers on more than the amount the hauliers loaded on the vehicle.

All the solutions rely on hauliers having digital assets, ScaniaCoins in case of the PD and CC solutions and bitcoins with a CD solution. If the hauliers do not have enough digital assets to finance e.g. reparations or towing, the same problem occurs as the one this thesis is trying to solve since no real-time payments can be made. However this only affects the hauliers desiring e.g. towing and reparations. Since there now exists a system where real-time payments can be made, the workshops for instance can refuse to perform reparations on vehicles they are unsure of will pay. The majority of hauliers are expected to have a sufficient buffer of digital assets considering that is profitable for them since their vehicles are avoided to be kept in custody. Consequently hauliers are forced to utilize the digital funds in case they want fast reparations. The consequences of not having enough digital assets are hence similar to if the haulier does not have any money, then the haulier's vehicle may be denied service until money is provided.

6.9 Trust

The trust to the third party is dependant on the initial trust to the issuer. It moreover is affected by the performance of the system. In a CC layout the third party is responsible for balance keeping, handling transactions, issuance and redemption of the asset. If the servers or databases are compromised users may lose trust. An authority is more trustworthy when it is distributed with separation of power [133], trusted third parties introduce security risks [134].

Using a blockchain an identity is bound to a set of key-pairs, meaning losing the keys loses the assets corresponding to the identity. With the keys only being stored on the device while a hardware fault occurs, the keys and thus assets may be lost [42]. If an attacker accesses one's private key, the attacker can steal all assets of the corresponding public key. Accordingly it is preferable to associate ownership with a person and not key-pairs [96]. This affects the trust of the system. A third party cannot control the actions on the blockchain, thus a wrongly issued transaction cannot be revoked [37] and may also affect the trust of the system.

Using a CD system, the payee needs to accept the currency Bitcoin for the payments. The trust in the system thus correlates with the trust of Bitcoin. Actors may not trust Bitcoin given that it is a volatile currency [4, 122, 135], particularly compared to other real world currencies [136]. With a central issuer of ScaniaCoins instead means the central party is responsible for maintaining the value of the digital asset. The trust regarding the value of the currency is thus moved from the decentralized Bitcoin to Scania. This introduces a third party risk, but the reliance on trust is minimized due to the security and auditability of the blockchain [96] and that the third party cannot control the blockchain, only the issuance and redemption. Moreover considering the magnitude of Scania the market generally trust Scania [6, 7].

6.10 Environmental Issues

Using a blockchain may affect the environment negatively. As can be read in more detail in *Appendix A.5*, Proof-of-Work wastes much resources when performing the Proof-of-

Work computations, but is currently still the most widely used consensus mechanism. A power consumption compared to Ireland's is required to power up the Bitcoin network [137]. The Bitcoin miners however position their sites close to the cheapest electricity sources for usage, which is power from hydropower plants [39]. On the contrary one cannot choose which electricity to receive from an outlet. With virtual mining as e.g. Proof-of-Stake the environmental issues should in the future be avoided as examined in *Appendix A.5*.

With the CC approach the power of the databases and servers is required, no Proof-of-Work is needed.

Chapter 7

Discussion

This chapter discusses and reflects upon the evaluation in chapter 6. Various aspects and consequences of the previous chapter's material are analysed.

7.1 Confidentiality

As written in *6.1* the blockchain does not ensure full confidentiality, inferring which actor is which is in theory possible even if it is difficult. Consequently it may be possible for unauthorized identities to track transactions made by others on the blockchain network. This is problematic if a haulier is having its vehicles pay for services or goods from profitable suppliers they do not wish to share with competitors. It may imply competitive advantages [59]. The competitors may derive and infer the suppliers from the blockchain transactions, this may affect the haulier's business negatively. On the contrary there is no proof a public address belongs to a certain actor, meaning one may draw the wrong conclusions from the interference, it is a pseudo-anonymous system [39, 121]. Moreover the solution motivated by the thesis problem statement is mainly focused on solving payments regarding towing, reparations and fueling. These transaction counterparties are not controversial to share. However sensible trades might not be suitable for a blockchain solution because of the risk of compromised confidentiality.

The upside of using a blockchain is that it is semi-transparent even in case of any potentially launched attack. However with a central issuer of colored coins using a blockchain, the central issuer has knowledge of the keys corresponding to the entities which redeem and obtain the ScaniaCoins as stated in *6.1*. If paper trails are stored, this information may be hacked meaning a successful attack on the central party exposes all identities that traded with the central party. On the contrary the third party may choose to not store this information since the blockchain works independently of it. Moreover since a wallet may contain several different keys, one cannot directly conclude the balances of the identities by only knowing the identity of one key and neither which parties transacted with each other. This is the inherited pseudo-anonymous property from Bitcoin [39]. Thus confidentiality is partly violated in case the central issuer is attacked, but the consequences are not major and the system will continue to be semi-transparent.

However a CC approach preserves confidentiality until the system is successfully attacked. An intruder may then access all the balances and the corresponding real identities as well as transactions made, thus counterparties that are traded with are exposed.

Hence the confidentiality of a blockchain is not optimal for all parties, while a CC solution relies on keeping the databases secure. Using a blockchain, the users may not appreciate the semi-transparency due to potentially exposed suppliers. The degree of disadvantage depends on the secrecy of the transaction counterparties.

A positive aspect with having the semi-transparency is that everyone can audit all the transactions and thus it should provide incentives to not cheat [3]. For instance the network may identify a performed 51% attack and one can thus avoid making business with the cheating party, meaning the attacker may experience a loss. On the contrary since everyone may audit the blockchain, one may not want to be coupled with transactions to parties with bad reputation. If an entity suddenly is coupled to a scandal, it might be unprofitable for the entity's counterparties indirectly.

Since the customers may want to avoid the semi-transparency, the CC approach should be the better one regarding this aspect [65]. The transparent transport project, mentioned in *3 Related Work*, utilizes a blockchain to track a supply chain to prove that goods were transported in an environmental friendly way. The project leader however highlighted that even if this is valuable to the overall audience and the end customers, different actors on the market might not want to expose their supply chains since it may entail a competitive advantage. This made the solution problematic to introduce in practice due to the blockchain [138]. However as stated e.g. towing, reparations and fueling ought to be uncontroversial, for that usage domain a blockchain solution is consequently not problematic.

Thus there are pros and cons regarding confidentiality and to use or not use a blockchain. It boils down to if the security of the central system exceeds the risk to potentially expose secret transaction counterparties on a blockchain. Managing to attack a central party implies the possibility to obtain everything, while the blockchain always provides the same information. Given the data concerning the use case the CC solution is preferred, however a blockchain solution is not a significant problem.

7.2 Integrity

As can be derived from *6.2*, a blockchain ensures integrity to a higher extent than a CC approach since it utilizes a hash chain and each transaction must be digitally signed by the adequate identity to be valid. Moreover a blockchain may entail a significant cost when modifying the database, for instance by using Proof-of-Work, implying it is difficult to tamper with data. With a centralized distributed database system the data is not bound to identities as written in *2.7*. If even one site is compromised integrity may be violated. With a blockchain one cannot spend nor claim assets of others unless having the private key corresponding to a certain public key as stated in *2.7*. Whereas a centralized system may use corrective approaches, meaning being able to handle faults afterwards, a blockchain solution relies solely on preventative security meaning to prevent unauthorized users to obtain the private key used for authentication [139]. As stated in *6.2* it is

however infeasible to find the private key by guessing. It may however be feasible to decrypt the password encrypted private keys stored on the device depending on how strong the password is. Using an offline attack one can find the password that encrypted the private keys, and thus obtain the private keys. With exponential backoff it ought to be very slow and time consuming. To launch such an attack the device holding the password encrypted keys must however be accessed first, and hence is an unlikely attack. If the device is stolen, the haulier ought to be notified and can thus unload the digital assets if the haulier has the stored key-pairs elsewhere. By the time the private keys are successfully decrypted by the attacker, the digital wallet does not hold any value. Moreover it is unlikely that the digital wallets hold significant amounts of assets since the idea is that a haulier only has a buffer for emergency cases, being able to transact to the vehicles in need when necessary.

In case the haulier stores their vehicles' private keys at other locations than solely on the vehicles, the storage must be secure as well. A successful attack to the storage implies control over the identities coupled to the accessed key-pairs, and thus violates integrity.

