Honn Lam

Enhancing Security and Transparency of User Data Systems with Blockchain Technology

School of Technology and Innovations
Master's Thesis
Automation and Information Technology

Vaasa 2024

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Honn Lam
Title of the Thesis: Enhancing Security and Transparency of User Data Systems with Blockchain Technology
Degree: Master of Science in Technology
Programme: Automation and Information Technology
Supervisors: Teemu Mäenpää & Jussi Kangas
Year: 2024 **Pages:** 67

ABSTRACT:
As blockchain is increasingly gaining popularity, interest in corporate use is also gaining traction. One area blockchain has seen an increased amount of use are systems involving management of sensitive information, such as user data systems. In this thesis commissioned by a stakeholder, blockchain is implemented in user data system as a proof-of-concept prototype aiming to prove that the implementation can enhance data security and transparency in user data systems.

The first half of the this will build a theoretical framework. First, fundamental theory of blockchain is examined, which includes an overview of blockchain's architecture and its security features. This includes architecture and functionality on general level, consensus methodology, and other security algorithms such as hashes. Second, already existing blockchain solutions that could benefit in designing the proof-of-concept prototype are explored, which included a patient data system, and an Internet-of-Things system. Patient data system's case provided a solution of implementing blockchain as a separate component in the patient information system, while Internet-of-Things' solution provided insight of storing functional data in the blockchain, while keeping the actual raw data in a separate database with a restricted access. These solutions formed an adequate foundation; however, the solutions couldn't be applied as-is, which lead to the need of applying and designing a new solution.

The latter half of reports the implementation process of blockchain. First, the research method used in this thesis, constructive research approach, is demonstrated - constructive research approach aims to create a practical solution for a real-life problem. The prototype's primary requirement is that it should be able to record a log of activity in the user data system, telling who did what and to whom, and without revealing any confidential information. The prototype is implemented in a test environment using a separate database for storing the actual user data, and blockchain for storing data about activity happening in the user data system. The prototype's validity was tested using software testing methods, more specifically integration testing and user acceptance testing.

The research will benefit the stakeholder with a working example showing a potential way of implementing a blockchain solution in a commercial software. The research aims to prove that with the implemented blockchain solution can adequately help monitoring actions committed by users, enforcing honest usage, and helping spot malicious activity, and this way improving transparency and security. The research has also value in scientific community with its practical approach demonstrating how could a blockchain system be implemented in sensitive user data systems, and what are its potential benefits in security. The next step for the study is evaluating the actual value of the implementation in the commercial software, or proof-of-value research.

KEYWORDS: blockchains, data security, transparency, confidentiality

VAASAN YLIOPISTO
Tekniikan ja innovaatiojohtamisen yksikkö
Tekijä: Honn Lam
Tutkielman nimi: Turvallisuuden ja läpinäkyvyyden kehittäminen käyttäjätietojärjestelmissä lohkoketjua hyödyntäen
Tutkinto: Diplomi-insinööri
Oppiaine: Automaatio- ja tietotekniikka
Työn ohjaajat: Teemu Mäenpää ja Jussi Kangas
Valmistumisvuosi: 2024 **Sivumäärä:** 67

TIIVISTELMÄ:
Lohkoketjujen kasvaessa suosiota myös kiinnostus yrityskäyttöä kohtaan on ollut kasvussa. Yksi osa-alue missä lohkoketjujen käyttö on nähnyt kasvua ovat luottamuksellista tietoa käsittelevät järjestelmät, kuten käyttäjätietojärjestelmät. Tässä sidosryhmän toimeksiantamassa tutkielmassa implementoidaan lohkoketju käyttäjätietojärjestelmään osana konseptitodistusprototyyppiä (proof-of-concept), joka pyrkii todistamaan lohkoketjun kykyä tietoturvan ja läpinäkyvyyden kehittämisessä käyttäjätietojärjestelmissä.

Tutkielman ensimmäinen puolisko luo teoreettisen viitekehyksen. Ensimmäisenä tutustutaan lohkoketjujen perusteisiin, johon kuuluu sen arkkitehtuurin sekä tietoturvaominaisuuksien tarkastelua. Tämä sisältää yleisen tason katsauksen toiminnallisuudesta, yhteisymmärrysmetodologiasta sekä muista tietoturva-algoritmeista, kuten tiivisteistä (hash). Tämän jälkeen syvennytään olemassa oleviin lohkoketjuratkaisuihin, jotka voisivat hyödyntää konseptitodistusprototyypin suunnittelemisessa. Tarkasteltavina toimivat esimerkit potilastietojärjestelmästä sekä esineiden internetistä. Potilastietojärjestelmän tapauksessa perusteltiin lohkoketjun implementoimista erillisenä komponenttina, kun taas esineiden internetin tapauksessa esitettiin toiminnallisen datan säilyttämistä lohkoketjussa, kun taas raakadata säilytetään erillisessä tietokannassa rajatulla pääsyllä. Nämä esimerkit loivat hyvän pohjan, mutta eivät ole sovellettavissa prototyyppiin sellaisinaan.

Tutkielman toinen puolisko selostaa prototyypin kehitysprosessia. Aluksi esitellään käytetty tutkimusmenetelmä, eli konstruktiivinen tutkimusmenetelmä, jonka ominaispiirteenä on luoda käytännön ratkaisu oikean elämän ongelmaan. Prototyypin ensisijaisena vaatimuksena on pystyä kirjata aktiviteettilokeja käyttäjätietojärjestelmässä, kertoen kuka teki mitäkin ja kenelle, kuitenkaan paljastamatta luottamuksellista tietoa. Prototyyppi implementoitiin testiympäristöön käyttäen erillistä tietokantaa itse käyttäjätietojen tallentamiseen, kun taas lohkoketjua käytettiin käyttäjätietojärjestelmän aktiviteettilokien tallentamiseen. Prototyypin toimivuus varmistettiin ohjelmistotestausmetodeilla, tarkemmin ottaen integraatiotestauksella ja hyväksymistestauksella.

Tehty tutkimus tulee hyödyttämään sidosryhmää toimivalla prototyypillä esittelemällä potentiaalisen tavan lisätä lohkoketjutoteutus kaupalliseen ohjelmistoon. Tutkielma pyrkii todistamaan lohkoketjutotetuksen tuomaa hyötyä käyttäjien tekemien muutosten tarkkailussa, täten kannustaen rehelliseen käyttöön ja samoin auttaa tunnistamaan haitallisen toiminnan, joka kaiken kaikkiaan johtaa kehittyneeseen tietoturvallisuuteen ja läpinäkyvyyteen. Tutkielmalla on myös tieteellistä arvoa esitellen käyttäjätietojärjestelmien tietoturvan kehittämistä lohkoketjutoteutusta hyödyntäen. Jatkotutkimusmahdollisuutena on arvioida toteutuksen varsinainen tuoma lisäarvo kaupallisessa ohjelmistossa.

KEYWORDS: lohkoketjut, tietoturva, läpinäkyvyys, luottamuksellisuus

Contents

1 Introduction 8

2 Blockchain, Security, and Trust 10

2.1 Architecture of Blockchain 11

2.2 Transactions and Consensus in Blockchain 13

2.2.1 Transactions Process in Blockchain 13

2.2.2 Consensus Protocols 14

2.3 Security and Trust in Blockchain 16

2.3.1 Blockchain Security and Challenges 16

2.3.2 Transparency and Traceability in Blockchain 22

3 Existing Blockchain Solutions 24

3.1 Prior Research 24

3.1.1 Research on Blockchain in Health Information Systems 25

3.1.2 Research on Blockchain in Internet of Things 25

3.2 Practical Solutions 26

3.2.1 Implementations of Blockchain in Patient Information Systems 27

3.2.2 Implementation of Blockchain in a IoT-based Logistics System 28

3.3 Key Takeaways 30

4 Research Approach 33

5 Execution 37

5.1 Motivation and Requirements 37

5.2 Design 38

5.3 Implementation 42

5.4 Testing 46

6 Results 52

6.1 Final Prototype 52

6.2 Test Results 54

6.3 Points of Improvements and Future Prospects 55

7 Conclusions 57

References 61

Figures

Figure 1. Block Structure (adapted from Zheng et al., 2017). 12

Figure 2. Overview of transaction process in blockchain (adapted from Laurence, 2017, p. 13). 14

Figure 3. Merkle tree diagram (adapted from Szefer & Biedermann, 2014). 18

Figure 4. Digital Signature Algorithm (adapted from Sheldon et al., 2002). 19

Figure 5. A simplified flowchart of how the blockchain system would be implemented. 42

Figure 6. Basic structure of the test environment. All the components are hosted locally. Frontend contains the user interface, where the end user can manage user data, while backend communicates between the user and the database. 43

Figure 7. A flowchart of how smart contract is implemented. 46

Figure 8. A rudimentary user interface to simulate editing of user data. 48

Figure 9. A view of a specific smart contract ‘DatabaseTracker’ in Ganache. The smart contract tracks changes in user data. 49

Figure 10. Test case, where the user has changed field ‘Email’ of another user. 50

Figure 11. Transaction details for changing the field ‘Login’ of another user. 51

Figure 12. Transaction details for changing both fields 'Login' and 'Email' of another user. 51

Figure 13. Architectural diagram of the final prototype. 52

Tables

Table 1. Examined prior research and use-cases and key takeaways summarised. 32

Table 2. Procedures of the study briefly reflected to Lukka's (2001) procedure descriptions. 36

Table 3. Main requirements of the prototype summarised. The requirement numbers are on the left column and the description on the right one. The requirements are not in any specific order. 38

Table 4. Requirements reflected with design choices. The left-most column stands for requirement number, middle column describes the description, and the right-most summarises the corresponding feature(s). 54

1 Introduction

Blockchain, popularised by cryptocurrencies and especially Bitcoin, has received attention extensively, which has resulted in blockchain technology rapidly increasing in appearance in various fields (Zheng et al., 2017). Rapidly increased use of blockchain technology also raises many concerns such as scalability and security, but also intrigue in how it could be used to enhance and optimise data security in commercial systems.

This study will concentrate on implementing blockchain as a way to make handling of user data more secure and transparent. Blockchain's viability in doing so is demonstrated in the study through a proof-of-concept prototype. The prototype is not tailored to any specific commercial product but is a general prototype built on a test environment simulating a real-life use-case. Ultimately, the research is commissioned by a stakeholder.

To better understand what is being worked with, basic theory of blockchains is first examined answering questions: What is blockchain? How does it work? What are its basic components? Blockchain's data security is validated by examining its security features, what makes it secure. Different pre-existing applications are also explored; examples where sensitive user data and transparency are in the centre of the solutions are investigated, which will help building an adequate foundation for designing the prototype.

The next step is to build the prototype as a part of constructive research methodology, a research approach aiming to create practical solutions to real-life problems built on top of a properly acquired prior knowledge of the field. The requirements for the prototype set by the stakeholder includes three main requirements: 1. the prototype should record a log of activity using blockchain; 2. the log should contain information of three aspects: 'who did', 'who it was done to', and 'what was done'; and 3. the prototype should not reveal any confidential information.

The prior research and use-case examples of blockchain used in patient data systems and Information-of-Things will work as the foundation for the research solution. They

provide insight especially on software architectural questions of the prototype, while emphasising adequate practices for handling confidential data. The outcome of the study will potentially help building better data security strategies for implementing blockchains in information systems handling confidential data with the research questions 'How can blockchain technology be used to enhance security and transparency in user data systems?' and 'How can blockchain be implemented in commercial software?'.

As blockchain and data security are broad fields, there are naturally some limitations to be acknowledged. Due to the scale and time constraints of the study, the theoretical framework will be restricted to general level – the architecture of blockchain will be explored in a level enough to provide a general understanding what is blockchain, and security aspects that are commonly found between different blockchain technologies are examined. This means that deeper understanding of hashing algorithms and algorithms will not be covered, as they necessarily do not provide a significant benefit in designing the prototype. The prototype itself will also not be tailored to meet commercial product standards but shall be a general level proof-of-concept prototype aiming to prove that the technology works.

This thesis is divided into six chapters. Chapter 2 will examine blockchain's architecture and its security features – the crucial components of a blockchain and its functionality will be investigated and blockchain's security and transparency for actual commercial use will be confirmed by inspecting blockchain's security features. Chapter 3 will review some pre-existing applications of blockchain in different industries. The industries relevant to especially this study include healthcare, and IoT systems. The aim of the chapter is to explore references for the practical research, what can be used and what to take into consideration. Chapter 4 demonstrates the research methodology used in this paper, or in other words, research process overview. Chapter 5 works as a report of required specifications and how the prototype is designed and constructed and the steps for conducting the tests, while Chapter 6 evaluates the results and discusses considerations for the future. Chapter 7 will discuss conclusions of the conducted research.

2 Blockchain, Security, and Trust

With the sudden rise in popularity of cryptocurrencies, consequently blockchain has also seen popularity and a rise in use in many industries according to Zheng et al. (2017). They summarise blockchain to be a decentralised public network which anyone can join, and where transactions are stored. They also mention security challenges to have been relevant since. In this chapter, crucial theory, components, and general security features of blockchain will be explored.

There are varying definitions of key characteristics of blockchain. Studies mention characteristics as key ones that are not mentioned in other studies, though there are a few general concepts that are a common theme in these studies, albeit are referred to with different terms. Some of the commonly mentioned characteristics include *decentralisation* and *security*, among others.

First commonly mentioned characteristic includes *decentralisation, which* means that data is handled among all participants instead of having a single organisation control the system (Liu et al., 2020). This means that all the peers who actively participate in maintaining the system, or commonly known as nodes, will have a copy of the blockchain stored in their individual devices.

Second characteristic is *data integrity*, which refers to that the data in blockchain is virtually permanent – as data is stored in the blockchain, deleting, or tampering the stored data later is nearly impossible (Panicker et al., 2016). Essentially, there is only ‘one truth’ in the transaction history, reducing the risk of fraud in the network.

