A Secure Patient Monitoring System Based on Blockchain Smart-Contracts and IoT for Healthcare

Project & Thesis-II

CSE 4250

A thesis Report

Submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Science in Computer Science and Engineering

Submitted by

Mohana Rahman	**170104101**
Sanita Shahrin Joyee	**170104138**
Muhtasim Sajat	**170104145**
Khondokar Riaz Mahmud	**170104147**

Supervised by

Assistant Prof Mr. Tanvir Ahmed

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

January 2, 2022

CANDIDATES' DECLARATION

We, hereby, declare that the thesis presented in this report is the outcome of the investigation performed by us under the supervision of Assistant Prof Mr. Tanvir Ahmed, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Mohana Rahman
170104101

Sanita Shahrin Joyee
170104138

Muhtasim Sajat
170104145

Khondokar Riaz Mahmud
170104147

CERTIFICATION

This thesis titled, **"A Secure Patient Monitoring System Based on Blockchain Smart-Contracts and IoT for Healthcare"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in January 2, 2022.

Group Members:

Mohana Rahman	**170104101**
Sanita Shahrin Joyee	**170104138**
Muhtasim Sajat	**170104145**
Khondokar Riaz Mahmud	**170104147**

Assistant Prof Mr. Tanvir Ahmed
Assistant Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Prof. Dr. Mohammad Shafiul Alam
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

First and foremost, we would like to convey our sincere gratitude to the honorable supervisor, Assistant Prof Mr. Tanvir Ahmed, for his guidance, enthusiasm, patience and motivation. Without his support and guide, our research cannot be done properly like this. Feedback from our teachers have been valuable in improving this research work. Also, thanks to our beloved friends for their continuous love and support. The completion of this research could not be accomplished without the effort and cooperation of our group members. Above all, praises and thanks to Allah, the Almighty, for his graces and blessings during the research process to complete the research successfully.

ABSTRACT

During covid-19 pandemic, the number of IoT based remote monitoring devices increased dramatically. IoT enabled healthcare system is effective for proper monitoring of patients including COVID-19 patients. However, some internal features of IoT result in a number of challenges, such as decentralization, privacy, and security vulnerabilities. As healthcare data is sensitive and often contains human survival related information, so security is a major concern. For the purpose of processing data generated by the remote monitoring systems, and to inform the healthcare providers securely according to the data anomalies, we propose the use of blockchain based smart contracts. With the help of our proposed system, many security vulnerabilities associated with remote patient monitoring will resolve. Also, our smart contract system will support real-time patient monitoring by sending notification and maintain a secure record for patients through blockchain.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

IoT based patient monitoring system has been increased because of its convenience and usefulness. During this pandemic people have become more conscious and aware of the high-risk patients. And so, the use of IoT devices has increased to a great extent. It is important to ensure proper security to data transaction as patient data is very sensitive and confidential. With the help of blockchain, more security can be provided and patient-controlled-access will be prioritized.

1.2 Motivation

The Covid-19 pandemic has shown us the limits of the healthcare system in all over the world. Considering humanity's past experiences with such devastating diseases, the truth is that we are no better prepared to deal with them now than we were a century ago. Though we have now better and advance technology but still stuck trying to flatten the deadly curve with little success.

The number of covid-19 infected patients and deaths are increasing rapidly. The number of patients is much more than the number of seats in the hospital. Hospitals are overcrowded and fully booked. Thus, all patients are not properly admitted and properly diagnosed. Therefore, when a patient has to go to the hospital in a critical condition, the hospital does not have enough seats or enough ICU for them. Besides, patients are not concerned about when they should admit in hospital or when not to admit in hospital. Many patients die for not being admitted to the hospitals on proper time. Because their vital parameters regarding covid-19 can't be monitored regularly and thus they need an

emergency consult later.

It is quite impossible to monitor a large number of populations manually during covid-19 and frequent hospital contact visits are being discouraged. However, some people's physiological critical needs yet needed routine monitoring in order to live a healthier life. For example, diabetes patients, lung patients, heart patients, kidney patients. Also, these people are considered as high-risk patients for covid-19. So, it will be better for them if they can be monitored inside their house. In this case, IoT makes a way to observe a patient remotely and provide proper information so that they can get a proper treatment. For example, if patient's oxygen saturation level is below 95 measured through an IoT device, then smart-contract can help notify the patient about his/her decreased oxygen level as well as the healthcare provider. Upon seeing the data from the monitoring devices, the healthcare provider can take necessary actions.

There is no doubt that IoT devices are useful for healthcare during this pandemic situation. But security issues of the existing system is an ongoing problem. In this case, alternative technologies should be explored. Blockchain as a stand-alone technology which offers an intriguing solution for IoT security. It uses cryptography encryption to protect every transaction. By doing so, patients and healthcare providers can enhance their existing security and transparency levels to a new height.

1.3 Thesis Contribution

Our smart-contract-based system would enable real-time patient monitoring and medical interventions by delivering notifications to patients and medical professionals, while keeping a secure record.This will eliminate several security flaws associated with remote patient monitoring while also automating notification delivery to all parties involved.

1.4 Thesis Structure

This paper is structured as follows: First, we provide a Literature Review in Section 2. Then, a short background on IoT, blockchain technology in Section 3 and our system model is in Section 4. Then, we provide our implementation and code in Section 5 and System analysis in section 6. Finally, we discuss our future work in Section 7, and draw a conclusion.

Chapter 2

Literature Review

2.1 Overview

The following five paper we take as our literature review.

[2] Healthcare Blockchain System Using Smart Contracts for Secure Remote Automated Patient Monitoring.

[3] Blockchain Use in IoT for Privacy-Preserving Anti-Pandemic Home Quarantine.

[4] Use of Internet of Things (IoT) in Healthcare - A Survey.

[5] Health Information Exchange with Blockchain amid Covid-19-like Pandemics.

[6] Smart contracts - security patterns in the Ethereum ecosystem and solidity.

[7] The Case of HyperLedger Fabric as a Blockchain Solution for Healthcare Applications

2.2 Related Works

[2] In this paper, a smart contract is implemented to ensure security to the data generated from IoT devices in remote patient monitoring. Using a private blockchain based on the Ethereum protocol, a system has been created where medical sensors communicate with a smart device that calls smart contracts and writes records of all events on the blockchain. At the beginning, the raw medical data is sent to a master smart device (e.g. smartphone, tablet) for aggregation and formatting by the application. The formatted information is sent to the smart contract for full analysis along with customized threshold values. The smart

contract will evaluate the provided data and issue alerts to both the patient and healthcare provider. All treatment commands from the smart contract and healthcare provider will be recorded as complete in a blockchain transaction. Only authorized viewers can read the blocks and only designated nodes can execute smart contracts and verify new blocks as the system has a private and consortium-led blockchain.