The haulier may thus decide for themselves what they would like to do with their key-pairs, freedom under responsibility. The power of keeping integrity of the hauliers' digital wallets is thus partly moved from the central party to the hauliers. With that said it is possible for the central party to provide some very secure key-storage that the hauliers can utilize if desired. Moreover to violate the integrity of the blockchain without unauthorized access to another's keys, is not possible due to the use of a hash chain and the mining power of the Bitcoin blockchain. Solutions building on the public blockchain should hence be considered a better choice than a CC solution regarding integrity since it provides a more secure system.

7.3 Availability

A CC system is vulnerable to attacks such as denial-of-service attacks, however attacking a public blockchain is infeasible considering it is completely decentralized. The consequence of it is that the payment system relying on the completely centralized approach may be inaccessible at times when being attacked meaning transactions cannot be performed. The same applies for more serious attacks, for instance if competitors successfully attack all the sites and destroy all data. The above is avoided with the public blockchain which due to the distribution and decentralization provides robustness. The approaches relying on the public blockchain thus do not risk to get availability compromised by attacks to the same extent, however routing attacks may deminish availability by attacking the Internet infrastructure. Attacks on the Internet infrastructure can nevertheless also be performed to affect the CC solution constituting of a distributed database, and should thus be neglected considering the availability comparison.

With a PD system using a centralized issuer of colored coins an attack to the central party can temporarily disrupt the issuance and redemption of the asset. Even if such an attack is strenuous, it would not be a major problem since transactions can still be performed in real-time. With a CC solution as a distributed database, it still provides ro-

bustness and good availability but not to the extent that the Bitcoin blockchain provides. Consequently a blockchain solution is preferred regarding this property, but a distributed database should be considered to be sufficiently robust.

With a blockchain solution an authorized user may lose the corresponding private keys, which in turn compromises availability. However given freedom under responsibility a blockchain solution is preferred in terms of availability due to the decentralization. The users may choose how to secure the keys, and Scania may provide a secure storage that can be used if desired.

7.4 Consistency

It can from *6.4* be derived that the blockchain contributes to consistency by avoiding unsolvable reconciliation processes. Thus the blockchain database is preferable since it is kept in consistent states without explicitly reconciling, which is a consequence of an easier data structure to reach consensus on as mentioned in *2.7*. In *2.6.2* it was pointed out that consistency issues are security threats to a distributed system, thus a blockchain is more secure regarding this aspect unless an eager commit protocol is used which on the other hand affects throughput negatively since a bottle-neck is introduced. However considering a traditional database has been tested and used in practise for decades, keeping consistency ought to not be problematic. Blockchains are on the contrary not equally mature, which is a disadvantage. No approach should accordingly entail any consistency problem.

7.5 Responsibility Issue and Immutability

Having transactions and balances handled by a completely decentralized network has pros and cons. Relying on the Bitcoin blockchain infrastructure to make transactions and keep balances is believed to be secure since Bitcoin provides significant security. However since it is decentralized it raises the question of who is responsible in case a fault occurs, for instance that a transaction is issued to the wrong receiver. A central party in the PD system cannot control the data on the blockchain, and may accordingly not revoke the faulty transaction. Similarly a CD system cannot revoke such transactions either.

The Bitcoin blockchain is practically immutable, thus the issued transactions are irrevocable. This is good when issuing a correct transaction, a payment should never be able to be revoked. On the contrary it is problematic in case of a faulty transaction such as assets being sent to the wrong address, since it cannot be revoked by anyone either. No one can be held responsible since no one and everyone are responsible together. That the system cannot handle this is impractical, and a limitation of such an approach. However it is freedom under responsibility, if a transaction is by mistake issued twice or to the wrong account it is the fault of the user. This can be compared to that a bank may in theory be able to withdraw a transaction, however it might not be legally able to if the receiver does not accept sending it back as stated in *6.5*. Thus a blockchain transaction is similar to a regular bank transaction regarding that aspect. However this may affect

the trust for the system negatively. On the contrary the possibility to mutate afterwards may be problematic as illustrated by the following scenario. X may buy goods from Y, X transacts via the bank to Y who then sends goods to X. If X now claims the payment was issued wrongly and that no goods were received, then it could become a legal process. It happened that Y lost both money and goods [7, 9, 10], this is something that is avoided with solutions building on the Bitcoin blockchain.

That a transaction is immutable might thus be a problem, it can be problematic if someone steals an identity's wallet of keys and spends the assets. One may compare it to a thief stealing a bank card and spending the money, then one would like if it was possible to restore. The case where one issues transactions two times by mistake can however be solved even with the immutable blockchain. Given that the payer and the payee have mutual trust, the payee will repay the extra faulty transaction since they want to keep the business relationship. For instance it might be unprofitable for a gas station to scam a haulier once, and then lose the business for all future. This kind of repayments are usual today using the bank system [7]. In fact this may be used when performing reparations without knowledge of the exact cost before the reparation. The haulier may deposit an amount that exceeds the expected cost. After the workshop has repaired the vehicle the remaining amount is transacted back.

It should be noted that most likely a vehicle is not loaded with significant value, which means the stake ought to be low in case a problem occurs and should thus decrease the trust loss slightly. Moreover such faults as double payments ought to occur rarely and Scania may compensate depending on the situation. In a central system the central party can handle the problem, however the possibility to mutate data also implies that data ought to be immutable also is mutable with the adequate access rights which creates a security issue.

The immutability property that is provided by the blockchain is valuable, however the trade-off is the responsibility issue. Considering that the assets in the system is believed to be of low value and be temporary it is a trade-off that should be considered to be adequate.

7.6 Cost

From *6.7* it can be derived that a CC approach to ScaniaCoins is cheaper than the rest of the proposed solutions. Only two bank transactions is required, and no blockchain transactions. However as mentioned in the same section, the high cost of a Bitcoin blockchain transaction is due to the supply of too few mined transactions and demand of too many issued transactions. Many actors want to issue transactions while only a limited amount of transactions can be put into blocks at the time, thus the transaction fees increase. With Lightning Network infinite number of transactions may be performed at any given time, which ought to solve the problem and consequently lower the transaction fees significantly meaning the cost for a transaction will be near zero [39]. Thus the CC solution should be cheaper, however both of the blockchain solutions will only be slightly more expensive which can be considered negligible. Accordingly from a cost perspective, the solutions should be approximately similar.

All the three solutions however outperform business credit cards as concluded from *6.7* given that the transaction amount is not negligible. The payee would be satisfied with a credit card payment since the payment is temporarily done by a bank, however the drivers are generally not trusted with such cards. Using invoice instead is preferable for the payer, but not for the payee since it implies a risk. This is where a third party at the moment has to ensure payment. With any of the three proposed solutions one can transact in real-time, removing the risk. The cost for this is one extra bank transaction of in total 3 SEK, 1.5 SEK for the sender and for the receiver respectively, plus potentially transaction fees on the blockchain which as reasoned above should be negligible in the future. Thus transaction cost-wise the three proposed solutions are slightly more expensive than a bank payment through an invoice, however the value of removing the credit risk should be considered to exceed the required extra fee.

7.7 Customer Usefulness

As can be derived from *6.8* all fair parties believe it is a useful system. The workshops and tow truck companies do not want to do anything without getting paid. From the payee's perspective they do not want to be victim to fraud, thus it is useful with real-time payments. Keeping the vehicle in custody as a security can thus be done until receiving the money, which can take several days. This is prolix for everyone involved. For instance one has to keep a spot for the confiscated vehicle. It moreover becomes costly for the hauliers since their vehicles ought to be out in traffic to earn money. Creditworthy hauliers are temporarily denied service on their vehicles, which they want to avoid. Thus a solution to the problem ought to be proposed, which is done through this thesis and implies customer usefulness. All three solutions should solve the problem.

Using ScaniaCoins, the $C bought by the hauliers are not affected by the interest rate, consequently one may view it as hauliers lending money to Scania for free. However the incentives for having ScaniaCoins is that it is possible to make instant payments, consequently it is avoided for hauliers to have their vehicles suspended temporarily until the payments are confirmed on the receivers' accounts. This is, as mentioned above, very costly for the hauliers and the value of instant payments ought to exceed the earnings from interest rate.