Third common characteristic mentioned is *security*. Cryptographic methods are used for ensuring security in a blockchain network (Yaga et al., 2018). Some of these applications include using a generated address for each user allowing anonymity, and for validating transactions (Zheng et al., 2017).

2.1 Architecture of Blockchain

According to Laurence (2017, p. 10-11) blockchain's main components are *block* and *chain*, and *network*. These main components again can consist of many other subcomponents. In this section, these components are given a general overview of, explaining each component's role in the blockchain.

The first main components of blockchain are *block* and *chain*. Vaguely put, a block is a list of transactions, where a transaction can be thought of as an action of transferring data from one user to another (Laurence, 2017, p. 10). Blocks in a blockchain are mathematically connected to each other by containing a hash (data transformed into an alternative form) created from preceding block's data – the blocks essentially refer to each other with these hashes, and this way create a *chain*.

A block can be divided into *block header* and *block body* as seen in Figure 1 (Zheng et al., 2017). A block body is fairly simple in structure; it mainly consists of a list of transactions to date, and possibly other data depending on the specific blockchain implementation (Yaga et al., 2018). On the other hand, Zheng et al. (2017) describe a block header to contain:

1) Block version, which 'indicates which set of block validation rules to follow'.
2) Merkle tree root hash, which is a hash value of the Merkle tree root of all the transactions in the block. In summary, Merkle tree is a list of hashed data linked to each other, creating a tree-like data structure (Yaga et al., 2018).
3) Timestamp, for when the block was validated and created in UNIX time.
4) nBits, which is the 'target threshold' of a block hash. In other words, it determines the difficulty of the target hash (Bitcoin Project, 2018). Relevant in blockchains where 'Proof-of-Work' algorithm is used.
5) Nonce, which is a 4-byte field used for mining, or creating new blocks; it is a random number which miners try to find, as it is used to generate the hash of a new block. This is also relevant in blockchains where 'Proof-of-Work' algorithm is used (Liu et al., 2020).

6) Parent block hash, which is a hash that refers to a preceding block.

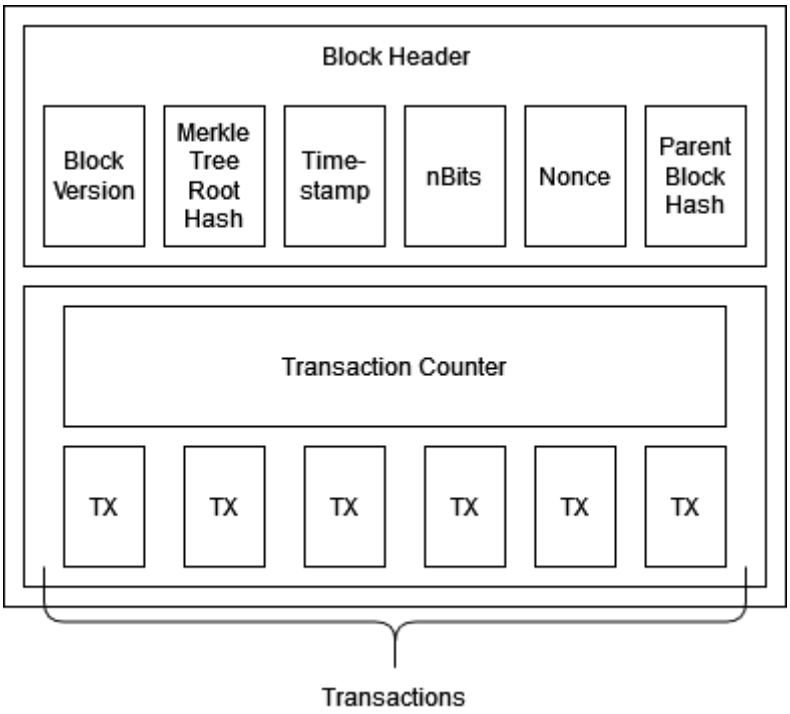

Figure 1. Block Structure (adapted from Zheng et al., 2017).

The other main component in a blockchain is *network.* Laurence (2017, p. 10) summarises a network in the context of blockchain to be a network of *full nodes*. Full nodes, or more familiarly *nodes*, are devices, such as computers, that maintain the blockchain network. She adds that each node maintains a copy of a complete record of all the transactions in the blockchain, and that operating a node is 'difficult, expensive, and time-consuming', which is why blockchain algorithms usually offer different reward incentives for participating nodes, such as cryptocurrency like Bitcoin.

In a blockchain network, nodes play a crucial role in maintaining the integrity and security (Park et al., 2019). They validate transactions through a consensus process, propagate transactions to other nodes. Nodes participating in maintaining the blockchain network contributes to decentralised nature of blockchain technology.

A notable component in blockchain networks is *smart contract.* In blockchain, various tasks can be automated using smart contracts (Mohanta et al., 2018). They are programmes tailored to do specific tasks and executed automatically in decentralised manner on the blockchain platforms. Smart contracts work by 'if-then' logic, meaning that the programmes trigger when a transaction of a certain condition set in is met (Wang et al., 2019). Like transactions, smart contract executions are also verified using consensus method, and this way ensure that the automated system is mutually trusted (Geng et al., 2021). Smart contracts have been utilised for many tasks, such as for tracking pharmaceutical supply chains and for automating interaction between devices in Internet-of-Things systems.

2.2 Transactions and Consensus in Blockchain

Transactions are an essential part of blockchain. Transactions is a term for the action of data exchange between users. They go through a unique process in blockchains called *consensus*, where they are validated and injected into a blockchain permanently inside a block, and virtually cannot be mutated afterwards (Zheng et al., 2017).

2.2.1 Transactions Process in Blockchain

In blockchains, committing transactions relies on the network, where nodes are in charge of validating the transactions. A typical transaction process in blockchains follows the steps seen in Figure 2.

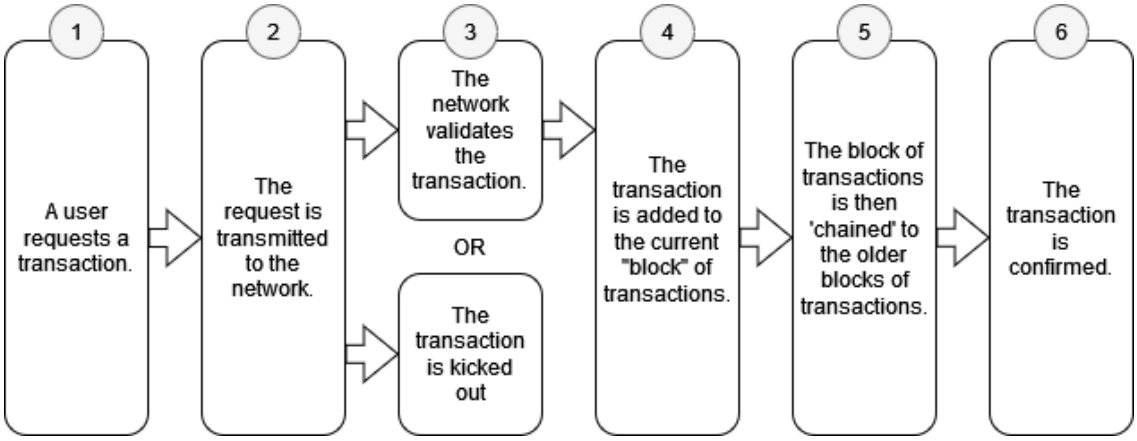

Figure 2. Overview of transaction process in blockchain (adapted from Laurence, 2017, p. 13).

Following the steps in Figure 2, Xie et al. (2020) give a general overview of the general steps of the transaction process:

1) A user requests a transaction, which can be for example sending virtual currency to another user.
2) The transaction is sent to all participating nodes in the blockchain network, which each node will collect into a block (Nakamoto, 2008).
3) The nodes in the network validate the transaction against the validation rules of the blockchain; in case the transaction does not meet the validation rules, it can be discarded, as described in Figure 2.
4) The validated transactions are then encrypted into hashed forms and then stored into a block.
5) The block is added to the blockchain, or 'chained', as other nodes validate the created hash.
6) The transaction is validated and a virtually permanent part of the blockchain, and it cannot be altered in any way.

2.2.2 Consensus Protocols

The main purpose of consensus system is to prevent malicious activity in the blockchain, as summarised by Yaga et al. (2018). Consensus protocols relies on the maintaining

nodes in the network to publish new blocks. In other words, in consensus process, mutually distrusting parties work together on deciding whether a new block shall be added to the chain or not.

Yaga et al. (2018) mention that there are many different distinct consensus protocols in different blockchain algorithms depending on many variables. A couple common protocols they mention are *Proof-of-Work (PoW)* and *Proof-of-Stake (PoS)*. Other commonly mentioned protocols include *Delegated Proof-of-Stake (DPoS),* and *Practical Byzantine Fault Tolerance (PBFT)* (Mingxiao et al., 2017; Zheng et al., 2017). Each protocol has its advantages and disadvantages, and dependent on the specific blockchain implementation's specifications and requirements which protocol to use.

The most popular consensus protocol is *Proof-of-Work* (PoW) (Chang & Wuthier, 2020). In PoW, nodes participate in solving hashes mathematically. It relies on computational power for solving the hashes, as the solving process is based on brute-force search by manually iterating nonces to find the correct hash. Additionally, the difficulty of the hashes is proportional to the number of nodes participating in mining. Once the correct hash is solved, the transaction is verified and added to the blockchain (Sriman et al., 2021). PoW is commonly used in public blockchains (see Chapter 2.3.2), such as cryptocurrencies like Bitcoin and Ethereum.

In *Proof-of-Stake* (PoS) on the other hand, validating nodes are randomly chosen based on how much of digital assets do they own and are willing to 'stake' as collateral; the more digital assets one possesses and is willing to pledge coins to be used for verifying transactions, the more likely they are chosen to be validators to mine the next transaction (Lin, 2023). This protocol is more energy efficient than PoW and is often transitioned into after using PoW.

In *Delegated Proof-of-Stake (DPoS)*, a variation of PoS, instead of randomly chosen validators, each node in the network can vote a set of validators by staking for a specific

node (Kumar R, 2021). The chosen validators take turns in validating transactions and creating new blocks, and dishonest validators can easily be voted out. DPoS has a benefit of greater efficiency of validating new blocks over PoS and PoW (Kumar R, 2021).

Practical Byzantine Fault Tolerance (PBFT) attempts to solve challenges concerning achieving consensus in the presence of faulty and malicious nodes, ensuring consistency (Xie et al., 2020). The nodes are divided into 'primary nodes' and 'backup nodes', where both have distinct roles in a three-phase protocol: 1) the primary node creates a request, such as a transaction, and sends it to the backup nodes; 2) the backup nodes broadcast their agreement of the request to all other nodes in the network; 3) the backup nodes broadcast a commit message in the network, ultimately confirming the request. This protocol resists malicious and faulty nodes by tolerating a certain number of faulty nodes in the network, requiring two-third majority of nodes to agree on the request, digital signatures and message authentication, and isolation of faulty nodes, to ensure consistent decision-making and consensus process in the network (Xie et al., 2020). PBFT is commonly used in permissioned blockchain networks (see Chapter 2.3.2).

2.3 Security and Trust in Blockchain

As a rapidly emerging technology, blockchain brings a lot of potential, but also concerns in terms of security and trustworthiness. In the paper's context, trustworthiness covers transparency and traceability. Blockchain's common security questions are examined, and potential commonly agreed solutions overviewed. Transparency and traceability are defined and how blockchain meets the definitions. Before implementing to a system containing sensitive information, it is necessary to understand whether the blockchain itself is secure.

2.3.1 Blockchain Security and Challenges

Cooper et al. (2023) mention common security features for blockchain including consensus mechanism, and different cryptographic techniques featured in blockchain's

architecture. A general overview of these techniques will be examined and discussed how do they contribute to blockchain's security. Challenges to be considered will also be examined.

Many features in blockchain architecture involve cryptography – a couple fundamental parts of blockchain are important contributors for blockchain security, which include *Hash Chained Storage*, and *Digital Signature* (R. Zhang et al., 2019).

Hash Chained Storage is formed by a *hash pointer,* and a *Merkle tree* (R. Zhang et al., 2019). Hash pointer is a piece of data which contains information of where the data is located in the blockchain, more specifically to the previous block, and, and also contains the hash of the data element that is being pointed to. They state that hash pointers can be used to check data integrity by showing whether the data has been tampered with or not.

Merkle tree is a binary search tree where nodes are linked to each other using hash pointers (Szefer & Biedermann, 2014). They state that a Merkle tree enforces data integrity thanks to how it works – visualised in Figure 3, when a user tries to tamper data in any node in the tree (for example, Node 5), the parent node's data (Node 2) will also have to be changed, and similarly its parent node again up until the tree's root node, as in data of Node 2 goes to Root Node. This is where a hash pointer exposes tampering, as the hash pointer of the root does not match with the stored hash pointer, meaning that the tampered data is not valid (Kabir et al., 2021).