[3] In this paper, they proposed a blockchain-based solution for safe home quarantine administration. Advanced cryptographic primitives are used to provide privacy and security for various events. Here, smart contract is created according to quarantine policy. A case study is given in an IoT system using a desktop computer, laptop, Raspberry Pi single-board computer, and the Ethereum smart contract platform to demonstrate the system's use. The results show that it is capable of meeting security, efficiency, and low-cost requirements. It also shows how the proposed method can obtain security margin under various threats by giving some potential attack scenarios such as Sybil attack, Denial of Service (DoS)/Distributed DoS (DDoS), Spoofing attack, Linking attack etc.

[4] In this paper, a survey is done on the increasing use of IoT devices and its advantages in the healthcare system. The paper describes the various Internet of Things (IoT) enable devices and its practices in the area of healthcare such as toddler, children, chronic care, monitoring of critical patients, operation theaters, medicine dispenser etc. This connected relationship between doctor and patient leads to a connected healthcare environment which promotes the quick flow of information and enables easy access to it. As a result of remote patient monitoring, a higher number of patients around the world will have access to adequate healthcare. In this paper, some IoT devices have been categorized based on their intended usage. All of these devices primarily communicate with users through mobile apps. The information generated by these devices is saved in the cloud. All cloud data is processed and delivered to the end user's mobile phone.

[5] In this paper, a blockchain-based data sharing system is described that takes advantage of immutability and decentralization qualities to offer a secure, user-centric approach to accessing and controlling sensitive medical data. The suggested system is based on a peer-to-peer network powered by the distributed Interplanetary File System and on-chain tagging, as well as the use of cryptographic generating techniques to enable secure medical data sharing. To ensure traceability and data integrity, the flow of information is orchestrated by a smart-contract deployed on a blockchain-based protocol. The framework's usefulness is proved by the application of the framework in a pilot study.

[6] In this paper, the predefined-code called smart contract is introduced along with the advantages, uses, limitations and the probable solutions to building efficient and secure smart contracts are discussed. Smart contracts are gaining traction in new corporate applications and the scientific community because they allow untrusted parties to express contract conditions in program code, removing the requirement for a trusted third party. The process of building high-performing and safe smart contracts on Ethereum is tough. This paper is developed numerous frequent security patterns based on an analysis of acquired data using grounded theory methodologies, which is presented in detail using solidity. The offered patterns described solutions to common security problems that lone developers can use to mitigate common attack situations.

[7] The main contributions of this research paper are the identification of healthcare application development requirements and the establishment of specific testing criteria, the design and implementation of instances on HyperLedger Fabric to analyze the identified requirements and criteria, and a critical analysis of HyperLedger Fabric adaptability for healthcare systems and the identification of future development requirements. They started by identifying key requirements of healthcare applications, and then presented the testing data in the form of scenarios which are implemented to test the identified requirements. Finally, they presented the details of the tools used to develop the proposed blockchain environment. Also, Various tests were developed to evaluate the network's important characteristics, ranging from security to fault tolerance. All the tests were done using HyperLedger Fabric and HyperLedger Composer. Lastly, the showed their experiment result and explained how security, regulation compliance, scalability and flexibility can be achieved with the help of hyperledger fabric.

2.3 Summary of Literature Review

Table 2.1 shows the summary of each paper's key features including the features that are not included.

Table 2.1: Literature Review Summary

Ref. No	Reference Paper	Main Features	Features Not Included
1	Healthcare Blockchain System Using Smart Contracts for Secure Remote Automated Patient Monitoring	Smart Contract, IoT in Healthcare and patient monitoring	Smart contract in Hyperledger Fabric
2	Blockchain Use in IoT for Privacy-Preserving Anti-Pandemic Home Quarantine	IoT, Privacy-Preserving, smart contract in Ethereum	Patient Monitoring, smart contract in Hyperledger
3	Use of Internet of Things (IoT) in Healthcare - A Survey	IoT, Healthcare and patient monitoring	Blockchain, Smart Contract in Ethereum and Hyperledger Fabric
4	Health Information Exchange with Blockchain amid Covid-19-like Pandemics	Electronic Health Records, Medical Informatics and Blockchain	Patient monitoring, Smart contract in Ethereum and Hyperledger Fabric
5	Smart contracts - security patterns in the Ethereum ecosystem and solidity	Smart Contract, Ethereum Virtual Machine, Solidity	IoT in Healthcare, Smart contract in Hyperledger Fabric
6	The Case of HyperLedger Fabric as a Blockchain Solution for Healthcare Applications	Blockchain, privacy preserve using Hyperledger Fabric	Smart contract in Ethereum and Hyperledger Fabric

Chapter 3

Background Study

3.1 The Internet of Things (IoT)

The Internet of Things (IoT) is a network of intelligent devices or objects that are connected to the internet. They should be individually addressable, programmable, and internet accessible. The Internet of Things (IoT) system's purpose is to link all "Things" together so that they can be programmable, intelligent, and communicate with humans. IoT is a collection of devices, including physical sensors, that are connected to the internet to gather and broadcast data without the need for human-to-human or human-to-computer contact. IoT has recently had a big impact on communication innovation [8].

IoT-based healthcare technology is becoming increasingly important in the medical profession, allowing for more effective and individualized healthcare. In the healthcare industry, IoT devices enable remote, continuous monitoring and maintenance of patients' health status in real-time rather than requiring patients to visit hospitals, which can result in a reduction in resources such as hospital beds, doctors, and nurses. Offering IoT-assisted individualized medical services improves the efficiency, efficacy, and cost performance of healthcare. Some advantages of IoT in healthcare sector are shown in fig. 3.1

Using IoT in the medical industry makes life easier, healthcare becomes less expensive, illness management becomes real-time, life quality improves, user end experience improves, patient care improves.The ultimate result is that people live longer and healthier lives. The progress of youngsters and elder parents in terms of disease management and prevention is closely evaluated. A significant change in the patient's health will trigger an immediate alert to various parties, saving lives and time [9].