7.8 Trust

A difference between the PD system and the CD system is a colored coin, ScaniaCoin, solution versus a Bitcoin solution. Both of the currencies are transacted and kept on the blockchain. For the colored coins solution there must be a trust in the central party to issue and redeem the ScaniaCoins. A solution building on the cryptocurrency of the blockchain does not have that problem since the currency is completely decentralized. On the contrary, that there is only one exchange party also has an advantage. Hauliers are usually restrictive with payment means for their drivers, to avoid fraud [7]. With ScaniaCoins the assets may only be utilized when trading with certain actors that have redemption agreements with Scania. No payee would accept ScaniaCoins as payment without be-

ing able to redeem them. The drivers may thus not use the payment means to anything else than Scania certified actors, which ought to reduce the risk of fraud. With a Bitcoin solution, the counterparties are not restricted. Moreover that the value loaded on the vehicle is low, increases the hauliers' trust for the system. With a low value loaded onto the truck, no significant loss can occur in case of fraud. This can be compared to the usage of a business credit card where one can shop anywhere up to a certain limit.

As written in *6.9* Bitcoin is a volatile currency, meaning a usage of the currency for a solution implies that trust must be placed in the currency. If the value of the currency suddenly significantly drops, the hauliers' assets may lose value. With a colored coin solution instead one has to place trust in Scania to maintain the value of the currency, there is a responsible party. However Scania ought to be trusted on the market [6, 7]. For instance one may institute a legal contract between Scania and the actors participating in the system that Scania ought to redeem all ScaniaCoins to a specific exchange rate as stated in *5.5 Currency Exchange*. Thus this increases trust.

Another trust issue is that since the Bitcoin blockchain is decentralized without a central entity, it opened up for criminal activities [125] since the flow of data cannot be controlled as mentioned in *6.5*. Scania getting coupled to criminal activities is not preferable, since that would affect the trust to the system negatively. With a CD solution it is difficult to track the assets, since the exchange of the currency is also decentralized. Thus criminals can trade without any paper trails [125]. With a central party as a colored coin issuer, all exchanges has to go through the third party and hence leave paper trails. This ought to minimize the threat for ScaniaCoins being used in criminal spheres compared to if Bitcoin was used [39], and the ScaniaCoins solution is thus preferable. More details about how to issue ScaniaCoins, coupled to trust, can be found in *Appendix A.8*.

7.9 Environmental Issues

The environmental issues, examined in *6.10*, suggest that building on the current Bitcoin blockchain is not environmental friendly and that the CC approach is preferred. A consensus algorithm not utilizing the power consuming Proof-of-Work should however in the future replace Proof-of-Work as an *industry standard*. An example is for instance a Proof-of-Stake algorithm, but it is yet unclear if it will happen in practice [39]. Much research is done within the area, e.g. Kiayias et al. claim to prove that their Proof-of-Stake protocol ensures rigorous security [140]. The environmental problems the current Bitcoin blockchain suffers from should thus not be an issue in the future, it ought to be solved. Accordingly the environmental aspect is not essential to the discussion.

7.10 Relevance of the Work

This thesis is mainly relevant to actors within the transport industry. Foremost it is relevant to hauliers, home detailers, tow truck companies and repair workshops since those are the involved parties in the problem motivating this thesis. The solution provided in the thesis may solve creditworthiness issues through the possibility of immutable real-time transactions, thus it is relevant to the transport industry as such.

Regarding the blockchain technology, a deeper understanding for the usage of it within the business is provided through the work. Problems with using a blockchain technology within the transport industry has been highlighted, in particular the confidentiality problem. Moreover the content is also useful and applicable for other businesses where it is to be decided if a blockchain solution is superior to other solutions. A foundation for it is provided through the material provided by the thesis.

Furthermore an understanding of the pros and cons for a PD system has been highlighted through this thesis, and thus the thesis provides information applicable to future projects in closely related domains.

7.11 Source Criticism

Considering blockchain is tightly coupled to Bitcoin, the vast majority of the material is Bitcoin centered which has several consequences. Partly it means that one unconsciously starts to connect blockchain to Bitcoin as well, even being aware of it. The thesis was however carried out using the Bitcoin blockchain meaning it might not be too problematic, especially since it was actively aimed to separate Bitcoin and blockchain.

Moreover blockchain is a term which is coupled to a hype, meaning the technology may often be exaggerated. The domain has become a big market where money can be made. This regards both implementations of blockchain solutions as well as consulting and idea brainstorming of how the technology can be used. In particular Bitcoin is a currency with many people heavily invested in, both emotionally and economically. This creates incentives to promote blockchain and Bitcoin impacting the material available which influences the thesis indirectly. However with the knowledge of it, the thesis still aims to keep the objectivity.

Chapter 8

Conclusion

This chapter summarizes the conclusions which can be drawn from chapter 7 and presents them concisely.

That the whole industry desires a real-time payment system within and across borders is concluded, and solutions are proposed in this thesis. The blockchain solutions, PD and CD, are more secure considering the robustness and strength of the decentralized blockchain. Integrity and availability are properties where the blockchain solutions have a clear advantage. Concerning the consistency, neither of the solutions is concluded to have a problem with keeping the consistency of the database.

Regarding the responsibility issue coupled to the immutability of the blockchain, it is problematic however the trade-off should be considered adequate since the immutability property is very valuable. Comparisons for cost and environmental issues between the different solutions should be considered negligible since it is believed the aspects will not be issues in the future. Concerning the cost it should be concluded that the extra required fee for enabling real-time payments is exceeded by the value of through this removing the credit risk.

As for the difference between a ScaniaCoin solution versus a pure Bitcoin solution, it is a matter of trust. Since Scania is considered a trustworthy entity the ScaniaCoin solution is considered the better since the currency is not equally volatile. Moreover the risk of criminal activity using ScaniaCoin is decreased versus the decentralized Bitcoin. However a disadvantage is that the exchange point being central makes it vulnerable to attacks, however it is not problematic that exchanging can be inaccessible temporarily while transactions still can be made. A single exchange point on the contrary implies an advantage since only Scania certified actors may redeem ScaniaCoins. Thus hauliers can be sure that assets loaded on the vehicle only will be used as payments means to certain actors, e.g. workshops, which reduces the risk for fraud. ScaniaCoin is concluded more suitable than Bitcoin.

The semi-transparency provided on a blockchain is problematic, and should not be considered perfectly optimal. However it depends on what the users will use the currency on the blockchain for. All controversial trades cannot be performed using a blockchain solution since the users may risk to expose transaction counterparties, for instance prof-

itable suppliers. This limits the blockchain solutions significantly. The CC solution has two advantages over the blockchain solutions. Firstly that wrongly issued transactions can be handled and secondly that it is not transparent as a blockchain solution. However regarding the latter, the use case motivating this thesis does not impose a big problem since e.g. towing, reparations and fueling ought to be uncontroversial. Thus a blockchain solution can be utilized instead of a CC solution. The advantages of using a public decentralized blockchain exceed the disadvantages and should consequently be considered most appropriate given uncontroversial trades.

To conclude and answer the problem statement of the thesis, the proposed PD solution fulfills security very well except for confidentiality considering the semi-transparency of the blockchain. This makes a blockchain solution problematic, and customers may avoid to use such a solution since they do not want to risk exposing transaction counterparties to competitors. This affects customer usefulness negatively compared to a CC solution. However given that the transactions made are uncontroversial this ought to be no problem. The solution is more robust, improving the customer usefulness compared to a CC solution slightly. The solution should also be preferable over a clean Bitcoin solution (a CD solution) for several reasons. Customer usefulness is improved since the currency is less volatile, to a higher extent prevents criminal activity and reduces the risk for fraud. Only having one exchange point, Scania, is not problematic since Scania is considered a trustworthy party by the market.

Chapter 9

Future Work

This chapter briefly highlights future work possibilities within the transport industry and the blockchain domain. Further some future work regarding the subject of this thesis is presented.

As mentioned in *1.3 Delimitations* only the payment application was implemented in this thesis. Thus to get the full system running, the exchange system needs to be implemented as well. The functionality required there would be similar to an ordinary Internet shop. The customers are today having accounts to Scania's Internet platform, the ScaniaCoin exchange service would also be added to the existing platform. The trustworthy customers would be able to buy the ScaniaCoins and pay by invoice, receiving the ScaniaCoins instantly. Moreover for the rest of the customers an option should exist for paying by invoice, however the ScaniaCoins would only be provided when the money is confirmed on Scania's account. Automatisation for this process would be preferable. The users would further require to enter bank details used when ScaniaCoins are redeemed for real money, thus when selling ScaniaCoins. When buying ScaniaCoins the buyer must specify the public key where the buyer wants to receive the ScaniaCoins.