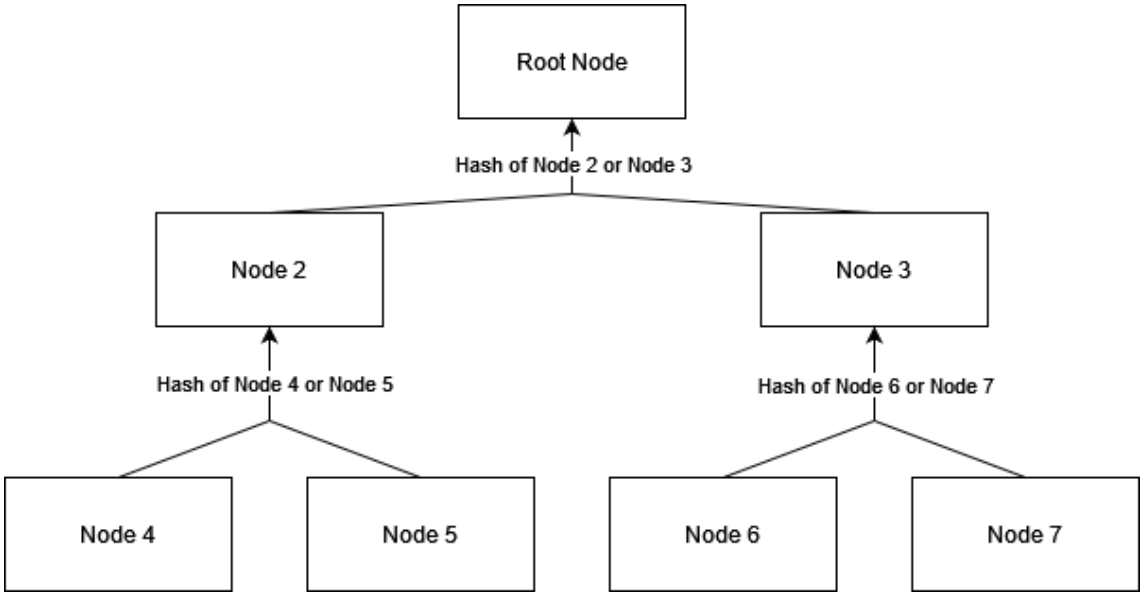

Figure 3. Merkle tree diagram (adapted from Szefer & Biedermann, 2014).

Digital signature is created using a cryptographic algorithm to verify the authenticity and integrity of data (Babiker, 2022). Digital signature is based on asymmetric public-key cryptography, which means that a sender encrypts the data using their own private key, which the recipient can verify using a public key by the sender. This way a transaction's validity can be verified, detecting any potential harmful transactions. The cycle of a digital signature architecture is visualised in Figure 4. First, a message digest, which is formed by applying a hash function on the message, essentially creating a hash (Sheldon et al., 2002). The sender then encrypts the message using a private key and the message digest sends a transmit to the receiver. Then using the sender's public key, the receiver decodes the digital signature. The signer certificates the transmission, which if successful will decode the transmission, and finally create a new message digest using a different hash function than when the sender creates a hash function.

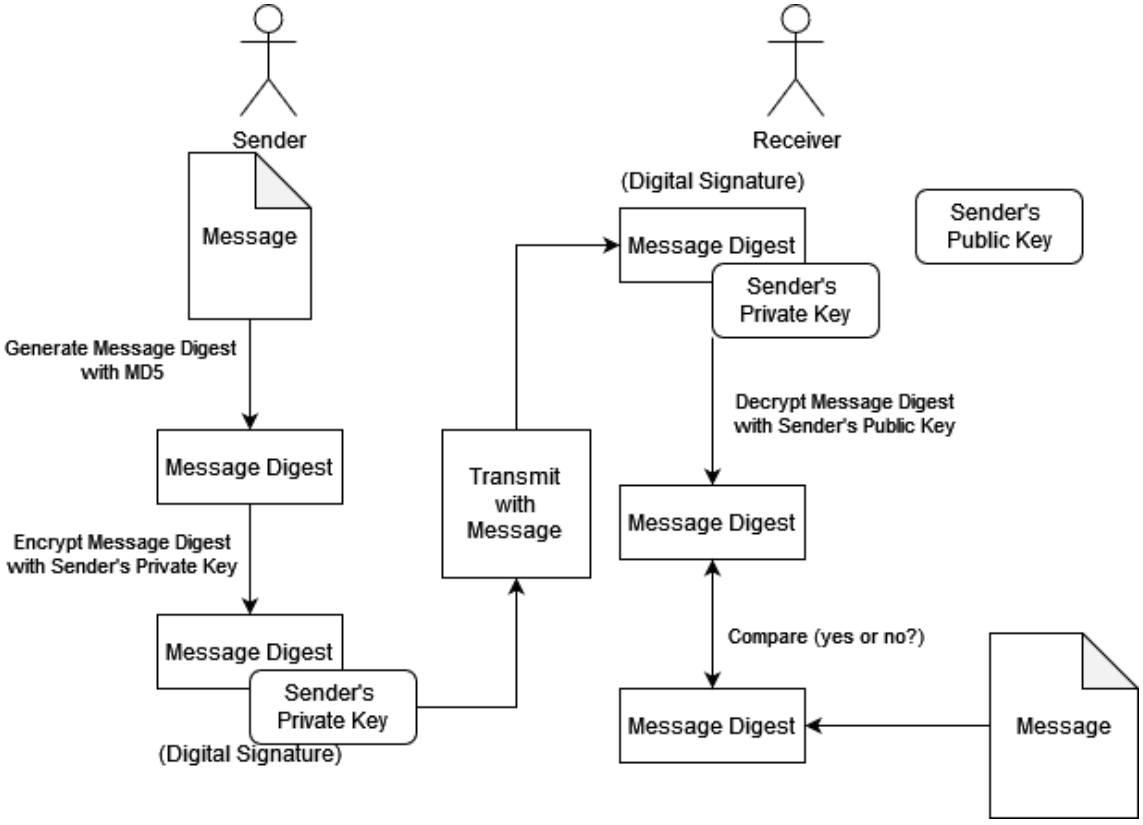

Figure 4. Digital Signature Algorithm (adapted from Sheldon et al., 2002).

In theory, consensus is an important process to ensure data integrity, as parties in the network participate in choosing whether a block is added to the chain or not, as summarised by Saqib & AL-Talla (2023). Due to the nature of common agreement of consensus mechanism, it prevents any malicious activity, and this way enforces security and correctness of the system. Additionally, it ensures reliability and immutability for the blockchain system, as committed transactions cannot be simply deleted or altered like in a conventional centralised database (X. Zhang et al., 2022).

As blockchain is an emerging technology, there are naturally many security questions and challenges to be addressed. In an investigation conducted by AlFaw et al. (2022) some common vulnerabilities and challenges involving cryptographic functions and consensus mechanism are examined. These observations will give a good idea of what to take into consideration in security features when implementing a blockchain solution.

First set of vulnerabilities are related to choices of cryptographic functions and the level of their implementation in blockchain technology. In case of Bitcoin as an example, a cryptographic digital signature algorithm called *Elliptic Curve Digital Signature Algorithm (ECDSA)* had exploitable vulnerabilities – due to error in implementation of the algorithm, an attacker was able to create a valid digital signature without knowing the private key, which allowed malicious validation of transactions (Campbell, 2019).

Another potential future vulnerabilities of cryptographic functions in blockchain are also relevant. Cryptographic functions used in blockchain architecture (such as SHA256-algorithm used in Bitcoin) have become more breakable due to rapidly increased processing power and advanced cryptanalysis, which can lead to vulnerability for attacks (Campbell, 2019). Especially advancements in quantum computing forces us to evaluate the cryptographic function's validity in the future.

Another vulnerabilities AlFaw et al. (2022) discussed are related to users addresses. In Bitcoin's example, identity fraud is possibly due to credentials not being authenticated. This leads to vulnerability for example to a *man-in-the-middle* attack, where the original intended Bitcoin address is changed to a different address.

The second set of vulnerabilities involves the consensus mechanism. The consensus mechanism's principles can also be its own vulnerability – Haque and Rahman (2020) found out that a single party can abuse the consensus mechanism and seize control of the blockchain by retaining more than 50% of the computing power in the blockchain network. They clarify that this vulnerability is present especially in PoW protocol.

Other vulnerabilities related to consensus mechanism AlFaw et al. (2022) found are *Finney attack* and *Race attack.* They define a Finney attack to be a process where an attacker can cause an asset to be spent twice, or being double-spent, by being disguised as a miner and conducting a transaction in a stealth mode and finally selling digital asset to a merchant who accepts the transaction despite it not being validated. In *Race attack* a

blockchain that relies on PoW as its consensus protocol, an attacker can abuse the time lag between a transaction's issuance and validation causing double-spending without the need of verifying the transaction (Rathod & Dilip, 2018).

Different potential solutions and improvements for these vulnerabilities have been proposed. For hash functions, it is advisable to actively look for potential stronger alternative algorithms as computing power keeps improving – Tomović et al. (2015) compared different hashing algorithms in context of security in cloud computing, though their findings have potential benefit in the context of blockchain security too. They examined potential replacements for SHA256 algorithm, used for example in Bitcoin, such as SHA512, or *Whirlpool,* which is relatively new and lacks practical implementation.

New principles overall can also be used as a potential solution. Peng et al. (2022) researched a potential alternative blockchain variation called *redactable blockchain*, which allows modification of the content of blocks, as opposed to conventional blockchain principle. This is allowed by using an alternative hashing technology called *chameleon hashing*, where a hash function has a *trapdoor* accessible using a secret *trapdoor key*. They emphasise redactable blockchain's ability to discard harmful or invalid blocks afterwards, which in theory solves vulnerabilities involving double-spending and frauds.

As for vulnerabilities in especially PoW consensus protocol, one way of improving its overall security is to adjust network-layer parameters - Gervais et al. (2016) noticed in their double-spending security comparison that Ethereum needs at least 37 block confirmations to match the security level of Bitcoin with 6 block confirmations, which suggests that Ethereum is more vulnerable to double-spending attacks.

Other proposed methods to improve consensus mechanism's robustness are to increase hashing power of nodes, which decreases chances of malicious actors catching up, to utilise Byzantine Fault Tolerance algorithms in the consensus protocol, which has a tolerance of malicious actors of certain percentage, or to implement a permissioned

blockchain, where each participant in the blockchain can select their own consensus node according to the rules (Hasan et al., 2023; Nijsse & Litchfield, 2020).

2.3.2 Transparency and Traceability in Blockchain

One of blockchain's characteristics is transparency; each user can see what data is being collected and how they are accessed (Zyskind et al., 2015). According to Dutta et al. (2020), blockchain is transparent due to how data is recorded and stored in the network – the data is stored in a 'public ledger', a network accessible by all of its users, which allows anyone to view the transaction history and verify the integrity of the blockchain, which. Other features they mentioned that contribute to transparency in blockchain are consensus protocol, which ensures that all the users are able to participate in validating transactions, decentralisation, which allows all the users in the network to have the same information real-time enforcing validity, and immutability, which ensures that the data virtually cannot be altered once recorded, ensuring that the full transaction history can be traced. With the combination of these features in blockchain, Dutta et al. (2020) state that blockchain is transparent and 'trusted'.

For traceability, Mitani and Otsuka (2020) state that there isn't a single agreed definition for traceability. According to their findings, some researchers' definition focus on the '"objects" moving between players', while Mitani and Otsuka (2020) themselves refer to transparency as a change of '"state" of the amount of an asset held' between a time interval. In general, despite varying definitions existing, a general concept of traceability can be concluded from their observations, which can be summarised as the ability to track the actions happening in the network, for example, movement of a digital asset from one user to another. Mitani and Otsuka (2020) emphasise that a transparent and tamper-proof record of transactions is important for blockchain to be traceable. Additionally, Mitani and Otsuka (2020) emphasise traceability is one enabler of transparency in blockchain, and is especially important in some industries, such as supply chain management.

Depending on the specific blockchain application, traceability and transparency can be limited by variable amount. This means that the access of which users can access the transaction history and validate them vary depending on the privacy level. Mingxiao et al. (2017) classify blockchains into three different categories depending on their access level: public blockchain, private blockchain, and permissioned blockchain. A public blockchain is 'completely open and decentralized' which anyone can access and contribute to, and where all transactions can never be changed or revoked. Private blockchain relies on nodes of the highest value in the network for transaction validation, though the owner of the blockchain has the highest authority to change the information. Permissioned blockchain also relies on designated nodes for consensus process but are chosen by mutually untrusted members instead of a single high authority. Additionally, Mingxiao et al. (2017) state that private blockchains are applied in for example in closed networks, such as intranets, while permissioned blockchains see use in semi-closed networks, such as enterprise settings.

It is important to find a balance between transparency and privacy especially in an enterprise setting where a supervising adversary wants only authorised users to access the blockchain, but at the same time wants to achieve high transparency. One way to achieve high transparency while maintaining adequate privacy is to utilise fully homomorphic encryption and zero-knowledge proof, as demonstrated by Mitani and Otsuka (2020) in their research. To summarise, traceability can still be achieved in a private setting by encrypting the data with homomorphic encryption, where data can be processed without decrypting it first and can be validated without revealing any data. This allows that a blockchain can function by its principles without revealing any private sensitive information, such as trade secrets.

3 Existing Blockchain Solutions

In this chapter, research and case studies are examined especially in terms of overall implementation, such as technology choices and how they affect the system, and how data security questions are handled. The specific industries examined are Internet-of-Things (IoT) and healthcare, as the stakeholder's use-case involves management of sensitive user data between mutually unknown parties in a commercial system, and monitoring their actions, which are both relevant in healthcare and IoT systems respectively. The example in healthcare industry provides insight on implementing blockchain in already existing information systems which also handle sensitive data, such as patient data, while research on IoT systems give insight on how to utilise the blockchain; what especially should be stored in the blockchain and what not, and how should the information for blockchain be generated.

3.1 Prior Research

Since the release of Bitcoin, research on blockchain technology has seen rapid rise. When searching for publications with keyword 'blockchains', around 21,000 results were found on the database of *Institute of Electrical and Electronics Engineers* (IEEE) alone as of December 2024. The top topics at the time after 'blockchains' with around 9,600 results were 'data privacy' with around 4,500 results, 'Internet of Things' with around 3,700 results, and cryptocurrencies with around 3,100 results.

Different ways of implementation for blockchain have been researched vastly. Some common applications include already mentioned IoT and healthcare sector. Some of the examined fields also include security challenges that are relevant also when designing the proof-of-concept prototype.