Data security and privacy are major concerns with IoT. Patients' medical data privacy is sometimes jeopardized and sensitive information is disclosed as a result of bad health

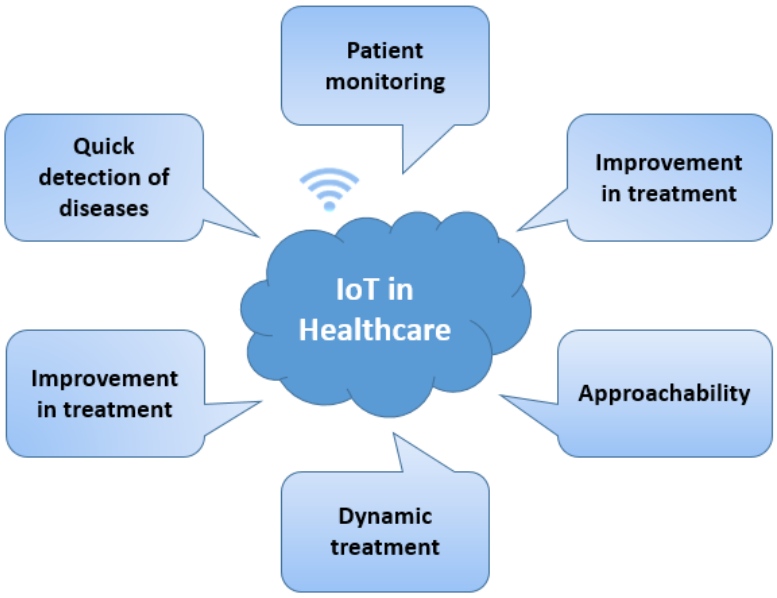

Figure 3.1: Advantages of IoT in Healthcare

information management. Several IoT devices, however, are lacking in data protocols and standards.

Many IoT devices collect data that is used to gain vital insights. However, the massive amount of data has an impact on decision-making quality, and privacy is a major concern in IoT. It's possible that the system will be hacked and patient data will be misused. As a result, maintaining security in communication sessions between IoT devices becomes the most important and difficult task.

3.2 Blockchain

Blockchain technology was introduced in 2008 by Satoshi Nakamoto [10]. It is a decentralized, distributed ledger consisting of blocks that hold records of transaction or tracking assets. The ledger is shared among all network participants (i.e., distributed) and the system's overall functioning is controlled by no single entity (i.e., decentralized) shown in Figure 3.2. Participants and miners are the nodes or agents that are involved in a blockchain network. The participants are the agents who carry out any transaction, whereas the miners are in charge of validating or rejecting a block. Contract records of anonymous parties, payment or reward histories, data ownership information may be found within a block.

A blockchain is also called a Chain of Blocks that are linked together using encryption. Each block within the blockchain is identified by a cryptographic hash of the previous block, a timestamp, and transaction data. Any changes within the block generate a different hash,

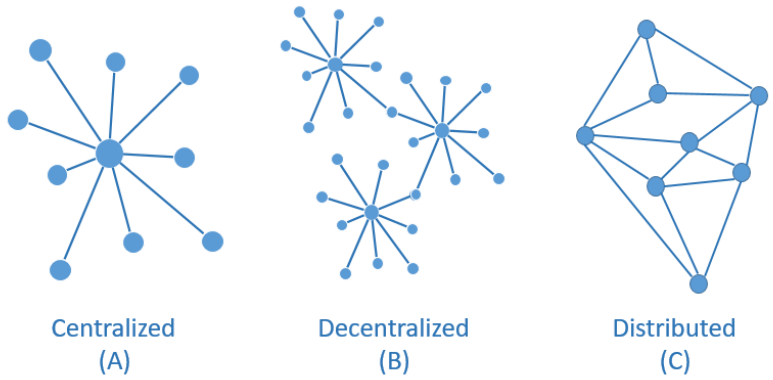

Figure 3.2: Centralized, decentralized and distributed network

which is generated depending on the block metadata. As a result, even minor changes within a block cause the entire chain to fail. Due to its decentralization, anonymity, and impossibility to be rewritten over, the popularity of blockchain is increasing day by day.

3.2.1 Blockchain Components

Blockchain technology consists of some core components, including cryptographic mechanisms (e.g., cryptographic hash function, cryptographic nonce) and data storage concept (e.g., ledger). The core components of a blockchain network are secured distributed ledger, asymmetric cryptography, transaction process, consensus mechanism etc shown in Figure 3.3.

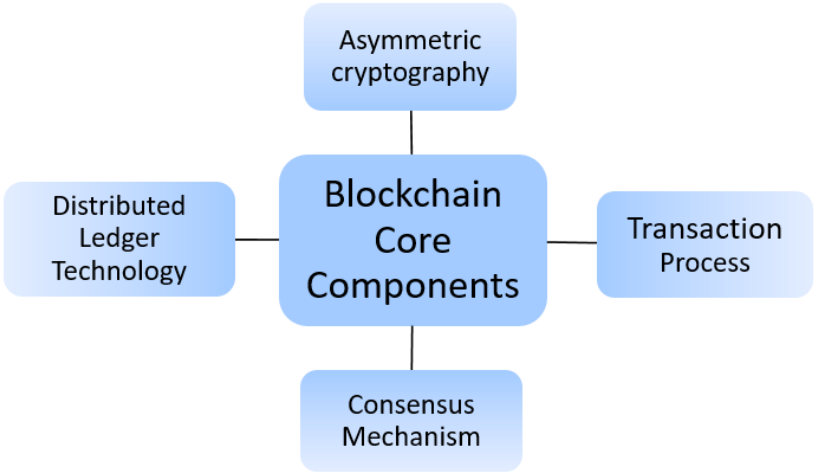

Figure 3.3: Four major components of blockchain operation

3.2.2 Cryptographic Hash Functions

The main concept that drives information validity and persistence in the blockchain is cryptographic hash functions. Hash functions convert any given data of any length to a predetermined output length. In the blockchain system, one-way hash functions are used. Consider the following equation:

$$n = h(m)$$

Where 'm' is a string of arbitrary length, 'h()' is a one-way hash function and 'n' is the result (Figure 3.3). It enables users to independently take input data, hash it, and return the same output, proving that the data has not changed. Even the slightest modification in the input results in a completely different output digest. Table 3.1 shows simple examples of this.

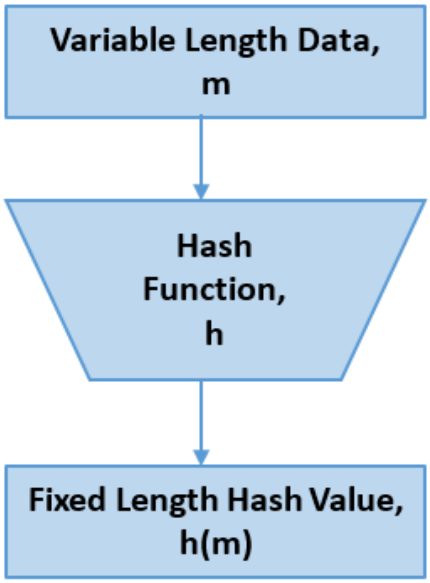

Figure 3.4: How Cryptographic Hash Functions work

3.2.3 Asymmetric-Key Cryptography

Asymmetric-key cryptography builds a trust relationship between blockchain users to verify transaction authenticity, non-repudiation, privacy, and integrity while keeping transactions public on a blockchain network [11]. For secure blockchain operation, the blockchain network makes use of the capabilities of public key cryptography. Apart from being on the same platform, users must have a digital wallet that is encrypted with the user's private key and accessible through suitable signatures generated with that private key. The public key of this wallet acts as the public bitcoin address, which is being changed with

each transaction to protect user privacy and anonymity. The user's private keys are used to digitally sign transactions and are kept private.