Concerning general future work, the blockchain applications out are mostly dealing with transactions of ownership. The reason is because the blockchain technology was invented in that domain. Thus future work should aim to find applications of the blockchain technology that separates it from use cases where ownership is transacted. During this thesis a suitable domain is however identified that the blockchain technology adds value in, compared to other distributed solutions. Given the properties of the blockchain, it is good at handling time series in a distributed database. Hence data which benefits from a proven chronological order of events is suitable for a blockchain, e.g. a datalog. The data must however be allowed to be public to all nodes participating in the blockchain since everything is viewable by everyone. With that said one can create a closed, *private*, blockchain with only invited participants as mentioned in *Appendix A.2*. Moreover the submitted data can be encrypted, meaning not everyone may access the plaintext. Thus analysis of the blockchain technology in the domain of time series is proposed for future work. For the transport industry this may for instance include service journals for vehicles and driving journals. Further it is suggested to examine storing tachograph data on a blockchain, since it is a problem with cheating in this domain. A problem here is to verify that the data submitted to the blockchain by a vehicle is correct.

Bibliography

[1] Bitcoin. Bitcoin - open source p2p money, 2017. URL https://bitcoin.org/en/. [Online; accessed 27-January-2017].

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. URL https://bitcoin.org/bitcoin.pdf. [Online; accessed 18-January-2017].

[3] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4, 2016.

[4] Melanie Swan. *Blockchain - Blueprint for a New Economy*. O'Reilly Media, Inc, 2015. ISBN 978-1491920497.

[5] Erik Hillbom and Tobias Tillström. Applications of smart-contracts and smart-property utilizing blockchains. Master's thesis, Chalmers, 2016. URL http://www.the-blockchain.com/docs/Applications%20of%20smart-contracts%20and%20smart-property%20utilizing%20blockchains.pdf. [Online; accessed 02-March-2017].

[6] Johan Nikolausson. Interview with the haulier Transab about payments and creditworthiness, 2017.

[7] Jan Björklund. Interview with the Managing Director of Scania Transportlaboratorium AB, Scania's own haulier about payments and creditworthiness, 2017.

[8] Michael Hedgren. Interview with IS/IT Manager | Commercial Operations, Scania Assistance | Scania CV AB about creditworthiness and payments for foreign drivers and hauliers, 2017.

[9] Donald Pihlblad. Interview with a chief over three towing companies driving for Assistanskåren about towing, reparations and creditworthiness, 2017.

[10] Emanuel Yacoub. Interview with the accounting and credit manager on Scania (Bilar) Sverige about towing, reparations and creditworthiness, 2017.

[11] ASB. International payments, 2015. URL https://www.asb.co.nz/help/how-long-it-takes-for-international-money-transfer-to-arrive.html. [Online; accessed 01-March-2017].

[12] Lloyds Bank. International payments, 2017. URL https://www.lloydsbank.com/online-banking/benefits-online-banking/international-payments.asp#collapse1-1438975317840. [Online; accessed 01-March-2017].

[13] Chin-Ling Chen, Wei-Chen Tsai, Yu-Yi Chen, and Woei-Jiunn Tsaur. Using a stored-value card to provide an added-value service of payment protocol in vanet. *Information Technology and Control*, 42(4), 2013.

[14] SEB. Överföringar mellan banker och bryttider, 2017. URL https://seb.se/privat/betala/betala-rakningar/overforingar-och-bryttider. [Online; accessed 31-March-2017].

[15] Oleg Mazonka. Blockchain: Simple explanation, 2016. URL http://jrxv.net/x/16/blockchain-gentle-introduction.pdf. [Online; accessed 26-January-2017].

[16] Sari Stern Greene. *Security Policies and Procedures Principles and Practices*. Pearson Prentice Hall, 1 edition, 2006. ISBN 9780131866911.

[17] Anoop MS. Elliptic curve cryptography, 2015. URL http://www.infosecwriters.com/text_resources/pdf/Elliptic_Curve_AnnopMS.pdf. [Online; accessed 26-January-2017].

[18] Edward W. Felten. Digital signatures. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/bx6si/digital-signatures. [Online Presentation; accessed 26-January-2017].

[19] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman & Hall/CRC Taylor & Francis Group, 1 edition, 2006. ISBN 978-1-58488-518-4.

[20] Edward W. Felten. Cryptographic hash functions. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/gFEJL/cryptographic-hash-functions. [Online Presentation; accessed 26-January-2017].

[21] Edward W. Felten. Hash pointers and datastructures. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/EYEAo/hash-pointers-and-data-structures. [Online Presentation; accessed 26-January-2017].

[22] Vitalik Buterin. Ethereum white paper, 2014. URL http://www.the-blockchain.com/docs/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf. [Online; accessed 26-January-2017].

[23] Joseph Bonneau. The bitcoin network. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/SKxAO/the-bitcoin-network. [Online Presentation; accessed 31-January-2017].

[24] Lykke. Colored coins exchange white paper, 2016. URL https://www.lykke.com/colored_coins_exchange_white_paper. [Online; accessed 10-February-2017].

[25] Ludvig Backlund. A technical overview of distributed ledger technologies in the nordic capital market. Master's thesis, Uppsala University, 2016. URL http:

//www.diva-portal.org/smash/get/diva2:947471/FULLTEXT01.pdf. [Online; accessed 15-February-2017].

[26] Joseph Bonneau. Bitcoin transactions. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/neykl/bitcoin-transactions. [Online Presentation; accessed 25-January-2017].

[27] Edward W. Felten. A simple cryptocurrency. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/rJ8KJ/a-simple-cryptocurrency. [Online Presentation; accessed 26-January-2017].

[28] Joseph Bonneau. Bitcoin blocks. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/I5uc7/bitcoin-blocks. [Online Presentation; accessed 27-January-2017].

[29] M. Tamer Özsu and Patrick Valduriez. *Principles of Distributed Database Systems*. Pearson Education, Inc, 3 edition, 2011. ISBN 978-1-4419-8834-8.

[30] Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. *Database Systems: The Complete Book*. Pearson, 2 edition, 2008. ISBN 9780131873254.

[31] Joseph Bonneau. The bitcoin network. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/SKxAO/the-bitcoin-network. [Online Presentation; accessed 27-January-2017].

[32] Daniel Drescher. *Blockchain Basics*. Apress, 2017. ISBN 978-1-4842-2604-9.

[33] Saveen Abeyratne and Radmehr Monfared. Blockchain ready manufacturing supply chain using distributed ledger. *International Journal of Research in Engineering and Technology*, 9(5), 2016.

[34] Richard Gendal Brown. On distributed databases and distributed ledgers, 2016. URL https://gendal.me/2016/11/08/on-distributed-databases-and-distributed-ledgers/. [Online; accessed 12-April-2017].

[35] Ludvig Öberg. Blockchain for business. Dataföreningen, 2017. [Presentation].

[36] Ghassan Karame, Elli Androulaki, and Srdjan Capkun. Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin, 2012. URL https://eprint.iacr.org/2012/248.pdf. [Online; accessed 31-January-2017].

[37] Peter Waher. Interview with an Internet of Things expert and e.g. the author of Learning Internet of Things (ISBN: 978-1783553532), 2017.

[38] Thibaut Lajoie-Mazenc. Increasing the robustness of the bitcoin crypto-system in presence of undesirable behaviours. Master's thesis, KTH, 2016. URL http://www.diva-portal.org/smash/get/diva2:1051879/FULLTEXT01.pdf. [Online; accessed 17-April-2017].

[39] Ludvig Öberg. Interview with Co-Founder Swedish Bitcoin Association and Satoshi Square Stockholm, 2017.

[40] Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun. Bitter to better — how to make bitcoin a better currency. In *Financial Cryptography and Data Security*, volume

7397 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-32945-6.

[41] Arvind Narayanan. Consensus without identity: the block chain. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/71F3l/consensus-without-identity-the-block-chain. [Online Presentation; accessed 30-January-2017].

[42] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten, editors. *SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies*, IEEE Symposium on Security and Privacy, 2015.

[43] Bitcoin. Bitcoin developer guide, 2017. URL https://bitcoin.org/en/developer-guide. [Online; accessed 30-January-2017].

[44] Kourosh Davarpanah, Dan Kaufman, and Ophelie Pubellier. Neucoin: the first secure, cost-efficient and decentralized cryptocurrency, 2015. URL http://www.neucoin.org/en/whitepaper/. [Online; accessed 01-February-2017].

[45] Andrew Miller. Essential puzzle requirements. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/5Gyi1/essential-puzzle-requirements. [Online Presentation; accessed 30-January-2017].

[46] Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake, 2012. URL https://peercoin.net/assets/paper/peercoin-paper.pdf. [Online; accessed 30-January-2017].

