

# Blockchain as a Technology to Facilitate Privacy and Better Health Record Management



Author: Tsigab A. Gebremedhin

Supervisor: Yngve Lamo & Svein Ivar Lillehaug

WESTERN NORWAY UNIVERSITY OF APPLIED SCIENCE

A thesis submitted in partial fulfillment  
of the requirement for the degree of  
*Master of Science in Software Engineering*

August 15, 2018



## **Acknowledgements**

I would like to express most profound gratitude to my supervisors Professor Yngve Lamo and Svein-Ivar Lillehaug of Western Norway University of Applied Sciences. Without their guidance and expert advice, this thesis would not have been possible. Furthermore, I would also like to thank my mother Alganesh, my lovely and supportive wife, Eyorusalem, and my two beautiful children Hosianna and Ariam for always encouraging and providing me with unending support.



## Abstract

Fear of stigmatization and discrimination from colleagues, friends and family drives patients with various type of mental health problems away from a traditional face-to-face therapy and enforces them to look for an alternative treatment methods. Internet-based mental health therapy helps patients to get their needed therapies and support from healthcare professional and peers, or as a part of automated online form of therapy. Conducting Internet based therapy anonymously is vital for the patient privacy. However, lack of trust, access permission, ownership control and traceability undermines patient safety and security. Blockchain technology is an innovative technology initially designed for a cryptocurrency. However, with the introduction of programming blockchain and smart contracts, the technology has extended its importance to other areas for developing decentralized application (DApp), such as mental health related information management, which is the primary focus of this thesis.

Privacy and security are very crucial for patient safety and to preserve patient's medical history from adversaries. Sharing of private medical information online between the patient and their respective provider contains sensitive information that can easily be compromised if a proper security measure is not put in place. Blockchain is consensus-based peer-to-peer distributed ledger technology that stores and maintains an updated copy of all transactions within the network. It makes trust more transparent and traceable by keeping auditable-logs of all transactions in the form of blocks.

In this thesis, Blockchain and its underlying technology are studied, and a prototype has been developed to explore the potential of the blockchain technology. Furthermore, we explore alternative distributed ledger technologies

and their respective security models such as consensus protocols, cryptographic techniques, privacy and scalability. The prototype was proposed based on Ethereum blockchain.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem Definition: Patient privacy, data ownership . . . . .	3
1.2	Motivation . . . . .	4
1.3	Purpose and Research Statement . . . . .	6
1.4	Thesis Outline . . . . .	7
<b>2</b>	<b>Technical Background: Blockchain and Ethereum</b>	<b>8</b>
2.1	Blockchain, a Distributed Ledger Technology . . . . .	9
2.1.1	Permissioned and permissionless blockchain . . . . .	11
2.1.2	Blockchain Consensus Models . . . . .	12
2.1.3	Public Key Cryptography . . . . .	14
2.1.4	Bitcoin: Peer-to-Peer Blockchain Technology . . . . .	16
2.2	Ethereum . . . . .	18
2.2.1	Ethereum Account Concepts . . . . .	20
2.2.2	Blocks in Ethereum . . . . .	21
2.2.3	Gas, Ether . . . . .	22
2.2.4	Mining . . . . .	22
2.2.5	Smart Contract . . . . .	23
2.2.6	Solidity - Ethereum smart contract language . . . . .	24
2.3	IOTA . . . . .	25
2.4	Hyperledger Fabric . . . . .	27
2.4.1	Membership Service Providers (MSP) . . . . .	29
2.4.2	Fabric Smart Contract : Chaincode . . . . .	29
2.4.3	Consensus and Ordering Policy . . . . .	29

2.5	Blockchain in Healthcare: Previous Works . . . . .	31
2.6	Chapter Summary . . . . .	32
<b>3</b>	<b>mHealth-BlockC: Design and Implementation</b>	<b>35</b>
3.1	Motivation for Developing a Prototype for Mental Health (mHealth) . . .	35
3.2	Choice of Technology . . . . .	37
3.3	User Stories and Requirements . . . . .	39
3.4	mHealth-BlockC Architectural Overview . . . . .	41
3.5	mHealth-BlockC Design . . . . .	44
3.5.1	Smart Contracts . . . . .	45
3.5.2	Front-End Web Interface Design . . . . .	48
3.5.3	Users Login and Ethereum Address Generation . . . . .	48
3.5.4	Ethereum Block Mining . . . . .	49
3.6	mHealth-BlockC Prototype and Code Review . . . . .	49
3.6.1	mHealth-BlockC prototype adoption and Ethereum Functionality	50
3.6.2	Front-End web platform . . . . .	54
3.6.3	Web3.js sample codes . . . . .	55
3.7	Prototype Testing . . . . .	56
3.7.1	Hardware Components . . . . .	56
3.7.2	System Components . . . . .	57
3.7.3	Patient-Provider Smart Contract Deployment . . . . .	57
3.7.4	DAPP resource usage : CPU and Memory . . . . .	58
3.8	Summary: Prototype Logical Sequence Code Review . . . . .	59
<b>4</b>	<b>Evaluation</b>	<b>65</b>
4.1	Security . . . . .	65
4.1.1	Gas and Transaction Fees . . . . .	67
4.2	Privacy . . . . .	68
4.3	Scalability . . . . .	69
4.4	Usability . . . . .	70
4.4.1	Elapsed time for Block Creation . . . . .	71
<b>5</b>	<b>Discussion and Future Work</b>	<b>73</b>
5.1	Discussion . . . . .	73



5.2 Future Work . . . . .	76
<b>6 Conclusion</b>	<b>77</b>
<b>Bibliography</b>	<b>79</b>

# List of Figures

2.1	Blocks in blockchain [1]	9
2.2	Digitally signed data verification and authentication using to Nodes A and B	15
2.3	Double spending problem	17
2.4	Block creation in blockchain	17
2.5	Ethereum block structure and block contents [2]	21
2.6	Solidity code snippet taking from our DApp implementation	25
2.7	The blue transaction on Tangle are fully confirmed and the nodes of newly added transaction indirectly participate on validation of transactions. For example peers for red and green transactions have been used to validate the blue transaction. The Red are on the process of validation while the blue are newly added transaction, unconfirmed and are called tips [3]	27
2.8	Transaction Flow diagram for Hyperledger Fabric based consensus process[4]	30
3.1	The mHealth-BlockC Data flow diagram	40
3.2	Simplified architectural overview of mHealth-BlockC and the the interaction between the patient/provider interface, frameworks and smart contracts in the EVM	42
3.3	Decentralized Application (DApp) structure	45
3.4	Patient and Provider contract structure	47
3.5	Events allows users to track whether they are registered successfully or failed by using logged message into the EVM	47
3.6	Creating an Ethereum account address for the user	49
3.7	An actual Ethereum block which is created during patient registration.	50

3.8	An overview of different DApp development libraries, frameworks and tools connecting to one another to execute transaction. . . . .	51
3.9	Truffle helps in structuring and organizing the DApp development environment . . . . .	52
3.10	Genesis Block structure with gas limit, nonce difficulty defined for our Ethereum test nodes [5] . . . . .	53
3.11	Provider setup for RPC connection . . . . .	54
3.12	user subscription form and patient profile during login with newly assigned Ethereum account address and initial Ether funds . . . . .	55
3.13	snapshot of patient’s medical record with profile data and generated randomly. . . . .	56
3.14	Creating an Ethereum account address for the user . . . . .	56
3.15	The code snippet first looks for the provider using hostname and port, and the ABI is loaded using web3 by calling GroupEtherapy.json file to interact with the EVM . . . . .	58
3.16	Logical Sequence diagram . . . . .	60
3.17	show how users registered to mHealth DApp. While registering they can select user type (patient or provider) from drop down list and also by calling registerBC() the system assigns them a blockchain account address. . . . .	61
3.18	Both patient and provider access the DApp by entering the user(email address and password) . . . . .	62
3.19	Patient and healthcare provider can modify their profiles after they login to mHealth and the patient can also share medical information with provider by setting the necessary access permissions . . . . .	63
3.20	Logical Sequence Diagram: patient profile flow . . . . .	64

# List of Tables

- 2.1 Comparising between the threes decentralized ledger technology . . . . . 34
- 4.1 Gas used during block creation in Ethereum network . . . . . 67

# Chapter 1

## Introduction

Digital technology in general and the Internet, in particular, has the potential to improve the quality of healthcare delivery and health outcomes for patients with mental and chronic diseases. A regular interaction between providers and patients using digital healthcare tools and services offers a unique opportunity to improve patient-provider communication and facilitate engagements [6]. E-health tools and services such as online discussion forum, secure instant messaging, and other synchronous and asynchronous applications have been used to provide online counseling and therapy for a large group of mental health populations. Furthermore, evidence-based support and interventions using digital technology have shown promising results in providing knowledge, emotional, social and practical help to patients with mental health challenges [7].

Online peer-support are guidance service and resources that help patients affected by mental illness and similar concerns to overcome common challenges. Also they can provide support by other patients that have experienced similar problems [8]. They have the benefit they that can include the provision of a safe environment to freely express and share emotions and ideas about one's current situation and challenges. Internet-based mental health interventions have been found to be highly effective for patients suffering from anxiety, depression and also less stressful than traditional methods [9].

Most of today's online mental health tools are using a simple web application with a client-server architecture where the security setting in most cases does not provide secure end-to-end encryption. Relying on centralized client-server architecture to store and exchange patient data poses a tremendous privacy and confidentiality threat to

medical data [10]. Providing continuous care, medical information and services have to be accessible for authorized users at any time and anywhere. Services running on a centralized network are vulnerable to security breaches, and a single point of failure in the system can destroy the accessibility of the entire communication network between endpoint nodes. To overcome security and privacy challenges such services needs to adopt a new emerging technology that supports a distributed system and protects the integrity and availability of medical data [11, 12].

Having access to a shared ledger, tamper-proof and transparent history of the transaction to all participant nodes of the network (for example patients can modifying the security permission and access rights to their personal medical informations) prevails the issue defined above. By providing the tool for patients to achieve anonymity and consensus among distributed nodes without depending on a single trusted third party, blockchain has the potential to ensure privacy, data security and facilitate patient-centric data management for Mental Health (mHealth) applications as well as for healthcare systems and decentralized application in general.

Blockchain known as distributed ledger is an emerging technology that has enormous potential in tackling transparency, availability, and confidentiality all critical issues facing mental health system. The blockchain is a shared peer-to-peer decentralized ledger that keeps an immutable continuous growing order of records in the form of blocks [13]. It relies on cryptographic hashes and incentives for ensuring data integrity and confidentiality of all transactions across distributed nodes [14]. In a healthcare setting, a transaction can be defined as a process of generating, exchanging, and uploading such as patient data, medication and other treatment data within the distributed nodes. In blockchain, there is no central trusted party. Instead, all records of transactions are hashed together into blocks and stored in a distributed network across all nodes.

The possibility of adopting and using blockchain technology in areas other-than cryptocurrency has attracted global attention in other fields, in particular after developing and introducing the concept of programmable blockchain: smart-contract and Decentralized Application (DApp). Smart contract and DApp have revolutionized the application of blockchain into other domains such as healthcare [15]. It has immense potential in healthcare for developing secure, transparent and tamper-resistant medical registries that can provide patients with enormous accessibility, trust and privacy capabilities.

## 1.1 Problem Definition: Patient privacy, data ownership

Mental illness is a leading healthcare problem across the globe. According to the World Health Organization [16], one out of four people are affected at least onetime at some point in their lives, and around two third of the population with known mental health problems have never sought medical support from healthcare professionals. Moreover, it is a known problem that lots o people with mental illness prefer not to disclose their mental health problem due to fear of job restriction, discrimination and stigmatization from the community and friends [17]. Being labeled mentally ill, and resulting in hopelessness, societal stigma is leading patients to look into alternative technologies that can provide them with a secure and anonymous communication [18]. Internet-based treatment and peer-support groups are some of the alternative solutions in which patients get support to deal with their physiological problems [19, 20]. However, most of the applications are running on the centralized network which uses insecure and unencrypted communication channels. Developing a system that deals with the patient agency using a centralized system might undermine the ability of the patients to maintain ultimate access and control of their data. The lack of privacy, confidentiality and ownership control over generated records leads to potential security breaches by malicious users for financial gains [21]. Furthermore, failing to mitigate privacy, security vulnerability of created medical records points to a possible security breach that might resulted in substantial financial and legal consequences.

People that are suffering from mental illness cannot defend themselves from malicious activities. Consequently, the can be become vulnerable to security breaches. Privacy is at the risk of being compromised, and identity cab be revealed from within the healthcare providers. As a result, private medical data have been shared and used for medical research without prior patient knowledge [22]. Those privacy and security breaches abstain patients from participating further in mental health-related treatments and furthermore, as they might resist disclosing their illness, family history, and other crucial information to providers due to bad malicious experiences.

In the age of digital currency and online banking, a patient with mental and psychological problems might be willing to self-manage their mHealth data transparently and confidently. However, the existing digital barrier that is facing mental health today

might create fear among patients and force them to abstain from disclosing sensitive medical episodes and seeking treatment. When designing a system to overcome those barriers, we must first prioritize those problems that have a direct impact on the patients' privacy and safety. Patients with mental health-related issues benefit from a medical system that offers an integrated, transparent and consistent overview of their medical history [23]. Setting like this are allowing them to build trust toward the technology and ensure their safety by providing the patients to maintain ownership and control over their own medical records. Furthermore, it encourages them to continuously engage in clinical decision processes and in preventing their medical records from adversaries.

In today's healthcare setting, huge amounts of medical data has been gathered by healthcare providers to improve the quality of care by enabling better care and developing new treatments and medications [24, 25]. In the absence of proper access control and permission, medical data that are shared and exchanged between healthcare providers and researchers might lead to a security vulnerability. Patients as the owners of the medical data need to be part of the decision-making process and, furthermore, they have to be concerned about they are sharing their data with, and for what purpose. However, exchange of sensitive patient information must be consent-based, otherwise patient privacy and data integrity is at risk [26].

## 1.2 Motivation

This thesis studies some fundamental issues that are challenging mental health today. Our first motive is to ensure patient's easy and transparent access to private medical data that have been generated (privately) during E-Therapy and while visiting health service providers across various treatment sites. In the current mental health setting, most medical records are stored in a centralized database system in which patient data remains mostly non-portable [27]. The patient access to mHealth data across providers and treatment sites are also limited. Furthermore, the trust relies on a single trusted third party and this might create a security risk. An ubiquity access with an auditable-logs of medical history improves the ability of a patient to monitor and control medical records. Moreover, the patient will be able to make a smart decision in granting or denying access to treatment data.



The second motive might to offer a patient the ability to communicate with their respective providers anonymously and keep the generated record encrypted. Patients as the owner of their medical data have the right to decide in disclosing, hiding and communicating anonymously with their respective peers or providers. In most cases, the proper degree of anonymity and confidentiality rely on patient preferences in compromising privacy and desire to receive informed medical care from providers [28]. The patient turns to the Internet for cyber-therapy and to get online group peer-supports because of their anonymity, which makes them compelling to discuss highly confidential personal issues [9]. However, they have concerns about privacy and data security and motivates them to look into alternative technologies that can offer them the immutable system, data encryption, and anonymous services [29].

Patient-initiated data sharing and privacy-preserving is our third motive. While exchanging medical data, patients and providers need a reliable method and tool for verifying and authenticating the identity of the parties involved. Cybercrime is on the rise around the globe, and healthcare data has also become the target of security attacks [27]. For example, in 2017 a ransomware hacking attack targeted over 150 countries and hundreds of thousands of machines all over the world [30]. The attack highlighted how vulnerable our healthcare system is to potential security threats. The inadequacy of the current infrastructure is compromising patient privacy and expose mental health data to possible malicious access without owner consent. To securely exchange medical records, the patient can provide a different level of permissions and access rights to providers based on their roles.

In addition to the above motives, the European Union (EU) has developed a data privacy law and directive called General Data Protection Regulation (GDPR) [31]. This framework is designed to protect and safeguard citizens personal information and also to change the organizational approach to data privacy across the EU. It has been in debate for four years before it was approved in April 2017. According to the EU [32], the directives will come into effect on 25<sup>th</sup> May, 2018. So, any organization whether or not in the EU who used to process personal data in the context of doing business or providing services to individuals within the EU must now comply with GDPR [33]. Healthcare sectors are one among many other areas that deal directly with personal data for providing treatment, doing research and for clinical trials. Based on the GDPR, processing

any private personal data such as medical data requires patient consent. Noncompliance with the law has legal and juridical consequences depending on the type of offense committed. This regulation provides patients with fundamental rights to defend their medical information from any privacy and security breaches. On the other hand, it encourages healthcare providers to explore and adopt an innovative technology which can help them to mitigate unsolved challenges described in GDPR, such as accountability, transparency, and confidentiality in processing and storing mHealth data.

### **1.3 Purpose and Research Statement**

The privacy, trust and verifiability concern imposed in currently available e-mHealth treatment demands a new approach to achieve privacy-preserving, confidentiality, and transparency with the needed requirements. The primary goal of this master thesis is to explore the potential of blockchain technology and how to apply the concept of smart-contract in developing a distributed application for monitoring and governing personal medical data in e-mHealth. Moreover, we will also investigate how secure the technology manages and preserves mHealth data in a more autonomous, confidential and distributed way without interference from a trusted third party. The preliminary design principles for our prototype throughout this work focus mainly on clarity and simplicity. We are building the application using blockchain and designing the user-interface using feature-rich web technologies. In this thesis, we try to address the privacy and the security concerns, and more specifically, the research questions focus on the following:

1. Does blockchain technology have the potential to mitigate confidentiality and privacy problems facing mHealth?.
2. What potential does the blockchain have for developing a transparent, traceable Internet-based decentralized environment for people with mental health issues?
3. Which cryptographic tools and methods are available that can provide a patient with a secure privacy preserving environment that can also can and anonymize the patient relationships with providers and peers?
4. How can the patient remain in control of their personal medical data?

## 1.4 Thesis Outline

The structure of this thesis is organized into six chapters.

- **Chapter 1 :** The first chapter gives an introduction to Internet-based treatments discusses the motive behind our research and presents the research questions of this thesis.
- **Chapter 2:** Briefly describes a different type of distributed ledger technology (DLT) by comparing the underlying protocols, cryptographic algorithms and technologies.
- **Chapter 3:** Describes the design concepts and the implementation of the prototype using smart contract and front-end frameworks.
- **Chapter 4:** Present the evaluation of our prototype and compares with other DLT technologies using the parameters such as privacy, security, scalability and computational costs.
- **Chapter 5:** Present and discuss the result from the evaluation of our prototype and suggests for future work.
- **Chapter 6:** Finally, the conclusions are presented in the last chapter.

## Chapter 2

# Technical Background: Blockchain and Ethereum

The primary objective of this research is to apply Blockchain technology to develop a prototype and explore the potential of a blockchain technology as a platform to support a secure peer-to-peer patient-driven anonymous and decentralized record management. From the perspective of privacy and security challenges in managing mental health records, the prototype should provide secure patient access and ownership in sharing and exchanging medical records with healthcare providers and families.

This chapter profoundly explores the core technologies that are closely related to this research and is divided them into six sections. The first section gives a brief introduction to blockchain and underlying building blocks such as peer-to-peer networking, mining, double spending and public key cryptography. The second section introduces Ethereum, a blockchain technology that is used to develop the prototype of this thesis project. The third and fourth section looks into other distributed ledger related technologies such as IOTA and Hyperledger Fabric that are of importance in designing and developing decentralized applications. Blockchain previous works in healthcare are also mentioned in the fifth section. The different DLT technologies described in this chapter are summarized in a single table in the final section.

## 2.1 Blockchain, a Distributed Ledger Technology

A distributed ledger technology (DLT) is a database which allows each node to store an identical copy of a record data across the entire network [34]. The state of the distributed ledger is independently maintained and updated by all participant nodes. A blockchain is one type of the DLT technology which employs a unique data structure for organizing and storing the data in the ledger. Data in the blockchain is organized in blocks (see Figure 2.1), and each block is linked to its previous block using a cryptographic secure hashes. Not all DLT applies a chain of blocks to store data in the system. For example IOTA is a DLT, and uses Directed Acyclic Graph (DAG) based data structure called Tangle [35]. Validated transaction is represented as a vertex in the Tangle graph.

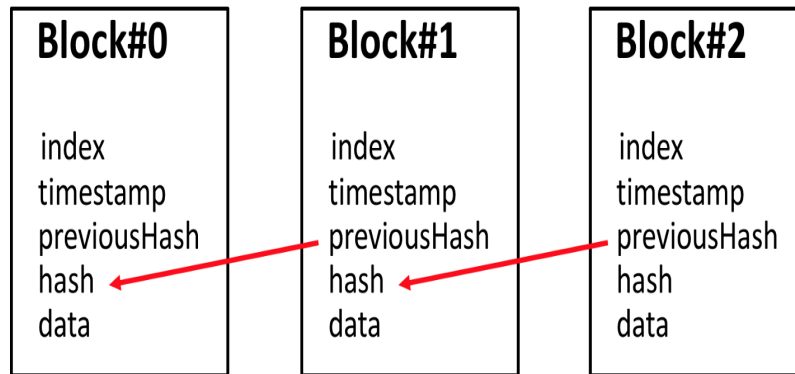


Figure 2.1: Blocks in blockchain [1]

Blockchain is the technology behind the world's cryptocurrencies such as Bitcoin, Ethereum, LiteCoin, etc. It is a shared Peer-to-Peer (p2p) decentralized ledger<sup>1</sup> that keeps an immutable continuous growing order of records in the form of blocks. Blockchain relies on cryptographic hashes and incentives for ensuring data integrity and confidentiality of all transactions across distributed nodes [14]. In blockchain, there is no central trusted party. Instead, all valid records of the transactions are hashed together into blocks and stored in the distributed networks across all nodes.

The original concept of blockchain first introduced in a white paper that was made

---

<sup>1</sup>Ledger is a collection of log

public in 2008 under the pseudonym Satoshi Nakamoto [36]. The groundbreaking paper is known by its title "Bitcoin: peer-to-peer electronic cash system" attracted global attention in academia, banking, and governmental organizations [37]. Millions of dollars have been invested in research and experimentation to adopt the technology since the paper was published. According to coinmarketcap [38], the current market capitalization of the whole blockchain technology is around 399 billion USD as of April 2018. Bitcoin is the first p2p electronic payment system that allows for two consenting parties to perform transactions without an intermediary. It is the first digital currency that solved the problems of double spending and Byzantine fault tolerance. Those problems have been topics for research for a long time since before Bitcoin. According to Brewer's theorem [39], it is impossible to achieve consistency, availability and fault tolerance all together in a system that supports distributed network. However, in a centralized system, problems like them never exist. As there is only one trusted part which is in charge of monitoring all communication between the nodes in the network.