Table 3.1: Examples of some input data sample and corresponding SHA-256 generated values

Input data	SHA-256 hash value
hash	d04b98f48e8f8bcc15c6ae5ac050801cd6dcfd428fb5f9e65c4e16e7807340fa
Hash	a91069147f9bd9245cdacaef8ead4c3578ed44f179d7eb6bd4690e62ba4658f2
Hash demo	e0b195ed51b552713a2ee42f66b54c77f67169ab7c4218d79f5a3e40e0f7466d
Hash_demo	c1987ba6ffe602f7467703fe5b01c7426e0ef675fea44f0727792122a35bf5b5

3.2.4 Transactions

Blockchain allows people to share and exchange information with one another on a peer-to-peer basis. For blockchain network users, a cryptocurrency transfer between users might be considered a transaction [12]. A block contains the valid transactions, and each block can hold zero or more transactions. A blockchain network user sends information such as the sender's address and public key, an asymmetric cryptography-based digital signature, transaction inputs, and outputs when making a transaction. After each transaction, the state of the blockchain changes [13]. With so many transactions being made every second, it's critical to validate and verify the legitimate ones while discarding the fakes.

3.2.5 Blocks

A block is made up of two parts: a header and a body. The previous block's hash, a timestamp, Nonce, and the Merkle root are all included in the block header. The previous block generated a 256-bit hash value. As a result of the prior block's hash being kept in the current block, blockchain continues to develop as new blocks are created and connected to it. The timestamp is used to keep track of when a block is created. In the construction and verification of a block, a nonce (number only used once) is used. The Merkle root is the hash of all the hashes of all the transactions in the block. A transaction counter and a list of transactions make up the block body.

The blocks in a blockchain are linked because each block contains the hash value of the previous block. A basic blockchain containing a sequence of blocks is shown in fig. 3.5. If a transaction record is modified, the hash value is modified as well. As a result, changing a single block disconnects all following blocks. And so, the chain will be broken.

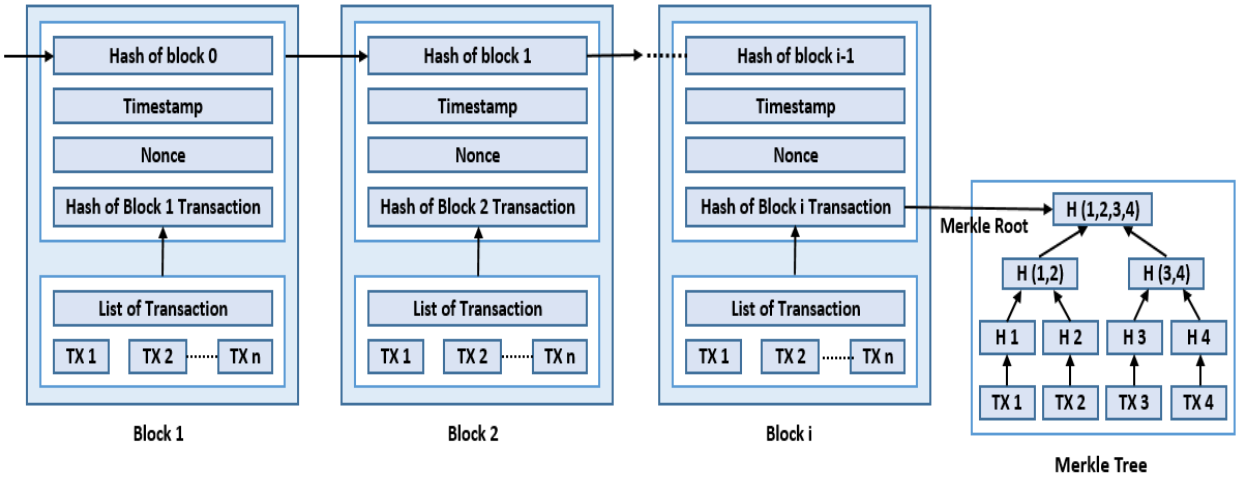

Figure 3.5: Chaining Blocks in a Blockchain Network

3.2.6 How blockchain transaction works

- A transaction is requested by the client. Cryptocurrency, contracts, records, or other information could be involved in the transaction.
- With the help of nodes, the requested transaction is broadcast to a peer-to-peer (P2P) network.
- A consensus mechanism is used by the network of nodes to validate the transaction and the user's status.
- The new block is then added to the existing blockchain after the transaction is complete. In a way that is both permanent and unchangeable.

3.2.7 Types of Blockchain

It is critical to be aware of and understand the various forms of blockchain that exist. Blockchain types are classified in a variety of ways, many of which are contradictory. This section will classify and compare various blockchain structures to aid in the development of blockchain-based software architectures. There are four different types of blockchain structures:

Public Blockchain

Public blockchains are similar to open source in that they allow anybody (permissionless) to join as users, miners, developers, or community members and view the network's transactions. On public blockchains, all transactions are entirely transparent, meaning that

anybody may review the transaction details. A public blockchain is completely decentralized in operation. Peer-to-peer techniques are used to carry out all operations and activities within a blockchain network. Currently, public blockchains are mostly utilized for cryptocurrency exchange and mining. Popular public blockchains like Bitcoin, Ethereum, and Litecoin may be familiar to everyone. Upon those public blockchains, nodes "mine" for bitcoin by solving cryptographic equations to generate blocks for the transactions requested on the network. The miner nodes are rewarded for their efforts with a small amount of bitcoin. Although public blockchains are incredibly safe, the total process is extremely slow due to the huge number of nodes.

Private Blockchain

Private Blockchains are permissioned blockchains that are controlled by a single entity. They are also known as managed blockchains. In a private blockchain, the central authority selects who can be a node. The central authority can also change any regulation of the blockchain (e.g., block consensus) and does not necessarily offer each node equal rights to perform functions. As public access to private blockchains is restricted, they are only partially decentralized and smaller in size. It's ideal for small businesses or groups that want to interact and share data but don't want their confidential information exposed on a public blockchain. Ripple (XRP) and Hyperledger are two popular instances of private blockchains. A token may or may not be associated with a private blockchain. It is less secure than a public blockchain due to the full power central entity, but it operates faster.