[47] Andrew Miller. Proof-of-stake virtual mining. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/UAFGS/proof-of-stake-virtual-mining. [Online Presentation; accessed 30-January-2017].

[48] Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. Cryptocurrencies without proof of work, 2014. URL https://arxiv.org/pdf/1406.5694.pdf. [Online; accessed 30-January-2017].

[49] Vitalik Buterin. Proof of stake faq, 2016. URL https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ. [Online; accessed 30-January-2017].

[50] Andruiman. Pos forging algorithms: multi-strategy forging, 2015. URL https://www.scribd.com/document/256072839/PoS-forging-algorithms-multi-strategy-forging-and-related-security-issues. [Online; accessed 01-February-2017].

[51] Arvind Narayanan. Putting it all together. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/serPk/putting-it-all-together. [Online Presentation; accessed 30-January-2017].

[52] George Bissias, Brian Levine, A.Pinar Ozisik, and Gavin Andresen. An analysis of attacks on blockchain consensus (draft), 2016. URL https://arxiv.org/pdf/1610.07985.pdf. [Online; accessed 23-February-2017].

[53] Maria Apostolaki, Aviv Zohar, and Laurent Vanbever. Hijacking bitcoin: Routing attacks on cryptocurrencies, 2017. URL https://arxiv.org/pdf/1605.07524v2.pdf. [Online; accessed 18-April-2017].

[54] Yong Yuan and Fei-Yue Wang. Towards blockchain-based intelligent transportation systems. volume 19 of *International Conference on Intelligent Transportation Systems*. IEEE, 2016. ISBN 978-1-5090-1889-5.

[55] Adi Ben-Ari. Blockchain for cross-organisation workflow, 2015. URL http://appliedblockchain.com/blockchain-for-cross-organisation-workflow/. [Online; accessed 07-February-2017].

[56] Dr. Gideon Greenspan. Why many smart contract use cases are simply impossible, 2016. URL http://www.coindesk.com/three-smart-contract-misconceptions/. [Online; accessed 07-February-2017].

[57] Bitcoin. Wallet, 2016. URL https://en.bitcoin.it/wiki/Wallet. [Online; accessed 24-March-2017].

[58] Jaume Barcelo. User privacy in the public bitcoin blockchain, 2017. URL http://inpluslab.sysu.edu.cn/files/Paper/Technology/User_Privacy_In_The_Public_Bitcoin_Blockchain.pdf. [Online; accessed 11-April-2017].

[59] S. Matthew English and Ehsan Nezhadian. Conditions of full disclosure: The blockchain remuneration model, 2017. URL https://arxiv.org/pdf/1703.04196.pdf. [Online; accessed 18-April-2017].

[60] Ross Anderson. *Security Engineering*. Wiley Publishing Inc, 2 edition, 2008. ISBN 978-0-470-06852-6.

[61] Dr. Gideon Greenspan. Ending the bitcoin vs blockchain debate. MultiChain, 2015. URL http://www.multichain.com/blog/2015/07/bitcoin-vs-blockchain-debate/. [Online; accessed 06-February-2017].

[62] Anastassia Ailamaki. Replication, 2002. URL http://www.cs.cmu.edu/~natassa/courses/15-823/F02/papers/replication.pdf. [Online; accessed 06-February-2017].

[63] Jim Gray and Leslie Lamport. Consensus on transaction commit. Microsoft Research, 2004. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2003-96.pdf. [Online; accessed 06-February-2017].

[64] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. Blockbench: A framework for analyzing private blockchains, 2017. URL https://arxiv.org/pdf/1703.04057.pdf. [Online; accessed 18-April-2017].

[65] Dr. Gideon Greenspan. Blockchains vs centralized databases. MultiChain, 2016. URL http://www.multichain.com/blog/2016/03/blockchains-vs-centralized-databases/. [Online; accessed 07-February-2017].

[66] BitFury Group and Jeff Garzik. Public versus private blockchains, 2015. URL http://bitfury.com/content/5-white-papers-research/public-vs-private-pt1-1.pdf. [Online; accessed 07-February-2017].

[67] Dr. Gideon Greenspan. Citychain17 - what is the difference between a blockchain and a database. MBN Solutions, 2017. URL https://www.youtube.com/watch?v=NK5Fz3w-H4o. [Online Presentation; accessed 25-April-2017].

[68] Gareth Peters and Efstathios Panayi. Understanding modern banking ledgers through blockchain, 2015. URL https://arxiv.org/pdf/1511.05740.pdf. [Online; accessed 27-January-2017].

[69] Xiwei Xu, Cesare Pautasso, Liming Zhu, Vincent Gramoli, Alexander Ponomarev, An Binh Tran, and Shiping Chen. The blockchain as a software connector. volume 13 of *Working IEEE/IFIP Conference on Software Architecture*. IEEE, 2016. ISBN 978-1-5090-2131-4.

[70] Trent McConaghy, Rodolphe Marques, Andreas Müller, Dimitri De Jonghe, T. Troy McConaghy, Greg McMullen, Ryan Henderson, Sylvain Bellemare, and Alberto Granzotto. Bigchaindb: A scalable blockchain database, 2016. URL https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf. [Online; accessed 21-February-2017].

[71] Marko Vukolic. The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *Open Problems in Network Security*, volume 9591 of *Lecture Notes in Computer Science*. Springer, Cham, 2015. ISBN 978-3-319-39028-4.

[72] Meni Rosenfeld. Overview of colored coins, 2012. URL https://bitcoil.co.il/BitcoinX.pdf. [Online; accessed 09-February-2017].

[73] Yoni Assia, Vitalik Buterin, Meni Rosenfeld, and Rotem Lev. Colored coins whitepaper- digital assets, 2016. URL https://docs.google.com/document/d/1AnkP_cVZTCMLIzw4DvsW6M8Q2JC0lIzrTLuoWu2z1BE/edit#heading=h.wxrvzqj8997r. [Online; accessed 10-February-2017].

[74] Assaf Shomer. The colored coins protocol, 2015. URL https://github.com/Colored-Coins/Colored-Coins-Protocol-Specification/wiki. [Online; accessed 10-February-2017].

[75] Feng Xia, Yang, Laurence, Lizhe Wang, and Alexey Vinel. Internet of things. *International Journal of Communcation Systems*, 25(9), 2012.

[76] Scania Group. The next big data thing, 2017. URL https://www.scania.com/group/en/the-next-big-data-thing/. [Online; accessed 10-February-2017].

[77] Andrew Miller, Iddo Bentov, Ranjit Kumaresan, and Patrick McCorry. Sprites: Payment channels that go faster than lightning, 2017. URL https://arxiv.org/pdf/1702.05812.pdf. [Online; accessed 23-February-2017].

[78] Joshua Lind, Ittay Eyal, Peter Pietzuch, and Emin Gün Sirer. Teechan: Payment channels using trusted execution environments, 2017. URL https://arxiv.org/pdf/1612.07766.pdf. [Online; accessed 07-March-2017].

[79] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, 2016. URL https://lightning.network/lightning-network-paper.pdf. [Online; accessed 13-February-2017].

[80] Joseph Poon and Thaddeus Dryja. Scaling bitcoin to billions of transactions per day. SF Bitcoin Devs Seminar, 2015. URL https://www.youtube.com/watch?v=8zVzw912wPo. [Online Presentation; accessed 10-February-2017].

[81] Aaron van Wirdum. Understanding the lightning network, part 2: Creating the network. Bitcoin Magazine, 2016. URL https://bitcoinmagazine.com/articles/understanding-the-lightning-network-part-creating-the-network-14653269 [Online; accessed 15-February-2017].

[82] Litecoin Association. The lightning network explained (litecoin/bitcoin), 2017. URL https://www.youtube.com/watch?v=MpfvhiqFw7A. [Online Presentation; accessed 15-February-2017].

[83] Bitcoin Core. Segregated witness benefits, 2016. URL https://bitcoincore.org/en/2016/01/26/segwit-benefits/. [Online; accessed 16-February-2017].

[84] Jamie Redman. The lightning network alpha release is ready for testing, 2017. URL https://news.bitcoin.com/lightning-network-alpha-release/. [Online; accessed 13-February-2017].

[85] Lightning Labs. Alpha release of the lightning network daemon, 2017. URL http://lightning.community/release/software/lnd/lightning/2017/01/10/lightning-network-daemon-alpha-release/. [Online; accessed 13-February-2017].

[86] Kristov Atlas. The inevitability of privacy in lightning networks, 2017. URL https://www.kristovatlas.com/the-inevitability-of-privacy-in-lightning-networks/. [Online; accessed 13-February-2017].