In a decentralized p2p network, no single entity has control over the entire communication, but all participant nodes play an existing role in the state of a transaction by verifying the validity and consistency across the distributed ledgers. When we talk about blockchain, it has a decentralized architecture, but logically it is centralized because the network holds the same copy of database across all nodes. The distributed system behaves like a single computer with a commonly agreed state.

Blockchain platform employs a range of consensus protocols to reach an agreement on the validity of a transaction. For example, Bitcoin and Ethereum use Power-of-Work (PoW) consensus protocol to mine a block difficulty (block hash) to reach consensus. PoW

Blockchain uses cryptography techniques and hash algorithms in place of trusted parties to verify, authenticate and sign transactions. Public-key cryptography [40]<sup>2</sup> uses a pair of keys: a private key and a public key to verify signatures as well as to encrypt and decrypt transaction in the blockchain. The private key is a secret key that is only known to the owner and used to sign messages digitally. The public key is derived from the private key using Elliptic Curve Digital Signature Algorithm (ECDSA) algorithm and

---

<sup>2</sup>Public-key cryptography is sometimes referred to as asymmetric key cryptography to differentiate from a symmetric key that uses one key.

is public to the network. Furthermore, it is used to encrypt transactions and to verify the integrity of signed transactions in the blockchain.

Blockchain is a decentralized digital ledger of blocks. Each block in the blockchain is connected with the previous block using sha256 generated hash and knows its **parent block**. Except the genesis block (the first block in the blockchain) every blocks holds a hash of its previous block in its block header, and characterizes the blockchain as an immutable ledger of transactions.

### **2.1.1 Permissioned and permissionless blockchain**

Blockchain can be categorized based on multiple criteria and standards [37]. They are widely classified into two categories: public blockchain and private blockchain.

As the name indicates, public blockchain also known as permissionless blockchain is an open network, in which every node enrolls into the network without verification and authentication from a third party. It is censorship resistant and maintained by all participant nodes in this distributed p2p. All nodes dictates the validity and fate of the block added into the blockchain. Power-of-Work (PoW) consensus protocol is employed to verify and validate the final state of the mined block in the blockchain. In the case of Bitcoin, it is the honest node with the longest chain that spends the most computation power that validates ledger for appending the block [41]. There is a various type of public blockchain based application such as Ethereum, Bitcoin [36] etc.

Unlike public blockchain, private blockchain is restricted and has a limited set of authenticated participants. Nodes requires permission to join the system and participate in the consensus process. Unlike permissionless blockchain, transaction in permission blockchain is only visible to authenticated nodes, and a replicated copy of the shared ledger maintains across all nodes within the network. Hyperledger and Multichain are both platforms for developing a permission blockchain. Hyperledger is an open source blockchain technology that was founded in 2015 by the Linux Foundation in collaboration with participants from various sectors, including IoT, Banking, Healthcare, manufacturing, and other areas [42]. Multichain [43] is another private blockchain platform for developing private blockchain that requires administrator authentication.

### 2.1.2 Blockchain Consensus Models

A decentralized ledger network consists of an unlimited number of nodes, where each node can be represented as an individual user in the distributed network. Nodes in the blockchain have a replicated copy of shared ledger which is maintained collectively in a distributed fashion by updating the order of state transactions. As an open and public network, all nodes are joining freely into the network without going through an authentication process. Because of this, all participant nodes are anonymous and not known to each other. In a distributed system, all nodes might behave differently. Some might be suspicious where others might be faulty or honest. The primary challenge in designing a distributed system is achieving consensus by ensuring that all nodes are willing to agree to the consistency of the state transaction at a global level. When a node transfers a value in a distributed network, all nodes must agree on the validity and authenticity before adding the newly created block into the ledger. Improper selection and implementation of a consensus protocol in blockchain might lead to security vulnerability and expose the network to malicious attacks, in particular to **sybil attacks** [44]. A Sybil attack occurs when a group of malicious entities manipulates the blockchain network by generating several false identities that are used to gain a substantial influence on the validation and the verification of the consensus process [45].

The applicability and efficiency of consensus protocol can be determined based on the three properties [46]:

- **Fault Tolerance:** In the case of a node failure, a consensus protocol needs to have the ability to maintain functionality, and recover from failure to continue operation and participate in the consensus process.
- **Liveness:** The consensus protocol ensures liveness of the distributed system by allowing all non-faulty nodes to reach an agreement on the state of a propagated block.
- **Safety:** The safety of the system is acquired, by allowing all nodes to agree on the validity of the discovered block based on the consensus process. Furthermore, each node keeps an identical updated copy of the shared ledger across the entire network.



According to Fischer, Lynch and Peterson [47], the above mentioned three properties of the consensus protocol are essential and known as the FLP impossible results. It states that no consensus protocol can ensure liveness, safety and fault tolerance in the asynchronous system simultaneously. While fault tolerance is essential in distributed networks to maintain functionality, liveness and safety depend on preference and system requirement. A blockchain consensus protocol has to be resilient to failures that are occurring in a distributed network environment.

Fault tolerance refers to the ability of a distributed system to recover from a breakdown and maintain functionality in case of disaster. There are two types of failure in a distributed system. A **crash failure** is a type of failure that hinders the node from participating in the consensus process and occurs due to hardware and software related problems. The second type of fault is called Byzantine failure. This type of failure may occur when a node misbehaves erratically by sending malicious and contradicting data to other nodes in the distributed system. According to Lamport [48], this type of problem is characterized as **Byzantine General's Problem**. Byzantine General's Problem occurs due to malicious activity from adversary or software bugs, which are misleading the consensus process by sending an ambivalent and suspicious response to other nodes in the distributed network. Consensus protocols allow a distributed network to operate correctly and update securely by achieving consensus in the existence of malicious and Byzantine nodes.

The blockchain has adopted different types of consensus algorithms in validating and appending transaction into the system. Ethereum and Bitcoin, for example, use a Power-of-Work (PoW). However, Ethereum uses a modified version of a PoW algorithm which is ASIC resistant and called Greedy Heaviest Observed Subtree (GHOST) [49]. The protocol is specifically designed to prevent fast transaction confirmation time that resulted in a discovery of stale block. The stale block occurs when two miner nodes discover a block at different time interval due to a delay in the transaction propagation. Besides that, it is also designed to combat the mining centralization that occurs by monopolizing the mining pool hashpower and by investing in highly expensive hardware such as ASIC. Moreover, the GHOST protocol improves the efficiency of the Ethereum PoW algorithm and uses less costly mining hardware (such as CPU and GPU).

Each node has to contribute a certain amount of computation work to validate the transaction. In Bitcoin, to discover a block and add into the blockchain a mining node competes with each other to discover the correct block hash value. The block difficulty is revised and adjusted at every 2016 blocks. The difficulty level is adjusted by the Blockchain protocol, as so to Bitcoin confirm the validity of a transaction at every 10 minutes. In comparison, traditional payment processing such as Visa, PayPal, and MasterCard confirm transaction within a few seconds. For example Visa process 2000 transactions per second on average, and can handle with a maximum peak 65,000 transaction messages per second sec [50]. It clearly shows that there exists an existential scalability gap between the mainstream payment systems and the blockchain technologies [51].

Permission Blockchain such as Hyperledger Fabric provides a modular architecture with a pluggable consensus model. It is designed and developed for an enterprise where each node is enrolled into the ledger by a membership service. The membership service is in charge of registering and issuing a certificate to peer to become part of the network. Currently, Hyperledger Fabric supports two types of consensus models which uses different approach on transaction validation from Bitcoin and Ethereum, Practical Byzantine Fault Tolerance (PBFT) widely used consensus protocol [52] and SIEVE [53] that mostly used for detecting the execution of non-deterministic chaincode. PBFT was adopted the concept of state-machine replica and achieving consensus by agreed upon a sequential order of a transaction to execute.

The consensus protocol in blockchain maintains the consistency of the data recorded in the blockchain by safeguarding the state of decentralized ledger under failure and adversary situations.

### **2.1.3 Public Key Cryptography**

Blockchain uses asymmetric-key cryptography to encrypt transaction and digitally sign messages while communicating with peers [36]. This cryptographic technique uses a pair of keys, a public and a private key [54]. The private key is secret key only known and visible to the account-holder while the public key is generated from the private key and shared publicly in the peer-to-peer network. An Ecliptic Curve Digital Signature

Algorithm (ECDSA) is used to generate the cryptographic keys [36, 55]. The ECDSA uses both cryptographic and hashing algorithms to digitally sign transaction across the blockchain network. Each nodes applies a digital signature to validate the authenticity and integrity of state transactions.

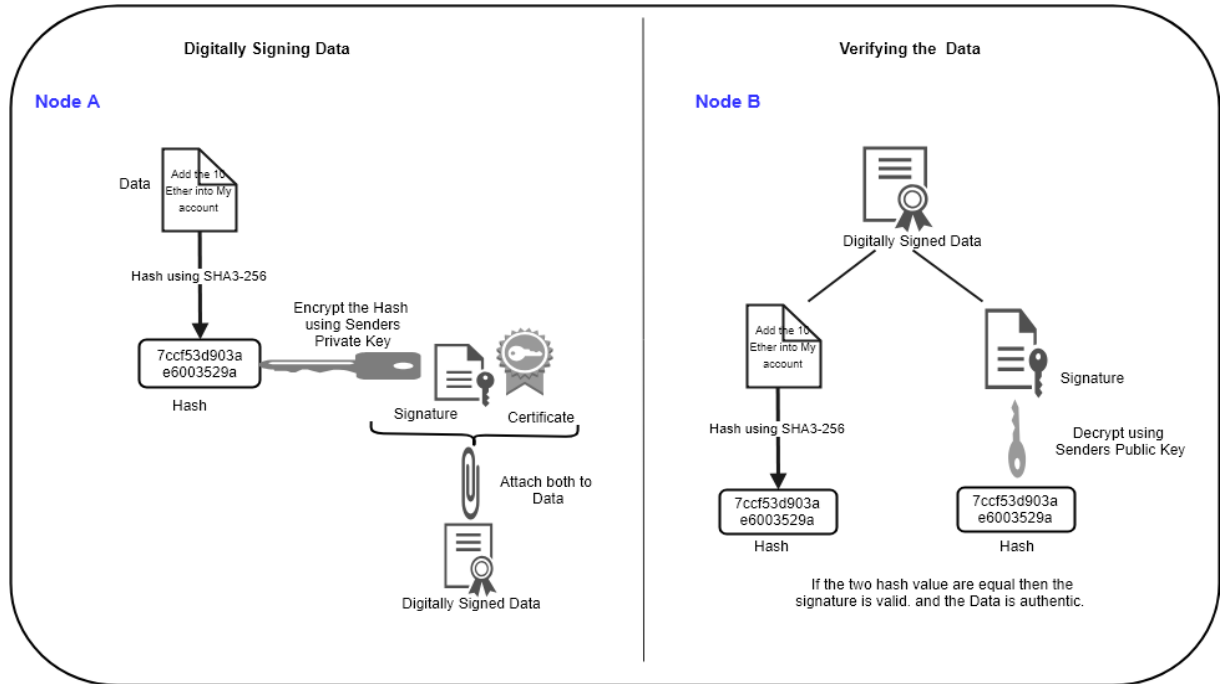


Figure 2.2: Digitally signed data verification and authentication using to Nodes A and B

Each node (patient or healthcare provider in our case) owns two cryptographic keys, the private and the public key. In Blockchain, transactions are encrypted and digitally signed before it was sent to other node or to a contract account in decentralized applications. In the figure 2.2, Node A first hash patient data into a unreadable fixed length characters using SHA256 algorithm and then encrypted the hash data using a his private keys. Both the digitally singed data using private key and the certificate are attached to the original data and sent over the distributed network. The other node on the network Node B verifies the authenticity and integrity of the data by decrypting using Node A's public key. And also Node B hashes the original data that have received from Node A and compares with the decrypted data. If the two matches the the node propagates for further execution to the network.

### 2.1.4 Bitcoin: Peer-to-Peer Blockchain Technology

To understand the concept behind the innovation of Blockchain, we need to go back in time to the first release of a mysterious white paper that was made public in 2008 and known by its title "Bitcoin: a peer-to-peer electronic cash system" [36]. The proposal and idea were to develop a decentralized electronic peer-to-peer digital currency which is cryptographically secure and transparent. A transaction between parties is conducted directly based on consensus instead of relying on a trusted centralized intermediate, such as a financial or a governmental institution.

Bitcoin is the first cryptocurrency and blockchain technology that has used a decentralized consensus algorithm. It was developed to solve a specific problem in the digital payment system, a double spending problem [36]. The problem was under research for many years before Bitcoin was proposed back in 2008. Double spending is the process of transferring digital coins or tokens into more than one node at the same time. In the real, the world double spending problem is solved through a trusted third party, such as a banking or another legal or financial institution. However, in the absence of a central trusted entity, addressing a double spending problem in a decentralized system is a very challenging task. Nakamoto proposed a solution to the double spending problem in the Bitcoin white-paper, by utilizing a p2p decentralized timestamp server. In a distributed system, a timestamp server maintains the validity and generate cryptographic proof of all transactions in the order list of blocks starting from the genesis block up to the latest block [36].

In Bitcoin, the block creation and appending it into the distributed ledger is a very complicated process. Since the network is public, it is vulnerable to security breaches. Moreover, a malicious users might compromise the confidentiality and integrity of the system to misuse personal information for financial and other gains. To mitigate such type of security breaches, Bitcoin utilizes asymmetric-key and cryptographic hashing algorithms to encrypt and digitally sign transactions. As shown in 3.12a when a participant node transfers a coin to another node, a sender node is required to digitally sign its previous hash and encrypted the transaction using the receivers public key. The sender node then broadcasts the digitally signed and encrypted transaction into the Bitcoin network [56]. Every transaction has to be verified before appending into the distributed

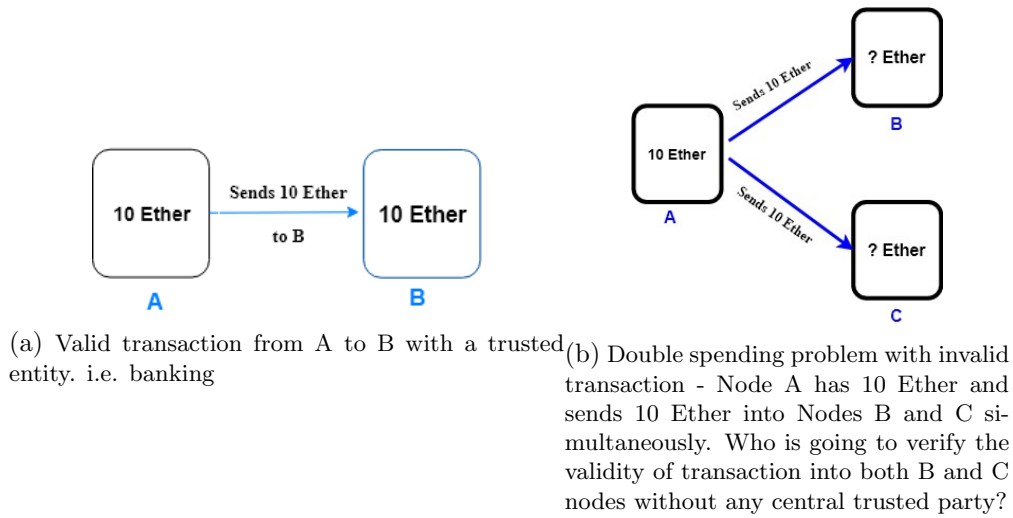


Figure 2.3: Double spending problem

ledger to prevent from being modified. Once broadcasted, the validity and verifiability of the transaction relies on the hand of mining nodes in the network.

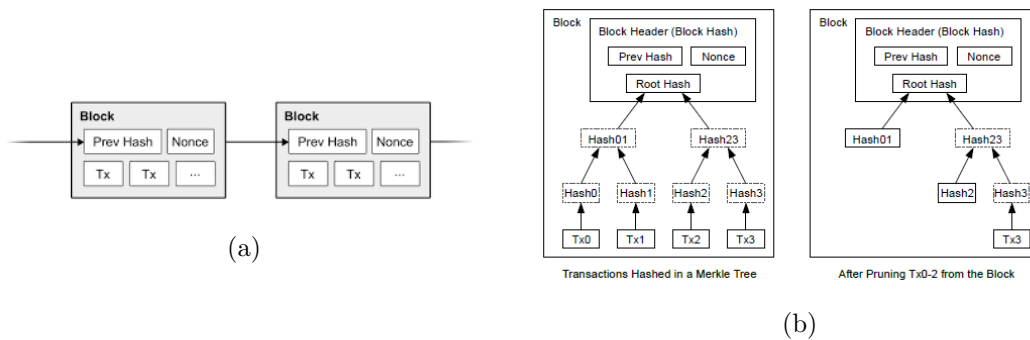


Figure 2.4: Block creation in blockchain

Figures **a** and **b**, show chain of data blocks in the blockchain. Each block except the genesis block includes the hash of the previous block [36].

Mining nodes computes mathematical puzzles to validate transaction using PoW algorithms [36]. PoW uses computational resources to predict the block difficulty by iterating through a random bit combinations to discover the block hash. In Bitcoin a nonce<sup>3</sup> is used to deviate the input data to a cryptographic hash functions and the miner randomly brute forced using the nonce to find a hash smaller than the target hash. In Ethereum nonce is simply a number transaction sent by an external account address or contract

<sup>3</sup>Nonce is a one time number but in cryptography defines as one-time code

addresses. In Bitcoin, finding a valid block that satisfies the target hash difficulty is a random process. When a miner creates a valid block, it is made public to the network. Majority of nodes in the network need to agree on the consensus by verifying the block hash. The block difficulty is determined by how often the block was found and how often the block difficulty was adjusted based on the network hash rate. In Bitcoin, a block is created every **ten minutes** and added into the blockchain. In comparison to a mainstream payment technologies such as Visa and PayPal. Bitcoin supports fewer transactions per minutes. The difficulty of the block is also adjusted based on decreasing or increasing hash rate after every 2016 blocks. The Timestamp<sup>4</sup> on the block is also validated to prevent attacks that attempts to introducing a false timestamps into the block. The mined block is added to the honest node with the longest chain in the network. The longest chain is the correct chain that demanded the most computation resources to create a block [57].

## 2.2 Ethereum

Ethereum is a second generation permissionless blockchain that transforms the way blockchain has perceived in a wider-world as a cryptocurrency and into to a platform for developing smart contracts and Decentralized Application (DApp) [58]. It is a general purpose blockchain that allows an individual users to interact with a distributed ledger using smart contracts and web interface frameworks. Unlike Bitcoin, Ethereum is both a cryptocurrency and programmable blockchain for building a decentralized application. According to the Ethereum white paper [58], the Ethereum kernel was first proposed in 2013 by Vitalik Buterin, and a year later Ethereum was introduced by three of it's founders Vitalik, Gavin Wood, and Jeffery Wilcke. Ethereum is described as the next generation blockchain and as a decentralized application platform [59].

As a platform for decentralized application, Ethereum has its own runtime environment that is completely isolated from the rest of the network called Ethereum Virtual Machine (EVM). It is a part of the Ethereum protocol that controls the internal state of the distributed system. Each node in the Ethereum network runs a local EVM and executes a copies of the same instruction across the entire network. In computational terms,

---

<sup>4</sup>Timestamp: the block timestamp should be at most 1 minute after the previous block

Ethereum is a **Turing complete machine**, because the EVM performs computational tasks that required coding and encoding arbitrary states transitions. It also allows to include programming languages that support an infinite loops [49].

Like any another blockchain technology, Ethereum supports a p2p network and the main function of the p2p network protocol is to conduct a node discovery and routing tasks. As described in the RPLx specification [60] [61] , Ethereum uses the **Kademila protocol**. The Kademila protocol is a distributed hash table that was developed for p2p file sharing. Each node that is connected to the Ethereum network maintains a consistent and updated copy of the distributed ledger database. In an Ethereum p2p network, nodes execute the same instruction across the entire distributed system. The instruction is running inside the EVM to maintain a consensus of all mined blocks in the entire ledger.

To conduct and facilitate a transaction in a trustless network Ethereum uses a token called **Ether**. Ether is cryptocurrency as well a crypto-fuel for running a decentralized applications. Ethereum includes a protocol for mining Ether and deployed smart contracts. Miners are nodes in the Ethereum network equipped with specialized mining software called **Ethash** [62].

Ethereum is a world of connected computers. The Ethereum network is running continuously on nodes all over the globe. Every state transaction in the network requires computation resources to solve block difficulty [63]. Ethereum utilizes the Power-of-Work (PoW) [64] consensus protocol to secure the network and process contracts. As a decentralized network, every node needs to agree on the state of a transaction. Nodes with a unique computational capability compute among each other by utilizing computer resources such as CPU power, memory, and GPU to solve the block difficulty. The nodes that are participating in this mining process are called miners [65]. Each miner who solves the computational puzzle and propagates a block into the distributed ledger is rewarded with a crypto-fuel: Ether. At every 15 seconds, a new block is mined and added into the blockchain. The amount of Ether paid to the mining node includes the gas consumed times the current price per unit of gas and transaction fee assigned by the sender of the transaction.

The gases that are consumed during processing and creating of a block by the winning miner is paid by the sender of each transaction. The PoW algorithm that is currently in use in Ethereum is called **Etash** [62].

### 2.2.1 Ethereum Account Concepts

Ethereum has many features that resemble Bitcoin in adopting core blockchain protocols. It has also developed additional creative innovations and modifications to the blockchain architecture in general. Accounts are the basic units in the Ethereum network in conducting a transactions [66]. EVM monitors the state and values of every account in the system. All state transition in the network holds an associated internal movement of crypto-values and records within Ethereum Virtual Machine (EVM). Ethereum account is represented by a 20-byte hexadecimal address that was derived from the user public-key using ECDSA hashing algorithms. Ethereum accounts contain four fields in it's 20-byte address [64]:

- **Nonce:** defines as the number of transactions sent from the current account address or the number of contract executions done by contract account.
- **Account balance** the number of Ether owned by the current account.
- **Contract code** of the account, if there is any executed code using the account.
- **Storage** : It is empty by default. content...

Ethereum has two types of accounts: external owned accounts (EOA) and contract accounts [66]. The network tracks the state of every account. External owned accounts are accounts that are governed by human users using private key and allow the account holder to perform a transaction by digitally signing the messages using the private key. On the contrary, contract accounts are administered by internal code and activated when receiving a transaction from an external owned accounts. An EOA can create a new contract by deploying a smart contract to the EVM. It is impossible for a contract account to perform a particular operation that requires permission to execute, such as API calls and random number generations. This is due to the nodes need to agree on computational outcomes that demand deterministic execution.