Consortium Blockchain

Private blockchains and consortium blockchains are sometimes considered independent categories. The most significant distinction between private and consortium blockchains is that consortium blockchains are administered by a group of organizations rather than a single entity. A consortium blockchain's consensus members are likely to be a group of preapproved nodes on the network. Consortium blockchains have more decentralization than private blockchains as they are semi-decentralized. It's appropriate in cases when numerous businesses collaborate on the same platform and need a common ground to conduct transactions and relay data. Consortium formation, on the other hand, can be a difficult process because it requires collaboration across a number of companies. Quorum, Hyperledger, and Corda are examples of consortium blockchains.

Hybrid Blockchain

A hybrid blockchain is one that attempts to combine the finest features of both private and public blockchain technologies. This means it combines the transparency and security benefits of a public blockchain with the privacy benefits of a private blockchain. The hybrid blockchain architecture is completely customizable, despite the fact that it is not accessible to the general public. Members of the hybrid blockchain can choose who can join the blockchain and which transactions are made public. This combines the best of both worlds and ensures that an organisation can work effectively with its stakeholders.

Permissionless vs. Permissioned Blockchains

All types of blockchains (i.e., public, private, consortium, hybrid) can be categorized as permissionless, permissioned, or both (shown in fig. 3.6). A permissioned blockchain requires prior authorization before it can be used, whereas a permissionless blockchain allows anybody to join.

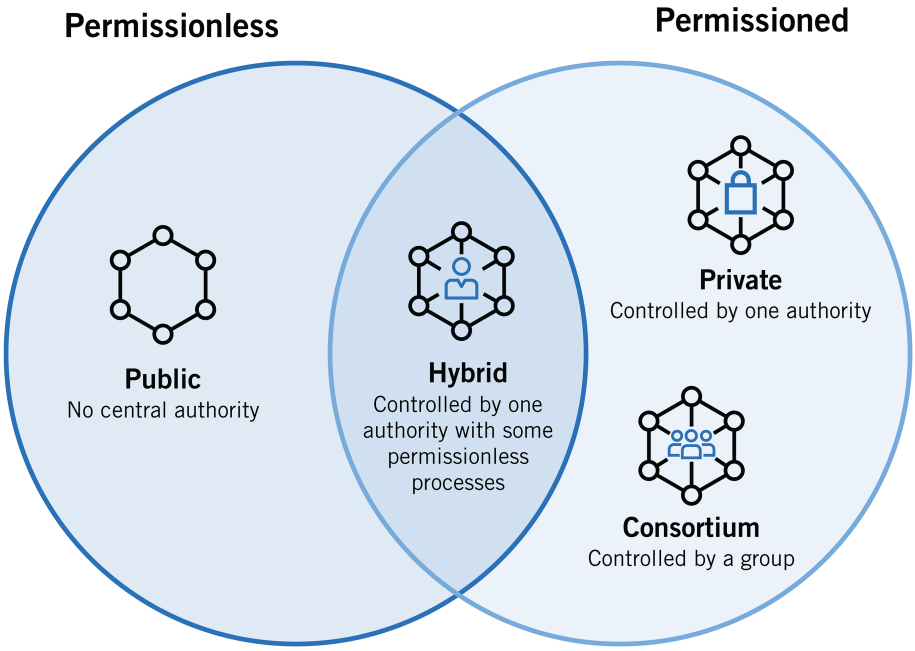

Figure 3.6: Type of Blockchain Network [1]

As there are more nodes to validate transactions, permissionless blockchains are more secure than permissioned blockchains. Due to the huge number of nodes and the large amount of the transactions, permissionless blockchains have long transaction processing times. Permissioned blockchains, on the other hand, are more efficient. As access to the

network is regulated, there are fewer nodes on the blockchain, resulting in faster transaction processing.

3.3 Consensus Algorithms

Any of the approaches used to obtain agreement, trust, and security across a decentralized computer network is referred to as a consensus mechanism.By assisting all nodes in the network in verifying transactions, consensus protocols serve as the backbone of blockchain.

In a blockchain network, confusion could develop if every node broadcasts blocks containing validated transactions. For example, we can see redundant transactions in different blocks if we consider a verified transaction that needs to be included into the block and numerous nodes broadcast about it. As a result, the ledger would be worthless; instead, a node must be chosen to be in charge of entering the transaction into a block. The nodes in a blockchain network must validate and agree to insert a new block to the main chain after successfully inserting all the verified transactions into the block.

Before adding a new block to the chain, each participating node must establish its validity and authenticity.The first node to reach consensus will be granted the right to insert the new block.It is impossible to edit or delete a block once it has been validated and added into the chain.

Various proof-based algorithms can be used to obtain consensus in a blockchain network [14].

3.3.1 Proof-of-Work (PoW)

Proof of work (PoW), which is utilized in Bitcoin, is the most frequently used and popular consensus mechanism. PoW has been a common choice in currency cryptography for a long time. A miner is a participant in the Proof-of-Work protocol. To earn the right to generate a new block, miners must complete a computational task, such as solving a puzzle problem. By finding a specific hash function, the nodes solve the problem. Each blockchain miner attempts to solve the hash value and determine a specific value to use as a nonce in order to match a predefined hash condition, such as determining the nonce value that sets the first 30 bits of the hash to zero. The complexity of the consensus mechanism can be increased by introducing a difficult puzzle. To reach consensus in the blockchain network, each miner looks for hash values that are less than or equal to a. certain target value [15]. The current block is broadcast to the whole network when a miner finds the target hash

value. Following that, all other nodes verify that the hash value is correct. If the block is valid, all nodes add it as a new validated block to their blockchain.

The major benefits of PoW are its high security, suitable scalability, and decentralization feature.But the PoW procedure consumes a lot of energy. More processing power is required as the puzzle becomes more challenging. As a result, resource-constrained nodes would be unable to solve a complex problem because the target hash generation time and success rate are dictated by the node's computational capability. PoW has a number of drawbacks, including a long block formation time, a high energy requirement, lower throughput, and hardware dependency.

3.3.2 Proof of Stake (PoS)

Proof of stake is the next widely used distributed consensus algorithm (PoS). The major goal of the PoS concept is to eliminate PoW's main flaw, which is energy inefficiency. The next block's creator is picked in a PoS-based mechanism employing various combinations of random selection, stake supply, and age, which can provide scalability. The node responsible for generating the next block is chosen using a quasi-random procedure that takes into account the assets stored in the nodes' wallets in this consensus mechanism. The miners do not earn any reward without the transaction fees because this approach does not require a lot of processing power to validate any proof. PoS has the following advantages: energy efficiency, high throughput, fast block creation time, scalability (less than PoW), and no reliance on or demand of special node hardware. Although PoS does not require as much compute power as PoW, it is dependent on the nodes with the greatest stake, therefore blockchain may eventually become centralized. Another typical PoS issue is "nothing at stake," which indicates that if a node in a blockchain network has nothing at stake and is misbehaving, the node will not be concerned about losing anything. As a result, there will be no barriers preventing the node from misbehaving in the network [16].

