University of Bergen
Department of informatics

An Overview of Blockchain-based Identity Management Systems

Author: Vegard Kjørberg
Supervisors: Chunlei Li (UiB), Guang Yang (HVL)

UNIVERSITETET I BERGEN
Det matematisk-naturvitenskapelige fakultet

November, 2022

Abstract

Modern systems are required to be developed very rapidly and have full functionality within a year or less. One major objective of identity management systems (IMSs) is to establish trust among entities that initially start with distrust. The existing solutions have however been rather centralized whereas the identity provider ends up performing too many roles. On one hand, centrally storing sensitive information, authentication, and authorization increases the risk of becoming central data silos, which in turn reduces the user's control over their own personal information. On the other hand, an identity management cannot be weakly decentralized where the user does not have full control, privacy and legal compliance.

As identity management systems have evolved over the years, they come with certain flaws and security concerns. While IMSs have problems with legal compliance, security and anonymity, the blockchain technology comes with these points as its main strengths. Blockchain can help adhere to compliance (GDPR); organizations can use it in order to generate trust amongst entities more efficiently and securely, and banks can also use blockchain in order to cut costs in Know-Your-Customer (KYC) processes, which in turn helps combat money-laundering.

This thesis studies several existing solutions of blockchain-based identity management systems. More concretely, I will investigate how they hold up in order to become a fully functional solution that can be used by everyone and where users feels like they have control over their information. The thesis reviews the evolvement, authentication methods and limitations of conventional identity management systems, and looks into several schemes of the emerging blockchain-based identity management systems. Finally I propose some real-world use cases based on two selected schemes and discuss the advantage and potential of the solutions.

Preface

This thesis is a conclusion of my two-year master's degree in Secure and Reliable Communications at the University of Bergen (UiB). Before my master´s degree, I also finished a bachelor's degree in Computer Security at UiB. A lot of the knowledge I used to write this thesis was learned through the courses I have taken, and some has been learned out of my own interest and while working on the thesis. I started my master's degree in january 2021 where I first had to take two semesters with only subjects. Early 2021 I also decided what kind of topic I would like to write my thesis about and I also had to choose a supervisor. The topic blockchain was presented and with Chunlei Li as supervisor I was immediately interested as I had both an interest in blockchain and I have had Chunlei as an instructor in different subjects during my bachelors. I first started on the thesis in early 2022 with some planning and how I thought I should structure the thesis. In the end would like to thank my supervisors Chunlei Li at UiB and Guang Yang at HVL for their excellent help in supervising during this process. I would also like to thank my cohabitant/girlfriend Clara Marcano for being great emotional and moral support while also pushing me to do better in addition to helping me go over grammar. Lastly, I would like to thank my parents Elly and Harald Kjørberg for moral and financial support during times of increased prices on everything.

Vegard Kjørberg

Contents

1 Introduction **1**

2 Identity Management Systems **5**

2.1 What is an Identity Management System? 5

2.2 Evolution of Identity Management Systems 6

2.3 Types of Digital Authentication 10

2.4 Challenges with Identity Management Systems 11

3 Preliminaries for Blockchain Technology **13**

3.1 Building Components 13

3.1.1 Peer-To-Peer Networks 13

3.1.2 Cryptography 14

3.1.3 Encryption and Decryption 14

3.1.4 Cryptographic Hash 14

3.1.5 Hash Chain 15

3.1.6 Digital Signatures and Timestamps 15

3.1.7 Merkle Tree 16

3.2 Characteristics 17

3.2.1 Decentralization 17

3.2.2 Transparency 18

3.2.3 Autonomy 19

3.2.4 Security 19

3.2.5 Traceability 20

3.2.6 Anonymity 20

3.2.7 Integrity 21

3.3 Challenges and Problems 21

3.3.1 Scalability . . . 21
3.3.2 Performance . . . 22
3.3.3 Cost of Decentralization . . . 22
3.3.4 Energy Inefficient . . . 22
3.3.5 Attacks . . . 23
3.3.6 Evolution . . . 23

4 Blockchain Technology and IMS **25**
4.1 History of Blockchain . . . 25
4.2 Blockchain Architecture . . . 27
4.3 Consensus . . . 30
4.4 Types of consensus . . . 30
4.4.1 Proof of Work (PoW) . . . 30
4.4.2 Proof of Reputation . . . 31
4.4.3 Proof of Stake (PoS) . . . 31
4.4.4 Delegated Proof of Stake (DPoS) . . . 31
4.4.5 Proof of Activity (PoA) . . . 32
4.4.6 Proof of Capacity (PoC) . . . 32
4.4.7 Proof of Elapsed Time (PoET) . . . 33
4.4.8 Proof of History (PoH) . . . 33
4.4.9 Proof of Importance (PoI) . . . 33
4.4.10 Results of consensus . . . 34
4.5 Smart contracts . . . 34
4.6 Why blockchain matches IMS so well . . . 35

5 Selected Use Cases of Blockchain-based IMS **37**
5.1 Smart Contract-based PKI by Mustafa Al-Bassam . . . 37
5.2 Sovrin Network . . . 41
5.3 uPort . . . 47
5.4 ShoCard . . . 50
5.5 EverID . . . 53
5.6 LifeID . . . 54
5.7 SelfKey . . . 54
5.8 Sora . . . 55
5.9 Civic . . . 55

5.10 Blockstack 57
5.11 Evernym 58
5.12 Discussion on selected solutions 59
5.13 Comparison Tables 64
5.14 Use Cases 66

6 Conclusion **73**

Bibliography **75**

Chapter 1

Introduction

A major issue that has started a lot of controversies in the cyber security landscape is the secure identity management. Traditionally, Identity management solutions have been delegated by trusted centralized organizations [1]. To name a few big ones, we have the likes of Facebook, Google, Apple, and Microsoft. These, among others, have been tasked with securely storing private data about its users and providing them with identity tokens, such as certificates, login credentials and a lot more [1]. For example, a lot of websites or games ask you to create an account on said website or game, or they ask you to log in with your Facebook account, Google account, Apple account, etc. With identity tokens created by this approach it has resulted in a lot of problems and there are four major ones: Individual user problems, information sharing problems, governmental information problems and at last privacy problems [1].

With major corporations sitting on such a large amount of information about people, we can have scandals similar to the Facebook-Cambridge Analytica data scandal. Personal data of millions of Facebook users, without their consent, was obtained by the British Consulting firm Cambridge Analytica for use in political advertising [2]. The data was collected through an app called **"This is Your Digital Life"**. The app managed to gather the data of up to 87 million Facebook profiles [3] and Cambridge Analytica used it to provide assistance to the 2016 presidential campaigns of Ted Cruz and Donald Trump [4]. Information about the data misuse was disclosed in 2018. Facebook apologized for their role in the data harvesting, and Mark Zuckerberg testified in front of the Congress. Facebook were to be fined 5 billion dollars by the Federal Trade Commission due to its privacy violations [5].

The Scandal sparked a massive increase of public concern in privacy and social media's influence on politics. We have even recently seen how social media can be used for politics, as Donald Trump has been using both Twitter and Facebook to spark tension and incitement of violence. All this to try to win the 2020 presidential election, which he said was 'stolen' due to voter fraud. This led to both Twitter and Facebook to make the decision of permanently ban him from their platforms as he spread a lot of misinformation.

To go from this kind of centralized Identity Management System (IMS) where all the big corporations have the whole power in controlling user data, researchers, and cryptographers have started working on ideas involving blockchain to decentralize them. Identity Management Systems usually come with pros and cons like legal compliance, security, and privacy (anonymity). The conventional digital applications come with high legal compliance due to the control of the service providers, but these come at the cost of lower user privacy and security. On the other hand, blockchain comes with high anonymity, privacy and security, but then again at the cost of legal compliance. We can see from this that the conventional IMSs could very well benefit from extending their use to the blockchain. Where blockchain provides high security, privacy and anonymity, in some cases where strong authentication is required it would also be beneficial to extend blockchain to the real world [1].

Distributed Ledger Technology (**DLT**) or more commonly called **Blockchain Technology**, refers to the technology behind decentralized databases by providing control over the evolution of data between entities through a peer-to-peer network. The use of consensus algorithms that ensure replication across the nodes of the network [6]. Blockchain was conceptualized by a person or group of people known as Satoshi Nakamoto in 2008 and even to this date nobody knows who this person or group even is. Nakamoto improved the design in an important way by using a Hashcash-like method to timestamp blocks. Doing this made it, so they did not require them to be signed by a trusted party and introducing a difficult parameter to stabilize the rate at which blocks are added to the chain [7].

The timestamp proves that the transaction data existed when the block was first published in order to get into its hash. Each block contains information about the previous block and each additional block reinforcing those that came before it. This makes blockchains resistant to modification of their data because once it is recorded, the data in any given block cannot be altered without altering all subsequent blocks [7]. In 2009 Nakamoto implemented the design as a core component of the crypto-currency called Bitcoin, where it serves as the public ledger for all transactions on the network [8].

By storing data across its peer-to-peer network, blockchain can eliminate a lot of risks that come with data being held centrally [8]. The decentralized blockchain can use ad hoc message passing and distributed networking. A big risk of a lack of decentralization is a so-called "51% attack" where a central entity can gain control of more than half of the network and can manipulate that specific blockchain-record and allow double spending [9]. Peer-to-peer networks lack the centralized points of vulnerability that computer hackers can exploit [10].

In this thesis, I will explore the differences of decentralized (blockchain) and centralized (conventional) IMSs. I will investigate and propose different solutions on how blockchain can be used to decentralize the systems in order to have better trade-off between security, privacy and legal compliance. Furthermore, I will go in depth on how blockchain is built up and how it works. I will also look into how different blockchain consensus algorithms are used, mostly **PoS**(Proof of Stake) and **PoW**(Proof of Work). This should eventually give more power back to the public and make them feel more secure that their private information does not get leaked or stolen.

Chapter 2

Identity Management Systems

In this chapter we will cover what an identity management system (IMS) is, the evolution of identity management systems, different types of authentication, and the challenges regarding IMSs. The authors in [1, 6, 11, 12, 13, 14] have all written about certain areas of these points in IMSs, we will provide a review of identity management systems by piecing their differences together.

2.1 What is an Identity Management System?

Identity Management is also known as **"identity and access management"** or IMS. It comprises the processes and technologies inside an organization. That identifies, authenticates and authorizes a person's access to certain services or systems that the organizations provide. As an example, we can use a campus setting where multiple systems such as library databases and grid computing applications require users to authenticate themselves. The person can either use a combination of a username and password to access a system, or they can use a card + pin code to enter certain areas within campus. Now, an authorization process determines which systems the authenticated user is permitted access to or if the person is allowed access into the area.

IMS is important for ensuring that the right person can access the data they need and have proper security clearances for what they need to accomplish their task. The cyber security industry became aware that user login credentials were a significant factor in data breaches, the importance of IMS solutions related to them grew. Identity management systems use standards and protocols to protect individually identifiable information and ensure that the user's credentials remain secure and uncompromised.

2.2 Evolution of Identity Management Systems

Before the World Wide Web was created, certain US universities and government institutions were connected through the first computer-based network called ARPANET. ARPANET was established in 1969, and it was meant to connect universities that worked for the US Department of Defense in order to make it easier for them to exchange information over long distances. Thus, ARPANET is considered the predecessor of the internet. The original protocols were hard to scale, so the TCP and IP protocol was implemented, and they are still essential to the World Wide Web of today. One major purpose of the IP protocol was to assign addresses to the devices that were connected to the internet. These addresses are what we know as IP addresses.

Originally, the list of devices and their IP addresses was managed over a text file called "HOSTS.TXT" by the Stanford Research Institute. This solution certainly did not work on a large scale, therefore, it got replaced by a new concept named The Domain Name System (DNS) which was proposed in the early 80s. DNS is managed by the Internet Corporation for Assigned Names and Numbers (ICANN) as a central institution even today. ICANN coordinates and manages the DNS and the IP addresses. The next big invention in the history of the internet was the invention of HyperText Transfer Protocol (HTTP) and the HyperText Markup Language (HTML) by Sir Tim Berners-Lee in 1989.

The ability to create websites in order to present content and interact with others more intuitively helped make computers and the internet easier to use. The increased user-friendly design made the internet more appealing to the public. A downside to all of this was that the website users could not be sure that the owner of the website was the owner and not a fraud. In order to solve this problem, a Public Key Infrastructure (PKI), Certifying Authorities (CA), and the HyperText Transfer Protocol Secure (HTTPS) were created. CAs are still, to this day, in charge of verifying websites in the same manner as they did back then.

PKI describes an infrastructure where the entities have a public key as an identifier to share amongst the internet and a corresponding private key in order to encrypt and decrypt messages. These messages are encrypted by using the public key, and only the corresponding private key is able to decrypt the message. The CAs manages the public keys, assign identities, and approves that a website is managed by the claimed identity. On the other hand, users have to trust that the CAs published data is correct and also the correct certifying of domains.

A central institution provides a framework for identity for all of its users, where a user ID and password can be created. The combination is used as an authentication to access different services that trust the identity provider. The very first institution who tried to enable a trusted federated identity was Microsoft back in 1999 with their "Microsoft Passport". The model was not successful because the system they had was inconsistent, and it lacked user experience. Microsoft acted as a central institution, this gave them power to determine which online enterprises could make use of this identity.