3.1.1 Research on Blockchain in Health Information Systems

Esposito et al. (2018) discuss gradual transition of healthcare data moving to cloud storage. There are several benefits on moving health information systems to a cloud platform. Some of the benefits include having complete medical history of a patient available real-time and regardless of geographical location, flexibility in data handling with how cloud systems are capable of handling big data, and interinstitutionality, where patient data can be accessed by different providers. However, the transition raises its own concerns, a major one being security and privacy.

Healthcare data is sensitive, and must be protected not only from external attackers, but also from unauthorised access even from inside the ecosystem, as stated by Esposito et al. (2018). They discuss the necessity of proper balance between security measures and privacy, and accessibility. There is also a challenge with regulations, as GDPR set by the European Union creates challenges with data permanence, which is one property of blockchain – GDPR lets individuals request their personal data to be erased from data systems, which is an important aspect to be taken into consideration. An example of Azaria et al. (2016) proves that with careful permission management and security measures, it is possible to maintain relatively high accessibility while enforcing proper privacy and security from unwanted parties.

3.1.2 Research on Blockchain in Internet of Things

As IoT, or a system where devices are connected to each other, is an emerging concept, Chanson et al. (2019) examined blockchain technology as a potential way to protect sensor data and ensure user privacy in the system. They showed concern in how risky it is to share sensor data with third parties – with third parties involved in data processing, unintentional access by malicious adversaries is a major security risk in IoT systems, which can in worst case scenarios affect global systems very rapidly. In their research questions they assess fundamental security challenges in the context of IoT sensor data, and the value of blockchain technology for sensor data protection systems.

Chanson et al. (2019) see blockchain as a potential security solution for IoT systems. They emphasise blockchain's key properties, such as decentralisation and consensus principle, as useful for mitigating data security issues arising in the IoT. They saw the potential of decentralisation, which is especially valuable for large systems where there might be conflicting interests between peers, allowing high security even in those situations. This is further enforced by consensus principle, which ensures data integrity in the system. However, there are some aspects to be considered when implementing blockchain into such systems, the biggest issue being privacy. Chanson et al. (2019) state that permissionless blockchain's would not be suitable for IoT systems that handle highly sensitive data, meaning that proper privacy measures would have to be taken into consideration. There is also another challenge for permissionless blockchains, which are restrictions. One has to take into account permissionless blockchains' scalability and potential prohibitive transaction costs.

3.2 Practical Solutions

As the blockchain has seen an emerging amount of interest in recent years, use in various industries of blockchain have been steadily increasing. Mettler (2016) examined different general examples where blockchain technology has been researched. As Bitcoin is assumed to be the first mainstream blockchain application, financial sector has strongly been intrigued on the concept of ownership of digital goods. Besides that, Mettler (2016) discussed different examples of blockchain applications in music industry, mailing systems, and healthcare, which Mettler (2016) discusses especially. Some notable fields of blockchain for healthcare include health management and patient information systems. In this case, especially the case of patient information systems is emphasised, as it correlates with the stakeholder's case where sensitive user information is being handled, and an implementation of IoT solution in a pharmaceutical supply chain, where data of different actions are the primary information being.

3.2.1 Implementations of Blockchain in Patient Information Systems

It has been proven that blockchain can potentially be used for patient information management. One such use-case of blockchain being used for handling sensitive information mentioned by Mettler (2016) is the case of Estonia's digital health infrastructure. He summarises that Estonia together with a company *Guardtime* implemented a blockchain-based healthcare platform, where all the information of medical treatments performed in Estonia are accessible by Estonian citizens, healthcare providers, and healthcare insurance companies. Mettler (2016) states that this vast example proves that a blockchain can be used to operate a public health infrastructure.

A more extensive example case of healthcare data management is a proposed solution called *MedRec*, as researched by Azaria et al. (2016). Like with Estonia's case, *MedRec* is also used for logging and accessing medical records. As sensitive information is being handled, security and credibility is something to be paid careful attention to. These security concerns are tackled by careful choice of certain blockchain properties.

When it comes to authentication in *MedRec*, a permissioned blockchain is used, which means only authorised parties can access the system (Azaria et al., 2016). These can be enforced by using smart contracts, which allow patients to control access to their medical information. Confidentiality is ensured by encrypting the medical records and storing them in a distributed manner between users, where each provider and patient has their own database. These encrypted databases are only accessible through a key available only to authorised parties. This kind of modular design in *MedRec* integrates with existing data storage solutions, promoting interoperability and making the system highly convenient and adaptable.

A typical blockchain requires miners to operate. Azaria et al. (2016) proposed two incentives for mining in *MedRec*, where the first is based on Ethereum incentivising model, and the second is more oriented for medical researchers and healthcare authorities. In Ethereum's incentivising model, transactions require digital asset 'Ether', which can be

earned by mining. This way care providers are incentivised to participate in mining to fund their activities in the blockchain network. The second model is based on setting 'bounties' on blocks containing records of desired type, which are then mined, ultimately rewarding the miner with the data in the block.

3.2.2 Implementation of Blockchain in a IoT-based Logistics System

Another field of application for blockchain is IoT data management. As IoT systems consists of a network of numerous individual sensor devices, blockchain can benefit by generally enhancing the systems' security, but also providing data integrity and traceability for crucial sensor data (Kale & Rathod, 2023). One example of blockchain-powered IoT systems being implemented includes pharmaceutical industry.

Bocek et al. (2017) introduced a blockchain-powered IoT solution for pharmaceutical supply chains. Fraud is a major problem with pharmaceutical supply chains, where blockchain is seen as a potential solution to. Some examples they mention for fraud detection include *Blockverify*, which introduces blockchain as a way to avoid counterfeit and forged pharmaceuticals, which cause a large number of deaths annually. They described blockchain to enable tracking and verification of ownership of the goods, as every change will be recorded in the blockchain. For identity management, Bocek et al. (2017) summarise that identification information can be stored in the blockchains, some examples being *UniqID*, which manages biometric data such as fingerprints, or *SolidX*, which is an identity management software for location access, authentication, fraud prevention, and anti-phishing, to name a few.

Bocek et al. (2017) examined more in-depth one real-life example of blockchain application in a pharmaceutical supply chain created by a start-up called *Modum.io AG*. To meet new EU regulations regarding logistics of pharmaceuticals created a demand for specialised services to meet the regulatory requirements. These include requirements to monitor temperature constantly, and requirement to include a serial number in every pharmaceutical package. *Modium.io AG* together with University of Zürich developed a

blockchain application to monitor temperature of each parcel during the shipment of the products, while reducing operational costs in the supply chain.

The system works by having IoT devices monitor the temperature of each parcel (Bocek et al., 2017). Especially in this case, there is a legal obligation to keep track of temperature history as well as possible, which is why blockchain's feature of data integrity comes in handy. The smart contract in this solution executes when a new shipment of drugs is delivered, which executes an action that checks that the shipment complies with the temperature regulations.

Just like in the MedRec example, Modium.io AG's solution also consists of separate database for sensitive data, and a separate blockchain component for functional data - the database stores user data and raw temperature data communicated by the end users, and the blockchain contains measurement results of whether the temperature complies with regulations (Bocek et al., 2017). The blockchain used in the solution is Ethereum, as it provides smart contract functionality, is cost-effective over developing and maintaining a proprietary blockchain, and is easy to implement thanks to its wide adoption and interoperability.

There were some things to take into consideration in the reported iteration by Bocek et al. (2017). Some of them concerned usability of the solution, such as including improving user experience for the end users. The major improvement points regarding the blockchain, however, were mainly performance related – one concern Bocek et al. (2017) noted was that the Ethereum clients were not stable, so server capacity had to be increased to prevent denial-of-service attacks on the Ethereum blockchain. Other concern shown was about smart contract's functionality. As Ethereum is computationally costly to run, modifications to the smart contract can increase the running costs to a point where the contract couldn't be deployed anymore.

3.3 Key Takeaways

To be able to design a prototype revolved around blockchain, prior research and case studies were investigated. Especially scenarios that correlate with the stakeholder's use-case were examined, them being healthcare industry with patient data, and IoT systems. The main key takeaways are summarised in Table 1, briefly describing the research topics of the examined research that had high emphasis, and the key takeaways that benefit this study.

The key takeaways from blockchains in patient data systems were how blockchain can be used to manage confidential data, and especially in settings where a proper software architecture has been long existing already. For example, Esposito et al. (2018) explored the idea of healthcare data being migrated to cloud using blockchain, while Azaria et al. (2018) proposed an actual blockchain implementation, where the blockchain would promote seamless data access between different health institutions.

Discussions on blockchain technology for protecting sensor data in IoT systems saw potential in one of blockchain's main principles, decentralisation. Chanson et al. (2019) proposed how in large IoT systems consensus principle would mitigate the risk of malicious activity thanks to the nature of consensus principle ensuring that mutually untrusting parties can trust the system. However, Chanson et al. (2019) pointed out that blockchain would not be suitable for storing sensitive data due to privacy reasons.

A more detailed case study on how blockchain is utilised in blockchain introduced blockchain as a component for functional data. Bocek et al. (2017) introduced a blockchain-based supply chain management system, where the IoT sensors would update on the shipment's status. A smart contract was used on delivery validating the shipment, which results were propagated to a blockchain system – simply put, the raw data from IoT sensors was kept in a conventional database, while the blockchain stored activity related data, which in this case is whether a shipment is valid or not.

Even though the solutions in practical examples investigated work as an adequate starting point, they cannot be directly applied to the prototype in the study. As the objective of the prototype is to be able to make user data systems more secure and transparent by monitoring activity instead of handling the user data itself, the healthcare systems' principle of migrating patient data to the blockchain is out of question, not to mention that the user data in this particular case should be kept confidential outside of any unnecessary exposure, which was also pointed out by Chanson et al. (2019).

The solution in IoT system's case had a more fitting solution where the data storage was divided depending on what type of data was to be stored: raw data, which is potentially confidential, was saved in a centralised database, while more functional data was saved in the blockchain. However, this solution also cannot be utilised without modifications. The first notable difference is that in the prototype in the study does not involve IoT systems. Outside that, the functional data saved differs with the objective of the prototype – the IoT solution's blockchain was used to validate whether the shipment meets regulations, while the prototype in this study is designed to log activity related to user data.

Research	Research Topic	Key takeaway(s)
Esposito et al. (2018)	Moving healthcare data to cloud using blockchain, and possible security risks.	Consider the balance between transparency and privacy when using blockchain for storing data.
Chanson et al. (2019)	Blockchain as a potential way to enhance security and privacy of IoT systems.	Blockchain is not suitable for highly sensitive data; consider what to store in blockchain.
Azaria et al. (2016)	Enhancing accessibility of medical records using blockchain.	Blockchain should be implemented as a separate component alongside a centralised database.
Bocek et al. (2017)	Enhancing shipment integrity in pharmaceutical supply chains using a blockchain-based IoT system	Blockchain should be used to store functional data, while a database should be used for storing raw data.

Table 1. Examined prior research and use-cases and key takeaways summarised.

4 Research Approach

This study is conducted using constructive research methodology. The primary aim of constructive research methodology is to develop practical solutions for real-world problems., or 'constructions', while simultaneously contributing knowledge to the whole field of study (Kasanen et al., 1993). In this study, a proof-of-concept prototype is created, and its functionality demonstrated and evaluated.

Constructive research methodology combines theoretical background with practical experience, which allows for solutions that are both theoretically robust and practically applicable (Oyegoke, 2011). The research process combines rigorous theoretical and empirical validation on proposed solutions by demonstrating their effectiveness through testing, justification, and validation. This way the process ultimately aims for innovation by creating solutions for relevant practical problems and can also have commercial value.

The practical solution in constructive research methodology doesn't exclusively have to be an actual usable product. Some other forms the solution can take are design and concepts (Crnkovic, 2010). In other words, the solution could also be an idea that fills a knowledge gap in more theoretical problems, or an analysis of use and performance, which aims to help researchers understand and improve systems.

This research approach is especially suitable for this research problem, as it is a real-life problem in the field of software engineering where a stakeholder wants a practical research and demonstration validated by theoretical background of whether blockchain can be used for enhancing security and transparency in user data systems.

According to Lukka's (2001) model, constructive research methodology can be divided into seven main steps. Constructive research starts by looking for a relevant problem, which the research attempts to find a practical solution to. The takeaways of the research should also contribute to the field of study. This is the first step of the methodology.

As constructive research aims to solve a practical problem, a stakeholder is usually needed (Lukka, 2001). The stakeholder should contribute an equal amount of effort in completing the research as the researcher. The stakeholder ought to provide the researcher with a research team, while the researcher has to be able to convince the stakeholder that the research is worth the effort. This is the second step of the methodology.

The third step in the methodology in Lukka's (2001) model is to acquire proper practical and theoretical background of the topic. A proper prior knowledge of the problem has to be obtained through literature review, exploring research and solutions that have possibly covered topics that are relevant to the research problem. This is to ensure that the researcher understands what has and has not been done in the field already, but also for the researcher to be able to recognise and analyse the theoretical contribution of the solution in the field of study.

The fourth step is to design the solution, or also known as a construction (Lukka, 2001). This step is critical for the research, as it is not worthwhile to keep on with the research if a unique and innovative solution cannot be designed – only applying already existing solutions to the solution is not constructive. The design process should be done together with the stakeholder and can be done iteratively.

After designing the solution, it should be implemented and tested (Lukka, 2001). This step is what makes constructive research methodology distinct from other methodologies. The solution is tested against the stakeholder’s requirements, which are usually in commercial standards. This can also be called ‘market testing’, and is the fifth step of the research methodology.

As the sixth step of the methodology, the test results are evaluated (Lukka, 2001). The results are examined and reflected upon the requirements, whether they meet them or not. If not, then naturally the reasons why the results didn’t meet the requirements are discussed, and can the unsatisfactory results be avoided in other settings.

The final step is to recognise and analyse the theoretical contribution of the research (Lukka, 2001). In this step, the researcher should conclusively be able to evaluate their contributions to the whole knowledge of the field of study. This can be done, for example, by reflecting the findings to prior knowledge. The contribution can be in form of innovation, or entirely new construction, which brings new knowledge to the field of study by its originality, or by creating a dependence on prior knowledge by applying and improving prior knowledge into the specific research problem.