## 2.2.2 Blocks in Ethereum

Blocks in Ethereum have similarities with Bitcoin, but also have some differences. The main difference is that Ethereum blocks contains both the transaction list and the Markle root hash of the entire network state tree. Conversely blocks in Bicoin holds only transaction list. Blocks size in Bitcoin is limited only to 1MB, while Ethereum block size is determined by the gas limit set by the network. Moreover, Block creation in Ethereum takes a few seconds compared to Bitcoin creates a block every 10 minutes.

Each block in the blockchain is linked together using cryptographic hashes, where each verified transaction in the form of data block recorded permanently in the network. Ethereum blocks consists of a set of valid transactions that are linked together and encoded into a Markle Patricia tree [67]. The block holds a hash of it's previous block in the blockchain, there by connecting the two blocks forms a chain of blocks. The cryptographic hashes that are linked all blocks back to the genesis block provides a tamper-resist property to the blockchain [68].

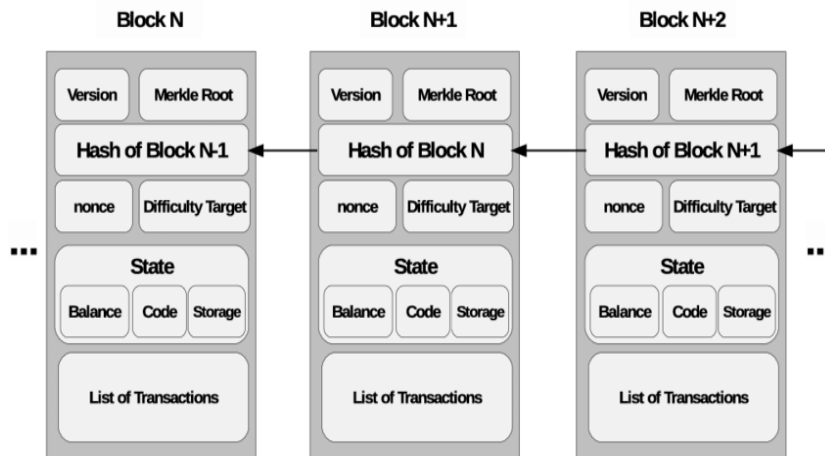


Figure 2.5: Ethereum block structure and block contents [2]

As shown in figure 2.5 a block in the Ethereum consists of a **block header**, a transaction and a list uncle blocks. Uncle blocks is a valid block, and in Bitcoin known as stale blocks [69]. However, in Ethereum unlike Bitcoin miners who generated a valid uncle block are partially incentivized for their computation works. The block header includes a hash of

the previous block, a timestamp i.e the time where the transaction is broadcasted to the network, the mining difficulty and a nonce.

### 2.2.3 Gas, Ether

Each operation in the Ethereum network has operational costs. The operations or OP-CODE are performed inside the EVM. Operations include ADD - adding two integers together, and CREATE - creating a new contract from a supplied smart contracts. Each of these operations has a unit of measurement associated with it called **gas** [64].

Gas measures how much work a miner spent on a executing a transaction or deploying a contract in Ethereum network. For instance, the minimum gas limit for a standard value transfer costs 21000 gases [70]. Every state transaction in EVM demands a certain amount of gas. Operations that require a high degree of computational resources (high use of memory and CPU power) such as a high hashrate costs more gas than other operations that demands less computational resources. To prevent any malicious activities by a miner and a smart contract, the EVM sets a gas limit for processing transactions and the deployment of a contract into the blockchain. An operation might run out of gas if the consumption exceeded above the specified gas limit. In this condition, the miner halts executing the operation and transaction is rolled back to its original state. Setting the gas limit prevents the network from an adversary which attempts to manipulate the system by launching Denial-of-Service Attack (DoS).

Gas and Ether are two different terms, but they are related to each other. A state transaction in the EVM has a constant gas limit, and the cost for a unit of gas fluctuates depending on market conditions. Gas price indicates how much **Ether** the user has to pay per unit of gas. The unit for measuring the gas price is specified in Gwei per unit of Gas. Ether is the cryptocurrency used in Ethereum to pay mining rewards [64]. Nodes in the network earns Ether by mining a new block or buying Ether from other nodes.

### 2.2.4 Mining

In a trustless distributed ledger, the security model depends highly on the consensus process based on Power-of-Work (PoW) algorithms. The PoW enables every node to

utilize a computational resource to prove that a transaction is valid by creating a new block with the highest possible difficulty, and in return, the node is rewarded with a token. The process is called mining and the node that participate in the consensus process is called a miner. The primary purpose of mining is to ensure the presence of a consistent copy of the block history across all nodes and to makes the distributed ledger computationally impossible to modify [71].

Mining in Ethereum is much the same as in Bitcoin and other blockchain technologies. Bitcoin used a Nakamoto consensus [72] PoW algorithm to solve the computational puzzle, while Ethereum used it's own modified version of Dagger-Hashimoto consensus model called Ethash [73]. To add a new block into Bitcoin, a miner node has to perform a certain amount of computational work using PoW and solves the problem by finding the hash value which is less the difficulty set by the network. The block difficulty set by the current Bitcoin protocol is creating a new block into the distributed ledger every 10 minutes. The miner node which successfully hashes the transaction incentives with rewards. The distributed nature of the system sometimes allows more than two honest nodes to solves the hash difficulties. The winning nodes propagate the proposed block into the peer-to-peer network for verification. However, the node with the longest chain has spent the most PoW and recognized as valid to included the block into the blockchain.

### **2.2.5 Smart Contract**

The concept smart contract was first introduced by cryptographer Nick Szabo [74]. The basic idea is to change the way contracts performed, away from mimicking the traditional concept of a contract in physical world by embedding the concept of a smart contract into the hardware and the software. Szabo described smart contract as a type of a cryptographic contract in which verification and contractual process are performed through self-enforced scripting codes [75].

Smart contract is a computer program implemented based on predefined rules and deployed into Ethereum as an ordinary transaction. The contract is executed without any third party intermediaries. Smart contract features allows to specify regulations and

policies on how to handle and administer transactions. The policy is structured to contain a set of rules that can govern a particular tasks. For example it deals on how to deal with access permission with both patient and provider. Ethereum is among the few blockchain technologies that offer a fully capable smart contract features [76].

As mentioned in the previous section, a smart contract contains codes and provides a particular function to other contracts. Basically, smart contract is a way of inserting programming logic into the Ethereum blockchain. It is governed by it's contract accounts. The contract account contains programs that trigger the EVM to execute an instruction such as storing state values of another contracts, and communicating with other contracts using internal messaging enable transactions. The Ethereum contract is initiated by external owned accounts and deployed into the Ethereum network, and it becomes operational forever until it is destroyed. In Ethereum, smart contracts are written using a **Solidity** scripting language which has a similar syntax to Javascript. Once the code is ready, it deployed into the EVM using Truffle [77], Mist or Ethereum wallet [78].

### 2.2.6 Solidity - Ethereum smart contract language

Solidity is the main smart contract language on Ethereum as well as on other private blockchain running on Ethereum. Solidity is an high-level contract-oriented programming language for Ethereum smart contracts [79]. It was derived and influenced by python, C++ and Javascript-and it's syntax is similar to that of JavaScript. Solidity is a statically-typed programming language and compiled into bytecode that executes in Ethereum Virtual Machine (EVM). EVM is a runtime environment for executing smart contracts. It is completely isolated and code running inside EVM has no access outside the perimeter such as to the file system, network and processors.

Like traditional programming languages, solidity as the code shown in figure 2.6 supports different data-types such as unsigned integer, integer, bytes, address and string . Solidity data type **address** is a unique from other data types and is using to represent the Ethereum accounts address. It is 20 **byte** long and derived from the public key which controls the account. Ethereum accounts uses to identifies users in the Ethereum network: the information about the value and state.

```

1
2  pragma solidity ^0.4.10;
3
4
5  contract GroupEtherapy {
6
7      address owner;
8      mapping (address => Patient) patients;
9      mapping (address => Provider) providers;
10
11     struct Provider {
12         uint providerId;
13         address adr;
14         bytes32 providerName;
15         bytes32 contactAddress;
16         bytes32 certificateNo;
17     }
18 }

```

Figure 2.6: Solidity code snippet taking from our DApp implementation

The first line in the code shows that the programming language used, solidity with a version number 0.4.10 or newer version. The keyword `pragma` is a special instruction for the compiler on how to handle the code. Solidity also supports multiple inheritance.

## 2.3 IOTA

Blockchain is actively evolving and has tremendous potential in providing auditable, transparent and secure transaction processing in a distributed environment. It does however have drawbacks for use as a platform for Internet of Things (IoT) as well as for use in other micro-payments. Some of the notable drawbacks of blockchain are high transaction fees, scalability problems and it's high demand for computational resources for mining blocks [35]. Internet of Things (IoT) is one of the areas that is mostly affected by the limitation of blockchain. Bitcoin and other cryptocurrency are not suitable for micro-payment of less valued transactions. For example, Bitcoin charges 0.68 US dollar [80] for all types of transactions, even for transfer of values less than the transaction payment specified. Millions of IoT devices are currently in use on a global scale, and

they require a consistent platform that can provide them with ubiquitous computing that overcomes the limitation and problems mentioned above. The IOTA project was initiated from a lack of some essential solutions in blockchain in particularly transaction settlements for IoT such as micropayment for machine-to-machine (m2m) services [81].

IOTA is a cryptocurrency specifically designed for IoT devices. **Tangle** was the main backbone for innovation of IOTA. Tangle is a data structure and distributed ledger database based on Directed Acyclic Graph (DAG) [35]. Unlike blockchain, IOTA is blockless and does not have separate miners for mining transactions. Each participant node is seen as an independent miner and has similar principles as a distributed database. It supports p2p network and consensus protocols to validate transaction in the Tangle network [35]. A transaction in Tangle represented as a vertex. When a new transaction arrives, Tangle forces the sender node to validate two previous transactions and decides on the new transaction. Nodes validates two transaction before getting approved their own transaction, the network uses the sender computational resources to validate in the two-for-one fashion.

Instead of a small set, all nodes in the IOTA network are responsible for overall consensus and approval of a transactions. Consensus in IOTA is not a separate entity as in blockchain, but an integrated part of the consensus process. All nodes in the network must agree on (in consensus) a newly created transaction, and this is done without involving a miner. As a result, the use of computational resources for transaction settlements is performed without any transaction fee. IOTA as opposed to the use of miners and payment for their services in other blockchain technologies.

IOTA is mainly designed for machine economy, privacy and security vulnerability are the primary concern in IoT and other sensor devices [82]. In traditional cryptocurrencies such as Bitcoin and Ethereum, network security is highly dependent on digital signatures based on elliptic-curve cryptography techniques. However, the development of quantum computers changes the ability of cryptographic algorithms to defend users data against security breaches [83]. Most of these cryptographic techniques are known to be vulnerable to quantum computing attacks. To combat against quantum computer-related security threats, cryptographers has developed an alternative hash-based digital signature that has shown promising results against quantum computing related attacks. IOTA as a cryptocurrency has adopted a **Winternitz** hash-based digital signature for

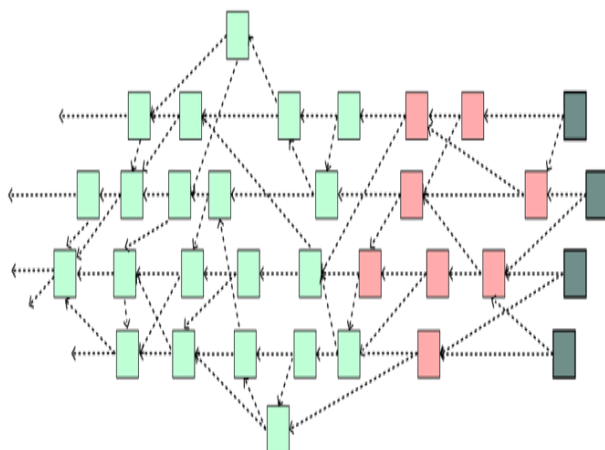


Figure 2.7: The blue transaction on Tangle are fully confirmed and the nodes of newly added transaction indirectly participate on validation of transactions. For example peers for red and green transactions have been used to validate the blue transaction. The Red are on the process of validation while the blue are newly added transaction, unconfirmed and are called tips [3]

singing and hashing transactions. It is a one-time signature algorithm that has properties of resilience to quantum computing related attacks. The algorithm generates only a single-use public/private key pairs that can only be applied to single transaction. Winternitz based signatures are an integral part of the IOTA protocol, and it is characterized as resistant to quantum computational attacks [35].

Even though IOTA is mainly focusing on machine economy, it can also applied in other areas. Currently, both research and case studies are running on Mobility, Smart Energy, Smart home an city, etc.

## 2.4 Hyperledger Fabric

Hyperledger is an enterprise-scale blockchain project which is developed by the Linux Foundation in collaboration with partners from within healthcare, banking, supply-chain management and IoT. **Hyperledger Fabric** is the one among many other Hyperledger projects currently under continuous development by IBM and the Linux Foundation. It is an open source a blockchain platform for developing a permission decentralized ledger technology using smart contracts called chaincode. Fiber is the first permission blockchain technology including a smart-contract language that supports a general-

purpose programming language such as Java, Node.js and Go [84]. Most existing smart contract languages that are used in developing decentralized applications today support mainly domain-specific languages, designed explicitly for the cryptocurrency [85].

Hyperledger Fabric (referred to as Fabric in the following) can be classified as a permission blockchain platform, with a modular and an extensible architectural design which can be used for a wide range of sectors such as healthcare, banking, and supply-chain management. It can easily be customized and extended by allowing the implementation of additional pluggable features such as a consensus protocols, identity management and a transaction functions [86].

In a permission blockchain, all participating nodes are identified and known to the network. Where transactions have to be approved through the use of consensus protocol. Fabric supports pluggable consensus protocols that efficiently run customizable trust models which fit particular businesses rules. Most consensus protocol implementations across blockchain technology vary from one another depending on their business rules. For example, Bitcoin and Ethereum support only proof-of-work consensus protocol. Fabric allows the for the network to choose the consensus protocol that best suits to the need of the existing participants. There are several different types of consensus protocols that are in use today. The implementation ranges from lottery-based consensus protocols such as Proof of Elapsed Time (PoET) and Power-of-Work (PoW) to voting-based methods including Redundant Byzantine Fault Tolerance (RBFT) [4]. Most of the existing protocols based on consensus algorithms have limitations in solving some of the challenges including performance, scalability, and efficiency. To overcome these problems, Fabric source code currently provides three different consensus algorithms implementations: Simple Byzantine Fault Tolerance (SBFT), SOLO, and Kafka. Besides, participant nodes can also choose the consensus protocols that best fit their business models.

A consensus in Fabric covers three different levels: transaction endorsement, ordering and validation. Endorsement policy is used to define the task of the individual peers that are participating in verification of a proposed transaction. The transaction proposal is sent to the peers for endorsement, in which it is executed and verified using a chaincode. Once the majority of the peers have agreed on the outcome of the transaction, the ordering service specifies the order policy and delivers the blocks to the committing



peers. The committing peers then validate the blocks and append them into the Fabric ledger.

Hyperledger Fabric has three main features which provides a unique functionality in developing an enterprise-grade distributed application: Membership service, Chaincode and Ordering service (known as consensus process).

#### **2.4.1 Membership Service Providers (MSP)**

The main difference between a Hyperledger Fabric and other blockchain technologies is that it is a private and permissioned based network. Fabric does not allow unidentified nodes to enroll and participate in the network. Fabric has implemented a membership service in order to authenticate, authorize and enforce security on a new member [87]. The service is responsible for securing the network by enforcing all nodes to acquire an enrollment certificate from Certificate Authority (CA). The digital certificate is designed based on Public Key Infrastructure (PKI), and CA is responsible for issuing certificates to all nodes that are allowed to enroll into the network. The membership service governs the entire network and controls the access level of the participant nodes based on their digital certificates.

#### **2.4.2 Fabric Smart Contract : Chaincode**

Smart contract in Fabric is known as Chaincode. It is a set of computer programs that contain instructions used to execute, validate and modify state transactions. The chaincode is event-driven, and is implemented using general-purpose programming languages such as Go, Java, and Node.js. Fabric Chaincode has a state, and replicated copies run on a shared ledger across the network[88].

#### **2.4.3 Consensus and Ordering Policy**

In Fabric, as shown in figure 2.8 verifying and adding a block into the ledger is performed through three different and distinct steps: Endorsement, Ordering, and Validation.

- **Endorsement Service:** is directed by a policy to decide upon a purposed transaction, in which a majority participant endorse the transaction (has got m out of n signatures).
- **Ordering Service:** receives the endorsed transaction and agrees to verify the order to be included in the ledger. Ordering service in Bitcoin occurs through PoW, but Fabric has a modular design and chooses an ordering service protocol that best suits to the system such as SBFT, Apache Kafka, SOLO.
- **Validation Phase:** peers that are responsible for verifying the correctness of the block including double spending and endorsement policy.

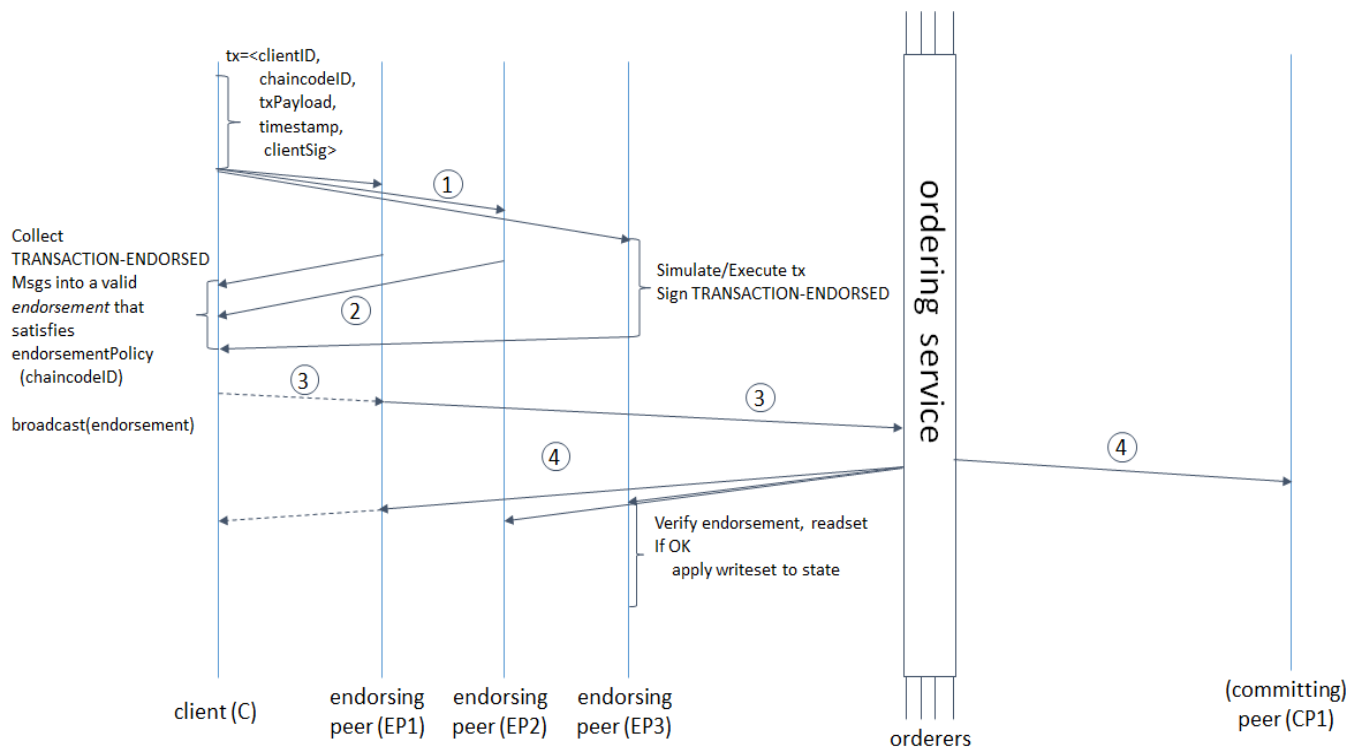


Figure 2.8: Transaction Flow diagram for Hyperledger Fabric based consensus process[4]

Hyperledger Fabric as a modular and flexible architecture it supports a pluggable consensus for all the three phases described above. Every Decentralized Application (DApp) may choose a different plugin for endorsement, ordering, and validation depending on their needs.

## 2.5 Blockchain in Healthcare: Previous Works

Currently, Blockchain is widely applied and used in cryptocurrency with a total market value around 450 billion dollars as of February 2018 [89]. Bitcoin and Ethereum are considered to be the leading and most well-known permissionless blockchain technologies. The technology has a lot of potentials that extends far beyond cryptocurrency, and it clearly shows how to achieve ownership of digital assets in decentralized and transparent manners. Blockchain is continuously evolving outside of financial domains and is met with great interest from areas such as healthcare [90], supply-chain [91] management, digital identity [92], IoT [35] and other private and governmental organizations across the globe.

The **MedRec** is a decentralized EHR that was developed by MIT Media Lab in cooperation with Beth Israel Deaconess Medical Center [93]. The blockchain enabled MedRec is designed to improve a patient-provider relationship that allows for sharing of a medical record. MedRec allows patients to have full control over their medical records and also provides them with functionality for permission management. It was the first practical DApp that was developed using the Ethereum blockchain. MedRec assists patients in assigning specific access rights and permissions for a particular medical record. For instance, a patient can authorize sharing of a medical record that was generated by a healthcare provider to other healthcare providers. A new provider receives an automated notification and can confirm and verify the authenticity of the record before accepting or rejecting the record. MedRec addresses four major problems that have faced healthcare such as system interoperability, patient agency, slow access to healthcare data and fragmented data.

As a result of a continuous cyber-attack on government infrastructure in 2007 [94], Estonia government in collaboration with **Guardtime** launched a project to develop a digital healthcare platform based on blockchain technology [95]. The concept was to secure the patient medical information and prevent the medical system from adversary. The system facilitates improved access to medical data for providers while guaranteeing patient privacy, data integrity, and confidentiality. The Estonian blockchain based EHR provides the citizens with the access to a transparent and auditable log of their medical data. In 2017, a US startup company **Gem** launched an Ethereum blockchain based

healthcare infrastructure, called the **Gem Health Network**. It's intention is to develop a shared patient-centric healthcare infrastructure that includes both researchers, individual patients and healthcare providers [96]. Gem Health Network seeks to connect all healthcare providers and industries by developing a shared ecosystem that can help to establish a trusted data exchange pipeline.