There was a demand for more control over someone's own identity even before the social networks became very popular. To cater to this, the Internet Identity Workshop (IIW) was created in 2005. They focused on making the user a central point in the identity process. This is what is called a user-centric identity. These user-centric identities should not be limited to access one service, they should be interoperably usable. OpenID, which is a decentralized authentication system for web applications, were created in 2005 based on the work of IIW. When a user had created a username and a password at an OpenID provider, it could then be used with partnered services, often called a relying party. If a website wanted to be able to use OpenID, they had to have their very own protocol in order to gain access.

OAuth was implemented on top of OpenID back in 2009 in order to solve the problem of authorization because of lack of standardized API access control for integrating third parties. OAuth exchanged the identity information to third parties, while data such as authorization information was not sent to them. This functionality became very popular with companies like Google and Facebook, who integrated the OpenID program. This resulted in a very widespread adoption of the format. OpenID Connect was implemented in 2010 and ensured a cryptographic authentication process on top of OAuth. OAuth enabled confirmation of the authentication based on authorizations. This is still a standard concept in a lot of online identity processes today, for example, using a Google or Facebook account for authentication. It is a very widely used process because the users only have to remember a single username and password instead of having multiple, in addition to avoiding repeatable online registration processes.

Because online identity providers like Facebook and Google have more than a billion users, it makes them centralized institutions and centralized identity providers. Due to their big amount of users and already existing personal information that is provided in the social networks, they are very privileged compared to other identity providers. As users can get access to websites with these identity accounts, the website providers get access to related personal information such as likes, social relationships, and preferences. This let them target their customers in a very efficient and personalized way. For example, if you are googling a product and you later on sign into some website with a Google account, this website could potentially start showing you ads for this product.

Even though IDPs (Identity Providers), websites, and users can benefit largely from this type of identity, it comes with very big issues in the form of trust and privacy. The identity is completely controlled by the identity providers, and, thus, it has the risk of the identity being banned or deleted, which could then result in the loss of said identity. In addition to that, the personal information of these users could be sold and used in a non-transparent way. This makes it so that third parties can get access to sensitive information about a user without said user's consent.

When a lot of users started to demand autonomy of their identities and control of how their identities information was shared and used in the mid 2010s, the term Self-Sovereign identity became used. The rise of blockchain technology now makes it possible to create a digital identity in a fully decentralized way without needing any central institutions. The use of cryptography plays a central role in this concept in order to enable pseudonymity and to secure connections between organizations and the users as privacy became an increasingly popular topic among users on the internet.

To establish self-sovereignty on the internet is a big task. It has to be independent of central institutions, and credentials have to be requested and securely stored. Trust in the system has to be established and many new processes and formats have to be standard in order to make Self-sovereign Identity an interoperable concept for online identities. Users will have their credentials stored in a digital wallet, just as in the real world, where they have their ID and such in their physical wallet. This make it so that they can identify themselves in a privacy-preserving manner to those who require them to identity themselves. Some examples or use cases for this is described later in the paper.

2.3 Types of Digital Authentication

With IMS, corporations and organizations can implement a variety of different digital authentication methods to prove digital identity and authorize access to their resources.

Unique passwords. The most common type of digital authentication is the unique password. Some organizations require longer and more complex passwords that require different combinations of letters, symbols, and numbers to make them more secure.

Pre-shared key(PSK). PSK is another type of digital authentication where the password is shared among different users who are authorized to access the same resources. This type of authentication is less secure than individual passwords, and a concern with PSK is that if you frequently change them it becomes inefficient and tedious for everyone.

Behavioral authentication. When dealing with highly sensitive information and systems, organizations can use behavioral authentication, which means they can analyze keystrokes or mouse-use characteristics. By applying artificial intelligence, which has become a trend in IMS systems, organizations can very quickly recognize if it is a person or a machine trying to access. A person will very often write in their credentials in sort of like a rhythm, so if suddenly the credentials are entered outside the normal rhythm, the systems can automatically be locked down.

Biometrics. Modern IMS systems can use biometrics for even more precise authentication. They can collect a variety of biometric characteristics like fingerprints, faces, irises, etc. Biometric and behavior-based analytics have been found to be more effective than passwords. However, when collecting these characteristics, companies has to consider ethics in different areas such as.

- Data security (accessing, using and storing the biometric data).
- Transparency (implementing easy to understand disclosures).
- Optionality (providing customers a choice to opt in or out).
- Biometric data privacy (understanding what constitutes private data and having rules around sharing with partners).

If a company's biometric data is hacked, then recovery can be very difficult. Users cannot swap out facial recognition or fingerprints like they can with passwords and other non-biometric information. An addition to this is that implementing biometrical authentications can be very challenging to do at scale and with different software, hardware, and training costs.

2.4 Challenges with Identity Management Systems

Typically, IMS initiatives have required several years of implementations before becoming fully functional. In recent systems, long time on IMS deployments are unacceptable and so, the trend has shifted to more agile IMS technologies that deliver full functionality withing 12 or fewer months. These include the likes of Office 365, Multi-Factor Authentication (MFA), Infrastructure as a Service (IaaS) for application migration and Single Sign On (SSO) plus numerous more. There are also quite the number of decentralized Identity Management Systems such as Microsoft ID2020, IBM Identity Management with Secure key Technologies, Estonian Citizenship Identity and Identity Chain to name a few.

The main goal of Identity Management Systems are to establish trust among entities that at first have a mutual distrust. These solutions have, however, either been centralized which causes the identity provider to perform too many roles such as storage of sensitive information, authentication, and authorization. This increases the risk of central data silos and reduces the user's control and privacy over their data or too weakly decentralized which means it does not provide full user control, privacy and or legal compliance.

How can we achieve a secure and functional decentralized identity who can also adhere to compliance (GDPR, for example) if needed? Organizations who are looking to extend their services can use Distributed Ledger Technologies (blockchain) to generate trust more efficiently and securely among entities with no prior information about one another. Banks can use it for their blockchain solutions to cut costs in Know-Your-Customer (KYC) processes and thus fight money-laundering. Blockchain can be used to transition from Centralized Identity Management Systems to more Decentralized Identity Management Systems.

Chapter 3

Preliminaries for Blockchain Technology

Blockchain is a new technology with a variety of attractive characteristics. In recent years there have been extensive researches on blockchain [1, 6, 15, 16, 17, 18, 19, 20]. For example, the authors in [20] provided a good review of blockchain. In this chapter, we will go through important building components of blockchain and the attractive characteristics it brings.

3.1 Building Components

3.1.1 Peer-To-Peer Networks

A Peer-To-Peer network also called a peer-to-peer network is a distributed network for sharing of resources amongst participants. The participants can share their resources that can range from processing power, link capacity, printers, and storage capacity and much more. Each participant is known as a node (peer) in the network, and they take on the role of both a client and a server. If a peer A requests a service or contents from peer B then A will be a client and B will be a server. This can be done without any intermediate entities. Vice versa, peer A can act as a server if peer B requests content or service.

3.1.2 Cryptography

In modern security protocols, cryptography is used to ensure confidentiality, integrity, and authenticity in communications. A mathematical value called a "key" is a central role in cryptography, and there are two general categories of modern cryptography: Symmetric-key and public-key (a.k.a. asymmetric-key) cryptography. Symmetric key cryptography uses the same key for sender and receiver for the cryptographic operations. In public-key cryptography, however, both communicating parties have two different keys that are called the public key and the private key. They are used for different cryptographic operations in different ways.

3.1.3 Encryption and Decryption

Encryption is used for the provision of confidentiality of security services and is a process to encode plain text (readable text) into cipher text (non-readable text). Decryption is the reverse version of Encryption, where cipher text is converted into plain text. Both of these processes can be implemented by using either symmetric or asymmetric cryptography.

3.1.4 Cryptographic Hash

To protect the integrity of data, a one-way mathematical function called a Hash can be used. This works by calculating a fixed-sized unique value called a "hash value" for every variable input. What is good about this is that the hash function is one-way, which means that the original data cannot be calculated based on the unique output. This one-way characteristic is what gives it its security strength, which is used to protect the integrity of data.

3.1.5 Hash Chain

When applying the hash function on data is successful, a hash chain is generated. For example, a hash value h_1 is generated by applying a hash function $f(x)$ on data x. The h_1 is input to the other hash function $f(h_1)$ to calculate the second hash value h_2 in the chain and so forth. These calculated hash values $h_1, h_2, ..., h_n$ make a chain of hashes with the length of n. As these hash functions are irreversible, h_1 cannot be computed from h_2 and so on. Hash chains have several applications in the protection of data integrity and plays a key role in blockchain.

3.1.6 Digital Signatures and Timestamps

For proof of authorship and contents, digital signatures are used. They are usually applied by using public key cryptography, where a signer uses its own private key to sign on a document and the recipient can then verify this signature by using the signer's public key. Digital signatures are considered authentic, unforgeable, non-reusable and non-repudiated. Digital signatures cannot be shifted for any other document or content and one cannot deliberately claim the signature which only the original signer can do, but even the original signer cannot repudiate it.

Timestamps are where the time at which the event occurrence is recorded by a computer rather than by the time of the event itself. Often enough, it records the date and time of the day when the event occurred, and it is accurate enough to be within a fraction of a second. The timestamp's data is recorded consistently together with the actual data for an easy comparison of two different records in order to track progress over time.

3.1.7 Merkle Tree

Merkle trees are also called hash trees, they provide efficient and secure verification of the data by arranging the data and its corresponding hash values in a tree form. Every leaf node in the tree structure is labelled with the hash value of its child nodes. In figure 1 the nodes in the bottom called Data1, Data2, Data3, and Data4 represent the data that is processed by this application. They are each summarized by their hash value Has1 to Hash4 as a leaf or Merkle leaf. The Merkle tree builds sort of a hierarchy by combining the hashes together until only one is left. The nodes combining the Merkle leaves are called a Merkle branch. At the top, when there is only one left, it is called the Merkle root. In other Merkle trees, there could be several more branches and leaves. In a Merkle tree, a modified data node can be located with complexity $O(log_2(N))$, where N is the number of data nodes in the construction of the Merkle tree.

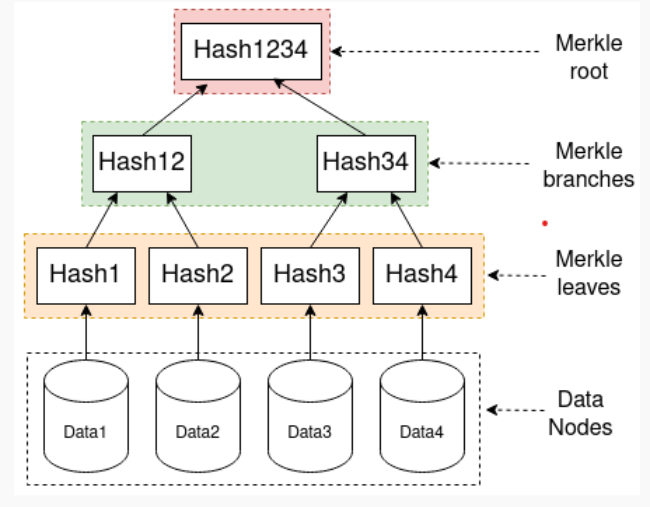

Figure 1. Example of a Merkle Tree

3.2 Characteristics

3.2.1 Decentralization

The most characteristic trait of blockchain is very likely to be decentralization. The blockchain ledger exists on multiple computers, which are often called nodes. The nodes form a blockchain network where several of them works in a peer-to-peer manner by validating access to information without the use of a centralized authority. The system uses a distributed system structure for recording, storing, updating, transmission, verification, maintenance and many other types of processes that are related to the information in the blockchain network.

This type of decentralization eliminates the need for central authorities and transfers the control to the individual user, thus making the system more fair and more difficult to tamper with. Information recording is performed and transactions are validated by using a set of rules and algorithms that are called consensus protocols among the blockchain nodes to make sure that the information is consistent and incorruptible. Consensus is achieved when enough devices have agreed about which of the new blocks will be appended to a blockchain. Later on, in chapter 4 "Blockchain Technology and IMS", we will take a look at different types of consensus protocols, and discuss their benefits and drawbacks.

Chain nodes can use a pure mathematical method involving asymmetric cryptography to establish trust among themselves without having a central authority or regulatory agency that can manipulate the data alone. Every distributed node in the network is relatively independent and have equal rights and obligations. It does not affect the entire network if there is a node-level corruption, thus it is guaranteed to have improved reliability and robustness of the blockchain system.

Many copies of distributed information on the blockchain can also prevent the risk of having the information be lost or destroyed due to the dependency on a centralized location. By removing a centralized body that is collecting, recording, maintaining and has unrestricted access to information. A lack of reliance on a centralized entity that executes and validates transactions can by a large margin reduce intermediary costs, and it can improve performance bottlenecks at the central servers.

3.2.2 Transparency

Transactions on the blockchain are completely transparent, as anyone can see all the details and history of the transactions. This is unique to blockchain, and it provides a very high level of accountability and integrity to the information, thus making sure that nothing is improperly altered, falsely added or removed. This type of transparency is achieved because a blockchain network have multiple validating peer nodes without a centralized authority. The holdings and transactions of each public address are accessible and open to be viewed by anyone. This results in traceable and transparent transaction records.

A key area where the transparency characteristic has found applicability is in healthcare and clinical trials. In healthcare, blockchain can be used by patients to easily view all of their claims, medical history, transactions and overdue payments. The data of clinical trials have usually been held from researchers, doctors and patients, thus resulting in lack of trust and credibility of the findings. Blockchain-based methods were suggested to trace the existence of documents that contain pre-specified end points in clinical trials. It has also been proposed to use smart contracts to act as a trusted administrator in order to address data manipulation issues that are common in clinical trials. Non-fraudulent public elections and increasing the voters' trust in electoral processes have also been proposed to benefit from the transparency characteristic.