[87] Amina Badzar. Blockchain for securing sustainable transport contracts and supply chain transparency. Master's thesis, Lund University, 2016. URL https://lup.lub.lu.se/student-papers/search/publication/8880383. [Online; accessed 27-April-2017].

[88] Hilda Hultén. Ny app för schyssta transporter. IntelligentLogistik, 2016. URL http://intelligentlogistik.com/forskning/ny-app-for-schyssta-transporter/. [Online; accessed 25-January-2017].

[89] Sean Rowan, Michael Clear, Mario Gerla, Meriel Huggard, and Ciaran Mc Goldrick. Securing vehicle to vehicle communications using blockchain through visible light and acoustic side-channels, 2017. URL https://arxiv.org/pdf/1704.02553.pdf. [Online; accessed 18-April-2017].

[90] Ali Dorri, Marco Steger, Salil Kanhere, and Raja Jurdak. Blockchain: A distributed solution to automotive security and privacy, 2017. URL https://arxiv.org/ftp/arxiv/papers/1704/1704.00073.pdf. [Online; accessed 18-April-2017].

[91] Pradip Kumar Sharma, Seo Yeon Moon, and Jong Hyuk Park. Block-vn: A distributed blockchain based vehicular network architecture in smart city. *Journal of Information Processing Systems*, 13(1), 2017.

[92] Olle Holm. Bosch skapar hjärnan till framtidens självkörande bilar. Motormagasinet, 2017. URL http://www.motormagasinet.se/article/view/420248/bosch_skapar_hjarnan_till_framtidens_sjalvkorande_bilar#. [Online; accessed 25-April-2017].

[93] LHV. Brief history of lhv, 2017. URL https://www.lhv.ee/en/about/. [Online; accessed 02-March-2017].

[94] Cuber. Cuber – lhv bank started public use of blockchain technology by issuing securities, 2015. URL http://www.cuber.ee/en_US/news/. [Online; accessed 13-February-2017].

[95] Aanchal Anand, Matthew McKibbin, and Frank Pichel. Colored coins: Bitcoin, blockchain, and land administration. Annual World Bank Conference on Land and Poverty, 2016.

[96] Alex Mizrah. A blockchain-based property ownership recording system, 2015. URL http://chromaway.com/papers/A-blockchain-based-property-registry.pdf. [Online; accessed 02-March-2017].

[97] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 2007. URL http://www.tandfonline.com/doi/abs/10.2753/MIS0742-1222240302.

[98] Balsamiq, 2017. URL https://balsamiq.com/. [Online; accessed 06-March-2017].

[99] Joseph Bonneau. Bitcoin as smart property. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/YJLvq/bitcoin-as-smart-property. [Online Presentation; accessed 12-February-2017].

[100] Bitcoin. Coinprism, 2015. URL https://en.bitcoin.it/wiki/Coinprism. [Online; accessed 06-March-2017].

[101] Flavien Charlon. Open assets protocol (oap/1.0), 2013. URL https://github.com/OpenAssets/open-assets-protocol/blob/master/specification.mediawiki. [Online; accessed 06-March-2017].

[102] Massimo Bartoletti and Livio Pompianu. An analysis of bitcoin op return metadata, 2017. URL https://arxiv.org/pdf/1702.01024.pdf. [Online; accessed 06-March-2017].

[103] Coinprism. Api blueprint, 2017. URL http://docs.coinprism.apiary.io/. [Online; accessed 07-March-2017].

[104] Fergal Reid and Martin Harrigan. An analysis of anonymity in the bitcoin sys-

tem, 2012. URL https://arxiv.org/pdf/1107.4524.pdf%3Forigin%3Dpublication_detail. [Online; accessed 07-March-2017].

[105] Coloredcoins. Documentation, 2017. URL http://coloredcoins.org/documentation/. [Online; accessed 07-March-2017].

[106] Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full aes-192 and aes-256. In *Advances in Cryptology - ASIACRYPT*, volume 5912 of *International Conference on the Theory and Application of Cryptology and Information Security*. Springer-Verlag Berlin Heidelberg, 2009. ISBN 978-3-642-10365-0.

[107] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full sha-1. In *Advances in Cryptology – CRYPTO*, volume 3621 of *Annual International Cryptology Conference*. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-28114-6.

[108] Morris Dworkin. Recommendation for block cipher modes of operation - methods and techniques, 2001. Special Publication 800-38A.

[109] B. Kaliski. Pkcs #5: Password-based cryptography specification version 2.0. The Internet Society, 2000. URL https://tools.ietf.org/html/rfc2898. [Online; accessed 20-March-2017].

[110] Bitcoin. Wallet import format, 2015. URL https://en.bitcoin.it/wiki/Wallet_import_format. [Online; accessed 07-March-2017].

[111] Hend S. Al-Khalifa. Utilizing qr code and mobile phones for blinds and visually impaired people. In *Computers Helping People with Special Needs*, volume 5105 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008.

[112] Google. Barcode api overview, 2017. URL https://developers.google.com/vision/android/barcodes-overview. [Online; accessed 07-March-2017].

[113] Edward W. Felten. Public keys as identities. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/HvhIA/public-keys-as-identities. [Online Presentation; accessed 26-January-2017].

[114] David Shrier, Weige Wu, and Alex Pentland. Blockchain & infrastructure (identity, data security) part 3. Massachusetts Institute of Technology, 2016. URL http://inpluslab.sysu.edu.cn/files/Paper/Summary/Blockchain___Infrastructure__Identity__Data_Security_.pdf. [Online; accessed 11-April-2017].

[115] Johnny Dilley, Andrew Poelstra, Jonathan Wilkins, Marta Piekarska, Ben Gorlick, and Mark Friedenbach. Strong federations: An interoperable blockchain solution to centralized third-party risks, 2017. URL https://arxiv.org/pdf/1612.05491.pdf. [Online; accessed 20-April-2017].

[116] Andrew Poelstra. Bitcoin birthday, 2013. URL https://download.wpsoftware.net/bitcoin-birthday.pdf. [Online; accessed 13-March-2017].

[117] George Watkins. Implementing the exponential backoff algorithm to thwart dictionary attacks, 2011. URL https://devcentral.f5.com/articles/implementing-the-exponential-backoff-algorithm-to-thwart-dictionary-att [Online; accessed 28-March-2017].

[118] Shayan Eskandari, David Barrera, Elizabeth Stobert, and Jeremy Clark. A first look at the usability of bitcoin key management, 2015. URL http://users.encs.concordia.ca/~clark/papers/2015_usec_full.pdf. [Online; accessed 13-March-2017].

[119] Arvind Narayanan. Distributed consensus. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/At1IC/distributed-consensus. [Online Presentation; accessed 30-January-2017].

[120] Kenji Saito and Hiroyuki Yamada. What's so different about blockchain? — blockchain is a probabilistic state machine. volume 36 of *International Conference on Distributed Computing Systems Workshops*. IEEE, 2016. ISBN 978-1-5090-3686-8.

[121] Aaron Wright and Primavera De Filippi. Decentralized blockchain technology and the rise of lex cryptographia, 2015. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2580664. [Online; accessed 11-April-2017].

[122] Jude Umeh. Blockchain double bubble or double trouble? *ITNOW*, 58(1), 2016.

[123] Rafael Pass, Lior Seeman, and Abhi Shela. Analysis of the blockchain protocol in asynchronous networks, 2016. URL http://eprint.iacr.org/2016/454.pdf. [Online; accessed 11-April-2017].

[124] Joseph Bonneau. Bitcoin as an append-only log. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/be6cd/bitcoin-as-an-append-only-log. [Online Presentation; accessed 02-February-2017].

[125] Edward W. Felten. Government notice bitcoin. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/VEEei/governments-notice-bitcoin. [Online Presentation; accessed 16-March-2017].

[126] Guy Zyskind, Oz Nathan, and Alex Pentland. Decentralizing privacy: Using blockchain to protect personal data. 36th IEEE Symposium on Security and Privacy Workshops, 2015. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7163223. [Online; accessed 11-April-2017].

[127] Nordea. Vanliga frågor och svar, 2017. URL https://www.nordea.se/privat/kontakt/kontakt/vanliga-fragor-och-svar.html#faq=Betalningar-och-overforingar+11974. [Online; accessed 03-April-2017].

[128] Konsumenternas. Betalning av misstag, 2015. URL http://www.konsumenternas.se/betala/olika-betalformer/betalningar/betalning-av-misstag. [Online; accessed 03-April-2017].