The procedures done in this study are reflected to corresponding steps described by Lukka (2001) in Table 2.

Procedure	Corresponding Procedure in this Study
Find a relevant practical problem	A real-world case commissioned by a stakeholder – design a blockchain application to track any changes in user data.
Look for a stakeholder	As the research is commissioned by the stakeholder, the stakeholder therefore exists.
Acquire proper practical and theoretical background	Literature review done about blockchain (Chapter 2) and investigation about applications in IoT and medical record handling (Chapter 3).
Design a solution	A proposed design was created based on knowledge acquired from prior designs (Chapter 3).
Implement and test the construction	A prototype application was created based on the design, simulating what the design potentially could be used for. The prototype's functionality was demonstrated to the stakeholder.
Evaluate the solution	The stakeholder saw the prototype as a sufficient proof of the design's potential in solving the research problem.
Recognise and analyse the contribution	The research contributed to blockchain research related to implementation in enterprise systems.

Table 2. Procedures of the study briefly reflected to Lukka's (2001) procedure descriptions.

5 Execution

This chapter gives an overview of the development process of the proof-of-concept prototype. The overview begins with defining specifications, following with a reflection of prior knowledge, and applying it into a design. Finally, the creation process of the prototype and testing procedures are documented.

5.1 Motivation and Requirements

The primary motivation for the prototype is that in a system owned by the stakeholder, potentially a massive amount of users is being managed in a distributed manner; the stakeholder's system is to be sold to different institutions and customers internationally, which creates a need for the administrating users to be able to track user activity in the system. This means that the requirements of the prototype are also set by the stakeholder, as the prototype will primarily benefit the stakeholder's system.

The main requirement of the prototype set by the stakeholder is to be able to prove to customers that any actions done in the user data system can be tracked. The requirements are summarised in Table 3. The first requirement (Requirement no. 1) is to implement the blockchain prototype which tracks activity in a user data system. The second requirement (Requirement no. 2) is related to the prototype's functionality; the blockchain prototype should give a detailed enough description of the activity. The information should contain three important points of information: 'who did', 'who it was done to', and 'what was done'. The final requirement (Requirement no. 3) is a complimentary requirement related to security, and correlates partly with the second requirement. When developing the prototype, an important remark to keep in mind is that the prototype should not reveal any confidential information when in use. Particular attention must be given to the design of activity logs to align with specified requirements, while also considering architectural solutions.

Requirement no.	Description
1	The prototype should record a log of activity using blockchain
2	The log should contain information of three aspects: who did, who it was done to, and what was done
3	The prototype should not reveal any confidential information

Table 3. Main requirements of the prototype summarised. The requirement numbers are on the left column and the description on the right one. The requirements are not in any specific order.

5.2 Design

The design process was started with evaluating the optimal way of implementing a blockchain system by reflecting on prior research. One major factor to be taken into consideration is that the stakeholder's system has a pre-existing user data system, which cannot be migrated completely into a blockchain-only system. Another factor to take into consideration that it is not necessary to store any user data in the blockchain, other than a description of committed activity in the user data system.

Research and case studies on healthcare patient systems had a lot of emphasis on confidentiality, and implementation on pre-existing patient data systems. Restricted access was also mentioned. As patient systems have been running as-is, simply migrating everything was not an adequate solution in the studies – not to mention that there were many security challenges if the actual confidential patient data as its entirety would was to be stored in the blockchain system. This led to a design where the blockchain would be implemented as a separate component.

The prototype's scenario has a similar setting as in patient system cases. One requirement is that the user data itself shouldn't be exposed, as the focus is on what's happening to the data instead of the data itself. The prototype should also be integrated to a pre-existing software, meaning that if blockchain is to be used as an integral part of the entire architecture instead of being a separate independent component, it would mean that the whole pre-existing software would have to be reconstructed. This leads to a conclusion where similarly to the prior research blockchain shall be implemented as a separate component in the architecture.

Second major point as mentioned is to consider what should be stored in the blockchain. The user data is confidential and should not be exposed to anyone outside authorised access, and thus it is out of question that the user data itself should be stored in the blockchain, as even though the blockchain would have restricted access, data permanence and its distributed nature can cause unnecessary challenge. In the investigated case where blockchain was used for IoT system in pharmaceutical supply chain, a database was used to save raw data, and the blockchain was used to save data related to the shipments' status during the logistical cycle. The data was generated to the blockchain using smart contracts, which triggered when a shipment was completed.

The model of pharmaceutical supply chain's IoT system's of separating raw data to a conventional database and functional data to a blockchain network fits this prototype well. The raw data, which in the prototype's case would be the confidential user data, would be stored in a database managed by the stakeholder, which allows the stakeholder to have complete control of who can access the user data as desired. Similarly, the blockchain should contain functional data.

The prototype is required to be able to track any changes happening in the user data system, or more specifically who edited whose data – naturally, to be able to properly utilise this kind of data, it should be stored somewhere. The blockchain is an ideal place to store this kind of data due to the persistency and permanency of data that are some

key attributes of blockchain, and the data being saved benefiting from these attributes, as the data is used to track what has happened in the user data systems. A smart contract should be implemented to automate the generation of data. The smart contract would trigger when a change in user data system is committed, creating a virtually eternal log of the transaction.

An additional concern is related to the blockchain itself. Prior research brought up evaluations whether to use a pre-existing blockchain or create a proprietary one, and it is a relevant question for this prototype too. Many factors lead to greatly favouring to use a pre-existing blockchain technology, and especially Ethereum. First, all prior research that were examined and mentioned which mentioned which technology they used mentioned using Ethereum. A notable feature of Ethereum is also the support for smart contracts, which are an important part of this implementation. Second, creating a proprietary blockchain would create a lot of security concerns to pay attention to; the proprietary blockchain should have at least the researched or equivalent security features to be able to be trusted in security standpoint. This creates a lot of additional work to the development process. As the primary goal of the prototype is to utilise blockchain technology to a perform specific task, and not the creation of blockchain itself, using a pre-existing blockchain, and especially Ethereum, is the preferable choice.

These points led to a conclusion that principles applied in healthcare and IoT systems found in prior research were an adequate foundation – healthcare systems in the prior research also relied on already existing systems, and similarly cannot simply be migrated, and additionally contain sensitive data that cannot afford to be compromised by storing them in a system where data is virtually permanent. IoT systems in the other hand, have showed a way of how to utilise blockchain in the system: in the IoT solutions, a database was used to store raw data, while blockchain was used to store functional data, enforced by smart contract. There were also evaluations whether to use a pre-existing blockchain technology or to create an own one. A major factor to choose a pre-existing blockchain technology Ethereum over creating a proprietary one included having to pay careful

attention to the security level of the blockchain, which would create a significant additional work outside the scope of the prototype.

After reflecting to these points a general design was created, as seen in Figure 5. The whole solution itself can be considered to fulfil Requirement no. 1. Overall, the blockchain network wouldn't disrupt or change the overall architecture of the already existing system, but instead, is implemented as a separate component. Smart contracts are implemented in the backend, or a software layer that interacts with components not seen by the user such as database, meaning that when a database call is created, the smart contract would trigger and store a description of the database transaction in the blockchain, leaving a trace of who did, what did, and to whom (Requirement no. 2). Confidentiality is also ensured, as the log will be implemented so that any user data is not revealed and will not be needed due to the nature of the design (Requirement no. 3).

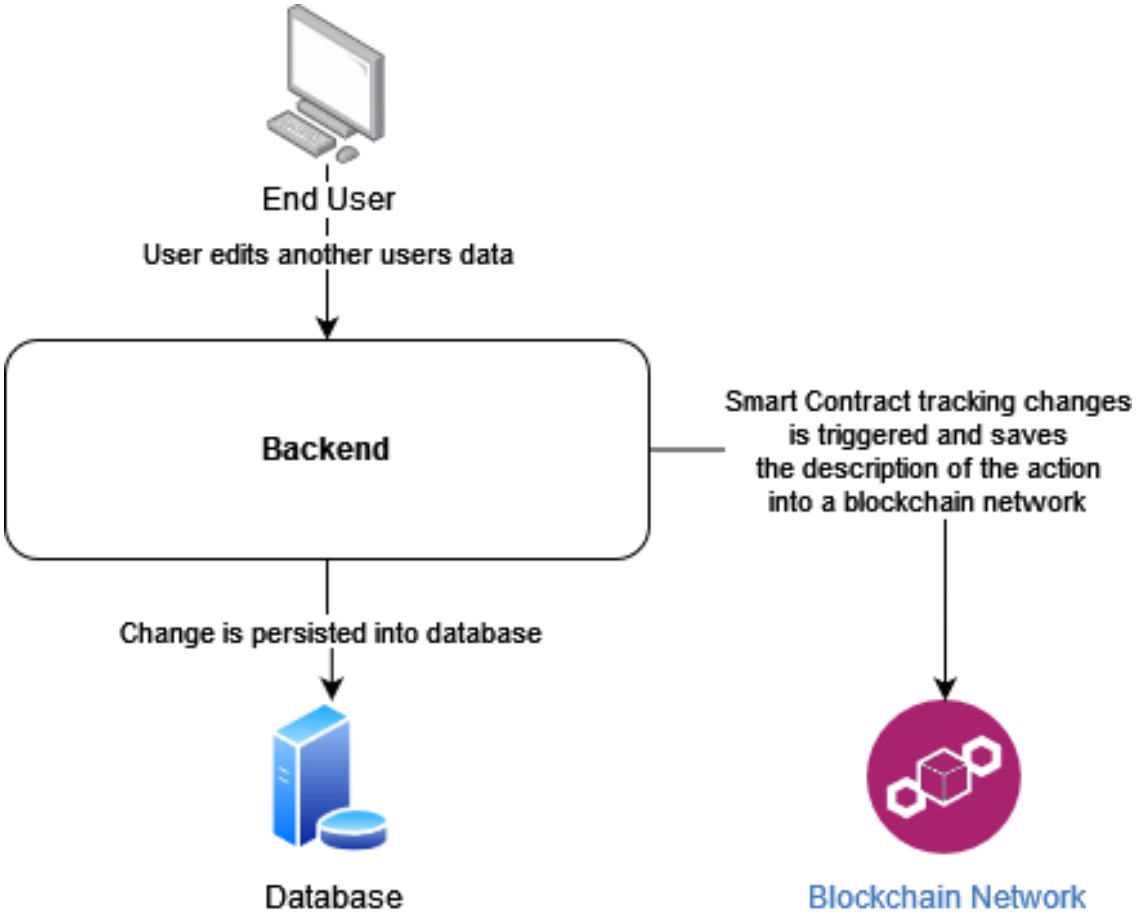

Figure 5. A simplified flowchart of how the blockchain system would be implemented.

5.3 Implementation

The implementation process is divided into two major phases, the first being preparation and second being the actual implementation. The preparation phase consists of building of a test environment. The second step consists of the implementation of more precise planning how the blockchain should be implemented in the test environment. The second phase also includes a brief evaluation of which blockchain technology should be used, and how should it be utilised.

To safely prototype the design without the fear of interfering the development of the production software, a test environment was created. The test environment consists of a locally run database, which includes dummy user data, along with simplified backend

(layer that communicates with the database) and frontend (the layer that communicates with the user, as in user interface) tailored to simulate a user data management system in a commercial software. The test environment uses a PostgreSQL database, JavaScript-based React user interface, and a .NET Entity Framework backend. The architectural diagram can be seen in Figure 6.

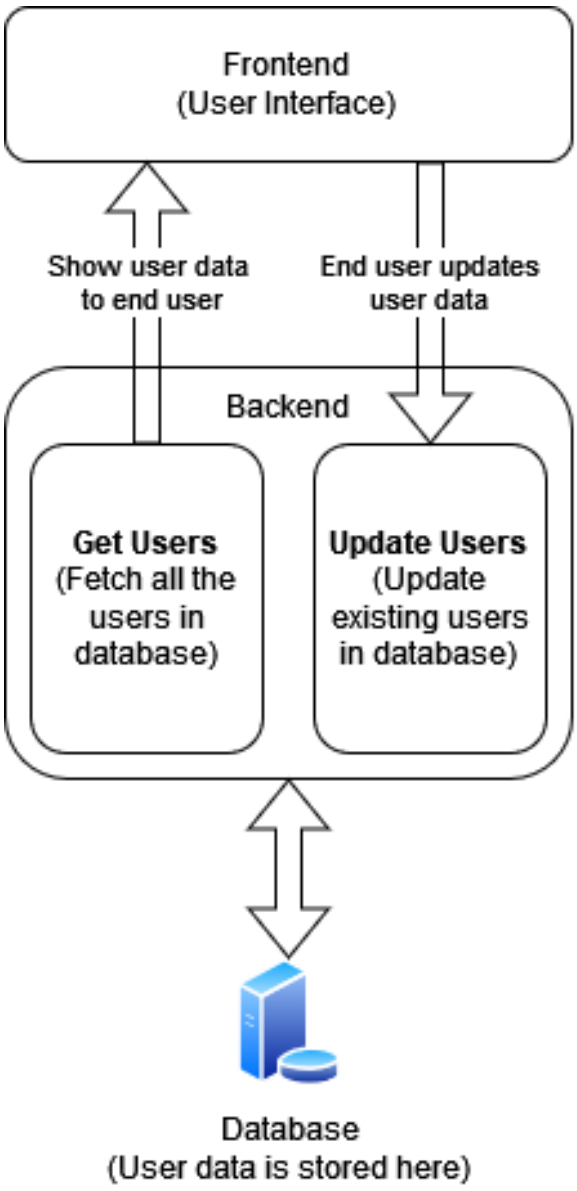

Figure 6. Basic structure of the test environment. All the components are hosted locally. Frontend contains the user interface, where the end user can manage user data, while backend communicates between the user and the database.