3.3.3 Practical Byzantine Fault Tolerance (PBFT)

Practical Byzantine Fault Tolerance (PBFT) is a replication technique that surpasses earlier approaches in handling Byzantine faults and inconsistency concerns generated by unreliable nodes in the system .Malicious software attacks today can be the result of faulty nodes' arbitrary or Byzantine behavior. The PBFT consensus algorithm is a collection of machine code that is used to solve Byzantine general problems. To add the next block to the chain via the PBFT method, all blockchain nodes must vote. More than two-thirds of the blockchain nodes must agree that the block and PBFT method are strong enough to resist

the opinions of the remaining one-third of nodes. Consensus will not be reached if this is not the case. As a result, PBFT obtains consensus faster and is more cost-effective than PoW [16].

3.3.4 Proof of Elapsed Time (PoET)

Another consensus approach that uses a fair lottery protocol is proof of elapsed time (PoET). The PoET consensus system prevents high energy consumption and resource utilization by blockchain nodes by using a fair lottery protocol. This consensus mechanism selects block winners and nodes that will receive mining rights based on a random elapsed time. Furthermore, the PoET consensus process promotes transparency by assuring that the PoET protocol's lottery results may be verified by outsiders or external participants. In conclusion, the PBFT workflow is comparable to the PoW consensus technique but without the significant node power consumption. By allowing a blockchain miner to sleep and nodes to move to other tasks, PBFT maintains power efficiency.

3.3.5 Proof of Importance (PoI)

The Proof-of-Importance (PoI) consensus model is used to determine which users are authorized to do required calculations when adding a new block to a blockchain and collecting fees. A PoI consensus model ranks blockchain miners according to the worth of their coins and the number of transactions they have completed in the respective cryptocurrency.The more transactions done from a node's wallet, the more likely it is that the node will be chosen to create the next block.

3.3.6 Proof of Capacity (PoC)

Miners in this method mine free coins using free space on their hard disk [17]. Before block mining begins, the algorithm uses the hard drives of the nodes, i.e. conducting calculation and storing solutions on the hard disks. The block is won by the node that stores the fastest (closest) solution to the block puzzle. The capacity of the user's hard drive can be utilized via this protocol. In this consensus approach, the more hard drive capacity a node has, the more likely it is to mine the next block and receive a reward.

3.4 Smart contract

Smart contracts are, in the context of blockchain, simply programs stored on a blockchain that run when predetermined conditions are met. It can receive or perform transactions like any address and can act as an immutable agreement. The concept of smart contacts can be traced back to the 1990s [18], which is much earlier than the blockchain technology.

The phrase "smart contract" was coined by cryptographer Nick Szabo in 1994 to describe a "computerized transaction mechanism that executes terms of a contract." Smart contracts in blockchain technology facilitate transactions between two parties and It only allows for the inclusion of validated transactions in the blockchain. However, to validate a transaction while performing transactions among blockchain nodes, proof-based consensus algorithms play a crucial role [19]. The basic idea behind using proof algorithms is that nodes in the blockchain network that perform and display adequate proof will be granted permission to add a new block to the main chain and receive a reward.

A smart contract is a crucial component that has aided the development of blockchain technology.Smart contracts are being integrated into blockchain technology with the primary purpose of reducing the need for trusted intermediaries, enforcement costs, fraud losses, and unintentional exceptions. When two parties agree on all of the conditions in a smart contract and begin a transaction, it is automatically written into the blockchain and carried out when certain clauses or criteria are met.

A smart contract's conditions are converted into computer code that can be executed and preserved in the form of program logic flows [20]. As transaction recorded is immutable in the blockchain, executed statements of the contract cannot be modified.

Smart contracts make it easier to control who has access to which smart contract functions. Smart contracts ensure that a smart contract's execution is predictable. When any smart contract condition is met, for example, the relevant function is immediately triggered. Furthermore, smart contracts assure contract enforcement, i.e., that all parties' actions are legal. Four stages of a smart contract's life cycle: creation, deployment, execution and finalization or completion [21].

3.4.1 Creation

Firstly, the parties involved negotiate the contract's rights, obligations, and prohibitions. After several rounds of negotiations and debates, all of the parties involved reach an agreement and prepare a draft of an initial contractual agreement. Then the natural-

language agreement is translated into a programmable language [21].

3.4.2 Deployment

The smart contract is validated before being deployed on top of a blockchain. And once it is published it cannot further be modified. This is because of blockchain immutability. Once a contract is published and implemented on the blockchain, it is accessible to all involved parties. Digital wallets store the digital assets of the parties involved and are also used to identify them [22].

3.4.3 Execution

The contractual clauses are monitored and reviewed when the smart contract is deployed in blockchain. Procedures that are sequential related are automatically executed once a contractual condition is met. Once a condition is triggered, logically connected declarative statements also automatically get executed and validated by blockchain miners. And the status is written on the blockchain.

3.4.4 Completion

The status of the parties involved is updated once a smart contract is evaluated. The blockchain records all completed transactions as well as all participants' current states.In the meantime, the parties involved transfer digital assets in accordance with the contract rules. The asset is unlocked after the digital asset transfer is completed, marking the end of the smart contract's life cycle.

3.5 Integrating Blockchain in IoT

The properties of the Internet of Things provide a number of research issues.

Heterogeneity

Heterogeneity in IoT-based systems refers to the variety of devices and their capabilities, as well as communication protocols and data formats. Heterogeneity also contributes to other important issues such as privacy, security, and interoperability.

Interoperability

Interoperability refers to the ability of IoT networks to exchange and interact with one another in order to collect extra and required data.Data interchange between multiple IoT networks becomes problematic due to the heterogeneity and decentralization of IoT systems.

Limitations in bandwidth, battery life, memory, and processing power

IoT devices may have limited resources in terms of bandwidth, battery life, memory, and processing power. Because of their resource constraints, IoT devices are vulnerable to malicious attacks.

Privacy

The technique of guaranteeing proper use of IoT data and information, i.e. not disclosing anything without user agreement, is known as privacy preservation. The structural complexity, decentralization strategy, varied devices, and limited resources make preserving the privacy of IoT networks difficult. Furthermore, combining cloud computing with IoT systems adds processing and computational power. Passing IoT data to a third party, on the other hand, could result in privacy breaches.

Security

IoT devices have a security vulnerability because of their heterogeneity, resource constraints, and decentralization [23]. Due to security countermeasures, traditional security solutions may not be viable for a resource-constrained IoT-based system. Furthermore, IoT systems may be exposed to a variety of attacks due to their limited resources and inability to update security firmware .