Immutable and tamper-proof properties of blockchain would allow patients easily to share and exchange their medical data for research purposes and in return might get monetary incentives in the form of digital token. For example, **healthbank** [96], a Swiss-based digital health data provider has developed a new user-owned data exchange platform . The platform gives the users the complete rights and ownership of the in own medical data that is stored on the **healthbank** platform. Data generated by individual patients during medical consultation, from laboratory reports, from their use wearable devices, and online-screening applications (e.g., heart rate, blood pressure, sleep pattern, eating habit, and etc.) can be securely stored in the blockchain [97]. The ownership and management of stored data lie in the hand of patients. The system does not only provide it's users with storage, but also with an intelligent consent management system. Users can review research details and monetary incentives in the platform and make their medical records, or parts of them, available for researches.

## 2.6 Chapter Summary

Distributed ledger technology provides a transparent and tamper-resistant environment for the transfer of data and records across untrusted decentralized Peer-to-Peer networks. All distributed ledger technologies have a common property that characterize them all together such as transparency, traceability, tamper-proof and cryptographically secure. Bitcoin was the first decentralized PoW consensus based cryptocurrency, and it is specifically designed for financial transaction. On the other hand, IOTA is a DAG based DLT developed for IoT. Hyperledger Fabric and Ethereum have however, extended the application of distributed ledger technologies into areas other than cryptocurrency. Hyperledger Fabric is a private/consortium blockchain used for developing Enterprise applications. Ethereum and Hyperledger supports programming languages with loops

The performance and scalability of distributed ledger technologies depends on the ability of the system to operate in public or private networks. Permission blockchain such as Hyperledger Fabric executes transactions in parallel to increase throughput, while Ethereum and Bitcoin validates transactions in sequential order due to computational works. Furthermore, the block size and the gas limit directly affect the security and scalability of Bitcoin and Ethereum. In Bitcoin, the block size is set to 1MB by the system. Miners in Ethereum and Bitcoin invalidates transaction with insufficient gas and bigger transaction size to prevent the system from Denial-of-Service Attack (DoS) attacks. Conversely, IOTA and Hyperledger Fabric are independent of miners and incentives [98] [99].

The security and privacy setting in Bitcoin, Ethereum, Hyperledger Fabric and IOTA differs based on the consensus model and cryptographic techniques that are utilized to validate and digital sign transaction. Both Bitcoin and Ethereum adopt the PoW consensus protocol to validate transaction on their permissionless p2p network. Blocks that are created using PoW protocols become computational impossible to modify. The protocol works best for open and permissionless network where race for computational reward (cryptocurrency) boosts security on the network. In terms of privacy, private and permissioned blockcahin has a better privacy preserving methods comparing to public blockchain. In Hyperledger Fabric for example user are authenticated to enroll to the system and also defined channels using chaincode policy to allow only member of the channel to execute transactions with in the channel. IOTA is developed for IoT and uses Tangle that is a DAG based database. Tangle improves the security of the system by employing single use public cryptographic hashing algorithms.

Ethereum is primary programming blockchain that support Turing complete programming languages in its architecture. The smart contract is a self-executed set of programming instructions that contains rules and directives about a specific contract agreements. Using the Solidity smart contract Ethereum supports the development of DApp. Conversely, Hyperledger is also the first permission blockchain that adopts a smart contract called chaincode. Unlike the Ethereum smart contract, the chaincode can be written in general purpose programming language such as Go, Java and NodeJS. However, In IOTA and Bitcoin smart contract is not a part of their architecture models.

Table 2.1 gives an overview of the three different distributed ledger technologies that we have presented in this chapter. The summary mainly focused on vital underlying protocols and concepts of distributed ledger technology (DLT).

Terms used	Bitcoin	Ethereum	Hyperledger Fabric	IOTA
<b>Platform type</b>	Specifically designed for CryptoCurrency	Generic blockchain platform	Modular architecture	Specialized distributed ledger technology (DLT) designed for Internet of Things (IoT) and machine-to-machine (m2m) transaction
<b>Mode of Operation</b>	Permissionless: only Public	Permissionless: Public and also support private network.	Permission private network. It requires authentication from Membership service to enroll the network.	Permissionless open network.
<b>Consensus Model</b>	Power-of-Work	Power-of-Work and also support Power-of-Authority on private network	Pluggable consensus model: PBFT, Apache Kafka	Directed Acyclic Graph (DAG) based consensus algorithm called Tangle.
<b>Cryptographic and Hash Algorithms</b>	Ecliptic Curve Digital Signature Algorithm and SHA256	Ecliptic Curve Digital Signature Algorithm and Keccak256 used to generate account from public key	Public Key Infrastructure (PKI) based public key cryptography and Certificate Authority (CA) .	Winternitz Signature Algorithm which is One time and quantum resistant
<b>Cryptocurrency</b>	Bitcoin	Ether	not cryptocurrency, but can implement tokens via chaincode.	IOTA
<b>Transaction Fee</b>	Yes, miner rewards Bitcoin	Requires Fee : it includes computational fee (compensation for gas consumed) plus transaction fee	No transaction fee	No fee required
<b>Miner</b>	Yes, highly depend on miners	Required miners to validate transaction	No miner	No mining but nodes validates two other transaction in order to get validated their own transactions.
<b>Smart Contract</b>	Scripts, Non-Turing complete language.	Support smart contract codes. (e.g. Solidity, Serpant)	Chaincode :general purpose programming language (e.g. Go, Java, Node.js)	Doesn't support Smart contracts

Table 2.1: Comparising between the threes decentralized ledger technology

## Chapter 3

# mHealth-BlockC: Design and Implementation

This chapter mainly focuses on the design and the implementation of a prototype for patient-centric record management based on Ethereum blockchain technologies: smart-contract and Decentralized Application (DApp). The prototype has been named mHealth-BlockC.

The design and implementation of patient managed medical records is based on Ethereum smart contract. The main idea is to create an environment that can be further tested and investigated in order to provide a better understanding of the blockchain and in particular Ethereum's potential in developing a secure auditable and tamper-proof decentralized Electronic Health Record (EHR).

### 3.1 Motivation for Developing a Prototype for Mental Health (mHealth)

Online therapy, also referred to as cybertherapy, or Internet-based treatment, can provide support for people with mental health problems. Within mHealth-BlockC, online therapy which includes online peer support, single-session online consultation, mental health screening and testing through online discussion forums, instant messaging and other synchronous and asynchronous communications [100, 101]. Such therapy reaches unprecedented number of mental health patients with information. Recent studies claim that the Internet is the primary place for many people with mental health problems to look for consultation and support [102].

Advancement in technology and introduction of Electronic Health Records has improved the quality of care and the safety of patients by facilitating healthcare delivery, reducing waiting time, keeping a better medical history and allow for patients to actively participate in the decision-making process [103]. Patients with mental illness are also benefited from the use of technology by allowing them to get better access to medical care and services using Internet-based treatments. Traditional face-to-face treatment

has some drawbacks, as a large portion of patients with mental illness do not seek frequent help from professionals. This is due to high treatment costs, fear of stigmatization and work-related issues [104, 18]. Moreover, negative stereotypes which are portrayed by both the healthcare providers and the communities prompts patients to avoid care. The inconvenience of face-to-face treatment forces patients to look for an alternative treatment method to alleviate their fear and mental health issues. Internet-based therapy and peer-support are said to be the best suites to solve some of the challenges by allowing patients to express their feeling freely behind a computer screen, communicate anonymously and get online-support from peers with the same problem. Various types of synchronous and asynchronous web-based mental health (e-therapy, web-based mental health screening) tools offer consumers and providers with a unique opportunity to engage and communicate effectively about the patients (the consumers) mental health conditions.

Despite it's importance, Internet-based support and therapy are web-based with the risk of centralized systems, usually alter over insecure and unencrypted communication channels, patient privacy being compromised and where identity is revealed [105]. Furthermore, current systems do not provide the patients with the ability to maintain ownership and access control over their own data. To preserve privacy in a transparent and auditable manner, blockchain as discussed in chapter two has a quite potential as a technology to develop DApps. As a distributed Peer-to-Peer network, it has an extensive capability in providing patients with the ability to maintain ownership, control access permission, communicate anonymously, and to keep an auditable-log of medical information.

Various different types of distributed ledger are in use today in developing DApps, and as discussed in chapter two table 2.1, the two of four presented DLTs have promising potentials in developing distributed applications. However, the third DLT, IOTA support smart contract not as part of its core architecture, and it is not viable for developing Decentralized Application (DApp)s. The requirements for the design of our prototype are focused mainly on the privacy and the security protocols of the underlying blockchain technology that can offers to the implementation of coherent DApp for mental health. Patients with mental health prefers to access online treatment services such as peer-support tools, online psychological testing and screening anonymously without revealing



their identities. Consequently, our system has to allow for each patient to maintain ownership and transparent access to all personal information without interference from an untrusted third party. In case of any malicious activity and modification to patient personal data the system must notify the owner by triggering an automatic notification message, in such a way that both the patient and other stakeholders might use the auditable log of block hashes to verify against any change on the replicated copy on the network.

To avoid monitoring and access restriction to the online mental health services, patients and providers enroll into the system without authentication from a central third party. In return a patient might be represented using a pseudonymous account. It encourages both the patient and the provider to build trust on the technology in which it allows to communicate confidently and where the patient can express their psychological problems freely without fear of discrimination from the community in general. Individual users can also decide on the fate of their personal medical informations whether they will grant or deny access to these depending on their treatment preferences. Access can also be restricted to a specific record with a limited range of time. For example, a provider might be granted access for a patient's personal medical data for the duration of the treatment time, where the access is invoked automatically afterwards.

Technology selection for our prototype is determined based in the table in chapter two table 2.1, and the maturity of the blockchain technologies during our project startup. Ethereum blockchain and underlying smart contract languages are chosen to develop and implement our prototype. Hyperledger Fabric, for example released in July 2017, was not part of the initial requirement gathering for our thesis. However, we have studied this technology later and is used the protocols and algorithms for comparison to our finding using the Ethereum blockchain technology.

## **3.2 Choice of Technology**

There are various types of distributed ledger technologies for developing a smart contract enabled Decentralized Application, ranging from permission to permissionless, and also from those using a pluggable consensus model such as Hyperledger and to the Bitcoin and the Ethereum with embedded consensus protocols. They do, however, all have some

common characteristics such as being based on distributed ledger technologies which are decentralized with peer-to-peer network architectures.

To develop a DApp in a distributed trustless network, we might need to genuinely assess the pros and cons of using a particular distributed ledger technology (DLT) to design an application that deals with sensitive personal medical information. There are specific criteria that have to be considered before applying a DLT technology, such as consensus models, privacy, security, ecosystem and the documentation that gives insight and a solid understanding of the technologies, underlying protocols, and tools involved.

Being the first cryptocurrency, Bitcoin has limited programming supports. Furthermore, the technology does not support programming languages with a loops. This allows Bitcoin to be deterministic and prevents against attacks that scripting language that supports an infinite loops in their implementation. Denial-of-Service Attack (DoS) is one such type of attack where an intruder launches an attack from multiple distributed sites using code that includes an endless loop in their implementation. Consequently, Bitcoin is not suitable for developing medical applications that make use of iterations and loops to store and retrieve a data by applying certain data structures. In contrary, Ethereum is a second generation blockchain that enables the EVM that supports Turing complete scripting languages and smart contracts for developing DApps.

As discussed in chapter two section 2.2, Ethereum is a programmable blockchain and designed to develop decentralized applications using smart contracts and have an Ethereum Virtual Machine (EVM) that supports Turing complete programming languages. Front-end web development libraries and frameworks can also interact with a deployed smart contracts in the EVM using Web3.js library which is explicitly designed for Ethereum based DApps. The tools and frameworks makes Ethereum more preferable over the other DLTs for developing DApps. Solidity is one of Ethereum's smart contract languages which is used for developing contracts that deploys in the EVM. It is a high-level programming language similar to JavaScript, easy to write and with extensive documentation. Moreover, the Truffle development framework is also used to speed up the implementation and integration process. Truffle is designed explicitly for Ethereum smart contracts. Also it supports the development process by facilitating for a fast and automated compilation, deployment, and testing process [77].

As mentioned in the previous chapter, IOTA is designed to overcome the blockchain scalability limitation problem when using IoT devices. IOTA uses a Tangle database which is a DAG based data structure for processing and storing validated transactions. The IOTA architecture is designed to facilitate micro-payments between IoT devices. However, the IOTA does not support smart contract as a core feature in its architecture [106].

### **3.3 User Stories and Requirements**

The practical part of this thesis is the design and the development of a prototype, mHealth-BlockC based on blockchain technology, that has been developed to address known concerns in current medical data management and to find a solution for sharing and managing personal medical data in a secure and privacy-preserving manner. A key issue has been to allow for the patients to have full control over their medical data, which they can share with various healthcare providers when needed. The project has focused on data privacy, security, and traceability of patient data in a transparent and auditable manner.

The design and implementation described are bounded to categories of stakeholders which have different interests in the data security and privacy: the patient or owner of the data, the healthcare provider, the researchers and various governmental organizations. All stakeholders are connected in a p2p secure and distributed environment. A patient is a private person and the sole owner of his personal medical data. The primary source of the medical data is from what is generated in patients online consultations. Depending on collected records and results, a patient may request a further traditional face-to-face consultation by visiting a therapist, or due to fear of discrimination he might request Internet-based therapy. To facilitate access permission for the treatment sites, the patient adds a personal number and share generated medical data with providers for further treatment. The whole concept is that the system gives the patient the ability to create, update and modify the medical data without external interference. Besides, a patient that has privacy concerns in doing a face-to-face consultation, might prefer to use online peer-support groups to discuss and raise issues related to his medical conditions.

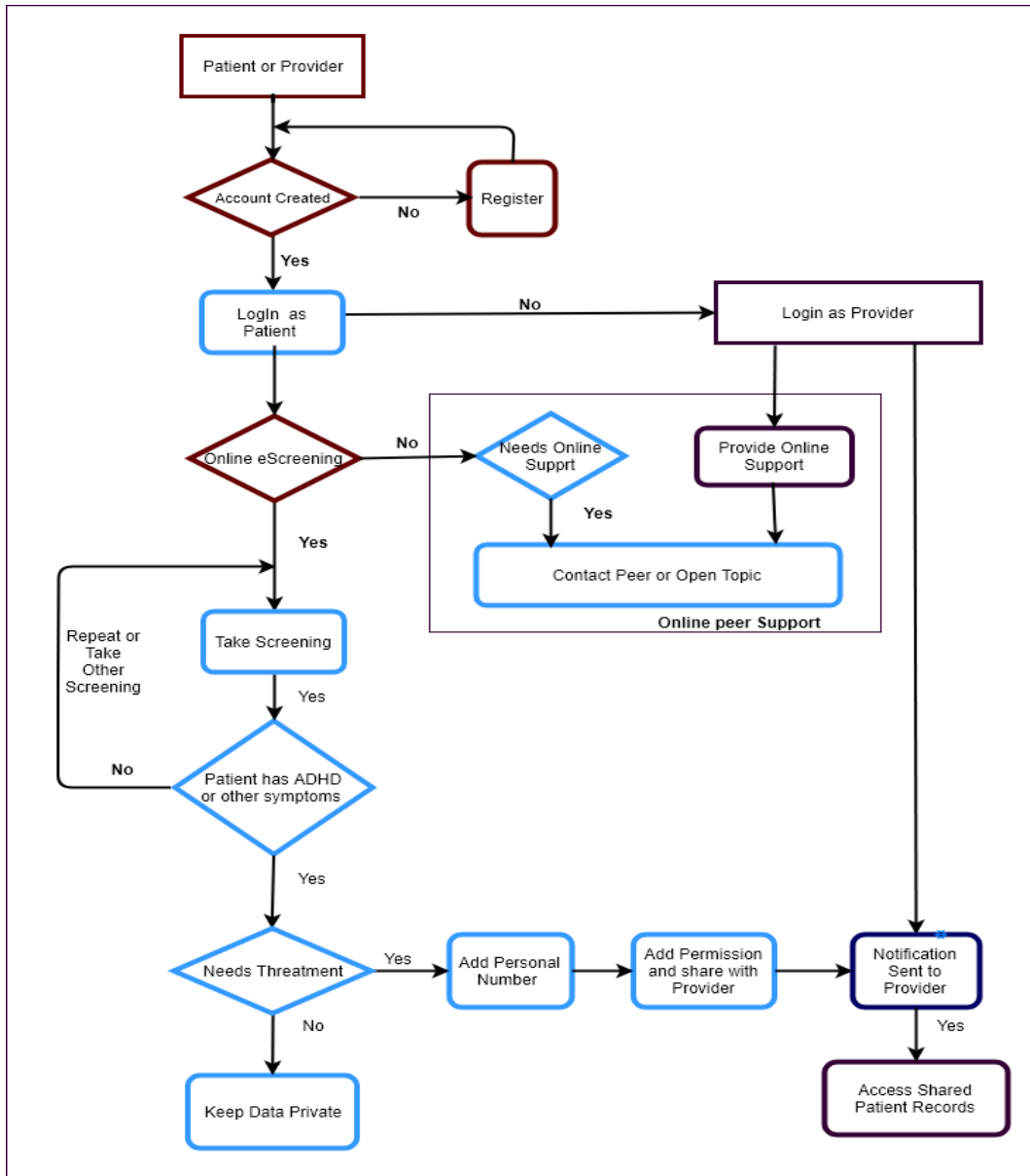


Figure 3.1: The mHealth-BlockC Data flow diagram

Let's take an example, Alice is a female suffering from a mental health related problems. She looks for psychological support and training in order to rehabilitate herself. However, getting support and training using traditional face-to-face makes her uncomfortable due to fears of being stigmatized from healthcare personals and colleagues. Instead, she looks for an alternative way that allows her to conduct training and therapy anony-

mously and securely. The possibility for online web-based mental health training and therapy favors her, and she thinks that such therapy suits better to her needs. She subscribes and starts online training by directly communicating with a therapist and online-support groups. Her chosen online web-based mental health training runs on a centralized database architecture, and all communicated information during training and therapy as well as all her profile data are stored on a central system. The database is maintained and managed by a central trusted party. Eventhough the online web-based support and training allows it's patient users to enjoy some sort of physical anonymity compared to traditional face-to-face, there are still some concerns such as data privacy, confidentiality and accessibility in case of system failure. Furthermore, her data might be modified, shared and exchanged without her consent. Those potential privacy and security breaches undermines her trust in the technology. Ethereum blockchain which is a distributed database, responds to and mitigates these identified issues regarding privacy, confidentiality and integrity of sensitive patient medical data.

### 3.4 mHealth-BlockC Architectural Overview

Figure 3.2 presents a simplified architectural overview of mHealth-BlockC an Ethereum based distributed ledger application, a prototype which has been designed and implemented in the domain of online mHealth therapy, to find out the potential of Ethereum and smart contracts as technologies to provide a platform for a secure, transparent and auditable-log of their medical history. As shown in the figure 3.2, the architecture includes of a patient interface, a provider interface, a local data storage, a smart contract deployed on the Ethereum network, Medical Record API's, a private Ethereum client and a web-interface for interacting with the blockchain using web3.js frameworks. Moreover both the provider and the patient might exchange and share personal medical data with external actors, such as a general practitioner(GP), researchers, hospitals and other governmental organizations. In some occasions the researchers and the medical personal might be the same persons.

In our prototype, the Ethereum implementation addresses three major issues raised in the introduction:

- Lack of data anonymization in using and accessing medical services,

- Lack of transparency in users medical history.
- Sharing of medical data without patient consent.

These are the main challenges that are investigated to address by building a prototype on a Ethereum private network. The blockchain technology might ensure the patients privacy and security by allowing them an extended control and ownership of their own medical data in a transparent and auditable manner.

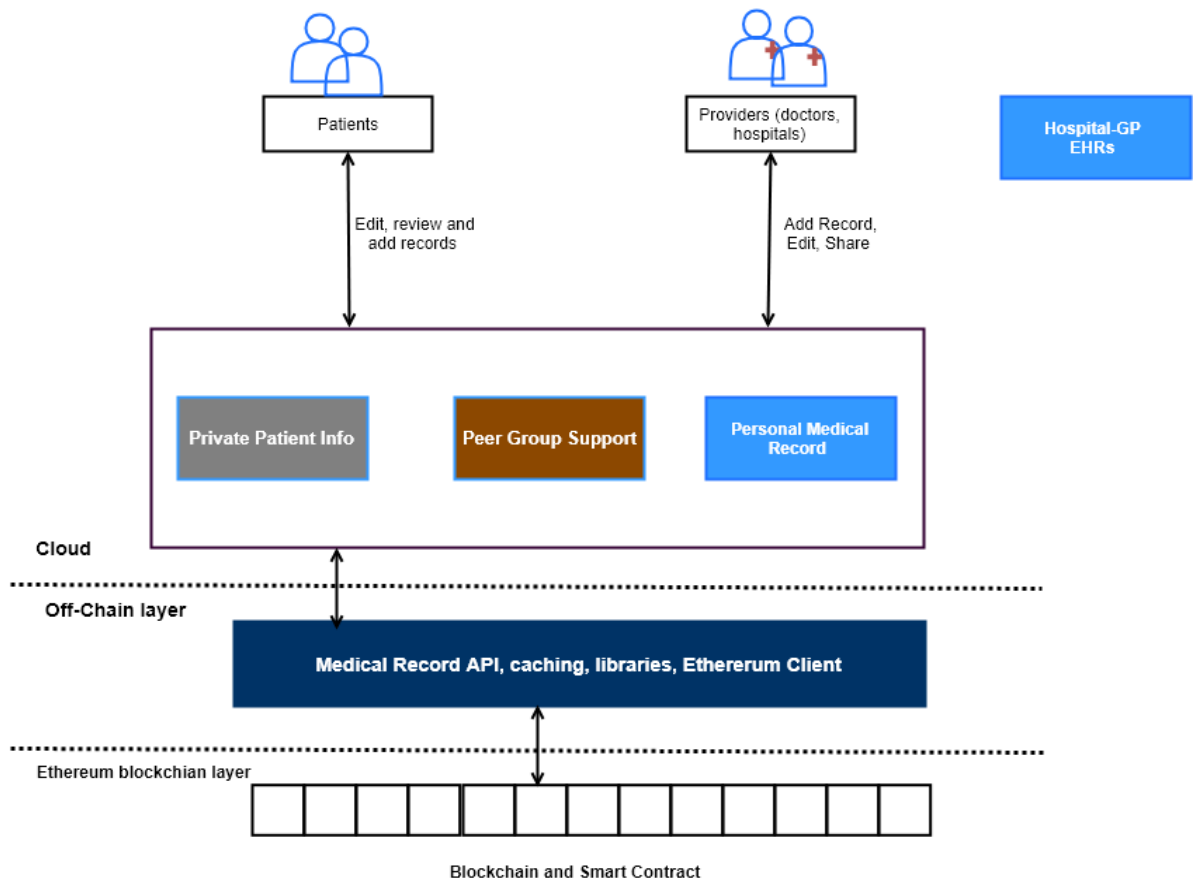


Figure 3.2: Simplified architectural overview of mHealth-BlockC and the the interaction between the patient/provider interface, frameworks and smart contracts in the EVM

As illustrated in figure 3.2, mHealth-BlockC is made to address the need of the three categories of users patients, healthcare providers and other governmental organizations. A patient generates private medical data by directly recording data into the system, which can be imported from external sources such as test results from mental health-related

consultation or online mental health screening tools. To access and use the system, both a patient and a provider are required to first register using a web-interface. The registration process allows users to enroll into the blockchain network by providing with 20 byte long Ethereum address. The generated Ethereum address is used as an account to identify the users. All registered users can log into mHealth-BlockC by entering their username and password without waiting for verification from a trusted third-party. As they log into the system it unlocks the Ethereum account and transfers an Ether into the the users account address. Both a patient and a provider user receive a transfer value of 150 Ether into the newly created address from the coinbase <sup>1</sup>. This initial fund is used for adding, updating and deleting records in the EVM. Ethereum miners are operating through incentives, and any state and value transition in the blockchain demands a fee. As such and without any Ether, a user cannot perform any form of a state transaction within the Ethereum network. The account address allows patients and providers to execute operations and conduct transactions in the Ethereum network. Furthermore, patients might create a personal medical records and share it with providers. Sharing and exchanging of medical data is based on a patient's consent. The provider has to request patient permission to get access to private data. On the other side, the patient might terminate access permissions to specific or all of his data on the blockchain. Even when a provider shares medical data without patient permission, the system sends an automatic notification to the data owner. The owner verifies the automated notification before accepting or rejecting the sharing requests. This triggered notification keeps the patient informed in case of any state transition or change in his medical records.