Implementation of the blockchain component was then started after creating the test environment. As designed, a local Ethereum blockchain was created for prototyping purposes using tools called ‘Truffle’ and ‘Ganache’. Truffle and Ganache deploy a simulated blockchain to run locally on a personal device for development purposes, opposed to blockchain’s principles of being a decentralised network. This allows safe testing and debugging of blockchain applications. Ganache provides a user interface tool, which is useful for development purposes, as it shows simulated users’ addresses, deployed smart contracts and transactions, to name a few. In this prototype it will be used as a tool especially for inspecting transactions, as the main functionality of the implementation revolves around them.

Next, a separate application programming interface (API) was created for the sole development and deployment of smart contracts. To summarise, an API is a boundary between different components in a programme; an example of this is a backend component, which communicates between the user interface and database (Garrod & Alrdich, 2014). APIs ensure modularity and that data can also be accessed from external sources outside the organisation (IBM Cloud Education, 2020). This kind of solution is ideal for this prototype, as we want to keep the blockchain implementation as a separate component instead of fully integrating it into the primary application. Using the API, smart contracts are deployed to the blockchain and can with little modifications be called from the backend functions.

A single smart contract was created in this prototype: a smart contract for tracking changes in database. In the test environment’s database, there is a table called ‘Users’, with columns, or in other words values to be saved, ‘Id’ (which is in GUID form, a 128-bit random series of characters), ‘Email’, and ‘Login’. The smart contract is programmed to save a description of the database transaction in the blockchain in a following form: ‘User [GUID] changed a value of user [GUID]: [changed fields]’. In this form, the users are only identifiable by authorised parties, theoretically eliminating potential security risks in case the information falls into wrong hands. It is worth noting that in this

implementation the description is formed in the backend instead of the smart contract itself, though due to the programmable nature of smart contract, it is not entirely out of boundaries to be able to make the smart contract form the description in the smart contract itself instead of backend, further improving modularity.

After creation of the smart contract, it is connected to the backend code, where the function handling a query for editing information of a particular user will call the smart contract when the conditions are met. To put it simply, when a user requests a change of information in for another user, the function checks which fields are changed, and creates a description for the smart contract to be saved in the blockchain (Figure 7). After the smart contract is connected to the backend successfully, the prototype will undergo testing whether it meets the expectations.

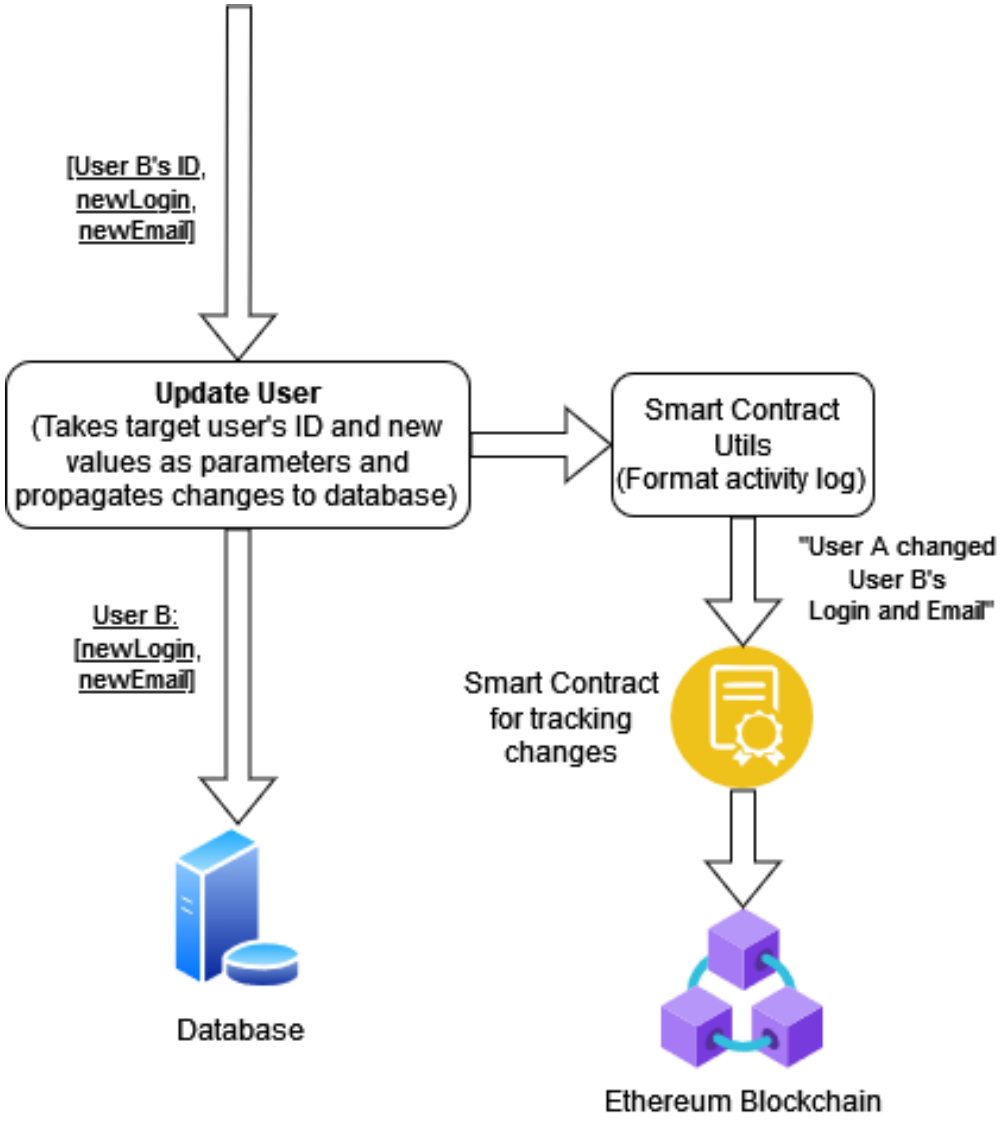

Figure 7. A flowchart of how smart contract is implemented.

5.4 Testing

To validate the functionality of the prototype, it must be tested. As the prototype is a piece of software, the best way to test it is by using software testing principles. Software testing ensures quality, reliability, and security of the software – software testing verifies the correct functionality of the software under various conditions and meets performance and security requirements (Spadini, 2021).

For software testing, there are many different methods. For this case, methods called *integration testing* and *user acceptance testing* are used. Integration testing intends to test the behaviour of the whole programme by combining individual components into one comprehensive group (Weikle et al., 2019). In user acceptance testing, the overall behaviour of the software is tested from end-user standpoint (Suman & Sahibuddin, 2019). Feedback will be given on whether it meets all the requirements set by the customer, which in this case is the stakeholder.

First, integration testing is conducted to ensure that the prototype works, which is followed by the user acceptance testing conducted together with the stakeholder. The integration testing will be conducted by testing the functionality of the components and their communication between each other by using the prototype how they're intended; some key aspects to keep an eye out for are whether the frontend successfully communicates with the backend, the smart contract triggers upon a database change, and a transaction is completed successfully and can be seen in Ganache. Integration testing will enforce especially the Requirement no. 1.

As mentioned, user acceptance testing will be conducted together with the stakeholder. After the functionality is confirmed through integration testing, the functionality is then demonstrated to the stakeholder. The functionality simulates real life use, which is to simply change user data as a user using the software. The user acceptance testing primarily looks out for Requirement no. 2 but will also confirm that the prototype complies with Requirement no. 3.

To begin the integration testing of the prototype, all the components are first launched locally, which includes the database, backend, frontend and then the blockchain network. The test is run to simulate a real-life usage as closely as possible, meaning that other user's data is changed using a user interface (Figure 8), and the transactions can be inspected using a separate piece of software, which in this case is Ganache (Figure 9). A

logged in user is simulated by injecting a certain user identifier into the code, as it is not in the scope of the prototype to develop a functional login system.

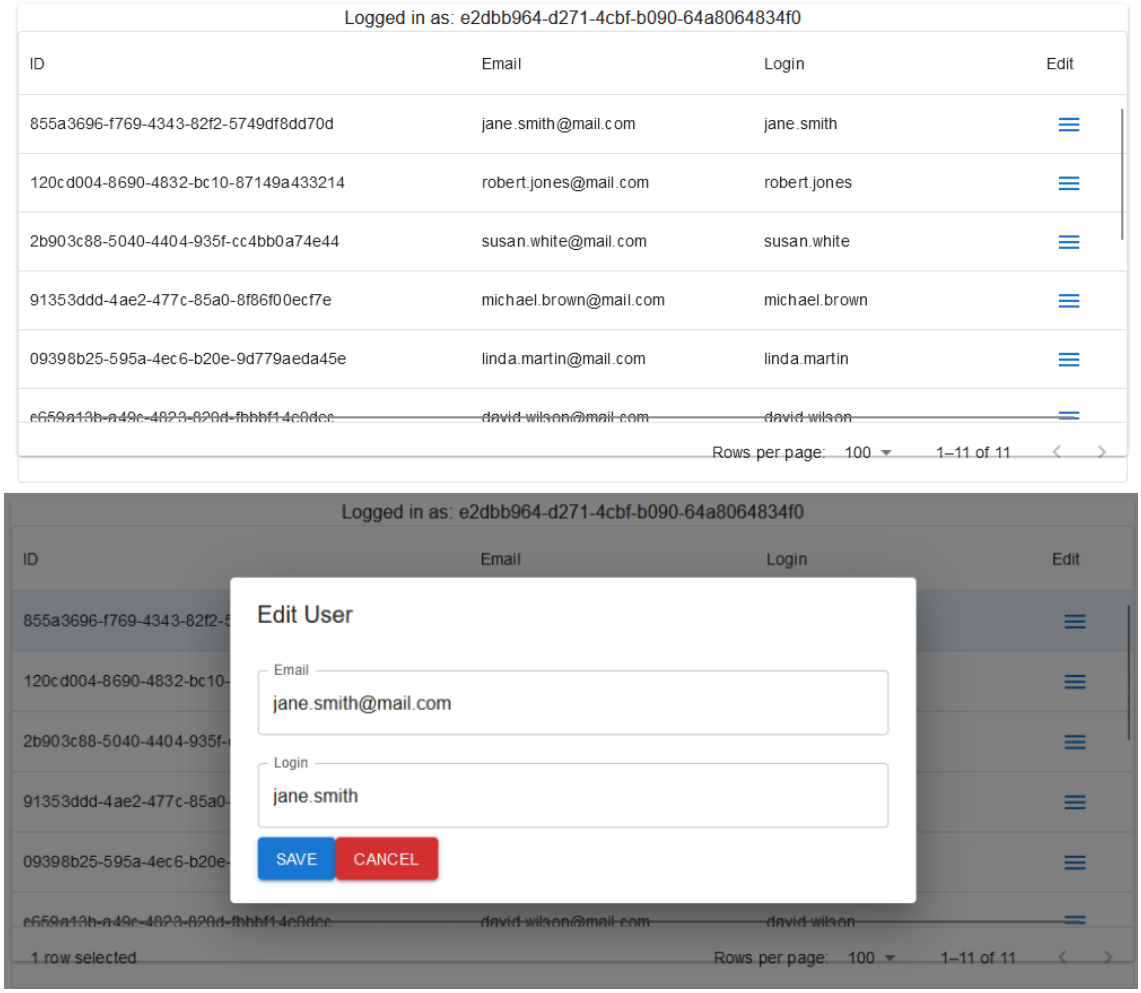

Figure 8. A rudimentary user interface to simulate editing of user data.

The rudimentary user interface contains the bare functionality needed to demonstrate the prototype (Figure 8); it contains a list containing user data from a database, and a pop-up where the data can be edited. Ganache contains a lot of different tools, but the most important tool for testing this prototype is smart contract inspector, which shows all the transactions that has happened using the smart contract (Figure 9).

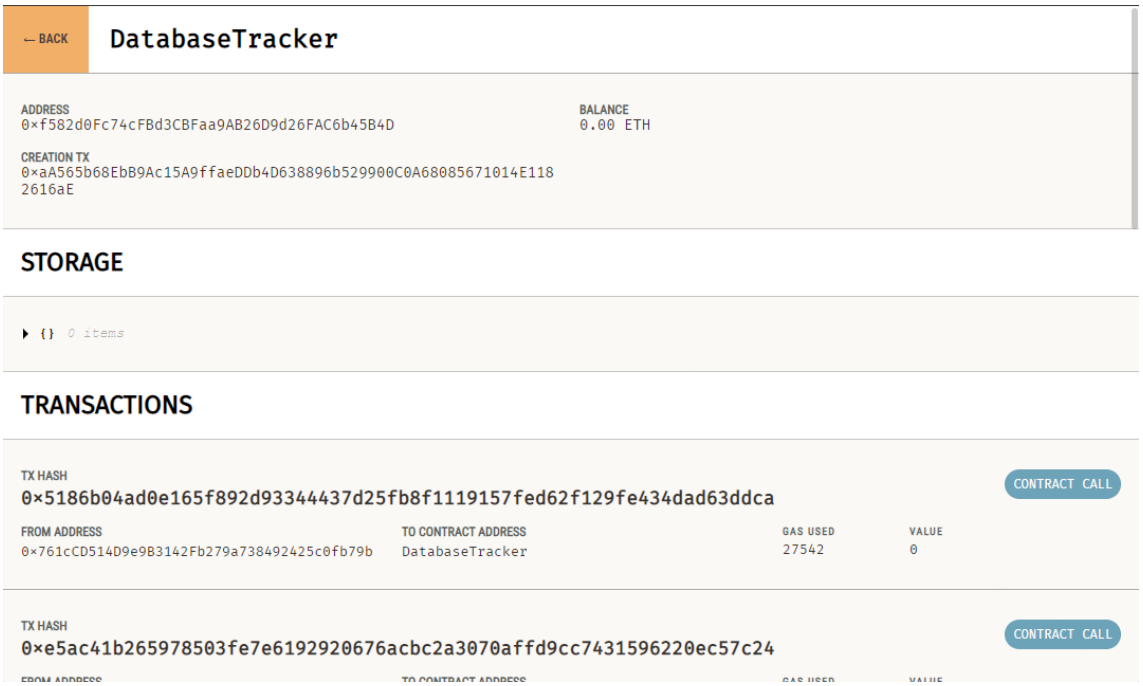

Figure 9. A view of a specific smart contract 'DatabaseTracker' in Ganache. The smart contract tracks changes in user data.