3.2.3 Autonomy

Usually, all the transactions are based on trust that guarantee that all parties involved can depend on each other in order to fulfill their commitments. Trust is no longer any issue within a blockchain system, which means that the system can work in a peer-to-peer manner without any reliable third party to ensure trust. Blockchain uses cryptography in order to completely replace third parties as governor of trust. By using the privacy and unforgeability of asymmetric cryptography, the system protects messages and verifies the identity of the sender, thus ensuring secure and reliable transactions.

Participating nodes in the blockchain system uses a complex distributed consensus algorithm to unanimously and securely add or update data to the ledger, while solving the problem of ownership confirmation in transaction process as well as maintaining the system integrity. The transactions are accomplished without intervention of a third party to ensure trust because of the fail-safe protocols that provides the basis for the trust. By eliminating the third parties to ensure trust, it also results in the cost of transactions decreasing.

3.2.4 Security

By using asymmetrical cryptography that consists of a set of public keys that are only visible to the owner, make blockchain systems are very secure. The keys are used to ensure ownership of a transaction as well as the un-tamperability of the transaction. Blockchain's security is related to integrity, confidentiality, and authorization of the transactions. By requiring a peer-to-peer consensus mechanism, it eliminates a single point of failure for the data compared to if it was centrally stored and would be far more likely to be compromised.

3.2.5 Traceability

Traceability means that you can track the source, destination, and sequence of different updates the data goes through between the different nodes. It is needed for data integrity and trust in the information. In addition to this, data traceability also have many other benefits such as data governance, conformity with regulations, it helps to understand the impact of changes, and it also helps improve data quality. Blockchain supports this in a way by time stamping the information as it is added or updated. This type of technology is used to add a time dimension for each block of data. The hash values that are stored in each block then correctly identifies the current block and its parent block. Data traceability have a very large impact on financial transactions, clinical trials and supply chain management.

3.2.6 Anonymity

Anonymity in blockchain is defined as to be protected from unauthorized intrusion and or observation. This can be achieved by authenticating transactions while at the same time not revealing any personal information of either party involved in the transaction. The data is exchanged between nodes by using a defined algorithm for establishing trust and thus the information of the nodes doesn't need to be revealed or verified and as such the information transfer can be carried out anonymously.

Blockchain system users can interact with a generated blockchain address to keep their real identities hidden, but blockchain cannot guarantee perfect privacy because of its nature of distributed and public environment. This is why some researchers have started to use the term pseudonymity in order to define this characteristic. Where anyone is able to create a blockchain address and not have it possible to connect this address to a person without information from other sources.

3.2.7 Integrity

Blockchain system are very resistant to changes in data, as its data integrity ensures that the data in the system remain accurate and consistent all the time. This is achieved because of the decentralized and immutable shared ledgers in the blockchain network. That means that when a block of data is agreed upon to be added to a blockchain, the transaction record of the block cannot be edited or modified. The data is permanently preserved in the system, with several copies in many nodes across the blockchain network. This is effectively guaranteeing the reliability and integrity of the data.

3.3 Challenges and Problems

3.3.1 Scalability

One problem that blockchain has today is that it have issues with scalability. This happens because of limitations with low throughput, high transactional latency, and very high resource needs. Its storage space requirement increases as the amount of transactions increases. Large blockchains growing bigger and bigger with new data nodes can become unwieldy when it comes to loading, computing and synchronizing data, and it could create difficulties for the client to run the blockchain system. Things like On-chain scaling and off-chain scaling have been some of the suggested techniques to address the scalability issue, however, they are in very early stages. To address high computational resource and storage requirements, edge computing have been proposed to approach this. This will offload the blockchain and mining computation from the nodes with limited power.

3.3.2 Performance

The blockchain systems suffer from performance problems such as a throughput bottleneck, transactions latency, and storage. As an example, in current systems are executed serially by the miners and validators. This significantly limits the throughput. While Bitcoin transactions are usually verified in an hour, it is acceptable but far from good enough. A proposed solution is the Lightning Network that uses Hashed Time lock Contracts (HTLCs) with a bidirectional payment channel that allows secure payment routing across multiple peer-to-peer payment channels. The blockchain community needs to explore the use of today's concurrent multicore and cluster architecture in order to address these performance issues.

3.3.3 Cost of Decentralization

Decentralization is one of the most important things of blockchain, it is not, however, without costs. There are open issues of consensus algorithms balancing between security and resource efficiency regarding adaptively controlling the replication factor in shards. On the other hand, append-only chains with historical data such as spent transactions continues to grow in size and thus make it so that ordinary nodes will one day run out of storage. This can lead to the blockchain network being controlled by just a few, but powerful nodes. A possible solution has been to investigate pruning out of date blocks that need to be forgotten, but at the same time without compromising its immutability. Except for some experiments, the data pruning problem remain an open issue. The monitory cost of using a public blockchain is another aspect of cost.

3.3.4 Energy Inefficient

Multiple consensus algorithms in blockchain systems are very energy inefficient. For example, the proof of work (PoW) consensus uses almost 15.77 terawatt hour. Put into perspective, this is around $0,08\%$ of the world's electricity consumption alone. Most of this is spent on computing the irreversible SHA256 hashing function. The extreme resource requirements for the system to verify its transactions and inefficient use of energy resources for these financial activities could pose an enormous threat to the global climate because of all the greenhouse gas emissions.

3.3.5 Attacks

Even though blockchain is very secure, there are still a couple of vulnerabilities that exist. Some attacks are: 51% majority manipulation of the PoW consensus, consensus delay because of distributed denial of service, selfish mining, pollution log, blockchain forking, orphaned blocks, de-anonymization, block ingestion, double spending attacks and also liveness attacks.

3.3.6 Evolution

The blockchain 1.0 technology is a part of Bitcoin, and it is associated with an unknown company or person going by the name of "Satoshi Nakamoto". Bitcoin used blockchain 1.0 to solve the problems of double spending of digital cash and that of the processing of digital transactions without the need of any trusted third party.

Chapter 4

Blockchain Technology and IMS

With the introduction to critical components of blockchain in Chapter 3, this chapter will now introduce the blockchain in more depth, including the history, architecture, consensus mechanism, and smart contracts of the blockchain technology [1, 6, 15, 16, 17, 18, 19, 21]. In the end, we will explore what makes the blockchain technology a good match for current identity management systems.

4.1 History of Blockchain

The blockchain technology was first described in 1991 by Stuart Haber and W. Scott Stornetta. They wanted to introduce a computationally practical solution for time-stamping digital documents, so they could not be backdated or tampered with. They developed a system that used the concept of cryptographically secured chain of blocks to store the time-stamped documents.

In 1992 the Merkle Trees were incorporated into the design. This makes blockchain more efficient by allowing several documents to be made into one block. They are used to create a "secured" chain of blocks. It stores a series of data records, and each data record that is connected to the one before it. The newest record in the chain contains the history of the entire chain. This technology went unused and the patent lapsed in 2004.

In 2004 the computer scientist and cryptographic activist Hal Finney introduced a system called **Reusable Proof Of Work(RPoW)** as a prototype for digital cash. It was a significant early step in the history of cryptocurrencies. The system worked by receiving a non-exchangeable or a non-fungible Hash cash based proof of work token in return and created an RSA-signed token that could be further transferred from person to person.

RPoW solved the double-spending problem by keeping the ownership of tokens registered on a trusted server. This server was designed to allow its users throughout the world to verify its correctness and integrity in real time.

In 2008 Satoshi Nakamoto published a white paper which introduced the concepts behind bitcoin and blockchain. Nakamoto is thought to be a pseudonym used by the individual or group of individuals who proposed the technology. Blockchain infrastructure would support secure, peer-to-peer transactions without the need for trusted third parties such as banks or governments.

The bitcoin/blockchain architecture introduced in 2008 was built on concepts from the previous decades. Nakamoto's design introduced the concept of a "chain of blocks." This made it possible to add blocks without requiring them to be signed by a trusted third party. Nakamoto defined an electronic coin as a "chain of digital signatures," where each owner transfers the coin to the next owner. This is done by digitally signing a hash of the previous transaction and the public key of the next owned, and adding these to the end of the coin.

Nakamoto's white paper was just the beginning. In 2009, bitcoin went from concept to reality. January 3rd. 2009, Nakamoto mined the first bitcoin block and thus validating the concept. This block contained 50 bitcoins and was known as "Genesis block" or block 0. January 12, 2009. The first bitcoin transaction took place when Nakamoto sent Hal Finney 10 bitcoin in block 170. On October 31. 2009 the first bitcoin exchange "Bitcoin Market" was established and thus enabled peopled to exchange paper money for bitcoin.

Nakamoto set up the system, so there would never be made more than 21 million bitcoin. More than 18 million have already been mined and based on the current rate of mining it should reach the 21 million limit sometime around 2140. Meanwhile, the value continues to grow despite continuous fluctuations in price. In October 2009, a single bitcoin was worth less than 1 cent and today as of writing, each bitcoin is worth, 28897,80 USD. On May 22, 2010, bitcoin made history when Laszlo Hanyecz paid 10000 bitcoin for two Papa John's pizzas. The pizzas were valued around 25$, a trade that would now be worth almost 290 million.

4.2 Blockchain Architecture

Blockchain is a distributed ledger technology (DLT) and consists of several various technologies. Mathematical computations, cryptography, game theory, peer-to-peer systems and validation protocols. Since blockchain eliminates the presence of a central governing authority, all of its transactions needs to be protected and the data must be securely stored on a distributed ledger. It works on a pre-set protocol, with different computers across its network (or it can also be called a set of nodes) having to arrive at a "consensus" to validate the transactional data. Each node adds, scrutinizes and updates the entries as they come.

Blockchain has a layered architecture to enable this unique way of authenticating transactions. There are five layers and all of them have a distinct functionality.

1) The Hardware Infrastructure Layer (Physical Layer)

Blockchain has its data securely stored in a data server. Our machines will request access to this data from the server when we surf the internet or use a blockchain app. This framework is known as the client-server architecture. Blockchain have peer-to-peer (P2P) networks that allow the clients to connect to other peers to make sharing data easier and faster. It is essentially a massive network of devices who communicate with each other and requests data from one another. In the end, this is how a distributed ledger is created and each device that communicate with another device on the network is a node. Each of these nodes can randomly verify transactional data.

2) The Network Layer (Peer to Peer Layer)

Blockchains are a long chain of "blocks" (hence the name blockchain) containing transaction data. When nodes validates a certain number of transactions, the data is bundled into a "block". It is then added to the chain and linked with the previous block. The first block in the chain is called the "Genesis Block" (mined by Satoshi Nakamoto) and because it is the first block it does not need to be linked with any previous blocks. However, the next block is linked with the Genesis block and this process is repeated every time a new block is added. This is how the blockchain form and grow.

At this layer, a protocol for a peer-to-peer network is established for communication between peers. The objectives of this protocol are to add new peers and assign GUIDs, which is a 128-bit text string that represents an identification [22], remove peers and more. Peer-to-Peer networks have several algorithms which includes the likes of Chord, Pastry, etc.

Every transaction is digitally signed with the private key of the sender's wallet, and only the sender have access to this key. This ensures that the data cannot be accessed or tampered with by anyone else. This digital signature also protects the identity of the owner, which is again encrypted and thus ensures maximum security.

If a blockchain is open, any computer can join in at this P2P layer and perform different actions like (mining and transactions). This is called a permissionless blockchain, and two of the most known examples of this are Bitcoin and Ethereum. In some cases, it is desired that some special privileges are reserved for special nodes and not everyone can join in on the physical layer of the block. This type of blockchain is called a permissioned blockchain. Here we have an example called Hyperledger.

3) The Content Layer

This layer operates with the data structures that hold blockchain data on each node in the peer to peer network. With Bitcoin and Ethereum the data is stored in blocks where each block points to the previous block. The first block (the genesis block) points nowhere and is typically hard coded. This type of data structure is similar to a linked list. In other blockchains like IOTA, a Directed Acyclic Graph(DAG) is used, and it is commonly known as a Tangle. Whatever data structure that is used in the content layer, it acts as the ledger that is distributed among the nodes in the blockchain.

4) Consensus Layer

The consensus layer makes sure that every peer in the network agrees on the same content at any given time by using algorithms called consensus protocols. Some protocols include the likes of Proof of Work, Proof of Stake, etc.

5) The Primitives Layer

A blockchain can be viewed as a State transition system. In blockchain, there is a conceptual state transition function F, such that $F(S, T[n]) \rightarrow S^I$, where S is a valid old state and S^I is a valid new state. T is an array of n transactions to be added to the new block. This state transition function enforces the properties of the consensus layer. In Bitcoin, F ensures that each transaction T is unspent, has a signature that matches the public address of the UTXO (Unspent transaction output). F also ensure the transactions does not create value, thus the sum of all input UTXO is greater or equal to the sum of all output UTXO in the transaction. Only Coinbase transactions create value that come in the form of miner rewards. If the input UTXO is greater than the output, the difference will be considered the transaction fee.

Each transaction has a unique ID that is usually a cryptographic hash of the content of the transaction. This ID can be used to verify the validity of any transaction. Not every node has the luxury of maintaining a database of all transactions. Such a node is called a full node. The majority of nodes, which are known as light nodes, use a technique called a Merkle tree to validate transactions instead of maintaining a database of all transactions. Merkle trees perform consecutive hashes of the transactions (which are the leaves of the tree), iteratively pairing and rehashing until one hash is formed. This is known as the root hash. For a light node to verify a transaction, they need the transaction ID and the Merkle Proof of the transaction, which they can get from a full node.