The prototype is designed and implemented to allow patients to maintain data ownership, unlimited access, and trace changes using an auditable-log of DLT. One of the properties of the distributed ledger is transparency, in which the content of a transaction that is added to the blockchain as a block is visible to the general public. However, the user's real identity is pseudonymous and represented by a 20 byte account address. To prevent disclosure of medical data, patient-provider generated medical data is stored on the local storage. An encoded hashed copy is added as a block into the Ethereum blockchain.

---

<sup>1</sup>Coinbase or Etherbase is primary account address that by default collects mined Ether in a single node Ethereum network

The block content in our prototype represents data ownership, consistency of user data using hashes, and access permissions which is granted or denied by members of the network. The EVM supports the use of a smart contract, which allows for maintaining state transaction such as a change in access permission, a creation of new records with data ownership. It is also designed to include a log of the patient and the provider associated with the record ownership and access permissions. To ensure data integrity and to prevent modification of patient data, our prototype applies cryptographic hashes. Furthermore, the system allows the patient to grant the provider permission to access the medical information associated with a specific treatment. As such, the provider can add a new record to the patient's medical history. The patient as the owner of the generated medical records can authorize an exchange of a medical data between providers. In such cases, the patient receives a notification from the provider and verifies the security risk of sharing the medical data before granting or denying access to it. This helps all parties to be engaging in the consensus process and improves patient privacy.

mHealth-BlockC is designed to unveil the potential of blockchain technologies in tackling the challenges that are faced sensitive personal health records particularly privacy, transparency and ownership. The prototype sets the patient at the forefront by enabling them to communicate anonymously with peers and healthcare providers.

### 3.5 mHealth-BlockC Design

Ethereum blockchain provides a transparent and a trusted environment for registration and exchange of information between participant nodes. The developed prototype combines both front-end, integrating libraries and deployed smart contracts. To interact visually with the blockchain, both front-end and smart contract are integrated together using web3.js, a Javascript framework specifically developed for Ethereum smart contracts. **Front-end web application** is developed using web technology frameworks which is used to interact with the Ethereum blockchain via client API's. The smart contract is business logic of the DApps. It is deployed and executed inside the Ethereum Virtual Machine.

The smart contract for our prototype is written in the Solidity programming language, but other blockchain technologies such as Hyperledger Fabric uses both Java and Go



to write chaincode. The code is compiled into bytecode, and it is deployed and runs inside the EVM. In the Ethereum network the deployed smart contract is assigned with a unique contract address which is 20 byte long, similar to external account addresses. Consequently, calling functions which are deployed as bytecode to the EVM using the front-end web interface is impossible. However, the Application Binary Interface (ABI) that is a .json format defines on how to calls functions and methods in the blockchain. The smart contract on the EVM receives function calls from external accounts such as patient or provider and other contract accounts. Incoming calls are processed according to the business logic of smart contracts.

The architecture of the front-end as shown in figure 3.3, allows both patients and providers to execute transactions directly through front-end user interfaces. The prototype can run on standard browsers and browsers that support Ethereum Metamask pluggins.

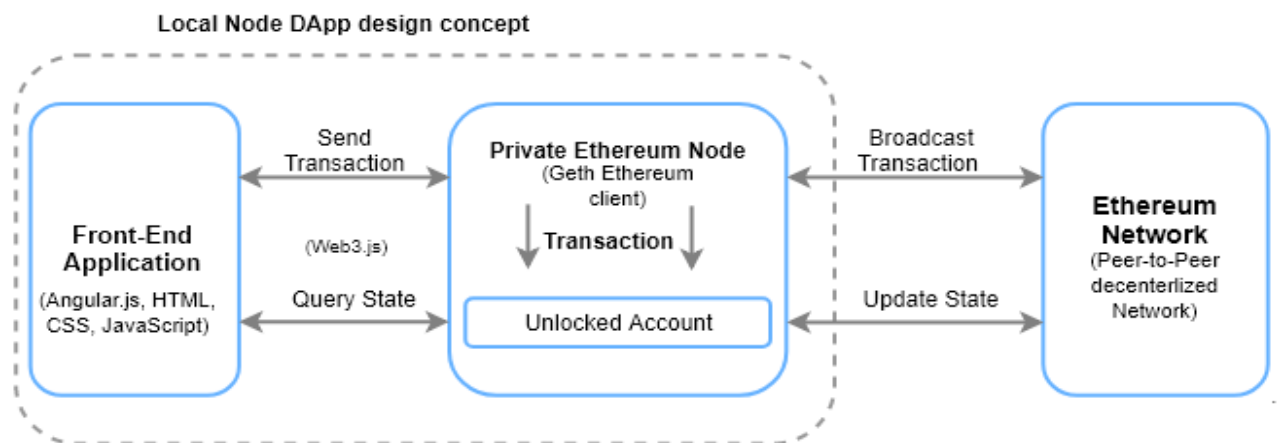


Figure 3.3: Decentralized Application (DApp) structure

### 3.5.1 Smart Contracts

The core of our design concept relies on a smart contracts, which is the business logic of our decentralized prototype. Solidity which is a smart contract language, is used to implement our contract prototype. It contains rules and instruction on how the patient-provider can interact with one another and with the system in a secure and transparent manner.

## Patient-Provider Contract

Our patient-provider contract is the leading smart contract of our Decentralized Application (DApp). The smart contract defines access permissions, record management by mapping the account address with different entities and attributes. The patient-provider contract generates Ethereum account addresses to newly registered users and links the unique account address to profile data. The patient-provider contract enables users to add, update, control access permission and share transaction by interacting using the user interfaces. In order to access a patient medical data, a provider must have the appropriate permission from the patient. The contract handles access request from the provider and other healthcare organization by sending a notification to the patient. The patient reviews the notification and the requested information based on the contract agreement and decides on to which specific information to grant or deny access to. Besides that, the contract prevents the provider to further share or exchange the data without the patient's knowledge by sending a notification to the patient. Access to the shared data is also restricted by limiting the duration of access time by setting the expiring time for the provider's access to the data.

The Patient-Provider contract is implemented using Solidity and the web3.js library, which is used to integrate the deployed smart-contract with the front-end. Web3.js is a javascript library designed explicitly to interact with the Ethereum smart contract. The smart contract is used to execute rules and instructions of our application which contains features including registering a patient and a provider, adding medical record, retrieving medical information, handling events and managing access rights and permissions.

The **registerPatient()** and **registerProvider()** function respectively, used to register both the patient and the provider into the distributed ledger and assigned an Ethereum account which can be used to identify the user in the distributed ledger. Both the private and the public key are generated using the Elliptic Curve Digital Signature Algorithm (ECDSA), and the keys are used to encrypt, digitally sign and hash the transaction. The account address, which is 20byte long, is derived from the public key using the keccak256 hash algorithm. The front-end web interface helps users to execute transactions easily by interacting with the smart contract graphically. Both the patient and provider can edit and update their profiles while login into the distributed ledger. To perform state transaction such as updating, adding and deleting in the Ethereum

Virtual Machine, a miner is required. Consequently, every node need to have enough Ether to pay for the computational work. In our implementation, an Ether was assigned to both patient and provider during login into the application. In the case of too little fund in a user account, we have implemented a function that transfers values into the account in need of funds.

```
1
2
3     pragma solidity ^0.4.10;
4
5
6     contract GroupEtherapy {
7
8         address owner;
9
10        mapping (address => Patient) patients;
11        mapping (address => Provider) providers;
12
13    }
14
```

Figure 3.4: Patient and Provider contract structure

Every transaction in our prototype logged in the EVM as **Events**. As shown in figure 3.5 implementing events is optional, and they don't have any effect on the state of the ledger. However, it helps both patients and providers to track that a record of transaction has conducted successfully, or failed, using the log from the Ethereum Virtual Machine (EVM). As shown in

```
1     event PatientRegistered(address patient);
2     event PatientRegistrationFailed(address patient);
```

Figure 3.5: Events allows users to track whether they are registered successfully or failed by using logged message into the EVM .

As shown in the code snippet in figure 3.4, an address is mapped to both patient and providers. The Struct is used to group different variables.

The contract contains functions that are used to

- getPatient
- getProvider

- insertPatientRecord
- getPatientRecords
- addPermission
- getPermission
- getPermissionType

Our prototype ensures that granting and revoking permission and access to personal medical information is performed only by the Patient. Furthermore, the patient can restrict access rights to providers by allowing them read/write, and similar restrict permissions by setting a time range for granted access to the personal records.

### **3.5.2 Front-End Web Interface Design**

The Model-View-Controller (MVC) software design pattern was adopted to develop the front-end web interface of our prototype. AngularJS is a JavaScript web application framework that supports the MVC design pattern to implement web applications. It extends the HTML syntax and attributes with directives. AngularJS is chosen as a front-end development framework for our prototype.

AngularJS provides users with a visual interface to execute smart contracts in EVM via web3.js library.

### **3.5.3 Users Login and Ethereum Address Generation**

Our DApp implementation allows a user to enroll into the Ethereum network without authentication from a trusted third-party. The account address, which is generated by the system, is used to identify both the patient and the provider uniquely. User authentication and registration to access mHealth-BlockC DApp relies both on the local login account and an associated Ethereum account address. The Ethereum account address is created and assigned to the user while subscribing into the system using the application web interface. The account address is then used to associate with the user's login accounts.

```

1
2  this.registerBC = function (user) {
3  console.log("The user to creat Account : " + JSON.stringify(user));
4  var addressCreated = $rootScope.web3.personal.newAccount(user.password);
5  console.log("New Account Created :"+ addressCreated);
6  return addressCreated;
7  };

```

Figure 3.6: Creating an Ethereum account address for the user

As shown in the code snippet in figure 3.14, the `RegisterController` function is responsible for establishing a mapping between the user profile and the 20 byte account address. The `registerBc` function generates a 20byte Ethereum account address using `web3.personal.newAccount()`. The `registerBC` function is invoked by `RegisterController` to create an account and link it with a user's login profile.


### 3.5.4 Ethereum Block Mining

Blockchain is an incentive-driven system which employs miners to engage in the consensus process by utilizing computational resources. We propose a blockchain mining model that is to be more inclusive by bringing involving node in to the network such as providers, researchers and other participate in the mining process. In return, the system rewards them with Ether. The amount of Ether earned during mining can be used to pay for the patients in exchange for personal medical information that can be used for medical research and government health related reports.

In our private Ethereum network, a Power-of-Work (PoW) consensus protocol is used to validate transaction and add the block into the ledger. Figure 3.7 shows the structure of the block and its content.

## 3.6 mHealth-BlockC Prototype and Code Review

With the prototype architecture defined and described above, We will explore an in-depth review of the code that has been implemented based on our design concept.

 Summary - information of block 13409

Block Number	<a href="#">13409</a>
Block Hash	0x0e267d1f1992e79d37e4a1d348ab0fd39a47c5eaf706ae2cc8f840c0f4a15ab1
Received Time	Fri, 03 Aug 2018 03:55:03 GMT
Confirmations	<b>50 Confirmations</b>
Difficulty	0.000 T
Nonce	0x743335abf745feea
Size	992 bytes
Miner	<a href="#">0xf4086a00e82ea1b2275f78ad8916806a3020a32e</a>
Gas Limit	4,712,388 m/s
Gas Used	84,000 m/s
Uncle Hash	0x1dcc4de8dec75d7aab85b567b6ccdd41ad312451b948a7413f0a142fd40d49347
State Root Hash	0x440d7150222d7cfc05c28395cc5c6befaaa41fab9f4c0283e7f471861181483
Parent Hash	0x08cba243cf988c24609e303fb49c01164e202f74c3cldb0424566a523e30dd9d1
Data	d983010802846765746887676f312e392e32877696e646f7773

Figure 3.7: An actual Ethereum block which is created during patient registration.

### 3.6.1 mHealth-BlockC prototype adoption and Ethereum Functionality

#### Initial Setup:

To build and develop a distributed applications are first needs to configure all the required development frameworks. We setup an Ethereum testnet which uses a single node to run and test the prototype. The relationship between different libraries and frameworks are briefly described in figure 3.8. To configure the nodes and virtual machines the following software packages are configured:

Non-Ethereum Frameworks and packages that are used for developing Decentralized Application are:

- **AngularJS:** a client side web framework which is written in JavaScript. It extends the HTML syntax with directives. It uses MVC design pattern to implement

the front-end web interfaces. The MVC makes applications developing using AngularJS easy to maintain and to test.

- **Node js:** a server-side Javascript runtime environment that allows for JavaScript codes to execute outside the browsers.
- **NPM:** the default package manager for JavaScript, and it supports code discovery and manage dependencies in our prototype projects.
- **Gulp:** a web application built tool for automating time-consuming development work flows and tasks.
- **Bower:** used to manage packages and to ensure that all packages are up-to-date. Furthermore, it works by automatically finding and installing packages.

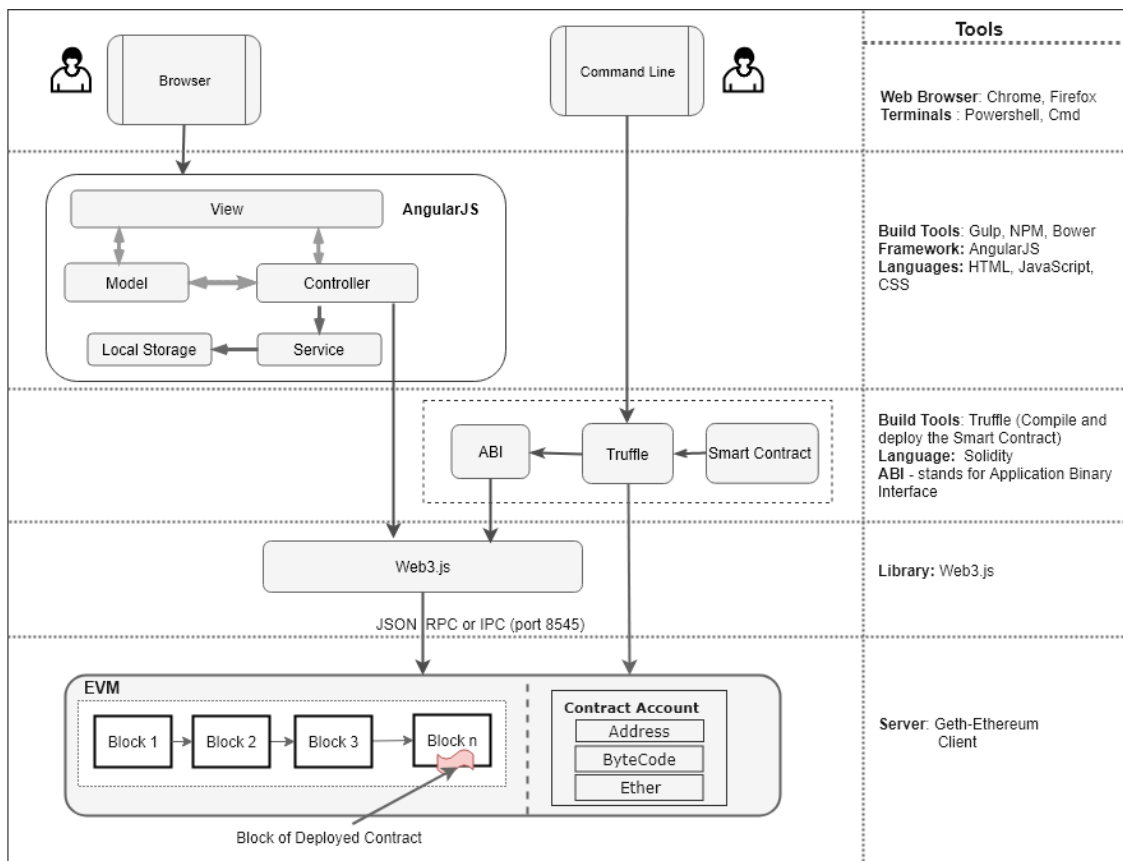


Figure 3.8: An overview of different DApp development libraries, frameworks and tools connecting to one another to execute transaction.

To fully function and run the DApp on Ethereum, certain development frameworks, libraries and tools are required. The tools and frameworks are used to facilitate the development process by simplifying the coding, compilation and deployment of the smart contract into the EVM. Ethereum based development tools and frameworks that are used in our Decentralized Application (DApp) are described below.

- **Go-Ethereum client:** offers a command line interface for managing transaction and executing smart contracts in the Ethereum network. It is implemented with the Go programming language and has the following functionality in the Ethereum network: used to mine Ether, to transfer tokens between account addresses, to create contracts and to provide commands that are used to extract information from the blockchain.
- **Solidity:** is a smart contact language . For more details see chapter two section 2.2.6.
- **Truffle:** is a development environment for Ethereum enabled which provides for an easier and faster development process. Furthermore, it offers automated contract testing, compilation, deployment and migration into the Ethereum network.

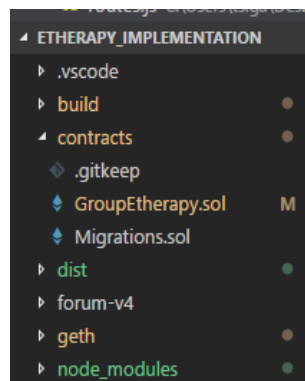


Figure 3.9: Truffle helps in structuring and organizing the DApp development environment

After configuring the Ethereum client tools and frameworks, the machines are prepared to run an Ethereum node. To initiate the Ethereum node, a genesis block should be configured with the required address, difficulties.



**Genesis block** is the first blocks in the blockchain, and it is the only block in the network that does not point to a predecessor block. As shown in figure 3.10, the genesis block is by default hardcoded in the form of JSON files and contains parameters that are used to instantiate a block. Previous hash for this block is zero because no other blocks were created before. The gas limit, block difficulty and intial Ether for the coinbase have also defined in the Genesis block. To start the private Ethereum network, the Geth client must initiate by including *genesis.json* file in commands, and the participant nodes starts to link validated block automatically to the genesis block.

```

1
2 {
3   "coinbase" : "0x0000000000000000000000000000000000000000000000000000000000000001" ,
4   "difficulty" : "0x3000" ,
5   "extraData" : "" ,
6   "gasLimit" : "0x80000000" ,
7   "nonce" : "0x0000000000000042" ,
8   "mixhash" : "0x0000000000000000000000000000000000000000000000000000000000000000
9     0000000000000000" ,
10  "parentHash" : "0x0000000000000000000000000000000000000000000000000000000000000000
11     0000000000000000" ,
12  "timestamp" : "0x00" ,
13  "alloc" : {
14  },
15  "config" : {
16    "chainId" : 15,
17    "homesteadBlock" : 0,
18    "eip155Block" : 0,
19    "eip158Block" : 0
20  }
21 }

```

Figure 3.10: Genesis Block structure with gas limit, nonce difficulty defined for our Ethereum test nodes [5]

**Start Mining:** Before mining starts, a default account is needed for deploying contracts and conducting transactions. The default account is called coinbase and is used to store initial minted Ethers, to perform transactions and to deploy contracts into the blockchain. Every transaction in the blockchain is minted and validated by participant nodes before they are appended into the ledger. The winning miner who solves the

computational puzzle incentives by the transaction sender node depending on the gas consumed. To perform any operation in EVM, the miner must be initiated. Otherwise no transaction or deployment can be executed and attained. The commands in figure 3.11 are used to start Geth-Ethereum client and initiate miners in the private Ethereum network .

```
1 // Private Ethereum blockchain starting commonad in console.
2 $ geth --port 3000 --networkid 15 --nodiscover --datadir="
3   private_blockchain" --maxpeers=0 --rpc --rpcport 8545 --rpcaddr
4     127.0.0.1 --rpccorsdomain "*" --rpcapi "admin,db,eth,debug,miner,
5     net,txpool,personal,web3"
6
7 $ geth attach http://127.0.0.1:8545
8 // Adding coinbase account
9 $ personal.newAccount
10 // Starting Mining
11 $ miner.start()
12 // to terminate mining
13 $ miner.stop()
14 // Starting Ethereum wallet to monitor account and deploy contracts
15 // graphically
16 $ & 'Ethereum Wallet.exe' --rpc http://localhost.8545
```

Figure 3.11: Provider setup for RPC connection

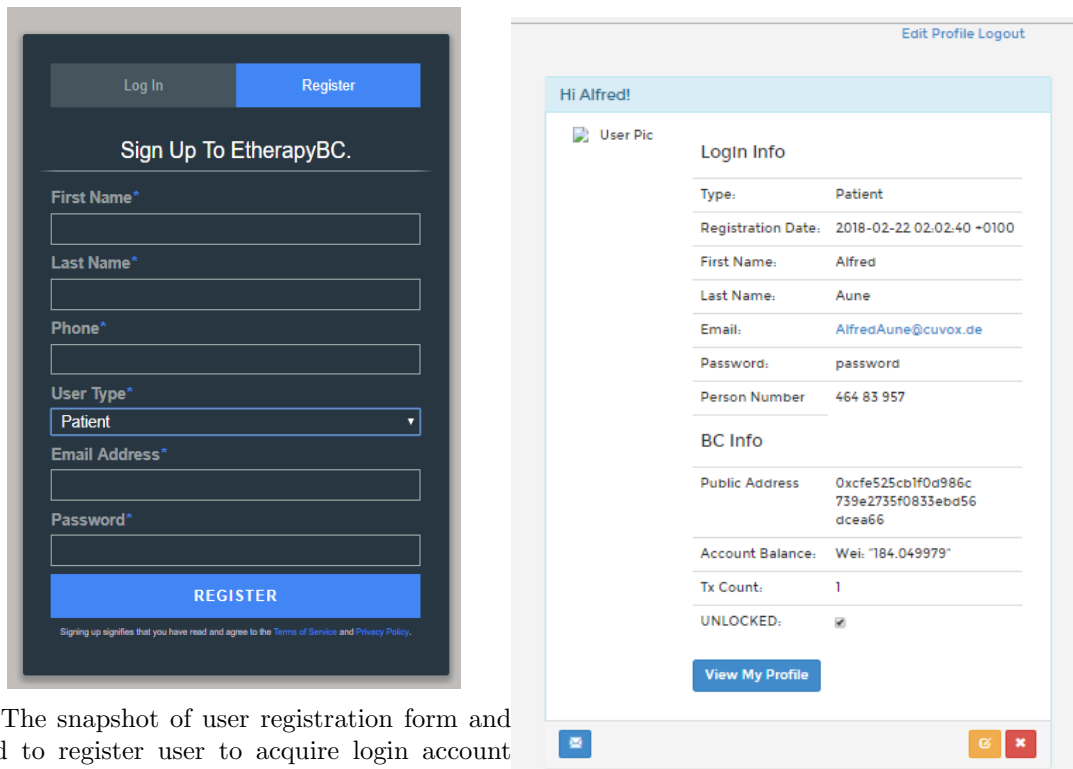
### 3.6.2 Front-End web platform

The Ethereum client uses a command-line interface to interact directly with the EVM. Ethereum is the first cryptocurrency to propose the concept of DApp and adopted into its architecture by enabling an end-user to interact with the EVM directly using web technology frameworks. Patients as the primary orchestrator of their medical data, it is very challenging for them to communicate and understand the resulting outputs using command line interface. A front-end framework allows a participant nodes to interact efficiently with the EVM using Web3.js. Web3.js is javascript library designed explicitly for Ethereum, and allow easy interaction with both local and remote Ethereum nodes.