The testing process is replicated a few times to prove that the system works indefinitely. The process is repeated each time with slightly different variations. In the first iteration, only the 'Login' field of another user is changed. In the second iteration, only the 'Email' field of another user is changed. In the third iteration of the test, both fields 'Login' and 'Email' of another user are changed. These same iterations are repeated for the stakeholder as a user integration test, assuming that the integration tests are successful.

After these changes were committed, Ganache should be showing the transactions, which it did. In Figure 10, Figure 11, and Figure 12, details of the transactions of all three different test iterations can be seen. The figures contain a lot of different information, but the information relevant to this test is the field 'Inputs', which shows the description of the tracked change in database, where the first series of characters is the editor's identifier, the second is the target's identifier, following with the changed fields listed.

The first test scenario was to change only the field ‘Email’ of another user. The results can be seen in Figure 10, where Ganache’s transaction inspector’s ‘Input’ field indicates us that a user has changed an email of another user; in ‘Input’ field a description can be seen, which indicates that which user did something and to which user, and which fields were changed. This verifies that the prototype works.

Figure 10. Test case, where the user has changed field ‘Email’ of another user.

The second test scenario was to change the field ‘Login’ of another user. Just like in the first test case, we can see the results in the ‘Input’ field in Figure 11, which indicates that ‘Login’ of the user has been changed.

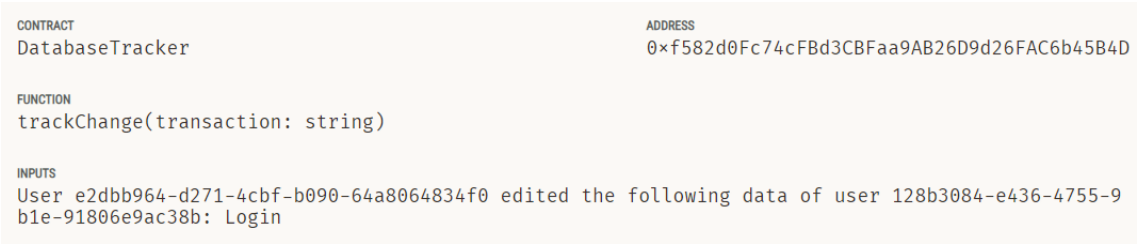

Figure 11. Transaction details for changing the field ‘Login’ of another user.

In the third and final test scenario both ‘Email’ and ‘Login’ of another user was changed, and the results can be seen in ‘Input’ field in Figure 12 verifying that the prototype successfully records a log of the activity.

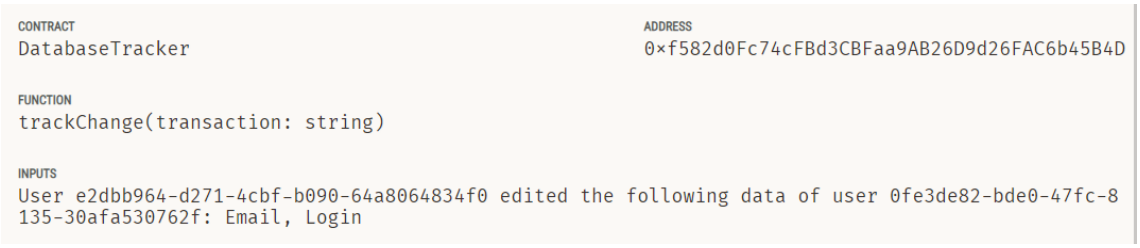

Figure 12. Transaction details for changing both fields 'Login' and 'Email' of another user.

6 Results

This chapter introduces the final design and implementation of the prototype and evaluates whether the test results were successful and satisfy the requirements of the stakeholder; the prototype and the test results will be reflected to the requirements of the prototype. As the level of implementation is only proof-of-concept prototype, there occurred points of improvements to be considered for further implementation, which are discussed in the latter part of the chapter.

6.1 Final Prototype

The final prototype's architecture is described in a diagram as seen in Figure 13. A change is made from the frontend (or User Interface), which then propagates the data to the backend, which has two key components: database connectivity for handling the user data between the database, and a smart contract for detecting activity related to user data, which then creates a transaction of activity log to the Ethereum Blockchain.

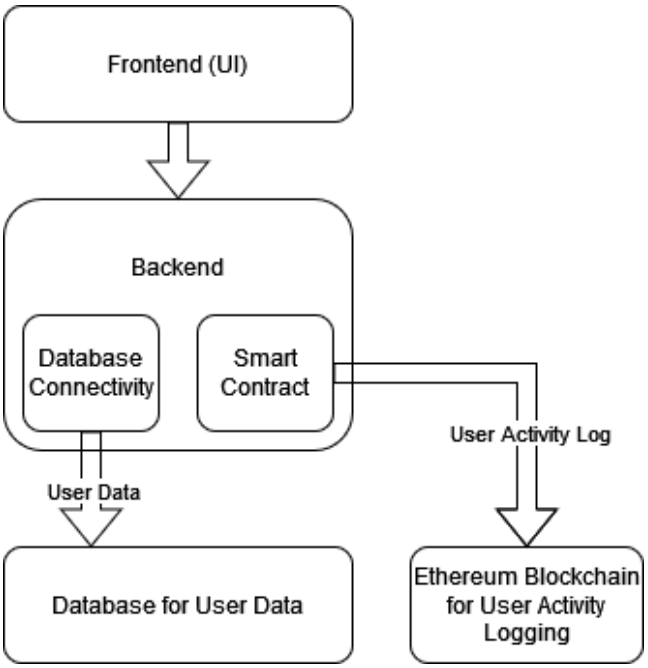

Figure 13. Architectural diagram of the final prototype.

The final prototype meets with the requirements – as the primary requirement of the prototype was that it should give a detailed description of who did what, and to whom, or in this specific scenario, who edited whose data, and more specifically what data. The prototype meets these requirements with the component: a smart contract which detects any changes happening through the backend, and finally creates a virtually permanent log of the activity in the blockchain.

Additional remarks on top of the primary requirement included that the actual user data should remain confidential, or in other words, should not be accessed by anyone than authorised parties. This is ensured in the prototype by keeping the user data in a centralised database with restricted access, while the blockchain only stores data related to user activity, and in a manner that even if the data leaked from the blockchain, the information would not reveal any confidential info thanks to the format the activity is logged as – the only identifying information shown in the blockchain is user identifier in encrypted form, which is only useful to users who have authorised access to the actual user data system, which again is separate from the blockchain component.

To summarise, the design choices are reflected with requirements in Table 4. As confirmed in the testing process, the prototype works as intended, fulfilling Requirement no. 1. The main components included the UI, backend for communication with database, which also triggers the smart contract, the database itself, and a blockchain network – as seen in integration testing, all the components work as intended, and the end-user can see a transaction record when inspecting the blockchain after a change is done using the UI, as expected. Upon inspecting the transaction data, we can see the three required aspects: who did, what did, and to whom (Requirement no. 2), albeit in a manner where any outsiders will not be able to extract any confidential information out of the log itself. The only piece of information that could potentially be used to access any kind of confidential information are the user identifiers used in the description; only authorised users who have access to the user data system will have any use for the identifiers in shown in the log (Requirement no. 3).

Requirement no.	Description	Corresponding Feature(s)
1	The prototype should record a log of activity using blockchain	A smart contract creates a transaction to an Ethereum blockchain upon user data change using the UI.
2	The log should contain information of three aspects: who did, who it was done to, and what was done	The smart contract is configured so that the transaction data to the blockchain contains a detailed description with the wanted information: 'User A changed data of User B: [fields changed].'
3	The prototype should not reveal any confidential information	The only identifying piece of information revealed is a user identifier string, which is not much of use for outsiders.

Table 4. Requirements reflected with design choices. The left-most column stands for requirement number, middle column describes the description, and the right-most summarises the corresponding feature(s).

6.2 Test Results

To summarise, the tests conducted to the prototype included integration testing to verify the intended functionality, and user acceptance testing to verify that the prototype also works in a desired manner in end user's standpoint. The integration testing process therefore verifies primarily Requirement no. 1 (Table 3), while the user acceptance testing process primarily verified Requirement no. 2 and Requirement no. 3.

The steps for testing documented in Chapter 5.4 were identical for both testing processes, albeit slightly different aspects of the results were evaluated in the respective

test. When observing the testing steps in integration testing standpoint, the prototype works in a desired manner. First, a change in user data using the UI is successfully propagated to backend. Second, as the backend has to be able to communicate with the database in order to further commit the transaction to the database in order for the transaction in to appear in the blockchain. As a transaction can be successfully seen in the blockchain, it is safe to assume that the prototype works as intended, verifying Requirement no. 1 is satisfied.

For user acceptance testing, especially the transactions in the blockchain are the aspects to be examined. The user acceptance testing process was conducted with the stakeholder, who also verified whether the functionality works as desired. The same steps were conducted as in the integration testing process. The transactions could be successfully examined in the prototype, and they contained the required information, 'who did', 'who it was done to', and 'what was done', verifying that Requirement no. 2 is satisfied. Additionally, the only identifying information available in the transactions were the parties' unique user identifiers, which rendered useless for parties who don't have access to the actual user data system. With this result, the Requirement no. 3 is satisfied. In conclusion, all the main requirements were verified to be satisfied in the testing process and thus also satisfied the stakeholder.

6.3 Points of Improvements and Future Prospects

There were a few aspects pointed out by the stakeholder for the future utilisation of prototype into an actual commercial product. First point of consideration are security standards – in this case, further implementation should consider the security standard IEC62443-3-3 along with the rest of the commercial system. This standard was intentionally ignored in this prototype, as it is out of scope for a prototype of this scale.

Another point of improvement pointed out is that in case this prototype is to be implemented in a commercial product the transactions should be presented in a more user-friendly format; one way to implement this is to create an audit log viewer. The scope of

transactions could also be expanded to include, for example, addition, deletion, and viewing transactions. On the other hand, it is also worth to consider what kind of things are worth tracking. However, thanks to the prototype's modularity, adding and removing actions to track should be a relatively straightforward process.

Finally, as the simulated blockchain network in this prototype is suitable only for development use, yet another remark upon implementation to a commercial product is to research alternatives suitable for a release product. The stakeholder showed concern that the blockchain logs shouldn't be openly accessible, so a viable blockchain development tool should be investigated.

7 Conclusions

As the popularity of blockchains is ever rising through the popularity of cryptocurrencies, potential applications in commercial use have also gained interest. Blockchain solutions have become increasingly common in different industries, which also leads to interest in how do blockchains work, and how can they be implemented in commercial software. These lead to forming of research questions, which were: 'How can user data systems' be enhanced using blockchain technology?' and 'How can blockchain be implemented in commercial software?'.

For the first question, this study first aimed to give a fundamental understanding of blockchain's architecture and inner workings, which was followed by discussion of its data security and whether it can be trusted to be used in commercial software. For the second question, an overview of already existing solutions was given. Practical research through a proof-of-concept prototype for a stakeholder was conducted; an implementation of blockchain was designed and developed to demonstrate how blockchain could be implemented in a pre-existing software as a way to enhance data security.

The study first gave an overview of fundamental theory of blockchain. A blockchain is a form of decentralised storage with its strengths lying in *persistency, anonymity,* and *audibility*. Blockchain can be roughly described as a 'list of transactions', a transaction being records of exchanges of assets, such as data, between users. Blockchain can be roughly divided into three main components: *blocks*, that contain transactions as data, *chain,* which are hashes that connect the blocks together, forming a chain of blocks, hence the name, and *network,* as the name implies, is a network of *full nodes*, or computers maintaining the blockchain network by each storing a copy of the blockchain and collectively validating more transactions and blocks into the chain through a process called *consensus*. Consensus process is what enables trust between users that are unknown to each other, with there being many different protocols – one protocol is based on calculating hashes in a brute-force manner, while other is based on random selection of validators among the network of nodes.

Blockchain security was overviewed and discussed. A comprehensive understanding of blockchain's data security was obtained through many viewpoints on what blockchain's security challenges and best practices. Blockchain's security is based on cryptographically encrypted data, and data integrity is maintained through collective validation of transactions with varying advantages and disadvantages between different protocols. When implementing a blockchain, one should also consider the privacy level of the data being managed by the blockchain; is the data something for the public to be seen, or confidential with restricted access. For each solution, a balance between privacy and transparency will have to be considered. Especially for commercial products aiming for the European market, regulations are also something to be taken into consideration.

Different pre-existing solutions were explored from healthcare and Internet-of-Things (IoT) systems' standpoint. Esposito et al. (2018) discussed whether blockchain could generally be used in healthcare data systems, while Azaria et al. (2016) looked more into a specific patient data system implementation. The key takeaways from Esposito et al. (2018) especially considering this study were the idea of implementing the blockchain as a separate component in the blockchain alongside conventional databases. They also brought up a challenge of privacy regulations; GDPR regulations is one thing to consider when considering utilising blockchain in systems involving user information. The use-case of Azaria et al. (2016) introduced a practical take of theory of Esposito et al. (2018); their use-case utilised a blockchain as a separate independent component among all the already existing patient systems, while adequately encrypting the stored data in the blockchain, mitigating the risk of malicious use.