When making a choice, the traditional centralized IoT device ecosystem faces significant communication costs. In the case of a smart manufacturing system, the present centralized IoT method would result in high maintenance costs, such as releasing and distributing software updates to thousands of devices. Furthermore, IoT-based systems are more likely to incorporate insecure and malicious devices, which could disrupt data collecting and decision-making. Blockchain allows for a trustless, scalable peer-to-peer approach that provides data security and transparency [24].

Before integrating blockchain into an IoT application, we must first determine whether it is the right answer for that application. It's important to highlight that blockchain isn't appropriate for all IoT scenarios.

Decentralization

When the central coordinator of an IoT system is untrustworthy, we must develop a decentralized feature. However, if the consumers of IoT devices have complete faith in the central coordinator, blockchain integration is unnecessary.

Immutability

In a real-world IoT system, malicious behaviors can alter the retrieved data and transaction history of IoT devices. If there is a requirement to assure the immutability of local data within an IoT device or a cloud server in such a case, we can use blockchain, which has the unique property of providing immutability within IoT systems.

Secured transactions

IoT applications may be required to make transactions among current IoT nodes or external parties on occasion. During a transaction, atomicity should be secured to ensure that both parties obtain their intended asset. Blockchain can be adopted for IoT systems in this case.

Sequential logging with timestamps

In some IoT applications, it is necessary to gather data and identify it with a correct timestamp before storing it consecutively. We can use blockchain to create a timestamped sequential logging for those applications.

Microtransactions

For auditing or traceability, some IoT applications may need to track every record of transactions. Blockchain can come in use in this scenario.

Peer-to-peer exchanges

The majority of interactions in IoT networks take place between nodes and gateways, which means that IoT devices send data to a remote central server. Peer engagement, on the other hand, is uncommon but may be necessary in particular situations. In these cases, blockchain can enable safe peer-to-peer contact to achieve the application's aim.

3.6 Some Applications of Blockchain and IoT in Healthcare

The blockchain enables hospitals and health providers to keep track of and share a patient's medical records. In Healthcare it has a wide range of applications:

Drug Traceability

Drug traceability is frequently done in a centralized way, which leaves several needs unmet, such as privacy, data authentication, and system adaptability [25]. Many decentralized solutions have been used to address drug traceability difficulties. A Blockchain-based system called Drugledger is suggested [26] for the authenticity and privacy of tracing data. For drug traceability, Drugledger connects Blockchain with the drug supply chain. The actual movement of medications in conjunction with the supply chain, as well as the information that flows to the drug ledger network in the form of a drug chain network, are both maintained by Drugledger [27], introduces a strategy for preventing drug fraud by tracing each medicine along the supply chain.

The Gcoin Blockchain model (G stands for global governance) was proposed in [28] for the transparent flow of drugs, and this model also changed the drug supply chain system from regulating to surveillance and inspection of drugs, combining the government model with the DAO (Decentralized autonomous organization). The blockchain is utilized to create a trusting environment between two parties. There are numerous ways to create Blockchain, however in this article, Gcoin Blockchain is implemented using Consortium proof of work. The Gcoin Blockchain monitors every medicine in the same manner as bitcoin's Blockchain does. It increases the level of trust between buyers and sellers. Gcoin's major goal is to increase the efficiency with which data is transmitted. Future study will include a regulatory effect analysis as well as a system simulation test.

Electronic Health Record (EHR)

Electronic health records, according to the International Organization of Standardization, hold patient data in a digital format, are shared securely, and are only accessible by authorized authorities [29]. It contains confidential information about a person's health difficulties, and its major goal is to keep the patient's records up to date and provide effective service. There are a variety of Blockchain-based EHR systems available:

(a) Medrec is a paradigm for decentralized record management. It's a Blockchain approach for data exchange, authentication, and secrecy [30]. This approach incorporates all of the characteristics of Blockchain-like smart contracts as well as the idea of decentralized data.

(b) OmniPHR, A public health record (PHR) allows patients to access their information. This approach was created to keep data up to date and to distinguish between electronic health records and personal health records (PHR).

(c) Healthbank is a platform for securely storing and managing health information. This is a new start-up that also offers patients incentives in exchange for their participation [31].

3.7 Ethereum

Ethereum is a decentralized computing platform network that is open-source. The Ethereum network, like the Bitcoin network, is based on blockchain technology, which is essentially a digital public ledger that allows financial agreements to be validated and kept solely by software without the need for a third party [32]

The Ethereum network may be thought of as a secure database that is available to everybody. When new blocks of data are added, they are cryptographically chained to a parent block, establishing an uneditable record of the previous alterations. Ethereum is a blockchain that includes a cryptocurrency called Ether as well as smart contracts. Ethereum is the world's largest blockchain running smart contract. A smart contract is a computer program that, once launched, acts autonomously and mandatorily according to predefined program logic and cannot be altered.

The network of Ethereum is fueled by computational power. In practice, this means that individuals and businesses use their computers to execute certain software, or nodes. Anyone may set up a node on their PC. The Ethereum network relies on node operators to

perform transactions. For running the technology and software required to conduct these transactions, these operators charge a fee. Because the costs keep the network functioning, they're termed gas fees. They're compensated in ether (ETH).

Consider the various applications for which a huge network of computers may be used. Ethereum utilizes it to perform peer-to-peer transactions and monitor who owns the ether money, similar to Bitcoin. On the network, developers may also develop and run decentralized applications (dApps). "Smart contracts" connect the dApps to the Ethereum network. On Ethereum, smart contracts are often written in a high-level language (such as Solidity) and compiled into bytecode. When the bytecode is invoked after being deployed on Ethereum, it will execute on the blockchain. Solidity has a syntax that is similar to JavaScript, although it is written in a completely different way. Like other high level language, solidity language supports inheritance, libraries, and other complex user-defined types. Damage control, modularity, check effects are the common security rules that have been acquiredn over the short time that Solidity has been in use.

DApps rely on the Ethereum network, which is decentralized and open-source, and cannot be controlled by a single entity. Once a dApp is published to the Ethereum network, it cannot be removed – even if the original creator wishes to do so or the company dissolves. Users may be able to utilize dApps under a pseudonym due to the decentralized system's privacy. Third parties, such as companies and governments, may have less control and censorship as a result.

Several research studies have been conducted on Ethereum's security and performance [33–37].

3.8 Hyperledger Fabric

Hyperledger Fabric is one of the Hyperledger blockchain initiatives. It's a development project that is open source. It has a ledger, uses smart contracts, and is a mechanism by which participants control their transactions, just like other blockchain technologies. The Linux Foundation established it in 2015 to develop cross-industry blockchain technologies. Fabric's architecture is highly modular and flexible, allowing for innovation, variety, and optimization across a wide range of industries, including banking, finance, insurance, healthcare, human resources, supply chain, and even digital music delivery [38]. Fabric is the first distributed ledger framework that allow smart contracts written in general-purpose programming languages like Go, Java, and Node.js rather than domain-specific languages (DSL).