Merkle trees are very efficient and secure depending on which hash function they use. In Bitcoin, a double SHA256 hash function is used to provide a higher collision resistance. After a successful attack (collisions that were found at 265 instead of the expected 280), there was fear among the Bitcoin designers. They believed SHA256 had the same weaknesses, since the design of SHA256 is also based on the same Merkle-Damgard construction as SHA1. Thus, two rounds of SHA256 is expected to provide a higher collision resistance. On the other hand, Ethereum uses a different hash function called Keccak256, which is based on sponge functions.

6) Advanced Protocols

At this layer, the primitives are used in innovative ways to create the value in use cases. Some use cases are:

- Smart contracts - accounts that are controlled by program code, such that each transaction to the Smart contract executes the code in the contract on all nodes.
- Atomic swap - Exchanges currency between the blockchains.
- Payment Channels - Creates an off chain to make transactions and execute settlements on the blockchain.

4.3 Consensus

Consensus mechanisms are used to verify transactions and maintain the security of the underlying blockchain. There are many types of consensus mechanisms with different benefits and drawbacks. The two most widely used consensus mechanisms are called Proof of Work (PoW) and Proof of Stake (PoS).

Consensus mechanisms form the backbone of blockchains, and they are what makes them secure. Each transaction in a blockchain is recorded as "block" of data, and this "block" needs to be independently verified by peer-to-peer computer networks before they can be added to the chain. This type of system help to secure the blockchain against fraudulent activity, and it addresses the problem of "double-spending".

Consensus is the process by which a group of peers or nodes on a network determine which blockchain transactions are valid and which are not. The consensus mechanisms are the methodologies used to achieve this agreement. They help to protect the network from malicious behavior and attacks.

4.4 Types of consensus

4.4.1 Proof of Work (PoW)

Proof of Work is essentially the pioneering consensus model for blockchain technology. The main purpose in PoW is for the nodes to "compete" for generating new blocks in the system

based on computing power. The miner is required to perform a computation and produce a value. The wining value is less than the predefined value that is set by the network. PoW has a possibility of forking, where two different nodes can produce the wining value. Many researchers have proposed many variations of PoW.

4.4.2 Proof of Reputation

This consensus mechanism increases the reputation of a node based on its participation, transactions, and assets. The node with the best reputation value generates a new block, which is then validated by voting in the blockchain system. Proof of Reputation also allow degradation in nodes' reputation in case of misconduct in the past, in addition to adding to the security of the blockchain.

4.4.3 Proof of Stake (PoS)

A huge advantage of the Proof-of-stake mechanism is that it does not require its nodes to have super expensive equipment in order to perform mining. In PoS the miners are required to pledge a "stake" of digital currency for a chance to be randomly chosen as a validator. The process is sort of like a lottery, in which the more coins you stake, the better your odds will be. Unlike PoW where miners get rewarded with block rewards, the people who contribute to PoS will simply earn a transaction fee. PoS is a more sustainable and environmental-friendly alternative to PoW, and it is more secure against a 51% attack. The system, however, favors entities with higher number of tokens. It has drawn criticisms for its potential to lead to centralization.

4.4.4 Delegated Proof of Stake (DPoS)

In Delegated Proof of Stake (DPoS) it is suggested to use voting from stakeholders in order to elect a witness node or also known as "block producer" to secure the network on their behalf. Only the top witnesses who have the most votes have the right to validate transactions. To do the voting, users can add their tokens to a staking pool, where the votes are then weighted

according to the size of the voter's stake. This makes it so the more the user has staked, the more voting power he will have. Those witnesses who end up successfully verifying transactions in a block will receive a reward that is usually shared with the people who voted for them. If the selected witness cannot produce a block, however, then this witness will not be allowed in future voting processes.

The top witnesses are always at risk of being replaced by people who are deemed more trustworthy and get more votes. They can be voted out if they fail to fulfil their responsibilities or if they try to validate any fraudulent transactions. This again helps to encourage witnesses to remain honest at all times and thus ensuring the integrity of the blockchain. Even though it is less used than PoS, DPoS is regarded by many to be more efficient, democratic and financially inclusive than PoS.

4.4.5 Proof of Activity (PoA)

Proof of Activity (PoA) is a hybrid of the PoW and PoS. In the PoA systems, the mining process begins like in PoW, where miners compete to solve mathematical problems by using large amounts of computing power. However, once the block is mined, the system swaps to resemble PoS, where the successfully generated block header gets broadcast to the PoA network. A group of validators are randomly selected to sign of the hash and thus validating the new block. As with PoS, the more crypto the validator holds, the higher is their chance of being selected. When every chosen validator has signed the block, it is then added to the blockchain network, and it is ready to record transactions. The block rewarded are then shared among the miner and the validators.

4.4.6 Proof of Capacity (PoC)

Proof of Capacity (PoC) bases its mining algorithm on the amount of space available on a miner's hard drive. The miners generate a list of all the possible hashes beforehand in a process that is called "plotting". These plots are then stored on a hard drive. The more storage a capacity a miner has, the more possible solutions. With more solutions, they have a higher chance of possessing the correct combination of hashes and thus winning the reward. Since it doesn't require expensive or specialized equipment, PoC opens up more opportunities

for average people to participate in the network. Thus, it is a less energy intensive and more decentralized alternative to some other consensus mechanisms. The problem as yet is that not many developers have chosen this system, and there are concerns about its susceptibility to malware attacks.

4.4.7 Proof of Elapsed Time (PoET)

PoET is used on permissioned blockchain networks and uses a lottery-based election model in order to randomly select a new leader for adding blocks in the blockchain. Trusted Execution Environment (TEE) is used so ensure a secure environment for this election process. There are three major steps in the process of electing a leader. The validator and the minder nodes run TEE by Intel SGX. Every node that validates requests a wait time, and the node with the shortest wait time wins the election in order to become a leader node. Because this relies on specialized hardware, it is the main drawback for using this consensus mechanism.

4.4.8 Proof of History (PoH)

Proof of History (PoH) requires nodes to provide a proof of history, and it creates a historical record in order to provide evidence that an event occurred at a specific time. This gives an alternative of trusting the timestamp on the transaction [20]. PoH is developed by Solana and the timestamping method is enabled by an SHA-256 sequential hashing verifiable delay function. It works by taking the output of a transaction and then use it as input for the next hash. This enables everyone to clearly see which event took place in a particular sequence. The verifiable delay functions can only be solved by a single CPU core. PoH severely reduces the amount of processing weight on the blockchain, and thus makes it faster and more energy efficient than many of the other consensus mechanisms. The downside is that since PoH is only employed by Solana, it has not yet been tested on a very large scale.

4.4.9 Proof of Importance (PoI)

Proof of Importance is used by NEM(XEM) which is a cryptocurrency, and they introduce the concept of harvesting which is pretty similar to mining. They use network theory in

order to define rating of each account based on vested and unvested coins. They rely on the number of days the coins are in the account of a node in the network in order to estimate "importance". It allows 10% of currently unvested amount vests every day. It is calculated based on the rank of the account within the network by considering the number of vested coins it holds.

4.4.10 Results of consensus

We have now seen many ways of achieving consensus in a blockchain network. Some are more secure, some are more environmental-friendly than others, some works better with scalability and some provide more fairness than others. All in all, they all have different strengths and weaknesses. Later in this paper, we will look into different solutions for using this blockchain technology together with identity management systems in order to create a more decentralized and secure identity management system.

4.5 Smart contracts

Simply put, smart contracts are programs that are stored on a blockchain that run when certain predetermined conditions are met. They are used to automate the execution of an agreement so that all participants can be certain of the outcome without any intermediary's involvement or time loss. They can automate workflow and trigger the next action when conditions are met.

Furthermore, they work by following simple "if/when and then" statements that are coded on a blockchain. The actions are executed when predetermined conditions have been met and verified. These actions could include releasing funds to the appropriate parties, registering a vehicle, sending notifications or issuing a ticket. The blockchain is updated when the transaction is completed. The transaction cannot be changed, and only parties who have been granted permission can see the results. Within a smart contract, there can be as many stipulations as needed to satisfy the participants that the task will be completed to their satisfaction. To establish the terms, the participants have to determine how the transactions and their data will be represented on the blockchain and then agree on the "if/when and

then" rules that govern the transaction. They need to explore all possible exceptions, and they have to define a framework for resolving disputes. After this, the smart contract can be programmed by a developer. Organizations that use blockchain for business have increasingly been providing templates, web interfaces and more tools to simplify their structuring of smart contracts.

There are several benefits of using smart contracts such as speed, efficiency, and accuracy because once a condition is met, the contract is executed immediately. Because they are digital and automated, there is no paperwork that needs to be processed, and no time will be spent on reconciling errors that result from manually filling in documents. There is better trust and transparency as there is no third party involved. Due to encrypted records of the transactions are shared across all participants, there is no need to question if the information has been altered for personal benefit or not. We have better security as blockchain transactions are encrypted and are hard to hack. Also, because each record is connected to the previous and subsequent records on a distributed ledger, the hackers would have to alter the entire chain to change a single record. Another big benefit is that smart contracts removes the need for intermediaries to handle the transaction. This cuts their time delays and also their fees.

4.6 Why blockchain matches IMS so well

Decentralized Identifiers (DID) are global, unique, and persistent. The identity owner entirely controls them. DIDs are independent of centralized registries, authorities, and identity providers. When an organization issues a verifiable credential, they attach their public DID to that credential. That public DID is also stored on the blockchain, an immutable record of data. If someone wants to verify the authenticity of the credential, they can then check the DID on the blockchain to see who issued it without having to contact the issuing party. The blockchain acts as a verifiable data registry that anyone can use to verify what organization a specific public DID belongs to. In identity management, a blockchain enables everyone in the network to have the same source for truth about which credential is valid and who attested to the validity of the data inside the credential. All this without revealing any of the actual data itself. This is what makes blockchain such a good addition to IMS.

When leveraging blockchain technology for identity management, it is important to note that there are three different actors: identity owners, issuers, and verifiers. The issuer is a trusted party who can issue a personal credential for an identity owner (the user). By issuing a credential, the issuer attests to the validity of the personal data in that credential. This will be the last name and the date of birth etc. of said person. The identity owner can store these credentials in their personal identity wallet and use them to later prove statements about his or her identity to a third party that is the verifier. Credentials are issued by second parties who attest to the validity of the data inside the credential. The usefulness and reliability of a credential all depend on the reputation or trustworthiness of the issuer.

With the infrastructure of blockchain, the verifying does not need to check the validity of the actual data in the provided proof, but they can use the blockchain to check the validity of the attestation and attesting party. From these, they can then determine to validate the proof. As an example, we have that when an identity owner presents a proof of their date of birth or last name etc. Rather than checking the truth of the date of birth itself, the verifying party can validate the signature who issued and attested to this credential and then decide if he trusts the assessment about the accuracy of the data.

The validation of a proof is based on the verifier's judgement of the reliability of the attestor. Leveraging blockchain technology establishes trust between the parties and guarantees that the authenticity of the data and attestations, without actually storing any personal data on the blockchain. This is very crucial, as a distributed ledger is immutable, which means that anything that is put on it can never be altered or deleted. Thus, personal data should never be put on the ledger.

Chapter 5

Selected Use Cases of Blockchain-based IMS

In this chapter, I will review eleven interesting blochain-based IMS schemes that have attracted significant interests in the community.

5.1 Smart Contract-based PKI by Mustafa Al-Bassam

In [23] Mustafa Al-Bassam has explained his proposal on a smart contract-based PKI which we will now summarize.

Mustafa Al-Bassam has written a paper on a Smart Contract-based PKI (Public Key Infrastructure) and Identity System, where he intends to use an Ethereum blockchain together with an identity system. Al-Bassam writes about how each Bitcoin transaction references other transactions (inputs) and creates outputs that are recorded on the Bitcoin blockchain. The Bitcoin in these transactions can be "spent" by other transactions in order to facilitate the creation of transactions. Bitcoin has a transaction scripting language that is used to specify "locking scripts" for specifying conditions that must be met in order to spend transaction outputs. Since the inception of Bitcoin came, other forms of blockchain-based systems have been made. This extends the scripting language beyond the purpose of a cash

system, to allow other types of applications to be expressed on the blockchain in the form of "smart contracts".

One of these systems for smart contracts is Ethereum. The white paper on Ethereum describes smart contracts as "complex applications involving digital assets being directly controlled by a piece of code implementing arbitrary rules". One potential application of smart contracts is the identity and reputation systems. An example of this, is that a smart contract can be created for mapping domain names to IP addresses to provide a decentralized domain name registration system. The smart contracts in Ethereum are written in a low level stack-based byte code language that is executed by an Ethereum virtual machine and is referred to as Ethereum virtual machine (EVM) code. These smart contracts can also be written in a high level language such as Serpent and then be compiled to EVM code.

Each smart contract have functions that have "gas" costs associated with them that depend on how many computational steps or storage space they require. These are paid for by using Ether, which is Ethereum's internal currency. Smart contracts can call functions in other smart contracts. Ethereum smart contracts can emit "events" that are an abstraction of the Ethereum logging and event-watching protocol. Events can have up to three indexes, and they can be watched and filtered efficiently by Ethereum clients. The primary proposition of SCPKI (Smart contract-based Public Key Infrastructure) is to write such a smart contract with the functionality of a public key infrastructure and identity management system. Where public keys and identity attributes are stored on the blockchain and can be managed by the smart contract.