On the other hand, Chanson et al. (2019) considered the potentials and challenges of implementing blockchain in IoT systems, while Bocek et al. (2017) introduced an IoT system used in pharmaceutical supply chains. Chanson et al. (2019) discussed how blockchain could secure IoT systems thanks to its decentralised nature, as IoT systems could potentially contain a large amount of mutually untrusting parties. However, they made a remark of how one should consider what to store in the blockchain, as it is not suitable

for storing sensitive information. These points are concretised in the case of Bocke et al. (2017). In their use-case, blockchain was implemented as a separate component alongside a conventional database – raw data to be hidden from the public was stored in the database, while functional data, information about shipments' validity, was stored in the blockchain.

Reflecting the results of this study with prior research of Esposito et al. (2018) and Chanson et al. (2017), and case studies of Azaria et al. (2016) and Bocek et al. (2017), the research objectives correlated. Both case studies and this study aimed to implement blockchain to improve some aspects in information systems, albeit the use-case differs enough to justify the design of a prototype. In the case of Azaria et al. (2016), the objective of the blockchain system was to use it for logging and storing health information in the blockchain, enhancing accessibility, while in the case of Bocek et al. (2018) blockchain was used for validation shipment integrity in a supply chain. As the point of the blockchain in this study was to enhance data security and transparency in user data systems, there is a knowledge gap between the case-studies and this study, creating the need for a new solution.

A proof-of-concept prototype was designed and developed based on prior knowledge. The prototype was built on a test environment, which simulated a real-life use-case of user data management. The stakeholder set requirements for the prototype, which were: 1. the prototype should record a log of activity using blockchain; 2. the log should contain information of three aspects: 'who did', 'who it was done to', and 'what was done'; 3. the prototype should not reveal any confidential information. Based on these requirements, the blockchain prototype was implemented as a separate component in the software architecture, containing activity logs propagated to the blockchain by a smart contract. The prototype didn't reveal any identifying information as required, and through software testing the set requirements were satisfied, which also satisfied the stakeholder.

Due to the limitations of the study, there naturally occurred points of improvements and possible future research problems. For further implementation in commercial software, proper security standard protocols must be met, and user-friendliness has to be considered. Suitable technologies for deploying the blockchain should also be explored upon further development, as the tools used in this study were suitable only for prototyping and testing purposes. There was also a gap in the theoretical knowledge. Even though research on security features used blockchain's architecture and use-case studies on specific fields such as healthcare, Internet-of-Things, and supply chain management are abundant, research on blockchain's security benefits or use-case studies of blockchain as a way to improve data security in user data systems or equivalent were scarce.

In conclusion, implementing blockchain can enhance security and transparency in user data systems depending on the implementation. Blockchain brings many benefits over a centralised database for logging user activity – blockchain's built-in data integrity ensures that the logs are permanent and trustworthy, as they are virtually impossible to tamper with, and they don't depend on the centralised storage's integrity. The activity logs enforce honest use and helps detecting malicious activity in user data systems. Additionally, thanks to the blockchain implementation's modularity, the activity logs can be easily tailored and scaled depending on the needs of a stakeholder.

References

AlFaw, A., Elmedany, W., & Sharif, M. S. (2022). Blockchain Vulnerabilities and Recent Security Challenges: A Review Paper. *2022 International Conference on Data Analytics for Business and Industry (ICDABI)*, 780–786. https://doi.org/10.1109/ICDABI56818.2022.10041611

Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016). MedRec: Using Blockchain for Medical Data Access and Permission Management. *2016 2nd International Conference on Open and Big Data (OBD)*, 25–30. https://doi.org/10.1109/OBD.2016.11

Babiker, A. G. A. (2022). Digital Signature from Syndrome Decoding Problem. *IACR Cryptol. ePrint Arch.*, *2022*, 1698.

Bitcoin Project. (2018). *Target nBits*. Bitcoin Developer Reference. https://btcinformation.org/en/developer-reference#target-nbits

Bocek, T., Rodrigues, B. B., Strasser, T., & Stiller, B. (2017). Blockchains everywhere—A use-case of blockchains in the pharma supply-chain. *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 772–777. https://doi.org/10.23919/INM.2017.7987376

Campbell, R. (2019). Evaluation of Post-Quantum Distributed Ledger Cryptography. *The Journal of the British Blockchain Association*. https://api.semanticscholar.org/CorpusID:88479840

Chang, S.-Y., & Wuthier, S. (2020). *Dynamic power control for rational cryptocurrency mining*. 47–52. https://doi.org/10.1145/3410699.3413797

Chanson, M., Bogner, A., Bilgeri, D., Fleisch, E., & Wortmann, F. (2019). Blockchain for the IoT: Privacy-Preserving Protection of Sensor Data. *Journal of the Association for Information Systems*, *20*(9), 1272–1307. https://doi.org/10.17705/1jais.00567

Cooper, E., Weese, E., Fortson, A., Lo, D., & Shi, Y. (2023). Cyber Security in Blockchain. *2023 IEEE Conference on Dependable and Secure Computing (DSC)*, 1–11. https://doi.org/10.1109/DSC61021.2023.10354161

Crnkovic, G. D. (2010). Constructive Research and Info-Computational Knowledge Generation. In *Studies in Computational Intelligence* (Vol. 314, pp. 359–380). https://doi.org/10.1007/978-3-642-15223-8_20

Dutta, P., Choi, T.-M., Somani, S., & Butala, R. (2020). Blockchain technology in supply chain operations: Applications, challenges and research opportunities. *Transportation Research Part E: Logistics and Transportation Review*, *142*, 102067. https://doi.org/10.1016/j.tre.2020.102067

Esposito, C., De Santis, A., Tortora, G., Chang, H., & Choo, K.-K. R. (2018). Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy? *IEEE Cloud Computing*, *5*(1), 31–37. https://doi.org/10.1109/MCC.2018.011791712

Garrod, C., & Alrdich, J. (2014). *Principles of API Design*. https://www.cs.cmu.edu/~charlie/courses/15-214/2014-fall/slides/17-api-design.pdf

Geng, T., Njilla, L., & Huang, C.-T. (2021). Smart Markers in Smart Contracts: Enabling Multiway Branching and Merging in Blockchain for Decentralized Runtime Verification. *2021 IEEE Conference on Dependable and Secure Computing (DSC)*, 1–8. https://doi.org/10.1109/DSC49826.2021.9346270

Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., & Capkun, S. (2016). On the Security and Performance of Proof of Work Blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 3–16. https://doi.org/10.1145/2976749.2978341

Haque, A. B., & Rahman, M. (2020). *Blockchain Technology: Methodology, Application and Security Issues*. https://doi.org/arXiv.2012.13366

Hasan, S. K., Zakir, Y. M., & Khondker, S. R. (2023). Permissioned Blockchain-Based Techniques for Refining the Data Security in Commercial Aviation. *2023 Tenth International Conference on Software Defined Systems (SDS)*, 81–88. https://doi.org/10.1109/SDS59856.2023.10329218

IBM Cloud Education. (2020, August 19). *Application Programming (API)*. https://www.ibm.com/cloud/learn/api

Kabir, R., Hasan, A. S. M. T., Islam, Md. R., & Watanobe, Y. (2021). A Blockchain-based Approach to Secure Cloud Connected IoT Devices. *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, 366–370. https://doi.org/10.1109/ICICT4SD50815.2021.9397000

Kale, D., & Rathod, S. (2023). Agriculture Food Supply Chain Management System based on BlockChain and IOT. *International Journal of Advanced Research in Science, Communication and Technology*. https://api.semanticscholar.org/CorpusID:256409026

Kasanen, E., Lukka, K., & Siitonen, A. (1993). The constructive approach in management accounting research. *Journal of Management Accounting Research*, *5*, 243–264.

Kumar R, N. (2021). Comparative Study of Proof of Work (PoW) and Delegated Proof of Stake (DPoS) Blockchain Consensus Algorithm. *International Journal for Research in Applied Science and Engineering Technology*. https://api.semanticscholar.org/CorpusID:237800914

Laurence, T. (2017). *Blockchain For Dummies*. Hoboken, NJ : John Wiley & Sons.

Lin, S. (2023). Proof of Work vs. Proof of Stake in Cryptocurrency. *Highlights in Science, Engineering and Technology*. https://api.semanticscholar.org/CorpusID:258023577

Liu, J., Peng, S., Long, C., Wei, L., Yunhao, L., & Tian, Z. (2020). *Blockchain for Data Science*. 24–28. https://doi.org/10.1145/3390566.3391681

Lukka, K. (2001). Konstruktiivinen tutkimusote. In *Metodix*.

Mettler, M. (2016). Blockchain technology in healthcare: The revolution starts here. *2016 IEEE 18th International Conference on E-Health Networking, Applications and Services (Healthcom)*, 1–3. https://doi.org/10.1109/HealthCom.2016.7749510

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., & Qijun, C. (2017). A review on consensus algorithm of blockchain. *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2567–2572. https://doi.org/10.1109/SMC.2017.8123011

Mitani, T., & Otsuka, A. (2020). Traceability in Permissioned Blockchain. *IEEE Access*, *8*, 21573–21588. https://doi.org/10.1109/ACCESS.2020.2969454

Mohanta, B. K., Panda, S. S., & Jena, D. (2018). An Overview of Smart Contract and Use Cases in Blockchain Technology. *2018 9th International Conference on*

Computing, Communication and Networking Technologies (ICCCNT), 1–4. https://doi.org/10.1109/ICCCNT.2018.8494045

Nakamoto, S. (2008). *Bitcoin: A Peer-To-Peer Electronic Cash System*. https://bitcoin.org/bitcoin.pdf

Nijsse, J., & Litchfield, A. (2020). A Taxonomy of Blockchain Consensus Methods. *Cryptography*, *4*(4). https://doi.org/10.3390/cryptography4040032

Oyegoke, A. S. (2011). The constructive research approach in project management research. *International Journal of Managing Projects in Business*, *4*, 573–595. https://doi.org/10.1108/17538371111164029

Panicker, S., Patil, V. K., & Kulkarni, D. D. (2016). *An Overview of Blockchain Architecture and it's Applications*. https://doi.org/10.15680/IJIRSET.2016.0511100

Park, S., Im, S., Seol, Y., & Paek, J. (2019). Nodes in the Bitcoin Network: Comparative Measurement Study and Survey. *IEEE Access*, *7*, 57009–57022. https://doi.org/10.1109/ACCESS.2019.2914098

Peng, C., Xu, H., & Li, P. (2022). Redactable Blockchain Using Lattice-based Chameleon Hash Function. *2022 International Conference on Blockchain Technology and Information Security (ICBCTIS)*, 94–98. https://doi.org/10.1109/ICBCTIS55569.2022.00032

Rathod, N., & Dilip, M. (2018). Security threats on Blockchain and its countermeasures. *International Research Journal of Engineering and Technology*, *5*(11), 1636–1642.

Saqib, N. A., & AL-Talla, S. T. (2023). Scaling Up Security and Efficiency in Financial Transactions and Blockchain Systems. *J. Sens. Actuator Networks*, *12*, 31.

Sheldon, F. T., Jerath, K., & Pilskalns, O. (2002). Case study: B2B e-commerce system specification and implementation employing use-case diagrams, digital signatures and XML. *Fourth International Symposium on Multimedia Software Engineering, 2002. Proceedings.*, 106–113. https://doi.org/10.1109/MMSE.2002.1181602

Spadini, D. (2021). *Supporting Quality In Test Code For Higher Quality Software Systems*. https://api.semanticscholar.org/CorpusID:233313090}

Sriman, B., Ganesh Kumar, S., & Shamili, P. (2021). Blockchain Technology: Consensus Protocol Proof of Work and Proof of Stake. In S. S. Dash, S. Das, & B. K. Panigrahi (Eds.), *Intelligent Computing and Applications* (pp. 395–406). Springer Singapore.

Suman, R., & Sahibuddin, S. (2019). User Acceptance Testing in Mobile Health Applications: An overview and the Challenges. *Proceedings of the 2nd International Conference on Information Science and Systems*, 145–149. https://doi.org/10.1145/3322645.3322670

Szefer, J., & Biedermann, S. (2014). Towards fast hardware memory integrity checking with skewed Merkle trees. *Proceedings of the Third Workshop on Hardware and Architectural Support for Security and Privacy*. https://doi.org/10.1145/2611765.2611774

Tomović, D., Ognjanović, I., & Šendelj, R. (2015). Security challenges of integration of hash functions into cloud systems. *2015 4th Mediterranean Conference on Embedded Computing (MECO)*, 110–114. https://doi.org/10.1109/MECO.2015.7181879

Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., & Wang, F.-Y. (2019). Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*(11), 2266–2277. https://doi.org/10.1109/TSMC.2019.2895123

Weikle, D. A. B., Lam, M. O., & Kirkpatrick, M. S. (2019). Automating Systems Course Unit and Integration Testing: Experience Report. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 565–570. https://doi.org/10.1145/3287324.3287502

Xie, M., Liao, Z., & Huang, L. (2020). Data Security Based on Blockchain Digital Currency. *2020 3rd International Conference on Smart BlockChain (SmartBlock)*, 5–10. https://doi.org/10.1109/SmartBlock52591.2020.00009

Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2018). Blockchain Technology Overview. *ArXiv*, *abs/1906.11078*. https://doi.org/10.6028/NIST.IR.8202

Zhang, R., Xue, R., & Liu, L. (2019). Security and Privacy on Blockchain. *ACM Comput. Surv.*, *52*(3). https://doi.org/10.1145/3316481

Zhang, X., Xue, M., & Miao, X. (2022). A Consensus Algorithm Based on Risk Assessment Model for Permissioned Blockchain.

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). *An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends*. 557–564. https://doi.org/10.1109/BigDataCongress.2017.85

Zyskind, G., Nathan, O., & Pentland, A. 'Sandy'. (2015, May). *Decentralizing Privacy: Using Blockchain to Protect Personal Data*. 2015 IEEE Security and Privacy Workshops, San Jose, CA, USA. https://doi.org/10.1109/SPW.2015.27