[129] Visa. Intra visa europe interchange fees - european economic area (eea), 2017. URL https://www.visaeurope.com/media/images/intra%20ve%20eea%20@%202017_01_26-73-17763.pdf. [Online; accessed 23-March-2017].

[130] Handelsbanken. Prislista företag, 2017. URL https://www.handelsbanken.se/shb/inet/icentsv.nsf/vlookuppics/a_betalningar_prislista_informations-_och_betaltjanster__villkor_foretag/\protect\T1\textdollarfile/p011.pdf. [Online; accessed 23-March-2017].

[131] Nordea. Prislista för våra vanligaste tjänster företag och föreningar, 2016. URL https://www.nordea.se/Images/39-90058/priser-vanliga-tjanster-2016.pdf. [Online; accessed 23-March-2017].

[132] Aaron W. Baur, Julian Bühler, Markus Bick, and Charlotte S. Bonorden. Cryptocurrencies as a disruption? empirical findings on user adoption and future potential of bitcoin and co. In *Open and Big Data Management and Innovation*, volume 9373 of *Lecture Notes in Computer Science*. Springer Cham, 2015. ISBN 978-3-319-25013-7.

[133] Stephen L. Reed. Bitcoin cooperative proof-of-stake, 2014. URL https://arxiv.org/ftp/arxiv/papers/1405/1405.5741.pdf. [Online; accessed 20-March-2017].

[134] Nick Szabo. Trusted third parties are security holes, 2001. URL http://nakamotoinstitute.org/trusted-third-parties/. [Online; accessed 20-March-2017].

[135] Mary Ann Callahan. 5 blockchain trends for 2017. EE Times, 2017. URL http://www.eetimes.com/author.asp?section_id=36&doc_id=1331571. [Online; accessed 11-April-2017].

[136] Pavel Ciaian and Miroslava Rajcaniova. The digital agenda of virtual currencies: Can bitcoin become a global currency? *Information Systems and e-Business Management*, 14(4), 2016.

[137] Karl J. O'Dwyer and David Malone. Bitcoin mining and its energy footprint. In *Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*, volume 25. IET, 2014. ISBN 978-1-84919-924-7.

[138] Magnus Andersson, 2017. Telephone conversation with the project leader of Transparent Transport Systems (http://www.vinnova.se/sv/Var-verksamhet/Innovationssatsningar/Digitalisering-av-svensk-industri/Sma-puffar/Transparent-transportsystem/).

[139] Arvind Narayanan. When is decentralization a good idea? Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/Tc83z/when-is-decentralization-a-good-idea. [Online Presentation; accessed 3-February-2017].

[140] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol, 2016. URL https://pdfs.semanticscholar.org/1c14/549f7ba7d6a000d79a7d12255eb11113e6fa.pdf. [Online; accessed 11-April-2017].

[141] Ittay Eyal and Emin Gün Sirer. Majority is not enough bitcoin mining is vulnerable, 2013. URL http://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf. [Online; accessed 31-January-2017].

[142] Joseph Bonneau. Mining pools. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/iIRi2/mining-pools. [Online Presentation; accessed 31-January-2017].

[143] Joseph Bonneau. Energy consumption & ecology. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/dE86F/energy-consumption-ecology. [Online Presentation; accessed 30-January-2017].

[144] Alexander Chepurnoy. Interactive proof-of-stake, 2016. URL https://arxiv.org/pdf/1601.00275.pdf. [Online; accessed 01-February-2017].

[145] Pavel Vasin. Blackcoin's proof-of-stake protocol v2. BlackCoin, 2014. URL http://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf. [Online; accessed 31-January-2017].

[146] Joseph Bonneau. Limitations & improvements. Princeton University, 2014. URL https://www.coursera.org/learn/cryptocurrency/lecture/7g8gI/limitations-improvements. [Online Presentation; accessed 31-January-2017].

[147] Pieter Wuille. Segregated witness and its impact on scalability. SF Bitcoin Devs Seminar, 2015. URL https://www.youtube.com/watch?v=NOYNZB5BCHM. [Online Presentation; accessed 16-February-2017].

Appendix A

Valuable Knowledge

This chapter presents valuable knowledge coupled to blockchain but not directly related to the thesis.

A.1 Transaction

A transaction is an isolated unit of work. Four properties must be fulfilled, commonly referred as the acronym *ACID* [30].

1. *Atomicity*, either the transaction is fully performed or not at all.
2. *Consistency*, the constraints for the database must be fulfilled after the transaction.
3. *Isolation*, each transaction must appear to execute independent of other transactions.
4. *Duration*, once a transaction is completed the effect of it must never be lost.

A.2 Blockchain Types

A.2.1 Private and Public Blockchain

The blockchain may be distributed to a set of participants in a peer-to-peer network. A public such network results in anyone being able to participate in the common blockchain, and thus read and issue transactions ought to be put on the blockchain. This is called a *public blockchain*. The blockchain can be restricted to a group of participants, consequently called a *private blockchain* [68].

A.2.2 Permissioned and Permissionless Blockchain

As mentioned in 2.2.7 a block holds batches of transactions which may be issued by various participants of the peer-to-peer network. The issued transactions propagate over the network [31]. Nodes may from the issued transactions create blocks, such nodes are

called *miners*. Nodes may further verify that transactions and blocks are valid, e.g. transactions being issued by the adequate issuer, such nodes are *verification nodes*. In a *permissionless blockchain* any of the participating nodes may mine new blocks and participate in deciding which blocks ought to enter the shared consensus version of the blockchain by verifying the transactions and blocks. In a *permissioned blockchain* these nodes are preselected by some authority [68], hence permission is needed to *mine* and *verify* blocks.

A.3 Mining Pools

Miners may form *mining pools* with miners having the same intentions. For honest miners the rationale for participating in a mining pool is that a single miner may infrequently mine a block and get rewarded. Hence joining a pool and splitting the rewards results in more frequent but smaller rewards [141], thus lower variance in the outcome [42]. The reward of a block mined by the pool is given to the *pool manager* who shares the reward to the pool members in proportion to the mining power applied by the respective pool members [42, 142]. A mining pool may also have other intentions, for instance attempt to partly control the blockchain in their favour [141].

A.4 Selfish Mining

A 51% attack requires the majority of mining resources, and may then control the blockchain. However Eyal and Gün Sirer later proposed the ability to partly control the blockchain with only 1/4 of the resources through a mining strategy called *Selfish Mining* [141]. The strategy aims to let honest nodes mine on blocks never making it into the long term consensus chain. If the actor creates a block, the block is not published to the entire network but instead creating a secret branch [42]. This gives an ahead start for the actor. Since the actor does not control the majority of resources, the public branch eventually catches up. Before it happens the secret branch is published providing the chain with the most work behind, hence the rest of the network adapts to the published secret chain. The *selfish mining* actor consequently partly controls the blockchain and mines blocks in higher proportion than its fraction of system resources [141]. Solutions to address this problem have been presented in the community.

A.5 Proof-of-Work versus Proof-of-Stake

Proof-of-Work (PoW) consumes much power in order to solve the puzzles, by mid-2014 the Bitcoin network using PoW was approximated to consume up to 1 Gigawatt. One of the biggest power plants in the world *Three Gorges Dam* at the time generated 10 Gigawatt. Power compulsory to run the network accordingly required a regular power plant [143]. Moreover power is needed to cool the mining devices. Further power and resources are required to create the hardware. The power consumption caused by PoW for Bitcoin is comparable to the power consumption of Ireland [137]. *Proof-of-Stake* (PoS) on the contrary does not need the heavy computations, and hence is more environmental friendly.

A consequence of the lack of computations using PoS is that since no computations are required it is practically easier to forge the blockchain [144], the computational time required to recalculate blocks with PoW is not a factor. Hence the cost of computations in PoW is the cost for security [47].

It is more difficult to obtain the majority of the resources in a PoS-system than a PoW-system [44, 47, 49]. It is improbable someone would invest all their wealth into tokens bound to the system in order to have more stake to attack it. Mining hardware on the contrary still exists in reality. Moreover investing in such tokens influences the supply and demand, making it more expensive to obtain the majority of stake than to obtain the majority of computing power [44, 47]. Hence it minimizes the risk of centralization which was a goal for the initial blockchain technology [2], consequently the risk for 51% attacks is also compromised [44, 47, 49]. Having the majority of stake it would still be unprofitable to attack the very PoS-system oneself secures, in case of such an attack participants might abandon the system implying a significant loss for the attacker [47].