An alternative to the centralized chain of trust certificate authority (CA) model of PKI is the web of trust. In web of trust, the concept is that over time you want to accumulate keys from other people that you want to designate as trusted introducers. Everyone else will choose their own trusted introducers, and everyone will gradually accumulate and distribute with their key a collection of certifying signatures from other people. It is expected that anyone receiving it will trust at least one or two of the signatures. By doing this, it creates a decentralized and fault-tolerant web of confidence for all public keys. In a web of trust model, there are no CAs. Instead, any user of the system can sign each other's public key, which means that there is no CA in the system that is "too big to fail" as public keys are by design intended to have multiple signatures. This indicates that if one signer is compromised and the signer's key is revoked, the impact on the trust network is limited.

Further on, systems like Ethereum are useful as a high integrity programmable database layer for decentralized applications. Due to gas costs, it is not economically practical to store large amounts of data on the Ethereum blockchain InterPlanetary File System (IPFS) has become a popular storage layer for decentralized applications. It is a peer to peer data distribution protocol where nodes in the IPFS network will form a distributed file system. The data in IPFS is addressed by its cryptographic hash, and so its links always stay the same regardless of which nodes serve the data. This will make it ideal for blockchain applications, as it makes it possible to address large amount of data from transactions in the blockchain by using permanent and immutable IPFS links.

The proposed SCPKI is a system hosted on the Ethereum blockchain and, it is controlled by a smart contract that allows entities to manage identities of itself and other entities. An identity is a set of attributes about an entity, such as cryptographic keys, names, or addresses. The design of SCPKI contains two primary components. The first component is the smart contract that dictates the protocol of the system and acts as an interface to the blockchain for the management of identities and attributes. The second is the client, which interacts with the smart contract and other systems such as IPFS to allow users to fully utilize the system by allowing them to search for and filter attributes.

The smart contract of SCPKI centers around the entity, which publishes a set of attributes, signatures, and revocations on the blockchain for its identity. Each entity is represented by an Ethereum address, which is controlled by a private key or smart contract the entity has control over. Publishing an attribute to an entity's identity binds the identity to the attribute. Attributes that represent cryptographic keys can also be reverse bound to the identity, thus creating a double binding.

The feature of double binding is realized by publishing a "binding proof" that consists of a cryptographic signature of the entity's Ethereum address. Using the cryptographic key represented by the attribute, proves that the owner of a private key is associated with an Ethereum address. This is useful for transferring trust relationships between different cryptographic keys. For example, if a user A trusts a user B's PGP (Pretty good Privacy) key and user B adds their PGP key to their SCPKI identity with a binding proof. User A can also trust user B's SCPKI identity without asking for more information.

Due to the expensive gas costs associated with Ethereum storage, SCPKI is designed to allow users to store large attribute data (such as PGP keys) off the blockchain on for example IPFS

to save costs. While at the same time maintaining the authenticity of the data by providing a cryptographic hash of the data with the attribute on the blockchain. Al-Bassam explains that two versions of the smart contract can exist. One full version that is designed in such a way where its information can be programmatically accessed by other smart contracts, and a lighter version where it can not. The trade-off is that the lighter version has lower gas costs because there is no need to store attribute data within the contract itself for retrieval by other contracts. Simply, it can emit events on the blockchain for clients to watch, but it is less extensible by other smart contract that interact with it.

Al-Bassam discusses some limitations and potential future work on his smart-contract based IMS. Al-Bassam explains that as the World Wide Web scales, rogue certificate attacks become more common and organizations demand more forgery-proof identity verification techniques and the need for transparent public key infrastructure systems will grow. The design of the system requires that all parties referenced by the system must already use the system. For example, if a university wants to issue a degree to a user in the system, then the user must first add a degree attribute to his or her identity before the university can sign it. This increases the adoption barrier for the system, but the upside to this is that the user has control over what attributes are or are not attached to their identity.

The system is only suitable for the publishing of attributes that the user wishes to make public. It is not suitable for the publishing of more private identity attributes such as personal address, as all attributes can be viewed by anyone in the system and there is no access control. Al-Bassam suggests that in future iterations of the system, Al-Bassam may address this by adding functionality to publish "zero-knowledge" or Zero-knowledge proofs ZKPs attributes for verification of privately shared data that the user distributes and has control over.

5.2 Sovrin Network

The authors in [24] have done a review on the Sovrin network, which we will now summarize.

The Sovrin Network is an open source framework for delivering a sovereign and decentralized digital identity to users, which is managed by the Sovrin Foundation. The Sovrin Foundation is a non-profit organization that provides the business, legal, and technical support for the Sovrin Network. The network is built on the Hyperledger Indy framework, which has a complete set of open source specifications, terminology, and design patterns that allow for the development of decentralized digital identity solutions. The Sovrin Network is a specific instantiation of Hyperledger Indy, which is a public permissioned distributed ledger.

In addition to the core blockchain Hyperledger Indy, some other Hyperledger libraries such as Aries and Ursa are used to provide different functionalities. Aries is used to provide verifiable digital credentials, and Ursa is used to provide a shared cryptographic library. Because it is a permissioned blockchain, only trusted institutions called Stewards can operate nodes while participating in the consensus process and abiding by the Sovrin Governance Framework (SGF). Using this network, users can securely publish their identity including transferring their credentials, sign transactions and control their keys and data in a secure peer-to-peer model. In the Sovrin Network all identity related operations are governed through the SGF which is developed by the Sovrin Governance Framework Working Group (SGFWG).

In the Sovrin Network, a digital identity is created by utilizing a DID for users, organizations and other resources. It permits Verifiable Credentials (VCs) associated with an identity to be privately issued, controlled, managed and shared using a security standard called Zero Knowledge Proof (ZKP). The ZKP is a cryptographic method for creating anonymous credentials to maintain user's identity as anonymous. Sovrin allows users to create several identities to maintain contextual separation for privacy purposes, where each identity has its own pair of private and public keys.

A user determines what type of attributes they want to associate with their identity. These identity owners can prove information about themselves to anyone through a secure peer-to-peer communication by using data that the other party can automatically cryptographically verify as being true. The identity is completely owned by the owner of that identity, and all identity related personal and confidential data is held by the owner in their digital wallet on the edge or the cloud. The identity is managed either by the user or by a user's appointed guardian service. The key distribution verification and recovery is based on the Decentralized Key Management System (DKMS) standard, which is an approach to cryptographic key management where there is no central authority.

The architecture of the Sovrin Network can be divided into four layers that explains its various components and functionalities.

The first layer is called the Ledger Layer, it runs on a blockchain and the underlying distributed ledger is Hyperledger Indy, which is open-source and is particularly designed to support identity related transactions. The Sovrin ledger is used to maintain the records of different types of identity-related transactions. It is publicly readable and writable so that anyone can access it without and intermediary. The ledger is permissioned, so all the ledger operations are performed in accordance with the Sovrin Governance Framework. It

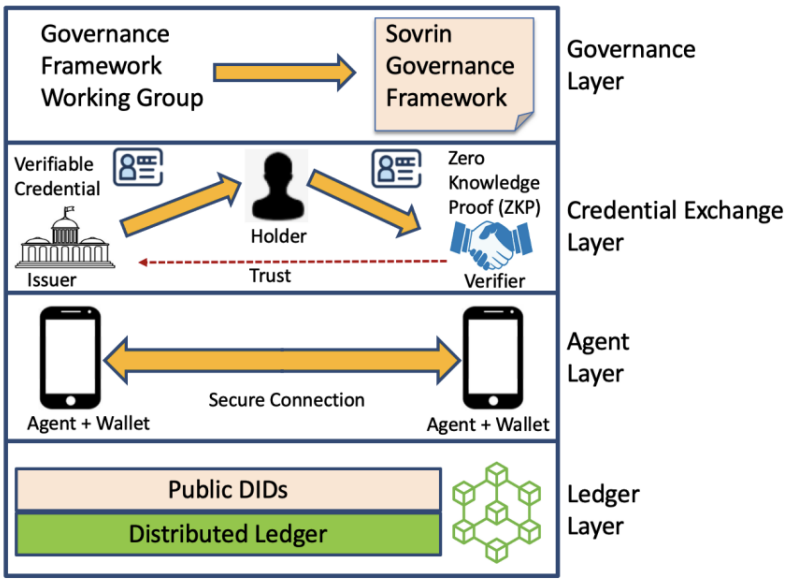

Figure 5.1: Architecture of the Sovrin Network [24]

is inexpensive to operate as transaction validation on the ledger is performed by known entities under proof-of-authority, making ledger access sufficiently inexpensive to support the mission of establishing Identity for All (I4A).

Sovrin does not require storage of any personal identifiable information in any form on the ledger. The ledger stored credential definitions, DIDs for issuers, schema definitions and revocation registries. These critical objects are stored on the ledger to avoid a single point of failure in the Sovrin Network and offer the blockchain security for digital identity. The objects are created by transaction authors and added to the ledger by Stewards that acts as transaction validators. The network is public and anyone can be a transaction author, but as it is permissioned, only organizations that function as Stewards can validate the transactions by enabling proof-of-authority consensus. Validation is done by using a modified Redundant Byzantine Fault Tolerance (RBFT) algorithm that is implemented as Indy Plenum.

The Sovrin ledger consists of several subledgers. The config ledger, node ledger, domain/main ledger and payment ledger. Only the domain ledger and payment ledger will accept publicly available transaction types. The Sovrin Network consists of server nodes that are located around the world, hosted and administered by a diverse group of trusted entities called Stewards. Each node contains a copy of the ledger, a record of publicly accessed information needed to verify the validity of credentials issued within the network. A node can either be a validator node or an observer node. The nodes can only act as validator or observer one at a time.

The second layer is called The Agent Layer, and it is concerned with the peer-to-peer connection between two entities or identity owners through the use of agents. An agent is a program that is required for an identity owner or any other participating entity to interact with each other in The Network. The agents work on the basis of a peer-to-peer model and share DID and other credentials with each other. They do not require access to the blockchain and communicates through signed and encrypted DIDcomm-messages. Agents utilize the DIDComm protocol as defined in an open and public process inside the Hyperledger Aries project.

Sovrin entities or identity owners normally have at least two agents. One that is on their device and one that is in the cloud. If an entity or identity owner have more than one device, then each device could have an agent installed on it. Each agent can access the wallet and perform cryptographic functions for that entity. The wallet securely keeps and retrieves

cryptographic key material, private keys, link secrets and other personal and confidential data related to any entity.

The Sovrin Network has two types of agents. The first one, is an edge agent that is hosted on the user's device (edge of the network) such as mobiles, tablets or laptops, and it always operates locally. The second agent is the cloud agent that can be hosted on the cloud directly by identity owners or hosted on behalf of them by third parties known as Agencies. The edge agent communicates with the cloud agent that runs perpetually and offers a store and forward service to route requests to and from the edge agent.

The Third layer is called the Credential Exchange Layer and it deals with an issuance, holding and, verification of credentials involving the three key roles of an issuer, holder, and verifier. It determines how the issuer issues credentials to the credential holder and how the credential verifier requests information from the credential holder, how the credential holder presents a proof of information from their credentials that the verifier can trust. The credentials are defined by their issuer using a credential definition which links the public DID of the issuer, the schema for the credential and a revocation registry for the credential. All of these, are stored on a distributed ledger that is used for decentralized discovery on the Sovrin Network.

Each credential holder or identity owner has a digital wallet that holds credentials containing certain information about that holder or identity owner. The digital wallet is an app that runs on a smartphone, tablet, desktop or another type of local device. The verifier verifies the holder's credentials by checking the issuer's public key from the Sovrin ledger and uses it to verify the issuer's digital signature on the credential. A credential holder can share information from multiple credentials with minimal disclosure about their identity using Zero Knowledge Proofs, aka ZKPs.

The ZKP is a cryptographic technique that allows a holder to share minimum information from their credentials. This is to prove a statement from the holder to the verifier without revealing their identity or any additional information that is not required by the verifier. A holder can hold the credential of another natural person (Guardian) or of a business (Credential Registry). Holders or identity owners can exchange credentials completely off-chain in peer-to-peer interactions through the Sovrin Network without interacting directly with the Sovrin ledger.

The last layer, called the Governance Layer The Sovrin Network, is decentralized in the sense that no central administrative authority or service provider completely controls the network. However, it is governed through the Sovrin Governance Framework (SGF) by the society consisting of trustworthy members. This SGF is developed and updated by the Sovrin Governance Framework Working Group (SGFWG). This governance framework is required to establish trust in the Sovrin Network as a global identity network Furthermore, the SGF is required to achieve the identity related governmental and jurisdictional requirements for data security, privacy, protection, and portability while at the same time preventing censorship and ensuring individual sovereignty over the sharing of identity data.

The Sovrin network works by having identity holders, credential issuers and verifiers access various identity related services using agents. These agents are generally a simple mobile app, and they are responsible for holding and processing credentials on the Sovrin Network. They can perform identity transactions on behalf of the identity owner and exchange information directly with other agents with secure encrypted peer-to-peer connections. The Edge agent accesses the edge wallet for the required confidential information and keys to process any identity related request. Here, the identity holder's actual proof of their credential is privately transmitted to a validator and only public identifiers of an issuer are anchored on the ledger.

The edge agent may not be a persistent agent and does not have service endpoint information to manage with other agents. Therefore, it communicates with the cloud agent, which is a persistent agent, and holds service endpoint information to manage other agents. The cloud agent accesses its own cloud wallet for required confidential information and keys to process the identity related request. It simply offers a store and forward service to route requests to and from the edge agent. The two cloud agents can securely communicate with each other, issuing and verifying distributed identity without accessing the distributed ledger. Any developer can develop an agent based on some specific instructions and code provided by Sovrin.