Both patients and providers uses the front-end web interface to register, retrieve, update existing data, and initiating access permission. Web interface has served an important tool for both patients and providers to interact visually with the blockchain. The current

implementation includes features for adding, updating, deleting and retrieving records. In addition, it includes also features for access and permission managements.

Every patient and provider use the interface to subscribe into the blockchain. Our prototype front-end interface is the center point for both patients and providers. and review review records and personal infr



(a) The snapshot of user registration form and used to register user to acquire login account linked to an Ethereum address. User Type allows to choose between a patient or a provider. (b) Patient profile after registration and accessing the network.

Figure 3.12: user subscription form and patient profile during login with newly assigned Ethereum account address and initial Ether funds

### 3.6.3 Web3.js sample codes

Web3.js library is used to interact and integrate both the front-end web frameworks with the Ethereum smart contract which is deployed and running in the EVM. Web3 contains web3.eth object to directly interact with Ethereum network.

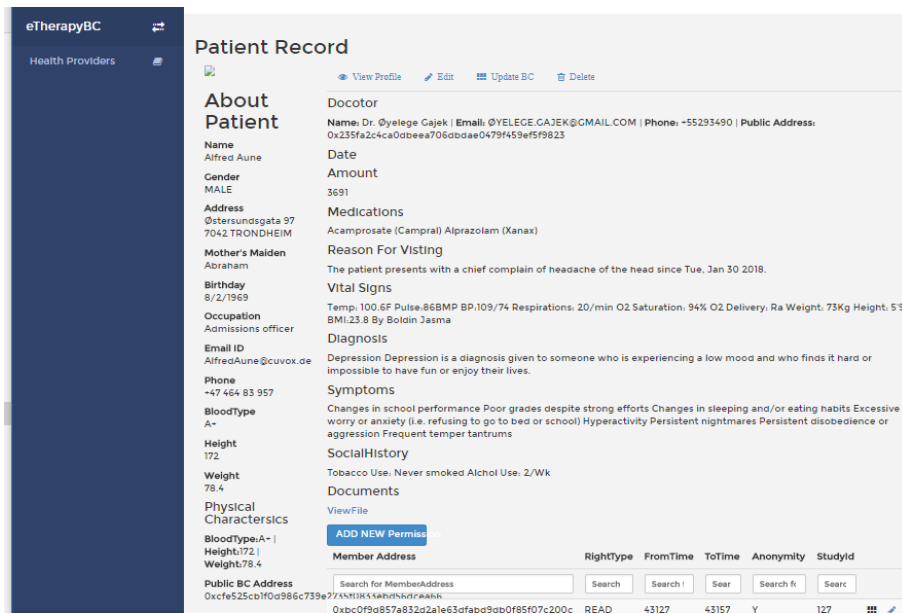


Figure 3.13: snapshot of patient’s medical record with profile data and generated randomly.

To communicate with the Ethereum network, the DApp needs to initiate a web3 provider instance.

```

1
2   this.registerBC = function (user) {
3     console.log("The user to creat Account : " + JSON.stringify(user)
4       );
5     var addressCreated = $rootScope.web3.personal.newAccount(user.
6       password);
7     console.log("New Account Created :"+ addressCreated);
8     return addressCreated;
9   };

```

Figure 3.14: Creating an Ethereum account address for the user

## 3.7 Prototype Testing

### 3.7.1 Hardware Components

Our prototype was developed and tested on Personal computer, which has 2.50Gz Intel-Core-i5 processor, 8GB RAM memory and Windows 10 operating system. All supporting

softwares and frameworks that are required to implement and test the prototype configured on single node. Both the application, the miner and the geth-client initiates in the same machine.

### **3.7.2 System Components**

Ethereum consists of distributed nodes of computers and virtual machines that runs Ethereum client by choosing a preferable programming language implementation. In this patient centric record management, we have used the go-ethereum client for running the Ethereum private network. Go-ethereum client offers a command line interface for managing transaction and executing smart contracts in the Ethereum network. Furthermore, it is responsible for a wide-range of tasks, such as interacting external users with the Ethereum network, connecting peer-to-peer nodes, creating accounts, transfer transaction, deploy smart contracts and update the state of the local-copy of the blockchain. The Geth console was modified to fit the design pattern that we have used on mapping the identity of patient-provider and related ethereum address.

### **3.7.3 Patient-Provider Smart Contract Deployment**

A Patient and a provider owned external account is used to deploy the smart contract to the EVM. Smart Contract deployment is similar to ordinary transfer of transaction, and it demands computational work and mining rewards. Truffle a building tool is used to compile and migrate patient-provider smart contract to the EVM. The Geth-Ethereum client and mining nodes must be initiated to successfully conduct the contract. Furthermore, users requires them to unlock their accounts while performing any type of transaction on the EVM. Furthermore, the gas limit also needs to be adjusted in case of happening any failure due bigger contract size or lack of fund for the miners.

The solidity compiler that is embedded with Truffle compiles the Solidity code into bytecode for Ethereum Virtual Machine and Application Binary Interface (ABI). Bytecode is machine readable and executed by the EVM while ABI is a json format of the the patient-provider smart contract. Both patient and providers interacts with the Ethereum ecosystem using a Web3.js library by loading using the ABI.

```

1
2   if (typeof web3 === 'undefined') {
3     console.log("Undefined web3 Defined!");
4     $rootScope.web3Provider = web3.currentProvider;
5     web3 = new Web3(web3.currentProvider);
6   } else {
7     console.log("Create web3 other than metamask");
8     $rootScope.web3Provider = new web3.providers.HttpProvider("http://" +
9       GETH_HOSTNAME + ":" + GETH_RPCPORT);
10    web3 = new Web3($rootScope.web3Provider);
11  }
12  $rootScope.web3 = web3;
13  $.getJSON('./contracts/GroupEtherapy.json', function (data) {
14    //get the necessary contract artifact file and instantiate it with
15    //truffle-contract.
16    var GroupEtherapyArtifact = data;
17    $rootScope.contracts.GroupEtherapy = TruffleContract(
18      GroupEtherapyArtifact);
19
20    //set the provider for our contract.
21    $rootScope.contracts.GroupEtherapy.setProvider($rootScope.web3Provider)
22    ;
23    console.log("Contract Loaded Successfully!...");
24    //user our contract to retrieve and mark the adopted patients.
25    return true;
26  });

```

Figure 3.15: The code snippet first looks for the provider using hostname and port, and the ABI is loaded using web3 by calling GroupEtherapy.json file to interact with the EVM

### 3.7.4 DAPP resource usage : CPU and Memory

Decentralized application on Ethereum includes browsers, front-end frameworks and libraries, the Geth-Ethereum client, the mining node and a local data storage. Collectively builds the DApp and the DApp allows both the patient and the provider to execute tasks such as subscribing , editing profiles, adding medical information, share data with providers and much more. All those activities in blockchain is called a transaction. Ethereum is performing a state transition and a block creation in a decentralized peer-to-peer fashion. Validating a broadcasted transaction across the Ethereum network is reached based on a consensus among the participant node. The system employed a PoW consensus algorithm for mining and validating a new block. This Power-of-Work (PoW) algorithm as its name indicated spends computational resources to solve the block difficulty and append the generated block into the distributed ledger. Our proto-

type runs on a single machine and all the development frameworks and tools are running on it. Running all those tools uses a computational resources, but the most computational resources on the machine is occupied by the miner and Geth-ethereum client. Sending records and deploying contracts is conducted with mining nodes. For example, in our Ethereum network when a miner initiates, the computer utilizes all the available computing resources such as CPU, Memory and storage to the mining node. 90.4% of CPU and memory resources are allocated to run our DApp in a machine with Core-i5 Intel processor and 2.50Gz computing speed.

### **3.8 Summary: Prototype Logical Sequence Code Review**

Our mHealth-BlockC prototype is built on an Ethereum test network, blockchain smart contracts, interface APIs and front-end web app for accessing the network iteratively.

In the mHealth-BlockC prototype the flow of transaction begins in the front-end web interface by a user triggering a send record or query button. The private Ethereum Geth-client used as an intermediary in providing command line supports such as to unlock accounts, executes a transaction in the local node and broadcasts it to the Ethereum p2p network for verification and validation. Then the web interface provides interactive graphic access to the system. As shown figure 3.3, the local node contains both the interactive web interface, which is designed using AngularJS frameworks and the Geth-Ethereum client.

The local node which contains the front-end interface and Geth-Ethreuem client interact with each other by establishing a connection using `web3.js provider`. To send a transaction to a user or a contract account requires a miner, and also gas which is used for the miners to perform their computational work. The newly generated block changes the state of the Ethereum Virtual Machine. However, querying of the transaction from the Ethereum network is a simple call and it is conducted without involving a miner and computational fees.

Ethereum offers patients the ecosystem for performing transactions on the distributed ledger. Every transaction that is conducted on the blockchain and all records associated with the patient are stored the blockchain.

The logical sequence diagram 3.16 shows the steps during the initial loading of the application. Initially, the routing service will load the login details of all active users by calling LoginServices getListUsers methods, which will be loaded from contacts-list.json file. Then the router will redirect the user to a login page shown 3.18, which displays the login form for the user to provide username and password. In addition to this a new web3 provider will be created by passing the blockchain server URL and then the compiled Ethereum contract artifact ABI as JSON file will be loaded using the Truffle contract and initialized the truffle contract which will make the system ready to interact with blockchain.

### 1.0 Logical Sequence Diagram: System Initialization

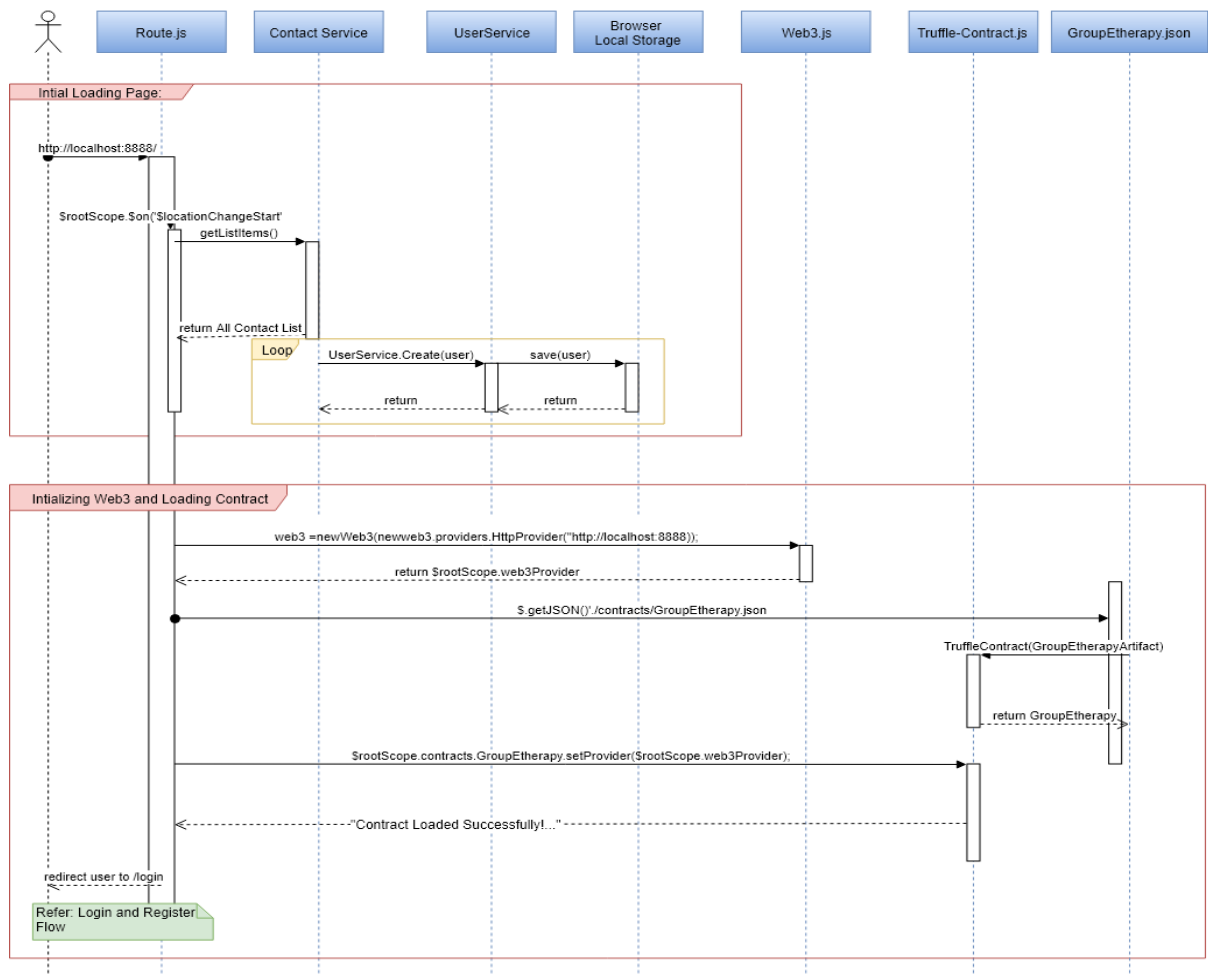


Figure 3.16: Logical Sequence diagram



The diagram on 3.17 shows steps for subscribing to the mHealth-BlockC. The Register-Controller allows the user to choose the user type (either patient or provider) and add personal details. To interact with the blockchain, RegisterController links the user details with Ethereum account address by generating an account using registerBC which uses web3.

### 3.0 Logical Sequence Diagram: Registration Flow

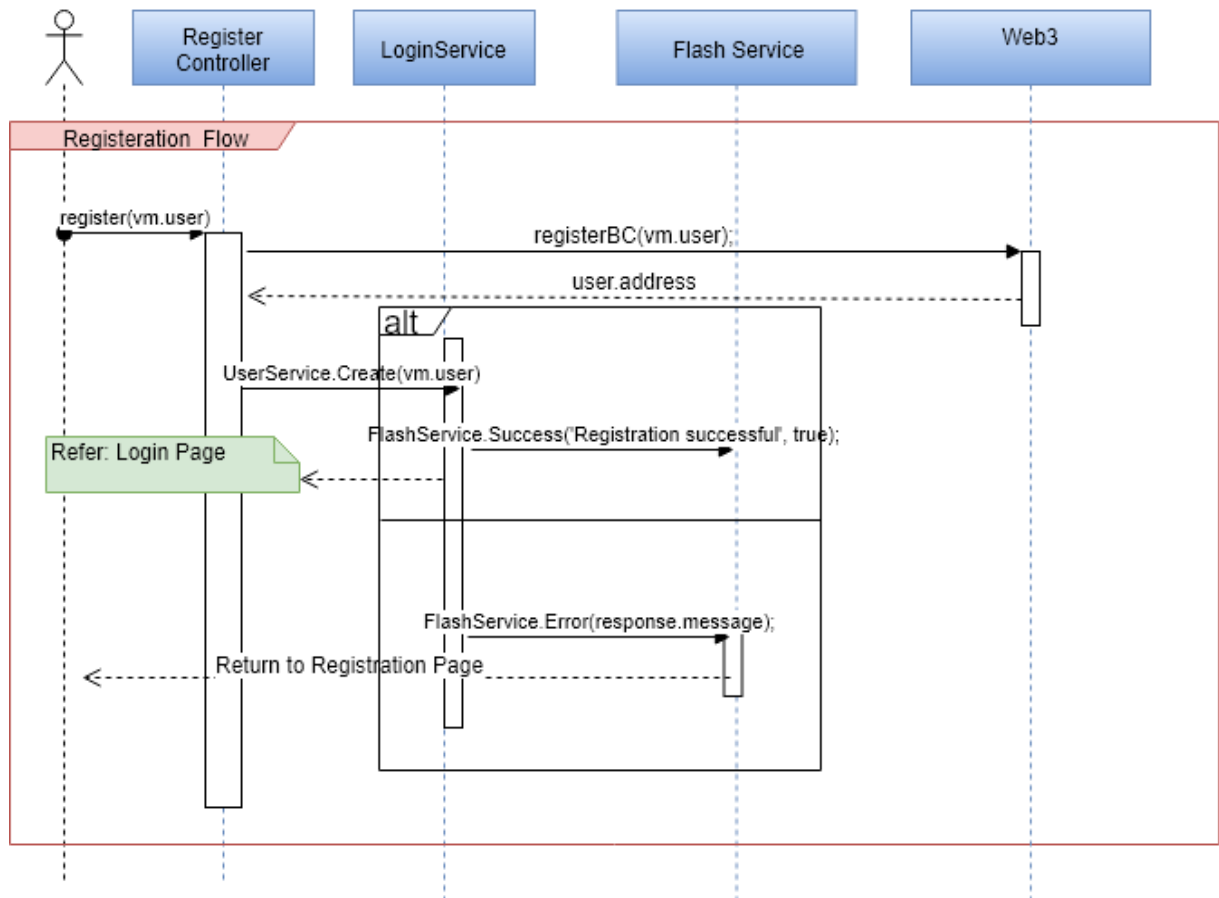


Figure 3.17: show how users registered to mHealth DApp. While registering they can select user type (patient or provider) from drop down list and also by calling registerBC() the system assigns them a blockchain account address.

The diagram on 3.18 shows the authentication process for accessing resources in mHealth-BlockC.