In the PoS-system the stakeholders and the miners are the same group, accordingly having the same interests. Stakeholders are keen of their stake and hence got incentives to secure the system. Using a PoW-system there might be a misalignment between the interests of the miners and the stakeholders, which thus affects the security of the system. In a PoW system miners usually instantly exchange their mined rewards to real money, consequently the miners might not be stakeholders to the same extent in a PoW system as in a PoS system [44].

It is yet unclear whether PoS can provide the same security as PoW, however much research is done within the area [42, 47]. NeuCoin, a PoS cryptocurrency, claims PoS is more secure than PoW [44]. A few other cryptocurrencies already use PoS, for instance BlackCoin which first served as an experiment for PoS in reality. Using PoS, BlackCoin provided security for 15-20 million dollars [145]. Further Bentov et al. have later claimed providing constructions of PoS protocols outperforming the current protocols [48]. Kiayias et al. claim to prove that their Proof-of-Stake protocol ensures rigorous security [140]. Ethereum, the second largest actor after Bitcoin, is very keen to swap PoW for PoS in their system [49].

A.6 Soft and Hard Fork

A fork is created due to lack of consensus, which is commonly solved with a consensus mechanism like PoW and PoS as stated in *2.5.2-2.5.3*. In previous sections of the thesis the forks were created by miners issuing different blocks, with all nodes running the same set of rules. Occasionally the rules change if the network decides to, consequently nodes may run different versions.

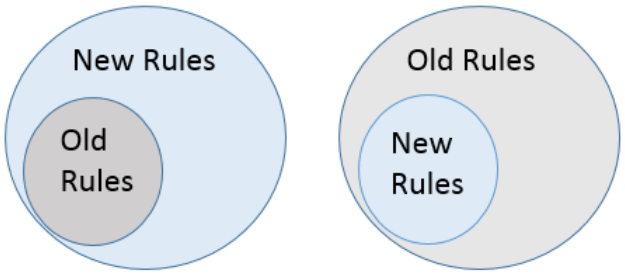

Figure A.1: *Illustration of extended and restricted rule set. A hard fork may appear in case Old Rules ⊊ New rules. A soft fork may appear in case New Rules ⊊ Old rules.*

A *Hard Fork* occurs when the rule set is extended, and a block violating the old rules but fulfills the new rules is published. Updated nodes accept the block while the older reject the block, creating two permanently divergent branches. Hard forks are thus not forward compatible, the old nodes have to update to not remain in their branch indefinitely [42, 43, 146].

A *Soft Fork* occurs when the rule set is restricted, the new rules being a strict subset of the old. A block fulfilling the old rules but not the new will cause temporary divergent branches. The fork is forward compatible since all blocks fulfilling the new rules are valid according to the old nodes. If the updated nodes have more mining power, their branch ought to be longer and thus the old nodes will accept that chain [42, 43, 146].

A.7 Segregated Witness

Segregated Witness (SegWit) unlike the initial blockchain protocols segregates the transaction signatures (witness) from the transactions themselves making it possible to not having to store all data and hence much memory may be saved. Thus also less data needs to be communicated over the network [147]. Segregated witness is a convenient tool for Lightning Network [83].

A.8 Reissuable or Non-Reissuable

Using a centralized issuer it has to be decided whether the asset should be reissuable or non-reissuable, meaning if it is possible to create new ScaniaCoins or if the ScaniaCoins are only created once and hence there is a limited amount of $C in the system. The advantage with only creating once is that if someone hacks the genesis with a reissuable approach, the attacker gains unlimited power to create new $C [73]. The disadvantage is that it could be problematic if there is too low amount of $C in the system. This is directly coupled to the trust of the system, if someone can issue unlimited $C it becomes problematic for the system. Thus it is proposed that ScaniaCoins ought to be non-reissuable, and the genesis issuance is issuing a significant number of $C that can be held on various Scania owned accounts.

Appendix B

Application Screenshots

This chapter presents the Android PoC application graphically with screenshots.

Main view, provides a navigation to all features. The balance of the user can be seen in the upper left corner.

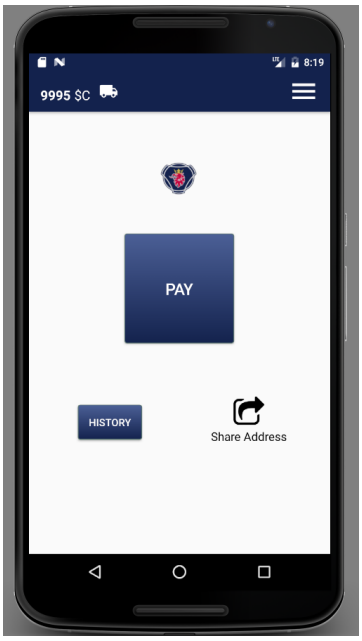

Figure B.1: *Main view.*

Share address view, one can share one's asset address for others to send assets to. It is also shared on a QR-code encoded format to allow other devices to scan the code.

Figure B.2: *Address view.*

Transaction view, supplying the address of the receiver and the amount to be sent one may transact.

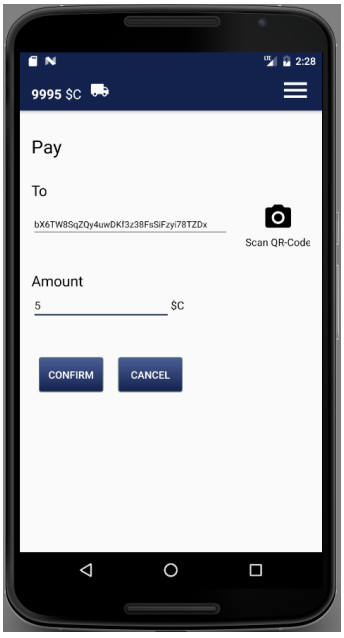

Figure B.3: *Transaction view.*

Scan view, to obtain the receiver's address the easiest way is to use the built in QR-scanner and scan the receivers QR-code address.

Figure B.4: *Scan view.*

Confirm view, to confirm the transaction the correct password must be provided to be able to sign the transaction with the adequate decrypted key.

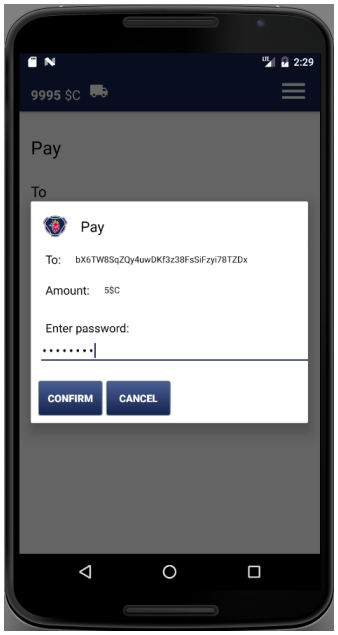

Figure B.5: *Confirm view.*

History view, the transaction history is presented as a list with the most recent transaction at top.

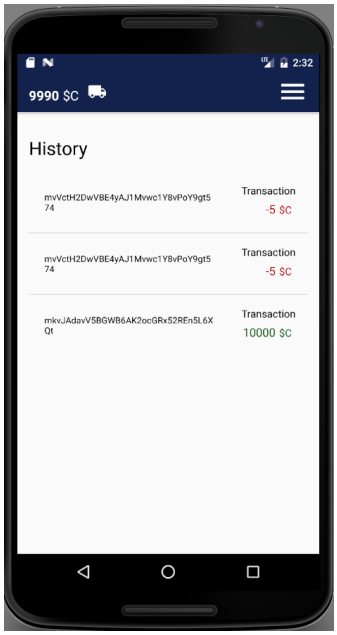

Figure B.6: *History view.*

Detailed view, clicking on one of the elements in the history list one may view the transaction details.

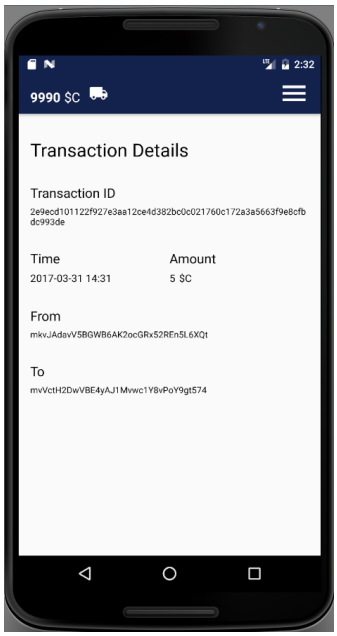

Figure B.7: *Detailed view.*