The Sovrin Network consists of server nodes that are located around the world and is hosted and administered by a diverse group of trusted entities that are called Stewards. Each node contains a copy of the ledger, a record of publicly accessed information needed to verify the validity of credentials issued within the network. The Stewards can write transactions to the appropriate ledger based on the guidelines given in the Sovrin Governance Framework. Stewards can cross-reference each transaction to assure consistency in whatever information

is written on the ledger and in what order by employing a combination of cryptography and a Redundant Byzantine Fault Tolerant consensus algorithm. The domain ledger is used to write identity related transactions and the payment ledger is used to write payment related transactions.

5.3 uPort

The authors in [25] have done a review on uPort which we will not summarize.

uPort is an open-source framework for delivering a decentralized identity for a self-sovereign identity. It is based on the public permissionless blockchain Ethereum, and it is utilizing its smart contracts. By employing this framework, users can securely publish their identity, transfer their credentials, sign transactions and control their keys and data. A uPort identity can be created for users, organizations and other resources. The identity is completely owned and governed by the owner of that identity and not by any third party In addition, all identity related personal and confidential data is held by the owner in their own digital wallet, thus securing it even more.

There are several components in the uPort identity management system. One of them being a component of a smart contract. The controller component is the overall control logic with the functionality of controlling the access to the proxy contract. Further, it allows the user to reclaim their identity if the user loses their mobile and private key. It maintains a list of recovery delegates, for example, certain family members or friends, who can help the user to regain their uPort identity. The Proxy Contract, is the permanent identifier of a user linked with the private key of the user, and it allows the user to replace their private key without affecting their permanent identity. Registry contract offer a cryptographic link between a uPort identifier and its data attributes or profile data stored off-blockchain (for example, InterPlanetary File System IPFS). IPFS is a peer-to-peer protocol for storing and retrieving data on a distributed file system The proxy contract can only update the Registry contract.

We also have some server components. One of them being Chasqui, which is the Message Server and manages all aspects of communications with any decentralized app and mobile app. Sensui is the Gas Fueling Server and avoids the requirement of a new Ethereum user to purchase Ether and paying fees to use the network. It pays the gas fees for the new user and thus, allowing them to create a new uPort account instantly. The Infura Ethereum RPC is an API that provides a standard RPC interface to allow uPort to communicate with the Ethereum network. And lastly, we have the Infura IPFS, which is an API that provides a standard interface that allows uPort to communicate with the IPFS network.

In the uPort identity management system, any user or app can interact with any application contract for identity related information. This process involves two main contracts. The first one, proxy contract which is a universal and permanent identifier for a user and a controller contract which is the main controlling program. The app interacts with the proxy contract through the controller contract, which then passes a request to the corresponding application, the smart contract. As a permanent identifier on the ledger, the proxy contract interacts with all application contracts and creates a layer between a user's own private key and application contracts.

uPort requires a standard RPC interface that is provided by Infura to communicate with the Ethereum network. A user can send a transaction without having any Ether in their sending account, by sending it to the Sensui server, which then supplies adequate Ether to pay the transaction fee. The uPort identity related attributes or profile data can be stored off the ledger on for example IPFS, Dropbox, OneDrive, or Google Drive. This can be accomplished by establishing a cryptographic link to an external data structure by using a registry contract. The registry contract can only be updated by the proxy contract, and this requires Infura IPFS interface to enable uPort to communicate with the IPFS network.

A DApp, called MyDApp, was developed in order to interact with the uPort mobile app. By utilizing this MyDApp there are three underlying IDM operations being performed. The first operation was login/connecting with the uPort mobile app, creating and issuing credentials and requesting and verifying credentials. In the first login operation, MyDApp generates a QR Code and the uPort mobile app scans this code to connect with MyDApp. Once MyDApp is connected with uPort mobile app, MyDApp generates a credential and sends it to uPort mobile app. When this credential is accepted, it is then stored in the digital wallet on the mobile device. This is one of the enhancements in SSI, where all credentials are stored in the digital wallet on the user's device.

The uPort identity management system is an open-source identity management system that offers decentralized self-sovereign identity, which is built on the public permissionless Ethereum blockchain. Currently, Ethereum's blockchain is using PoW Ethash as its consensus algorithm, and it is a not a very efficient one. They plan to switch to PoS later on, which is much more effective. uPort's identity management system offer several important features that is related to identity and its management such as sovereignty, access control, storage control, recovery management, security, privacy, safeguard, user-friendliness, and cost-effectiveness. It offers no support or limited support for several important commercial features such as availability, transparency, portability, interoperability, and governance in order to establish it as a commercial and successful SSI system.

As SSI is an emerging IDM model and uPort is an emerging identity management system. The implementation of the commercial and operational features require the development and adaptation of a set of common protocols and standards that is provided by organizations. Such as the Decentralized Identity Foundation (DIF), the World Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS). Currently, one of the biggest issues for uPort is the scalability feature and is being resolved by employing various design optimization techniques to fulfil the growing demands of sovereign identity globally.

5.4 ShoCard

The authors in [26] have done a review on ShoCard which we will now summarize.

ShoCard is a digital identity card on a mobile device. It binds a user identifier, existing trusted credential and other identity attributes together by using cryptographic hashes that are stored in Bitcoin transactions. ShoCard uses Bitcoin as a timestamping service to sign cryptographic hashes of a user's identity information, which will then be mined into the Bitcoin blockchain. They have a fixed central server as an essential part of its scheme, and this server intermediates the exchange of encrypted identity information between a user and a relying party. The scheme has three different phases: A bootstrapping phase, a certification phase and a validation phase.

The Bootstrapping phase occurs with the creation of a new ShoCard. The ShoCard mobile app creates an asymmetric key pair for the user and scans their identity credentials by using the camera of the device. Then, the scan and the corresponding data are encrypted and stored on the device. The digital signature of this data is then embedded into a Bitcoin transaction to later be used for data validation purposes. The resulting Bitcoin transaction number constitutes the ShoCardID of the user, and it is retained in the mobile app as a pointer to the ShoCard seal.

When a ShoCard is bootstrapped, the user can interact with service providers to gather additional attributes that rely on the seal in a certification process. For an identity provider to associate certificates to a ShoCardID, they must first verify that the user knows both the data that is hashed to create the seal and the cryptographic key that is used to create its signature. In person this can be achieved by having the user provide the original identity data from the seal from their mobile device, a digitally signed challenge and the original trusted credential.

In a Bitcoin transaction created by the provider, the certificate takes on the form of a signed hash of new attributes. The provider must share the Bitcoin transaction number together with a signed plaintext of the new attributes directly with the user. The user will later need to provide the attributes to relying parties and may not want to lose them if the mobile device is lost. A ShoCard server offer storage for symmetrically encrypted certifications that

are known as envelopes. ShoCard will never learn the encryption key, which enables the user to share certifications only with selected parties.

The validation phase happens when a relying party has to verify a certification to determine if a user is entitled to access a service. To validate the envelope, the user has to provide the relying party with the reference and encryption key for the envelope. After getting the envelope from the ShoCard servers, the relying party checks that the envelope signature was indeed produced with the same private key that signed the seal. They check that the certification signature was created by a trusted entity and that the plaintext certification corresponds to the one that was hashed and signed in the blockchain. Lastly, they check the textual details that were presented by the user in the pending transaction match those that are embedded in the seal.

The ShoCard central server functions as an intermediary to manage the distribution of encrypted certifications between ShoCard and its users. This makes it so ShoCard have less risk than if it was stored and distributed in plaintext identity data. The end user controls the secure storage of identity information and appropriate sharing of it with relying party ShoCard having an intermediary role creates uncertainty about the longevity of a ShoCardID existence. If ShoCard just stopped to exist, then its users would be unable to use the system with the certifications they had acquired. In a way, this makes ShoCard more centralized in practice than its open reliance on DLT might suggest. Every ShoCard identity has to be bootstrapped with an existing trusted credential, e.g., a passport or a driver's license. This approach requires the users to provide personal information from outside in order to create a ShoCard seal. This extra information may make ShoCard much less appealing for low-value online accounts.

As the user is in full control of initiating sharing activities and as ShoCard only stores encrypted data, we can have fairly good confidence that only justifiable parties are involved when it comes to the sharing transactions of identity data. The ShoCard server could however, be able to associate a single ShoCardID with requests made by the relying party because envelopes have to be retrieved from the server by the relying party. ShoCard only support unidirectional identifiers, and they do not support a public registry for ShoCardIDs. They might need omnidirectional identifiers in the future in order to realize their vision of an ecosystem with reusable certifications.

ShoCard support several identity providers through their certification functionality. These providers have to create bespoke integration with ShoCard's own web services in addition to Bitcoin, which can lead to a barrier to uptake. The positive perceptions of trustworthiness of its identity proofing and its users and the resulting value of a ShoCard can help in the decision to leverage ShoCard in future applications.

Scanning of QR codes is a dominant interaction of the ShoCard user experience because it is simple and consistent. While this is good for the user, it is unclear why they would adopt this new type of digital identity and how they would be educated about the implications of referencing identity data on a blockchain. In addition to this, the users are not supported with cryptographic key management and the overall deployability of ShoCard is debatable. As Bitcoin transactions take on average 10 minutes to be mined into the blockchain, and the fact that, it is recommended to wait for 6 more blocks to be mined before assuming the settlement of a transaction. This makes the waiting time of around an hour rather cumbersome, and it could make real-time settlements of certifications pose a challenge for the user experience and to those who wish to build applications that leverage ShoCard.

The authors in [27] have done a review on EverID, LifeID, SelfKey, and Sora which we will now summarize.

5.5 EverID

EverID is a transparent and independent identity network that runs on a permissioned Ethereum instance, which is hosted on a server operated by EverID. EverID supports various currencies in its payment system and is a large, scalable decentralized system for digital identity. This identity platform is made to secure user identity data and allow its users to access their data. EverID has indicated that the private information of a user cannot be transferred or used without the consent from the owner of the data. Supernodes store an EverID datagram for every entity in the system, and this can be accessed by a smart contract. To access information on an identity network, EverID uses biometric verification.

EverID is built with many components like Datagram, Decentralized Application (Dapp), Application Programming Interface (API), Core Smart Contracts, Ethereum Blockchain and Supernode. The datagram is storing specific files on user identity data, which resides in Dapp on the user's mobile device and in the Supernode. A datagram can contain any datatype with an unlimited size. As EverID uses biometric identifiers to create an identity in the network, the user's can only make a single identity account in the system Dapp is a code, and it allows the user to create a wallet on the Ethereum network with a biometric together with a password.

By using Dapp a user can register themselves directly as well as store and access their identity information. The EverID API is useful for integrating with other applications and services, and it is secured by using a hash-based message authentication code system. EverID is built on five primary smart contracts; Creation and management, validation, transaction, remote management, and operational EverID. Ethereum has a private blockchain records, and they store all transactions that are performed on any identity. They provide tamperproof series of transactions in the EverID identity framework, and Supernode is used to perform coordination in a decentralized identity network.

5.6 LifeID

LifeID is an open identity platform based on a decentralized permissionless blockchain using smart contracts. With LifeID any institution or company can build an identity layer for their applications. It supports many secure features for its users in the form of biometrics, data privacy and recovery, backup and verifiable claims with zero knowledge proof. In addition to this, LifeID offers three different key recovery options such as self-backup, secure organizational restore, and backup through a trustworthy community of people. LifeID store on chain data such as DID, DID document, and optionally it stores the hash of verifiable claims and doesn't store any user information on the network. LifeID provides identification of different entities such as humans, organizations and IOT devices.

The platform supports a large amount of users and is able to bridge older identity platforms with blockchain based identity frameworks. This bridge technology only provides functionality for reading operations, and the components include OpenID connect identifier, and DID resolver. OpenID connect is an interoperable protocol used by many organizations and websites to authenticate users. The protocol allows web clients, JavaScript clients, and mobile clients to have access to services provided by the service providers. The DID resolver can search for a DID and get its related DID document.

5.7 SelfKey

SelKey is an open identity system built on the Ethereum public blockchain. It provides a free and open source identity wallet for the identity owner. The user first generates a public private key pair for their identity wallet and after that the user can request to legitimate claim an issuer to attest a claim based on the submitted document for notarization. Then these attestations can be used for verification of identity information. The attestations are designed in such a way that only a minimum of information is shared with the requesting party. Validating Ethereum nodes are the central part of SelfKey. A user's identity documents are stored on a user controlled mobile device SelfKey uses uPort's key recovery mechanism to recover the key of a user in case it gets lost.

5.8 Sora

Built on a blockchain technology to identify an entity in a network, Sora is used to store and validate verifiable claims about personal identifiable information. It uses a permissioned blockchain called Hyperledger Iroha for temper-proof sharing of transaction details of a user's information. The JSON-LD standard is used to structure identity data with a list of attributes and its associated values, which then allows a user to share a particular attribute to the verifier. Sora allows the user to create any number of identities on a network, and it is not possible to relate different identity attributes of a user.

Sora stores a key pair on a centralized server in an encrypted form for key recovery. Any user or organization can issue a verifiable claim, either for themselves or for any entity. The verifiable claims contain personal information in public and private part. The blockchain maintains public data and stores hashes of the claim and information regarding the user. The private part is shared with a verifier with the consent of the user when required. The user selects a personal 8-digit password for the encryption, which must be a combination of small and capital letters and digits. This password is the master key to decrypt the identity information. However, there are mainly two problems with this identity system. One of them being, that the system is not fully decentralized because of the storage of keys on a centralized server, and the second is the security of the master password could be vulnerable to various attacks.