## 2.0 Logical Sequence Diagram: Login

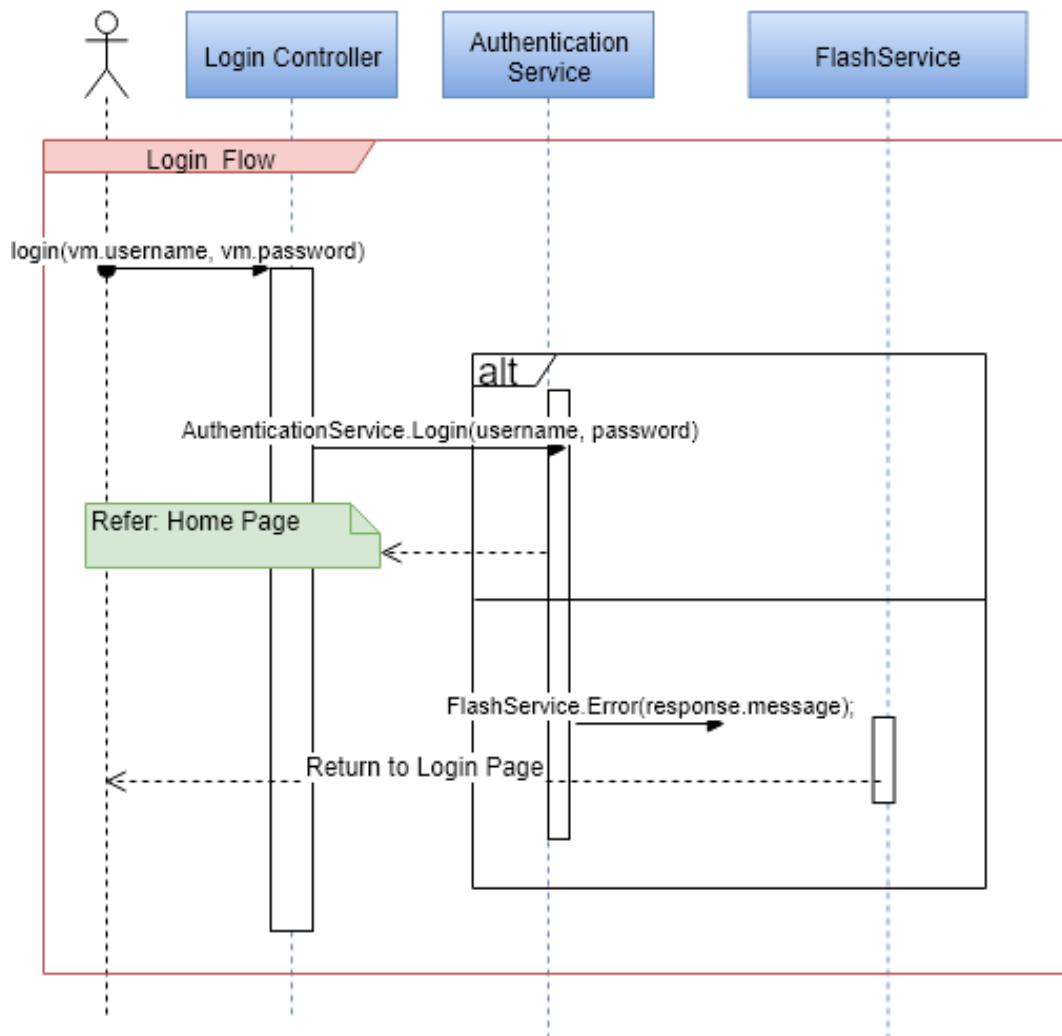


Figure 3.18: Both patient and provider access the DApp by entering the user(email address and password)

The logical sequence diagram 3.19 shows the major technical ingredients, like controllers, service, repository, model classes and methods related to patient profile management. If the user is initially registered as a patient then the patient form will be presented to the user after login. Once the user filled the patient form and submit it, the patient controller update method will be called with the populated patient model as param, which will subsequently call the service layer update method with the same patient

#### 4.0 Logical Sequence Diagram: Home/User Profile Page Flow

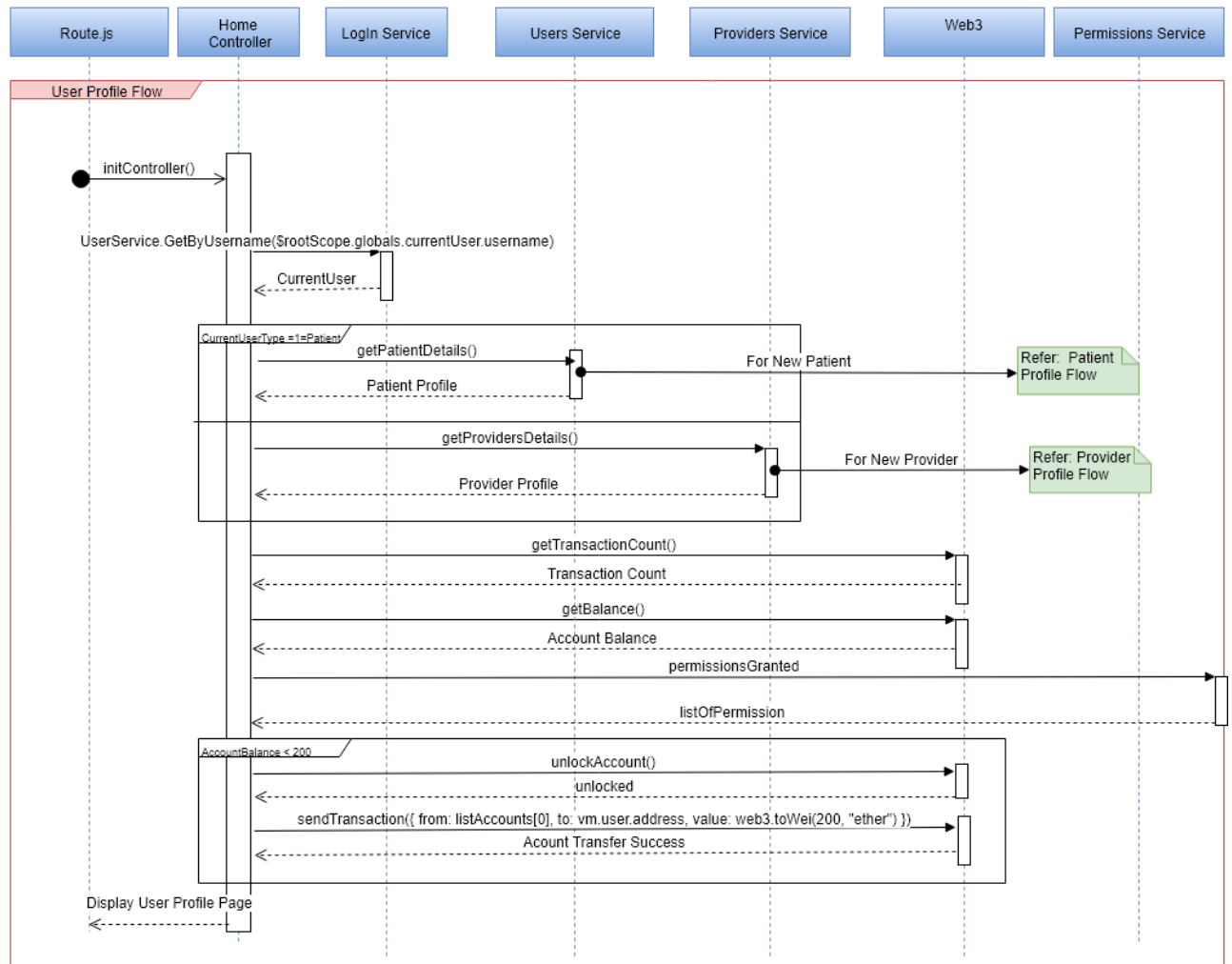


Figure 3.19: Patient and healthcare provider can modify their profiles after they login to mHealth and the patient can also share medical information with provider by setting the necessary access permissions

model. The service layer will call the Patient Repository update method which will manage all CRUD operation on patient data using browser local storage. Later if the user clicks the UpdateBC button on the patient profile page, the complete patient profile will be pushed to a blockchain using web3.js.

## 5.0 Logical Sequence Diagram: Patient Profile Flow

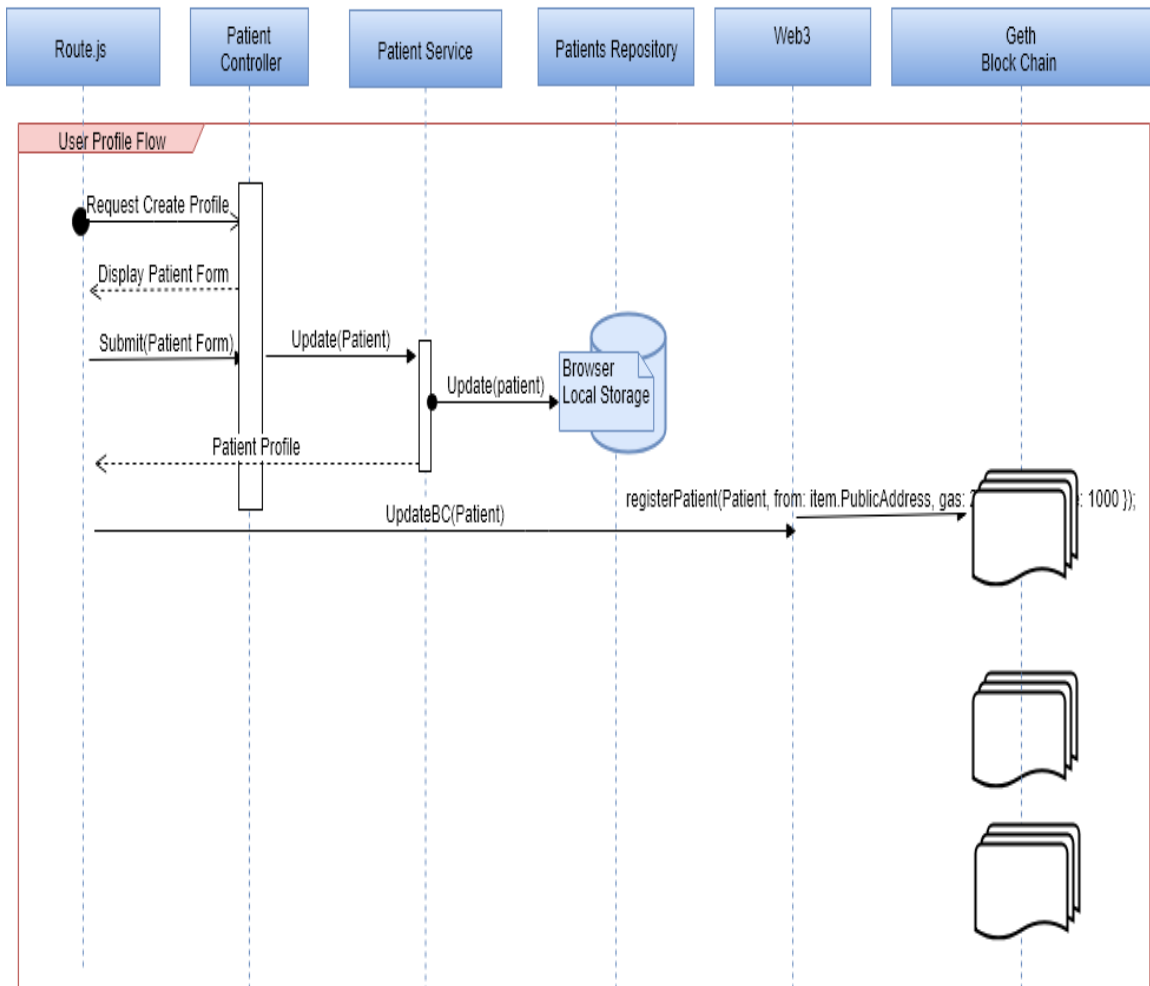


Figure 3.20: Logical Sequence Diagram: patient profile flow

## Chapter 4

# Evaluation

The mHealth-BlockC prototype was designed to explore and understand the potential of blockchain platform to provide a secure and an anonymous exchange of medical data between patients, and others. The mHealth-BlockC gives patients the ability to control and monitor their medical record by allowing them with a secure access, an immutable-log of medical history as blocks and certain level of privacy. The permission management allows patients to share and exchange medical records with the provider and keep appending auditable-log of shared record, state transaction into the distributed ledger. Any transfer of values and records that changes the state of the blockchain is appended as a block in the distributed ledger and has relevance for both the patient and provider and a patient can grant permission and to healthcare providers to trace back the recorded history and verifies for any security breaches such as modification of off-chain data and can verify with the hash data in the Blockchain.

In the following we will discuss privacy, security, scalability and computational cost based on the finding from deployment and testing of the Ethereum DApp and compare with other related blockchain technologies discussed in Chapter 2 such as Hyperledger Fabric and IOTA.

### 4.1 Security

As a decentralized ledger, Ethereum utilizes vital security models that help to mitigate the treats concerning patient safety and security related to mHealth. We built our

Decentralized Application (DApp) on top of the Ethereum blockchain, and its security model enjoys extensive properties of decentralization where individual patients can keep their mHealth records secure, accessible and if needed share with their healthcare service providers. The DApp relays on PoW consensus protocol and participating nodes to prevent a single point of failure which can be resulted from malicious nodes who tried to manipulate the network. The patient record is stored both locally on patient and provider nodes, in addition to that each node has a copy of authorized data in its distributed ledger. Any modification to the data automatically verifies against the authorized copy on each node. Every block in Ethereum contains a timestamp and a hash which provides the ledger a tamper-proof property.

Ethereum uses a cryptographic tool known as the Elliptic Curve Digital Signature Algorithm (ECDSA) to prevent the system from adversaries. In our private Ethereum network, every account address is defined using a pair of keys, a public and a private key generated using ECDSA. This approach enables secure communication within untrusted nodes, and also validates the authenticity and integrity of any records in a private network. The identity of the participant nodes is represented by Ethereum account address, and it is derived from the public-key using a Keccak-256 hashing algorithm. The mHealth data of both patients and providers are stored in a local database, and an encoded hashed copy of the data is stored in the ledger as a chain of blocks. Participant nodes on the distributed ledger can verify the consistency of patient records by comparing the hash of the actual data stored on a local database with the hash of the ledger.

Ethereum as a blockchain technology stores validated transactions in the form of blocks, where blocks are linked together using a hash of the previous block. The block hash is minted by spending computational work using PoW. Modifying a single block in the network becomes computationally expensive. This is because each transaction in the Ethereum blockchain uses a miner with a substantial computational work and fee, as such, it becomes extremely hard to reverse and modify any data in the blockchain. To reverse or modify the whole ledger, however, it requires nodes to control a 51% mining power so that consensus is reached by using the majority votes.

Hyperledger fabric membership service enroll users into the network using a certificate authority. Membership service defines the user enrollment rules, identities verification,

authentication and access control. The Certificate Authority is a pluggable interface of the membership service, and it is responsible for assign certificate to user after verifying their identity.

#### 4.1.1 Gas and Transaction Fees

Our private Ethereum network, it's security model is incentive driven. Transaction execution in Ethereum requires gases to proceed operation and is used as anti-denial of service attacks. Miners which participate in a transaction validation are rewarded for their computational work. In our Ethereum network, the miner applies Power-of-Work (PoW) algorithm to generate the right block with the highest given difficulty to reach consensus. To achieve a consensus on the state of a transaction, miners utilizes computer resources such as CPU and memory to solve the computational puzzle. The winning miners who created the right block propagate into the network for verification and incentives with Ether accordingly. The reward includes a transaction fee and an additional fee for gas consumed by the miner to create a new block. As shown in table 4.1, our contract deployment into the Ethereum private network consumes 1,367,155 units of gas and each unit of gas costs 18 Gwei.

Ethereum permissionless network use gases and PoW consensus models to allow fair node participation and to secure the system against DoS and Sybil attacks. However, in private and permissioned p2p network where all participant nodes are know, the underlying Ether and mining has no value. The Ethereum PoW is designed for private permission blockchain that validates transaction without the need for a miner and computational fees.

Block Number	Data Recorded	Gas Limit	Gas Used	Gas Price (in GWei)	Transaction Cost (in Ether)	File Size (in byte)
13951	Registering provider	4712388	106,039	18	0.0037	813
13668	Registering patient	4712388	187,090	18	0.0052	877
14027	Adding P. Record	4712388	271,834	18	0.0049	989
13286	Contract deployment	4712388	1,367,155	18	0.0246	5232

Table 4.1: Gas used during block creation in Ethereum network

As shown in Table 4.1 above, a gas limit is the maximum amount of gas allocated for performing a transaction. The amount of gas consumed is directly proportional to the block size. Ethereum has two different type of account: external owned accounts and

contract accounts. Transferring a patient record to an external account consumes less amount of gas which is 21000 unit of gas. However, sending records into the contract account consumes much higher gas depends on the type of recorded data and contract codes. In our DApp, all kind of patient records are sent into the contract account and consumes a high amount of gas. For example in table 4.1 adding patient record into the contract account consumes 271,834 unit. Deploying and executing a decentralized application for mHealth in Ethereum network is computationally expensive if a miner is involved.

Conversely, Hyperledger fabric as permission blockchain it is independent of miners and cryptocurrency. Nodes perform a transaction without any incentives, in which validation conducted using agreed peers in the network. Fabric supports chaincode, it is a type of smart contracts to develop DApp. Compared with Ethereum, Fabric have a zero transaction execution fee, and it makes more attractive for processing and validating transaction that contains a vast amount of medical data.

## 4.2 Privacy

Blockchain in general and Ethereum, in particular, have several limitations regarding privacy. The transfer of medical data within the public blockchain network is conducted at the ledger level and the activities are visible to all the participant nodes. However, in a private permission blockchain these activities are performed at the transaction level and they are only visible to the authenticated nodes within the channel. The transfer of values and records within the distributed ledger are not entirely anonymous. Our prototype enjoys a certain level of anonymity by securely hashing and concealing some of the user's personal information. Users in the system are not known to each other and identify themselves using an account address that is pseudonymous. The pseudonymous property preserves the privacy of the patient by not revealing their real identity to the public, as so the patient and the provider uses 20-byte account address that was derived from the public key using a Keccak-256 hashed algorithm. However, Ethereum is transparent by design, meaning the pseudonymous identity of users are known to then nodes in the network. Malicious users can link the attributes in the content to analyze the network and to disclose the patient's real identity.



Ethereum blockchain is transparent by design which means transaction of medical data executed at the ledger level and all users is noticed a transaction without knowing the real identity. Consequently, de-anonymize attack links the content on the transaction with Ethereum address to successful disclose real identity. However, there are methods to obfuscate the content of transaction data. One of the potential solutions for preserving privacy in Ethereum is to adopt a ZCash technology [107], which uses a zero-knowledge cryptography proof called Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zkSNARK). The zkSNARK protocol preserves the privacy of participant nodes by verifying the correctness of the record data without revealing the content to the public. It uses two cryptographic random values to hide the likability between the content of the transaction and the real identity.

Hyperledger Fabric as a permission blockchain, It allows only authorized nodes to enroll into the network. Fabric membership service is responsible for issuing certificates and controls the nodes identities while sending and validating transactions. Compared to the Ethereum and IOTA, Fabric uses a fine-grained security control on the enrolled nodes. Nodes in Fabric uses a channel to establish a secure communication among selected nodes within the network by defining a chaincode policy. The channel is designed to restrict access to a transaction, and makes it available to the members of the channel only. Other nodes in the network are unaware of the transaction and the availability of other channels. In Hyperledger Fabric consensus is reached at the transaction level, in which only members of the same channel participates to validate the transaction. In our medical record scenario, patients might establish a channel by defining the chaincode so that medical information can be exchanged with only members of the same channel.

### 4.3 Scalability

Scalability is the primary issue in Ethereum and other blockchain technologies. As a permissionless and open blockchain, patients and healthcare providers enroll into the network freely without an authentication and verification process. Patients and providers who joins the system are participating in a consensus process and engages in state transactions by sending and receiving medical records and values. Unlike Hyperledger Fabric and IOTA, Ethereum executes transactions in sequence at a ledger level. All

those activities greatly affect the scalability of the system to the level at which the blockchain cannot process more transactions. Each activity and state transaction on the network is carefully monitored and managed by all nodes. Moreover, the entire outcome of state transaction is highly depending on the miners and the individual nodes collective validation. In general perception, a network with thousands of nodes have a high throughput and performance, but in the case of the Ethereum, the nodes works independently and do not share computing resources to scale-up the performance of the system.

The mHealth-BlockC runs on a single machine and utilizes all available resources to perform multiple tasks including mining nodes, sender nodes and validator nodes simultaneously. Scalability in Ethereum encompasses different factors that includes transaction costs, throughput, latency and bootstrap time [51]. Those factors hinders the system to scale-up by utilizing the available computation resources. The average block-creation time is approximately 15 seconds, and comparing with Bitcoin it is comparatively low. However, with increasing network size and transaction, it affects highly the scalability of the system to conduct with high throughput and low latency.

IOTA uses Tangle as a protocol to validate transactions. The scalability of Tangle network depends on it's numbers of participating nodes. IOTA validates transaction in parallel and scalability is not a major issues. . In IOTA a node that needs to send a transaction validates two other transactions before getting his own transaction validated by the network.

## 4.4 Usability

The idea of Gas adds some complexity to the users and creates obstacle to users to engage with the Ethereum network and execute transaction. From a medical data point of view, the gas limit restricts the ability of the miner to process huge record of transaction informations, and sometimes the system responds with error messaging due to an insufficient amount of gas assigned to the network.

The account address that is used to identify the users in the Ethereum blockchain is driven by from the private/public key, where each transaction is encrypted and digitally

signed using cryptographic hash before broadcasting into the p2p network. The private key, also known as the secret key, is only known to the owner. The key is locally stored on the owners node. However, lack of knowledge on private key management exposes both patients and healthcare providers to security vulnerable and theft. Malicious users might use the private key to decrypt messages and to access medical data without patient permissions.

Running mHealth-BlockC in Ethereum blockchain requires a lot of works and computer resources. The Geth-client and the miner need to be operational and always ready to execute and validate transaction that are sent by the patient and the provider nodes. Without starting the miners, the Ethereum network do not execute transaction and and block creation will not happened at all. Patient and provider nodes must initiate the Geth-Ethereum client and the mining nodes in order to verify and validate transactions. Running all the development tools on one machine troubles the performance of the machine. As a result, patients and providers have to wait for some time before they receive transaction validation from the network.

The mHealth-BlockC development frameworks and libraries are at early stage. Developers experienced delays due to the lack of Ethereum based automated and Integration tools. The solidity language do not implement nested string data types and becomes a huge burden for for developers. The front-end has a visual user interface that enables the user with easy access to the EVM. By using the web interface the users easily add, update and delete records of data from the Ethereum network.

#### **4.4.1 Elapsed time for Block Creation**

One of the drawbacks of applying blockchain in implementing a DApp is the amount of time it takes to create a block. As per our implementation, the application uses an average of 15 seconds to process a transaction and generate a block. Block creation in a distributed ledger starts from the time when a record is broadcasting into the p2p distributed ledger network. Once committed into the system, the transaction goes through the mining and verification operations. The whole process depends on the miners hashrate capacity and also on the blocks difficulty. The higher the difficulties are, the longer the it takes to solve the computational puzzle.

In general, the purpose of the Ethereum PoW algorithm is to create a cryptographically secure and tamper-resisted distributed ledger technology. Block creation and verification is also part of the Ethereum security procedures. Block creation time can vary from one blockchain technology to another depending on the consensus process and hashing power used to solve block difficulties. Compared to other blockchain technologies, Ethereum has a lower transaction processing time. However, short block time does also have some downsides in Ethereum that resulted in creating orphaned blocks. Those orphaned blocks are blocks that do not make it into the main blockchain.

## Chapter 5

# Discussion and Future Work

The aim of this thesis was to study the potential of blockchain as a technology and a platform to implement a health record system capable of providing patients with the ability to perform online treatments anonymously, to maintain ownership and to preserve the integrity of patient data using cryptographic hashes. Furthermore, we have investigated the contribution of the underlying blockchain protocols such as consensus protocols in building trust among participant entities in Peer-to-Peer distributed ways.

To do so, we have implemented a prototype, mHealth-BlockC based on Ethereum and smart contracts that could be further investigated and evaluated while in use.

This prototype addresses the domain of online mHealth application as this is an area with a high demand for a technology that can improve the patients requirement to anonymity and data integrity.

### 5.1 Discussion

Ethereum and other blockchain technologies have been designed to perform transactions in a transparent and an auditable way. Transfer of records within the blockchain is conducted at a ledger level nodes follow the transaction pattern. However, the real identity of the sender node is secret to the public. Instead uses a derived account address from the public key using a cryptographic hash algorithm. In essence, the privacy of a patient's medical information is not well-preserved. Moreover, by carefully analyzing the

network and the links between the user account address and its respective transactions the real identity of the node might be revealed. Conversely, in Hyperledger Fabric, nodes are authenticated and assigned a certificate from a CA to enroll to the system, as such all nodes are identified. By using chaincode a channel can be established group nodes with the same interest among an associated node so that any transfer of records within the channel becomes only visible to the member of the channel. For example, in mHealth-BlockC, patient, therapists and researchers with a common interest on specific medical information can create a channel by defining the chaincode policy. Every transaction that is sent by any member of the channel is visible only to the member of the channel nodes. Furthermore, it only allows the three to participate on validating the transaction. Using channels, Hyperledger Fabric preserves transaction privacy while revealing the identity of users nodes to the network. In general, to preserve privacy for both the user identity and the content of medical data, blockchain requires to adopt zkSNARK protocol that executes and validate transaction without disclosure of either identity or content of the data.