The authors of [28] have written a review on Civic, Blockstack, and Evernym which we will now summarize.

5.9 Civic

Civic is a blockchain-based identity authentications system where third party wallets creates key pairs and stores identification information on a user's computer. Civic and its blockchain only accept data hashes that are stored on the Ethereum network as ERC20 tokens. It supports three different independent groups in the network which are; consumers, validators, and service providers. It is based on the Ethereum blockchain, and uses smart contract to track proof of attestation. The identity in Civic utilizes the validated identity for websites

and mobile development without requiring the username and passwords for authentication. The users monitor their protected data and only share the information they are willing to share. Civic's app is used to store identity information on a mobile device in an encrypted form. The Hash value of the attached identity information is stored in the Merkle tree and collected in the blockchain. The Merkle tree sections can be exposed selectively, thus increasing the user's control by enabling identity owners to disclose personal information selectively.

Civic allows trustworthy identity authentication providers, who are also known as validators, to participate and sign transactions in public blockchain nodes. Not entirely decentralized, but it reconfigures the centralization function and provides an interactive open system for the validators. Civic has the same consensus mechanism as Sovrin, and the authenticator can revoke identity records. If a user wishes to change their last name, the authenticating agency cancels the blockchain's previous last name, thus making the users dependent on authentication authorities to establish a protected digital entity, which results in a lack of portability. As Civic is a transparent permissionless blockchain, it does not have software or infrastructure for its network.

Civic's system have many benefits which include a strong relationship among financial institutions, public agencies, and intends to build a market among banks, utility organizations, local state, or federal governments verifying individual or business identity attributes on the blockchain. Validators can price identity authentication and sell the identity to stakeholders using smart contracts. Civic's system is effective as it plays a vital role in the ecosystem and uses validators to verify identity data that are accessible through mobile apps. Civic plans to launch the Civic wallet by integrating identification with other applications where users can interact more securely and efficiently using standard cryptocurrency applications compared to other wallets. The downside is that the development of this project is at a very early stage.

5.10 Blockstack

Blockstack is a decentralized network of computers that handle identity data. Blockstack ID is a decentralized user ID that connects decentralized applications. Blockstack public benefit corporation is an open source organization that is interested in developing core Blockstack protocols and applications. The application developed on Blockstack provides users with the control over their own identities and eliminates failure reference points. Used data credentials of a user cannot be stored at a centralized server, where the content sharing is carried out by using encryption. The collection of profiles can be seen and tracked globally through a blockchain, which may leak information and endanger the user's privacy.

Blockstack business logic and data processing operates on a computer instead of centralized servers that are hosted by service providers. The decentralized storage scheme of Gaia ensures that users own and operate private data lockers. Cloud users can use these lockers as additional storage. The Key recovery protocol is unavailable in Blockstack and, therefore, users cannot reset their keys in the event of failure or stolen ID, which leads to noncompliance with the persistence principle. Blockstack operates on the top of the Bitcoin network and is an open-source repository that offers programming libraries on a variety of platforms. The portable nature of Blockstack allows developers to adapt and integrate other technologies. Blockstack has a fullstack approach that provides all layers that are required to build a decentralized application, besides allowing customers with a single username to operate across all applications without a password. As the Blockstack system is in early development stage, it only offers desktop versions of the Blockstack browser.

5.11 Evernym

Established in 2013 by Jason Law and Timothy Ruff who is a well known player who aims to facilitate SSI introduction within various industries. Evernym achieves universal accessibility by using Sovrin to claim that SSI is a global and public utility to meet everyone's identity needs. Evernym can store all personal information on the customer's smartphone, while control in an Evernym solution is enabled by biometrics. The Evernym solution provides an easy way to import a private key and handle SSI. An individual can import a private key into a digital wallet through a text file or scanning a QR code. By using Sovrin, Evernym have a concept of "guardian" as a trusted third party to protect an exposed individual's identity. Evernym uses a hybrid open source framework that provides access to a permission ledger where the guardian organizations have to behave according to the criteria the Sovrin Trust Framework set out.

Evernym observed that the Sovrin network architecture, management, and operation could provide members with portability of their public and private data in compliance with other principles. Evernym connections within the Sovrin network will be connected by comparing a "fairly-pseudonymous identifications" or a single DID in each relation. The Evernym system is unable to provide flexibility, which results in a lack of interoperability. In addition to this, a small amount of information is available for the user control of the issuer's credential. In Evernym's Connect Me DApp, the user's biometrics are necessary to access a given identity and the related details in all situations. Individuals may also be expected to provide biometric information to establish peer-to-peer contact networks with other individuals and organizations in accepting credentials from an issuer to exchange credentials.

5.12 Discussion on selected solutions

Al-Bassam's [23] solution has several strengths and weaknesses including:

- Users have control over the attributes that are attached and not attached.

- Limited as everyone who is not referenced by the system has to use the system.
- Only suitable for publishing attributes that the user wishes to make public.
- Not good for publishing private identity attributes.

Sovrin [24] has several strengths including:

- Joining process of network is easy and friction-less. Users can also create several identities without any cost.
- Offers password-less authentication and single sign-on functionality.
- Offers user-centric sovereign identity that is fully controlled by the owner.
- Offers easy-to-use data management and control functionality where the user can store, control and share identity data.
- Sovrin complies with GDPR and privacy-preserving policy.
- In addition, Sovrin is publicly available all over the globe and can be accessed by all users through smartphones and an app to manage their identity.
- Sovrin is also based on open source projects and thus the code is open for anyone to use and to contribute as they wish.

The weaknesses of Sovrin include:

- Private key that is associated with a Sovrin identity is one of the possible attack opportunities if it is compromised, then the related confidential information can also become vulnerable.
- Regulated through a governance model called Sovrin Governance Framework therefore the identities and other services are governed through its members which leads to additional rules and constraints.

- As it is based on a public permissioned ledger, not everyone can operate nodes. Only trusted institutions (Stewards) can operate nodes while at the same time participates in the consensus process.
- Only offers limited portability, interoperability, and scalability The number of public repositories are also limited.

uPort [25] has several strengths including:

- Many of the same strengths as Sovrin with the likes of: Easy joining process and user-centric sovereign identity with full control by the user in addition to password-less authentication and single sign-on functionality.
- Offers user-friendly data management and control functionality where the user can store, control and share identity data.
- Different strengths over Sovrin: uPort's registry is a shared contract and the identity data is stored by each identity and solely controlled by the uPort identity, which means it is impossible to censor or block.
- uPort has a simple key management functionality where the users hold their private keys on their device and use social recovery methods by assigning recovery delegates who they chose on their own.
- Offers user-friendly mobile app and interaction with blockchain apps, as well as compliance with GDPR and privacy-preserving policy. It Also has an open-source identity management system, which means that the code is open for use by anyone.

The weaknesses of uPort include:

- Private key is only stored on the user's mobile device and if it is compromised then the identity and other related personal and confidential information could be at risk.
- The authentication process on the user's mobile device is not completely secure. The recovery delegates could be at risk and be subject to attack vectors, as their identity is connected to the user's identity and may be traced on the blockchain.
- uPort currently uses Proof of Work for the underlying Ethereum blockchain. This is not the most efficient algorithm, but Ethereum is planning on replacing it with a Proof of Stake consensus algorithm.

- The organization's own network could have different service and the smart contracts are smaller in terms of size and has limited capacity.
- The identity attributes are encrypted, but the analysis of their meta-data in the JSON structure might provide indications.
- In addition, uPort has limited portability, interoperability, and scalability. The numbers of public repositories of the identity management system are also limited. uPort have asserted that its basic identity management system will always be free to users, but all transactions have involved a cost.

Shocard [26] has several strengths including:

- QR scanning is simple and reliable.
- User is in control of secure storing and any sharing of data with a relying party.
- ShoCard supports many identity providers and have a very strong identity proofing.
- They also have their envelope technology where they store symmetrically encrypted certifications in case a user loses their device .

The weaknesses of ShoCard include:

- The hash of a user's attributes is available to the public, which means that the ShoCard server is able to associate a ShoCardID with the requests that are made by a relying party. With this information, they can technically track a user.
- If ShoCard stopped existing, the users would be unable to use the system
- As the users have to provide an existing credential like a passport or driver's license, it is very tedious for low-value online accounts.
- The Bitcoin transactions take too long, as it can reach upwards of an hour. This makes real time settlements of certifications a big challenger for the user experience.

EverID [27] has several strengths including:

- Uses biometric verification and a user's private data cannot be transferred without consent from the user.
- The API is good for integrating EverID with other applications and services.
- Provides tamperproof series of transactions.

The weaknesses of EverID include:

- User need to unveil full information in order to claim verification This does not go along with the SSI principle of minimization.
- As EverID uses biometric verification, a user can only make one identity account on the system.

LifeID [27] has several strengths including:

- Security features such as biometrics, data privacy and recovery.
- Key recovery options like self-backup, secure organizational restore and backup through a trustworthy community.
- Supports a large amount of users on its platform.

Sora [27] has several strengths including:

- Structures identity data with a list of attributes and its associated values.
- Allows users to create many identities, and it is not possible to relate different attributes of a user.
- Has key recovery methods in case someone loses their keys This is done by storing an encrypted key pair on a centralized server.

The weaknesses of Sora include:

- Encrypted key pair is in fact stored on a centralized server and to recover them the user needs a personal 8-digit password that could be vulnerable to various attacks.

Civic [28] has several strengths including:

- Utilizes validated identity for websites and mobile development without requiring username or passwords for authentication.
- Users can monitor their protected data and only share what they want to share.
- Users can selectively expose the Merkle tree sections, which increases their control and enables identity owners to disclose personal information selectively.

The weaknesses of Civic include:

- Users depend on authentication authorities to establish a protected digital entity if they wish to change for example their last name. This make it not entirely decentralized, but it also makes Civic lack portability.

Blockstack [28] has several strengths including:

- Provides users with control over their identities, eliminates failure reference points.
- A user's used data credentials cannot be stored at a centralized server where content sharing is carried out by using encryption.
- Gaia storage scheme allows users to own and operate private data lockers These let cloud users use it as extra storage space.
- Operates on top of the Bitcoin network and offers programming libraries on a variety of platform.
- Blockstack's portable nature allows developers to adapt and integrate other technologies.

The weaknesses of Blockstack include:

- Collection of profiles can be seen and tracked globally through a blockchain, this can potentially leak information and endanger the user's privacy.
- Blockstack does not have a key recovery protocol, so users' cant reset their keys in the event of failure or a stolen ID, which can lead to noncompliance with the persistence principle.

5.13 Comparison Tables

The comparison of selected solutions is summarized in the following tables.

Table 5.1: Comparison of Selected Solutions

	Al-Bassam's Scheme [23]	Sovrin [24]	uPort [25]	ShoCard [26]	EverID [27]
Blockchain platform	Ethereum	Hyperledger Indy	Ethereum	Bitcoin	Ethereum
Consensus mechanism		Redundant Byzantine Fault Tolerant consensus mechanism	Proof of Work, but plans on switching to Proof of Stake	Proof of Work	
Blockchain type		Public permissioned	Public permissionless	Permissioned	Permissioned
Smart contract content	Public key infrastructure		Users can sign attestations for other identities		Enables Supernodes to access EverID datagrams
Stored information	Identity attributes Cryptographic keys,names or Ethereum addresses	User decides what to publish	User decides what to publish	Existing IDs like passport or driver's license	Biometric identifiers
Roles		Sovrin Foundation, Sovrin Governance Framework Establishes trust Required to achieve certain identity requirements		ShoCard central server Acts as an intermediary to manage distribution of encrypted certifications between ShoCard and its users	

Table 5.2: Comparison of Selected Solutions (continued)

Progress	Concept proofed Work in progress, not yet released	Released as an application in 2016	Released as an application in 2017	Released as an application 2016	Released as an application in 2021
Strengths	Users have control over what they want to publish	Easy to join.Create multiple identities Password-less authentication Single sign-on functionality User-centric sovereign identity Controlled by owner	Same as Sovrin, but also offers: user-friendly data management Control functionality Identity data stored by each identity, controlled by the uPort identity	QR code scanning User in control of storing and sharing Support different identity providers Strong identity proofing	Biometric verification Private data of users can't be transferred or used without users' consent API for integration with other applications
Limitations	Parties referenced by the system must already use it Not suitable for publishing private identity attributes	Private key has attack possibilities if compromised Regulated through Sovrin Governance Framework, could possibly lead to rules and constraints	Private key, only stored on the user's mobile If compromised, identity, personal, and confidential information at risk Authentication process on mobile not completely secure	ShoCard can technically track a user If ShoCard stopped existing, users would be unable to use the system Slow Bitcoin transactions	Users unveil full information to claim verification Does not follow SSI principle of minimization Because biometrics, only one identity account on the system.

5.14 Use Cases

ShoCard

Some use cases for ShoCard can be for users to use their device's camera to store their identity and credit card information on the system. Systems like Airbnb and car rental sites like Getaround could use ShoCard to allow their customers to just with the tap of a few buttons, register their identity and payment information. The user logs in to the website with their ShoCardID and then proceed to find the car or place they want. Then they choose the date and time they want to rent it from, the system calculates the price cost and show it to the "customer". The user can then press the "rent" button Because the user logged in with their ShoCardID the calculated cost of rental will automatically be paid. For safety reasons, there should be another "confirm" message and button appearing after pressing the "rent button" asking if the person is sure they want to rent from this date to this date.

The people who rent out their houses or apartments on Airbnb could have a special lock or safe on the building which contains the keys to said building. The person who is going to rent it can then simply show up and scan their QR code on the lock to open it and retrieve the keys to enter the building. This makes it so the person who rent it out can be sure that the key is safe and also be sure it is the same person who rented the building that enters the building. The same could be applied for Getaround. The customer arrives to the place with the car, if it is in a garage, the person can scan their QR code on a device in front of the gate to open it When entering the garage, pictures of the car are required to be taken and scanned on the safe or box containing the keys to the car. Once the person deliver the car after rental, he or she is required to take new pictures and scan them on the box before putting back the keys. These pictures are for both the customer and the person renting out the car to see if the car was damaged during rental. The person renting will have proof that he or she did not damage the car. At the same time, the owner will have proof if they did damage the car as the pictures are connected to the identity that rented it.

Another use case for ShoCard could be to replace passports entirely, as it can be very dangerous and tedious to carry a passport around while travelling. If you are in another country and lose your phone that was in your backpack that somehow got stolen, the thief still need passwords and or biometrics to log into the ShoCard app containing your ShoCardID. ShoCard also has its envelope system in place to help you recover in case you lost your device. With the current version of passports, if they get stolen, the thief have everything right there at once.

Having ShoCard replace passports could speed up passport control in airports and automate it instead of having personnel checking every single person. You could have closed gates where you scan your ShoCardID and then have to look into a camera that confirms that your face corresponds with the scanned ShoCardID.

As Facebook is already sitting on data on all of their users, there could be implemented a different version of Facebook that uses ShoCard as its building block. In this case, the users sign up to the website using their ShoCardID. This not only makes it, so the users are in control of their data, but it also limits or entirely removes fake profiles as you can only have one account per ShoCardID. Now you can be sure that the person is real, and the platform is more safe as it will be harder for viruses to be spread around. People will not be able to create a bunch of fake profiles in order to send "special links" containing viruses or malware to unsuspecting victims in order to get their passwords or other types of credentials. Last but not least, we will not get another episode of the Cambridge Facebook Analytica Scandal because there is no user data to sell as all the users are in full control of their own data.

The BankID app we have in Norway could be remade with ShoCard and be used worldwide. You simply put your identity credentials onto ShoCard together with facial recognition, and everything that requires you to log on or sign in with BankID can be done with a single face scan with ShoCard. Whenever you are required to show ID proof like logging into your bank, opening online mail from for example Posten, logging into Studentweb, buying stuff online and much more. You can simply tap the open with ShoCard option and take a face scan with ShoCard and just like that you are logged on. It takes away the tedious job of putting in your phone number, date of birth and then an additional pin code on the code word they send you. All this secured on ShoCard with full user control and no company or organization sitting on all of your personal data.

A small problem occurring in Norway is that a lot of banks are no longer putting identity credentials together with a photo on their credit cards. As a lot of people don't have a driver's license, they have to carry around their passport. Having to carry a passport around for whenever you go to the store to buy alcohol or when you go out to restaurants, bars, or nightclubs etc can be quite dangerous. If the wrong person sees you showing off your passport to enter the club and then putting it back into a purse or pocket. That person can then later on try to steal it if you are too drunk or not paying close enough attention. It is easier to have your items stolen from a bag or a purse due to how "separated" they are from your own person.

In contrast, most people keep their phones in their pants or hold them on their hands throughout the night, making it less accessible to others. We can take a group of young women sitting down at a bar drinking as an example. At some point during the night, the group decides to get up and dance after having a few drinks. Because of the alcohol, the heat of the moment, or simply because it is more comfortable to dance without carrying a purse, one of the women decides that she will leave her purse, which has her passport and other valuables inside, on the table.

This could be a great opportunity for the wrong person to take the purse or simply steal the contents of the purse, particularly the passport and credit cards. Some may argue that getting your passport stolen is more severe than getting your credit card stolen. Blocking and replacing a credit card is as easy as contact the bank and have them do it for you. However, often with passports, it is a rather tedious and, in some instances, long process just to be able to have a new passport.

This is where ShoCard comes in. As you can put your identity credentials and a photo of yourself on ShoCard you now have a much easier way of carrying around a valid ID. You can simply open up the ShoCard app and show your ID to whoever checks it at the door at the club, or to whomever checks for your ID at the store when buying alcohol. Grocery stores and liquor stores could even keep up with this and have you scan your ShoCardID at the self-service in order to reduce the amount of time each customer uses. This would also make it so the employee could focus on other tasks than having to run around and check everyone's ID when they are buying alcohol.

In addition to putting your identity credentials on ShoCard, the user can even put their credit card information on it. Thus, the user has eliminated one more item that they must carry around without worrying about its location, namely, a credit card. By looking back at the example of the girls at a nightclub, we can assume the girls have their phones on the pockets of their pants. In the case that there is a malicious individual at the club who is looking into stealing their phones, they would now have to be closer to the girls and reach inside their pockets in order to steal the phones. The location of the desired item, the phone, already makes the stealing process harder where the malicious individual to succeed. Nevertheless, if the phone or phones were to be left inside of the purse of the owner, the person stealing it would still require a password, pin code, facial recognition or fingerprint in order to access the phone. In the case of the person being able to access the phone, they are now having to go through a second protection layer, namely ShoCard, which also requires a password, facial recognition and a fingerprint to log into it. By looking at this, we can see that it makes it harder for any malicious individual to be successful when trying to steal a person's identity or banking information. The maliciuos individual would have to directly approach the owner of the phone, something that can be quite challenging, particularly while in a public space.

ShoCard has so many possibilities, but also weaknesses that need to be either fixed or improved upon in order to realize these use cases that I have mentioned. First, ShoCard needs to go from their Bitcoin based transactions to something that does not take an hour for information to be added. This is important for the use case about Getaround and having pictures being uploaded quick and easy. In the case of it replacing passports and serving as ID for buying alcohol and entering bars etc you somehow need to make sure it is not possible to create fake profiles, so some sort of governmental involvement seems to be a must. In addition, it needs to remove the fact that if it stopped, existing users would no longer be able to use the service it provides. This could cause problems on a global scale that have massive impact on the society and would shut down a lot of travel etc as people now use ShoCard instead of passports.

Sovrin Network

The Sovrin Network could be used together with mittuib to handle the identity management parts. Its users, both students and employees etc can put on just the information that is required about them like name, date of birth, address, study line and which institute they belong to. For example, I could put Ola Nordmann, born 17th of May 1990, live at Alègaten 66, study Cybersecurity at the institute of informatics. Now I know that my data is safe and that I have full control over it as no other entity can get it.

Sovrin could be used as a type of identification system like BankID. You put on the necessary information like first name, last name, social security number and date of birth. Sovrin's use of ZKPs ensures that as little as possible is known about the user, and yet they can prove their identity to whoever needs it. For example, when logging into your bank account you simply choose to log in with your Sovrin identity and with a secure password-less single sign-on you now have access to your bank account. This could also be used for logging in to Posten to view sensitive information or sign things. You can use it to log into healthcare to view messages from your doctor, or log into laanekassen to apply for stipend for studying. For all of this, Sovrin needs to fix or update some of its problems.

Sovrin could be used as the underlying identity handler for Facebook. This way Sovrin's ZKP system makes sure that as little as possible information about the user is revealed and the encryption mechanisms makes the user account secure. The user can share only what they want to share. All of this ensures that the users' data cannot be misused, spread around or sold for various use such as marketing, campaigning (Facebook Cambridge Analytica) or even identity theft/fraud. Facebook has tons of fake profiles going around pretending to be other people. These fake accounts will try to spread viruses or other types of harmful malware in order to achieve different things like getting the innocent user's username and password, etc. With a decentralized system like Sovrin working in the background handling the identity management, this will be a lot harder to achieve.

The Sovrin network looks like a good solution that could be implemented in a lot of applications, but they do have problems. They need to find a solution so that their key pairs does not have attack possibilities because this will cause the users' confidential information to be compromised. In addition, they need to improve on their scalability in order to support a lot more users. In addition to this, their system is regulated by the Sovrin Governance Framework that is governed through their trustworthy members, this could lead to additional rules and constraints. This could lead to people questioning Sovrin as a decentralized system in its entirety.

Chapter 6

Conclusion

In this thesis, I have reviewed different concepts within both identity management systems and blockchain, as well as the evolution of identity management systems (IMSs). IMSs have significantly changed a lot over the years. Early on, you only needed to have a username and a password in order to log in to certain things. IMSs have evolved to the need of longer and more complex passwords. For example, the identity provider usually requires users' passwords to have at least 8 characters, at least one number, one special character, and at least one upper case letter. Nowadays, this evolved to requiring multi-factor log-on on top of username and password, where a user needs to connect his/her phone number to the account. Nonetheless, modern IMS have a lot of flaws with the likes of having to be fully functional fast, or centralized in which they perform too many roles such as storage of sensitive information, authentication, and authorization. This has led to the risky situation that provider sitting on too much information about user data, where the user will not be in control of their own private data.

By reviewing the existing blockchain-based schemes for IMSs throughout my thesis, my understanding of how they can be applied in real life has improved. Each case can have certain impact on society. Not only can they help and improve certain areas like the use case of most bank cards no longer having ID on them in Norway, but they can also secure and protect users' identities. With a more secure and protective blockchain-based IMS, users would be in better control of their identity and personal information sharing. We can believe, with a higher degree of certainty, that the less access to users' information, the less an adversary would gain by performing specific actions.

Bibliography

[1] KA Nyante. Secure identity management on the blockchain. Master's thesis, University of Twente, 2018.

[2] Rosalie Chan. The Cambridge analytica whistleblower explains how the firm used facebook data to sway elections. *Business Insider*, 5, 2019.

[3] Sam Meredith. Facebook-Cambridge analytica: A timeline of the data hijacking scandal. cnbc, 2018.

[4] Allan Smith. There's an open secret about cambridge analytica in the political world: It doesn't have the 'secret sauce' it claims. *Business Insider*, Mar 2018.

[5] Julia Carrie Wong. Facebook to be fined $5 billions for Cambridge analytica privacy violations–reports. *The Guardian*, 12, 2019.

[6] Blockchain identity management: The definitive guide (2021 update). *Tykn*, Dec 2021.

[7] Blockchain. *Wikipedia*, Sep 2022.

[8] T Economist. The great chain of being sure about things. *Economist*, 2015.

[9] Jeff John Roberts. Bitcoin spinoff hacked in rare '51% attack'. *Fortune*, 2018.

[10] Jerry Brito and Andrea Castillo. *Bitcoin: A primer for policymakers*. Mercatus Center at George Mason University, 2013.

[11] Identity and access management. *EDUCAUSE*.

[12] Understanding the importance of iam (identity and access management). *AuditBoard*, Dec 2021.

[13] Sandra Gittlen and Linda Rosencrance. What is identity and access management? guide to iam. *SearchSecurity*, Aug 2021.

[14] Martin Schäffner. Master's thesis martin schäffner. Master's thesis, University of Mûnchen, 2019.

[15] Nick Szabo. Smart contracts: building blocks for digital markets.

[16] History of blockchain - javatpoint.

[17] R Sheldon. A timeline and history of blockchain technology. *URL: https://www.techtarget.com/whatis/feature/A-timeline-and-history-of-blockchain-technology*, 2021.

[18] How does blockchain work and what are its different layers, explained. *cnbctv18.com*, Mar 2022.

[19] Consensus mechanisms in blockchain: A beginner's guide. *crypto.com*, May 2022.

[20] Muhammad Nasir Mumtaz Bhutta, Amir A. Khwaja, Adnan Nadeem, Hafiz Farooq Ahmad, Muhammad Khurram Khan, Moataz A. Hanif, Houbing Song, Majed Alshamari, and Yue Cao. A survey on blockchain technology: Evolution, architecture and security. *IEEE Access*, 9:61048–61073, 2021.

[21] Iddo Bentov Computer Science Dept., Iddo Bentov, Computer Science Dept., Charles Lee Litecoin Project, Charles Lee, Litecoin Project, Litecoin ProjectView Profile, Alex Mizrahi chromawallet.com, Alex Mizrahi, Chromawallet.com, and et al. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract]y: Acm sigmetrics performance evaluation review: Vol 42, no 3, Dec 2014.

[22] Alexander S. Gillis. What is guid? *SearchWindowsServer*, Aug 2021.

[23] Mustafa Al-Bassam. Scpki: A smart contract-based pki and identity system. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pages 35–40, 2017.

[24] Nitin Naik and Paul Jenkins. Sovrin network for decentralized digital identity: Analysing a self-sovereign identity system based on distributed ledger technology. In *2021 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–7. IEEE, 2021.

[25] Nitin Naik and Paul Jenkins. uport open-source identity management system: An assessment of self-sovereign identity and user-centric data platform built on blockchain. In *2020 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–7. IEEE, 2020.

[26] Paul Dunphy and Fabien A.P. Petitcolas. A first look at identity management schemes on the blockchain. *IEEE Security & Privacy*, 16(4):20–29, 2018.

[27] Jayana Kaneriya and Hiren Patel. A comparative survey on blockchain based self sovereign identity system. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pages 1150–1155, 2020.

[28] Mohammed Shuaib, Noor Hassan, Sahnius Usman, Shadab Alam, Surbhi Bhatia, Arwa Mashat, Adarsh Kumar, and Manoj Kumar. Self-sovereign identity solution for blockchain-based land registry system: A comparison. *Mobile Information Systems*, 2022:1–17, 04 2022.