Blocks in Ethereum are linked together using cryptographic hashes, and they are miners that creates the block by spending a computational work. To modify or reverse the block becomes impossible because it requires high computational work. However, miners that controls 51% of the system mining pool might alter data by dominating the consensus process. An immutable property of the blockchain ensures the integrity and consistency of patient information. Furthermore, the chain of blocks stores a hash of medical information, and those hash of blocks create and auditable-log transaction history that helps a patient to trace and verify against any change by comparing with off-chain medical information.

Permissionless p2p decentralized network allows everyone to join to the system without verification and authentication. To build trust among participant nodes Ethereum employs PoW consensus protocol to reach an agreement on state of a created block. The protocol uses miners. and utilizes computational resources (CPU, memory and Electric powers) to solve the block difficulty assigned by the system to verify the validity of state transition. The winning miner who solves the puzzle is rewarded with a tokens. Patient medical information contains huge amount of data, involving miner to validate every transaction costs them extensive amount of cryptocurrency. To prevent unnecessary

payment for miners, Hyperledger Fabric supports a pluggable consensus models that are independent of miners and fees to validate transactions. Furthermore, Microsoft have released the Ethereum based Power-of-Authority (PoA) consensus protocol on Azure that ends the reliance of the consensus process on the miners [108]. The protocol is designed for private Ethereum blockchain where all participant nodes are identified and known to one another, and consensus to validate transaction is reached without involving miners and null fee.

The security of a patient's medical data in untrusted decentralized p2p network depends on public key cryptographic techniques. Every user owns a pair of keys: public and private key, and the medical data is encrypted using private key, hashed and digitally signed using receivers public key before broadcasted to the network. The receiver's uses the private key to verify the authenticity of propagated block. Both Ethereum and Hyperledger Fabric assigns a pair of key to all enrolled users to the network. Using the pair of keys patients, providers and other participants verifies the authenticity and the integrity of the personal medical information for any change or modification. The blockchain ECDSA allows patients to verify ownership, to preserve privacy and to keep a consistent copy of replicated data by using hash and encryptions.

Unlike IOTA that validates transaction in parallel, Ethereum creates blocks by executing transaction sequentially. Every record of transactions waits until previous created block validated and appended into the ledger. Personal medical information contains entire history of a patient information. To execute transactions effectively, there are factors that needs to address such as block size, transaction cost, throughput and latency. All directly affects the scalability of the blockchain. Our study shows that Ethereum executes transactions sequentially, as a resulted transaction processing time becomes high and affect the scalability of the network. Deploying a contract into Ethereum for example takes a certain amount gas, confirmation time and block propagation delay.

Traditional healthcare information exchange have challenges in interpreting correctly a shared medical data. Heterogeneous medical information across different treatment sites creates additional barrier in sharing medical informations. Lack of Interoperability in healthcare is the main challenge in providing a quality of care across geographically scattered environment. Blockchain as a distributed ledger provides a platform that allows all participants such as patient, providers and other healthcare organization to

exchange data in peer-to-peer fashion using a standard data exchange formats such as FHIR and OpenEHR.

## 5.2 Future Work

Our study has focused mainly on private Ethereum blockchain that uses a Power-of-Work consensus protocol to validate transactions and its underlying protocols. Transaction validation involves miners. Consequently, the sender nodes (such as a patient, provide and healthcare stakeholders) pay a transaction fee to the mining node for its computational work. Reaching consensus using PoW is computationally expensive and less affordable for patients and healthcare stakeholders to reward cryptocurrency for every confirmed transaction. Extending research into other blockchain technologies that employs consensus models which are independent of miners and cryptocurrency could be a great idea for developing enterprise applications such as healthcare applications. Hyperledger Fabric and Ethereum that employs Power-of-Authority consensus model are technologies that might have a potential that can alleviate such PoW computational concern.

Most blockchain technologies provides integrated and tamper-proof log of a patient's history that are traceable. However, privacy, scalability and cost remains the primary concern in our Decentralized Application. Hyperledger Fabric and Ethereum Proof-of-Authority are both private/consortium blockchain technologies that might have some potential in preserving a patient's and other healthcare provider's privacy while performing transactions. Moreover, Hyperledger Fabric represents model-based architecture with pluggable features, and adding ZeroCash and Zero-Knowledge Succinct Non-Interactive Argument of Knowledge enhances a better privacy to the system and the individual patients.



## Chapter 6

# Conclusion

This research has examined the different blockchain technologies such as Bitcoin, Ethereum, IOTA and Hyperledger Fabric by investigating its underlying models and protocols for developing a mHealth-BlockC prototype. Two out of the four DLTs that we have used to study proves their potential for developing decentralized applications. However, Bitcoin and IOTA are permissionless DLT with two different and specific architecture and design purposes. Bitcoin designed for cryptocurrency and keeps validated transactions in the form of blocks while IOTA is developed as a platform for Internet of Things. Instead of blocks IOTA uses Tangle a DAG based that is used for storing transactions. Furthermore, IOTA designed to overcome some of the drawbacks of Blockchain such as scalability and computational costs.

The Ethereum based mHealth-BlockC prototype demonstrates how the blockchain technology and its underlying protocols contributes in developing a self maintained, secure and tamper-resist decentralized applications. The mHealth-BlockC smart contract serves as the backbone for our DApp by self-executing instruction triggered by patient and provider without a trusted third party. Pseudonymous user identity and an auditable-log of record history allows users enjoy a certain level of privacy and it provides to undergo compressive review and permission management on their medical data.

Hyperledger Fabric and Ethereum are both a programmable blockchain that allows the development of DApps using smart contracts. Despite some limitation, the DLTs that supports smart contract as part of their architecture have the potential for developing

decentralized application that demands fine-grains control, anonymity, confidentiality and data integrity such as personal medical data.

# Bibliography

- [1] Saurel Sylvain. Create your own blockchain in 30 minutes. <https://medium.com/@ssaurel/create-your-own-blockchain-in-30-minutes-dbde3293b390>, January 2018.
- [2] Minhaj Ahmad Khan and Khaled Salah. Iot security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82:395–411, 2018.
- [3] ELM Labs. Iota tangle eli5. <http://elm.nyc/research-1/2018/2/15/iota-tangle-eli5>, February 2018.
- [4] Hyperledger Architecture WG. Introduction to hyperledger business blockchain design philosophy and consensus. *Hyperledger Work Group*, March 2017.
- [5] Ethereum. Private ethereum network. <https://github.com/ethereum/go-ethereum/wiki/Private-network>, 2017.
- [6] Janice Genevro. Engaging patients and families in the medical home. *Mathematica Policy Research*, June 2010.
- [7] Mary B., Kelly A., Joelle F., Lee U., Michael K., and Thomas G. Use of computers and internet among people with severe mental illnesses at peer support centers. *Psychiatric Rehabilitation Journal*, 40(4), December 2017.
- [8] Natalie Drew Michelle Funk. Creating peer support groups in mental health and related areas. *Department of Mental Health and Substance Abuse, World Health Organization*, 2017.
- [9] Meredith L. Urowitz Sara Barbera Lisa Wiljer David O’ Rinn Susan Ferguson Sarah E. Classen, Catherine C. Chivers. Psychosexual distress in women with gynecologic cancer: a feasibility study of an online support group. *Psycho-Oncology*, 4:930–935, 2013.
- [10] Patricia R Recupero and Samara E Rainey. Informed consent to e-therapy. *American journal of psychotherapy*, 59(4), 2005.
- [11] D. A. Ludwick and John Doucette. Primary care physicians’ experience with electronic medical records: Barriers to implementation in a fee-for-service environment. *International Journal of Telemedicine and Applications*, 2009, January 2009.
- [12] Gunther Eysenbach, Tormod Rimehaug, Ioannis Mavridis, Eva Skipenes, Kylie Bennett, Anthony James Bennett, and Kathleen Margaret Griffiths. Security considerations for e-mental health interventions. *Journal of Medical Internet Research*, 12(5), December 2010.
- [13] Juri M. The disruptive potential of distributed consensus architectures. *Berkeley RoundTable on The International Economy ,University of California, Berkeley*, 1, 2016.
- [14] M. Swan. *Blockchain: Blueprint for a new economy*. O’Reilly Media, Inc., 2015.

- [15] Vaughn & Jeff Shelton & Alex Cahana Chris, Burniske & Emily. How blockchain technology can enhance ehr operability. *Ark Invest & Gem*, page 13, September 2016.
- [16] Communications NMH. World health report : Mental disorders affect one in four people. [http://www.who.int/whr/2001/media\\_centre/press\\_release/en/](http://www.who.int/whr/2001/media_centre/press_release/en/), 2001.
- [17] Katy Kaplan, Mark S. Salzer, Phyllis Solomon, Eugene Brusilovskiy, and Pamela Cousounis. Internet peer support for individuals with psychiatric disabilities: A randomized controlled trial. *Social Science & Medicine*, 72(1):54–62, 2011.
- [18] Julio Arboleda-Flórez and Heather Stuart. From sin to science: Fighting the stigmatization of mental illnesses. 57:457–63, 08 2012.
- [19] Magdalena Berger, Todd H. Wagner, and Laurence C. Baker. Internet use and stigmatized illness. *Social Science & Medicine*, 61(8):1821–1827, 2005.
- [20] F Griffiths, A Lindenmeyer, J Powell, P Lowe, and M Thorogood. Why are health care interventions delivered over the internet? a systematic review of the published literature. *Journal Of Medical Internet Research*, 8(2), 2006.
- [21] Kurt Roemer. Innovations for healthcare that ensure patient privacy, while transforming care delivery. *Health Management Technology*, 38(10), October 2017.
- [22] Mark Griffiths and Gerry Cooper. Online therapy: Implications for problem gamblers and clinicians1. *British Journal of Guidance & Counselling*, 31(1):113–135, 2003.
- [23] Kenneth D. Mandl, Peter Szolovits, and Isaac S. Kohane. Public standards and patients’ control: How to keep electronic medical records accessible but private. *BMJ: British Medical Journal*, 322(7281):283–286, February 2001.
- [24] Alex W Goodby. Clinical data as the basic staple of health learning : creating and protecting a public good : workshop summary, 2010.
- [25] The Office of the National Coordinator for Health Information Technology. Report on health information blocking. [https://www.healthit.gov/sites/default/files/reports/info\\_blocking\\_040915.pdf](https://www.healthit.gov/sites/default/files/reports/info_blocking_040915.pdf), 2015.
- [26] L O Gostin, Turek-Brezina, Powers, and Kozloff. Privacy and security of health information in the emerging health care system. *Health Matrix: The Journal of Law-Medicine*, 5(1), 1995.
- [27] Medicalchain: Blockchain technology for secure storage and transfer of electronic health records. [https://longcatchain.com/data/files/181217050\\_686448280.pdf](https://longcatchain.com/data/files/181217050_686448280.pdf), 2017.
- [28] Kenneth D Mandl, David Markwell, Rhona MacDonald, Peter Szolovits, and Isaac S Kohane. Public standards and patients’ control: how to keep electronic medical records accessible but private. *BMJ*, 322(7281):283–287, 2001.
- [29] Jan Copeland and Greg Martin. Web-based interventions for substance use disorders: A qualitative review. *Journal of Substance Abuse Treatment*, 26(2):109–116, 2004.
- [30] Davey Winder. Ransomware. *PC Pro*, (261), July 2016.
- [31] Union European. Regulation (eu) 2016/679 general data protection regulation. [http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L\\_.2016.119.01.0001.01.ENG](http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2016.119.01.0001.01.ENG), 2017.
- [32] Commission European. Data protection in the eu. [https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu\\_en](https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en), May 2018.

- [33] GDPR EU. Frequently asked questions about the incoming gdpr. <https://www.eugdpr.org/gdpr-faqs.html>, 2018.
- [34] Nathan Dudgeon and Gareth Malna. Distributed ledger technology: From blockchain to icos. *Banking & Financial Services Policy Report*, 37(2):4–9, February 2018.
- [35] Popov Serguei. The tangle. *IOTA website*, 1.4.2, February 2018.
- [36] Nakamoto Satoshi. Bitcoin: A peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf>, 2008.
- [37] B Imran. *Mastering Blockchain*. Number 16-35. O’Reilly Media, Inc, March 2017.
- [38] CoinMarketCap. Cryptocurrency market capitalizations. <https://coinmarketcap.com/>, April 2018.
- [39] N Gilbert, S & Lynch. *Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services*, volume 33(2). 2002.
- [40] W. Diffie. The first ten years of public-key cryptography. *Proceedings of the IEEE*, 76(5):560–577, May 1988.
- [41] Swanson Tim. Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems. Technical report, 2015.
- [42] Linux Foundation. Hyperledger: building open source blockchain for business. <https://ibm.com/blockchain/hyperledger>, 2018.
- [43] Greenspan Gideon. Multichain private blockchain — white paper. <https://www.multichain.com/download/MultiChain-White-Paper.pdf>, 2015.
- [44] Zied Trifa and Maher Khemakhem. Sybil nodes as a mitigation strategy against sybil attack. *Procedia Computer Science*, 32:1135 – 1140, 2014. The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014).
- [45] K. Haribabu, Chittaranjan Hota, and Saravana. Detecting sybils in peer-to-peer file replication systems. In Dasun Weerasinghe, editor, *Information Security and Digital Forensics*, pages 123–134, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [46] Baliga Arati. Understanding blockchain consensus models. <https://pdfs.semanticscholar.org/da8a/37b10bc1521a4d3de925d7ebc44bb606d740.pdf>, April 2017.
- [47] Michael Fischer, Nancy Lynch, and Michael Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, April 1985.
- [48] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, July 1982.
- [49] Community Ethereum. A next-generation smart contract and decentralized application platform. <https://github.com/ethereum/wiki/wiki/White-Paper#ethereum-accounts>, 2018.
- [50] Visa. Visa annual report. Technical report, Visa, 2017.
- [51] Kyle Croman et al. On scaling decentralized blockchains. In *Financial Cryptography and Data Security*, pages 106–125, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

- [52] Cachin Christian. Architecture of the hyperledger blockchain fabric. Switzerland, July 2016.
- [53] Christian Cachin, Simon Schubert, and Marko Vukolić. Non-determinism in byzantine fault-tolerant replication. March 2016.
- [54] W. Diffie and M. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, November 1976.
- [55] Mayer Hartwig. Ecdsa security in bitcoin and ethereum: a research survey. *CoinFabrik*, 2016.
- [56] Dai Wei. b-money. <http://www.weidai.com/bmoney.txt>, November 1998.
- [57] Hartikka Lauri. Naiveoin: a tutorial for building a cryptocurrency - proof of work. <https://lhartikk.github.io/jekyll/update/2017/07/13/chapter2.html>, 2017.
- [58] Ethereum community. Ethereum next generation blockchain. <http://ethdocs.org/en/latest/introduction/what-is-ethereum.html#a-next-generation-blockchain>, 2016.
- [59] V Buterin. A next-generation smart contract and decentralized application platform. <https://github.com/ethereum/wiki/wiki/White-Paper#ethereum>, 2014.
- [60] Petar Maymounkov and David Mazières. Kademlia: A peer-to-peer information system based on the xor metric. In Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*, pages 53–65, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
- [61] Wiki Ethereum. Kademlia peer selection. <https://github.com/ethereum/wiki/wiki/Kademlia-Peer-Selection>, October 2015.
- [62] Ethereum Homestead. What is mining? <https://ethereum-homestead.readthedocs.io/en/latest/mining.html>, 2016.
- [63] Ethereum Wiki. Mining. <https://github.com/ethereum/wiki/wiki/Mining>, 2018.
- [64] Wood Gavin. Ethereum: A secure decentralised generalised transaction ledger eip-150 revision. 2016.
- [65] Ethereum Foundation. How do i mine ether? <https://www.ethereum.org/ether>, 2018.
- [66] Community Ethereum. Ethereum homestead documentation. <http://ethdocs.org/en/latest/>, 2016.
- [67] Buterin Vitalik. Merkle in ethereum. <https://blog.ethereum.org/2015/11/15/merkle-in-ethereum/>, 2015.
- [68] Oyedeji Oluwoye. Digital cryptocurrencies: The design and network analysis of the bitcoin infrastructure. <http://search.proquest.com/docview/1876549389/>, January 2016.
- [69] Buterin Vitalik. Uncle rate and transaction fee analysis. <https://blog.ethereum.org/2016/10/31/uncle-rate-transaction-fee-analysis/>, October 2016.
- [70] go ethereum. Protocol parameters. [https://github.com/ethereum/go-ethereum/blob/master/params/protocol\\_params.go](https://github.com/ethereum/go-ethereum/blob/master/params/protocol_params.go), 2015.
- [71] Bitcoinwiki. Mining in bitcoin. <https://en.bitcoin.it/wiki/Mining>, 2018.
- [72] Szabo Nick. The dawn of trustworthy computing. <http://unenumerated.blogspot.com/2014/12/>, 2014.

- [73] community Ethereum. Mining in ethreum. <http://ethdocs.org/en/latest/mining.html>, 2016.
- [74] Szabo Nick. Smart contracts: Building blocks for digital markets. <http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/1996>.
- [75] Szabo Nick. The idea of smart contracts. <http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html>, 1997.
- [76] Jude Umeh. Blockchain double bubble or double trouble? *ITNOW*, 58(1):58–61, March 2016.
- [77] CONSENSYS. Migrating with ganache. [http://truffleframework.com/docs/getting\\_started/project#alternative-migrating-with-ganache](http://truffleframework.com/docs/getting_started/project#alternative-migrating-with-ganache), June 2018.
- [78] Wiki Ethereum. Ethereum wallet and mist. <https://github.com/ethereum/mist/releases>, June 2018.
- [79] Solidity.readthedocs.org. Solidity 0.2.0 documentation. <http://solidity.readthedocs.io/en/latest/types.html>, 2017.
- [80] CoinDesk. Bitcoin transaction fees. <https://bitcoinfoees.info/>, May 2018.
- [81] Schiener Dominik. A primer on iota. <https://blog.iota.org/a-primer-on-iota-with-presentation-e0a6eb2cc621>, May 2017.
- [82] Ali Dorri, Salil S. Kanhere, and Raja Jurdak. Blockchain in internet of things: Challenges and solutions. *eprint arXiv:1608.05187*, August 2016.
- [83] D.J. Bernstein and T. Lange. Post-quantum cryptography, 2017.
- [84] Hyperledger. A blockchain platform for the enterprise : Hyperledger fabric. <http://hyperledger-fabric.readthedocs.io/en/latest/index.html>, 2017.
- [85] Fabric Hyperledger. Hyperledger fabricdocs documentation. <https://media.readthedocs.org/pdf/hyperledger-fabric/latest/hyperledger-fabric.pdf>, May 2018.
- [86] Vikram Dhillon, David Metcalf, and Max Hooper. *The Hyperledger Project*, pages 139–149. Apress, Berkeley, CA, 2017.
- [87] Elli Androulaki et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. January 2018.
- [88] Brown Richard. A simple model for smart contracts. <https://gandal.me/2015/02/10/a-simple-model-for-smart-contracts/>, 2015.
- [89] CoinDesk. Blockchain total maket value. Accessed February 2018.
- [90] Gem. Gem i health. <https://gem.co/health>, February 2018.
- [91] Skuchain. Blockchain technology for collaborative commerce. <https://www.skuchain.com/>, February 2018.
- [92] ShoCard. Identity for a mobile world. <https://shocard.com/>, February 2018.
- [93] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman. Medrec: Using blockchain for medical data access and permission management. In *2016 2nd International Conference on Open and Big Data (OBD)*, pages 25–30, Aug 2016.

- [94] Jaan Priisalu and Rain Ottis. Personal control of privacy and data: Estonian experience. *Health and Technology*, 7(4):441–451, Dec 2017.
- [95] O. Williams-Grut. Estonia is using the technology behind bitcoin to secure 1 million health records. March March 2016.
- [96] Axel Schumacher. Blockchain & healthcare - 2017 strategy guide. 2017.
- [97] Peter B. Nichol. Blockchain opportunities are changing healthcare globally – innovative leaders see the change. <https://www.cio.com/article/3042603/innovation/blockchain-applications-for-healthcare.html>, March 2016.
- [98] Singh Gari. How hyperledger fabric brought blockchain to business. <https://ibm.com/blogs/think/2018/02/business-blockchain/>.
- [99] Community IOTA. Introduction the iota project. <https://docs.iota.org/introduction>, July 2018.
- [100] Gerhard Andersson, Jan Bergström, Fredrik Holländare, Per Carlbring, Viktor Kaldo, and Lisa Ekselius. Internet-based self-help for depression: randomised controlled trial. *The British journal of psychiatry : the journal of mental science*, 187, November 2005.
- [101] Azy Barak, Britt Klein, and Judith Proudfoot. Defining internet-supported therapeutic interventions. *Annals of Behavioral Medicine*, 38(1):4–17, August 2009.
- [102] Amanda Fitzgerald, Caroline Heary, Colette Kelly, Elizabeth Nixon, and Mark Shevlin. Self-efficacy for healthy eating and peer support for unhealthy eating are associated with adolescents’ food intake patterns. *Appetite*, 63:48–58, 2013.
- [103] Ronald F Dixon. Enhancing primary care through online communication. *Health affairs (Project Hope)*, 29(7), July 2010.
- [104] Ronny Bruffaerts, Anke Bonnewyn, and Koen Demyttenaere. Delays in seeking treatment for mental disorders in the belgian general population. *Social Psychiatry and Psychiatric Epidemiology*, 42(11):937–944, November 2007.
- [105] Joseph L Hall and Deven McGraw. For telehealth to succeed, privacy and security risks must be identified and addressed. *Health affairs (Project Hope)*, 33(2), 2014.
- [106] Tien Tuan Anh Dinh, Rui Liu, Meihui Zhang, Gang Chen, Beng Chin Ooi, and Ji Wang. Untangling blockchain: A data processing view of blockchain systems. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1366–1385, July 2018.
- [107] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy (SP)*, volume 00, pages 459–474, May 2014.
- [108] Born Cody. Ethereum proof-of-authority on azure. <https://azure.microsoft.com/en-us/blog/ethereum-proof-of-authority-on-azure/>.