However, the Hyperledger Fabric architecture differs significantly from other blockchain architectures. Hyperledger Fabric is a permissioned and private architecture. As a result, it's an ideal solution for any organization. Organizations prefer privacy and are unable to rely on public platforms to provide it. Because there is a lot of sensitive information in enterprise organizations that they can display to their market competitors. As a result, having Hyperledger Fabric architecture on their side may truly assist them take advantage of all of the technology's benefits while also maintaining their integrity.

Hyperledger Fabric consensus is a flexible model. It also has a variety of consensus mechanisms, pluggable options, multiple ledger formats, and many other features. It allows for more effective customization of the platform to match specific use cases and trust models. Fabric can reap the benefits of consensus mechanisms that don't involve the use of a native cryptocurrency to incentivize costly mining or fuel smart contract deployment.

Hyperledger Fabric has been designed with a modular architecture in consideration. As a result, the Hyperledger Fabric consensus, ledger types, tokens, and other features can be changed. The platform was built from the ground up to be adjusted to fit the needs of a wide range of organizational use cases. Pluggable ordering service, pluggable membership service provider, optional peer-to-peer gossip service, pluggable endorsement and validation policy enforcement are some of the modular components of hyperledger fabric.

Transaction processing of hyperledger fabric is separated into three phases: distributed logic processing and agreement ("chaincode"), transaction ordering, and transaction validation and commitment. This separation confers several advantages: Fewer levels of trust and verification are required across node types, and network scalability and performance are optimized.

3.8.1 Key Characteristics of Hyperledger

Here are some of Hyperledger Fabric's primary characteristics and how they vary from other distributed ledger platforms.

- Permissioned architecture
- Highly modular
- Pluggable consensus
- Low latency
- Open smart contract concept — gives opportunity to develop any desired solution model (account model, UTXO model, structured data, unstructured data, etc)

- A flexible approach to data privacy: use 'channels' to isolate data, or use private data 'collections' to share private data only with those who need to know.
- Support for smart contracts in multiple languages: Go, Java, Javascript
- Developed for continuous operations, including asymmetric version support and rolling upgrades
- Governance and versioning of smart contracts
- Flexible endorsement approach for gaining consensus across required organizations
- Data can be queried (key-based queries and JSON queries)

Chapter 4

System Model

4.1 Overview

In this pandemic, patients at higher risk are being monitored more and more as the days go by. IoT devices are being used to do the monitoring because of the ever-advancing technology. Patients who have diabetes, heart disease, asthma, lung disease etc. are being monitored by devices such as, pulse oximeter, heart rate monitor, insulin pump, blood pressure monitor etc. This raises the concern of security of health information of patients.

The medical state of such patients can worsen anytime. It can be because of being infected by covid-19 or not. Either way the patient will require medical attention. During this deadly pandemic, the hospitals are bound to follow a lot more protocols and formalities leading to time consuming tests as there are isolated covid wards for the covid positive patients. But in many cases, there might not be enough time to follow all the protocols. In such times, the data provided through IoT devices will come in handy for the doctors to have a primary assumption of the cause of degradation.

We propose a model which is a deployable smart-contract implemented on blockchain to ensure more security to handling sensitive patient data while triggering necessary emergency alerts and taking additional useful information of the patient in need of medical attention as early as possible.

Many institutions, such as hospitals and pharmacies, are available to provide health services. Human lives are frequently lost to dangerous illnesses. It is necessary to continually monitor the patient's health in order to offer resources to the patient on time. There are various scenarios in which patients are compelled to wear IoT-based monitoring devices in order to continually check their health. These gadgets capture and store medical data on the cloud. Healthcare has been quite popular in recent years, but the key challenge is

ensuring the security of patient data. Many examples of healthcare data leakage have been reported when data is stored in the cloud [39]. Many countries place a high value on the privacy and security of patient information. The Health Insurance Portability and Accountability Act (HIPAA) is utilized in Europe to protect patient data and ensure secure medical data transfer [40].

This framework's goal is to use healthcare devices to read patients' vital signs and exchange that information with approved doctors and hospitals over a secure Blockchain network. When blockchain and IoT are combined, the result is always a more secure network. The notion of distributed ledger, cryptography, and consensus algorithms are some of the Blockchain elements that make IoT networks more secure. The Blockchain's decentralization idea allows for data transparency. This Blockchain-based architecture is presented for a remote patient monitoring system. The framework aims to remove roadblocks and create a more secure network.

4.2 Proposed Model

The patient will be equipped with various medical devices such as oximeter, heart rate monitors etc. The data from these devices is sent to a device like a smartphone that is capable of sending the data to a smart contract. The formatted data will then be analyzed based on the smart contract which shown in Figure 4.1

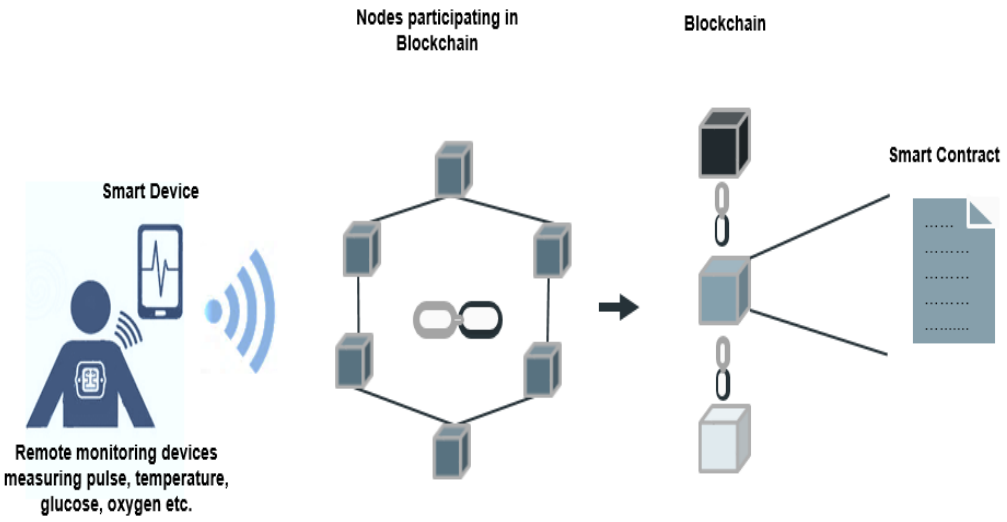

Figure 4.1: Formatted data obtained based on edge computing is being sent for analysis using smart contract in the blockchain system

Our written smart contract will analyze the data for any anomalies. If no anomalies

found and the patient data is within the predefined range then no alerts are sent assuming the patient is stable. But if any anomaly is found then necessary alerts will be sent to both the patient and the healthcare provider as shown in Figure 4.2.

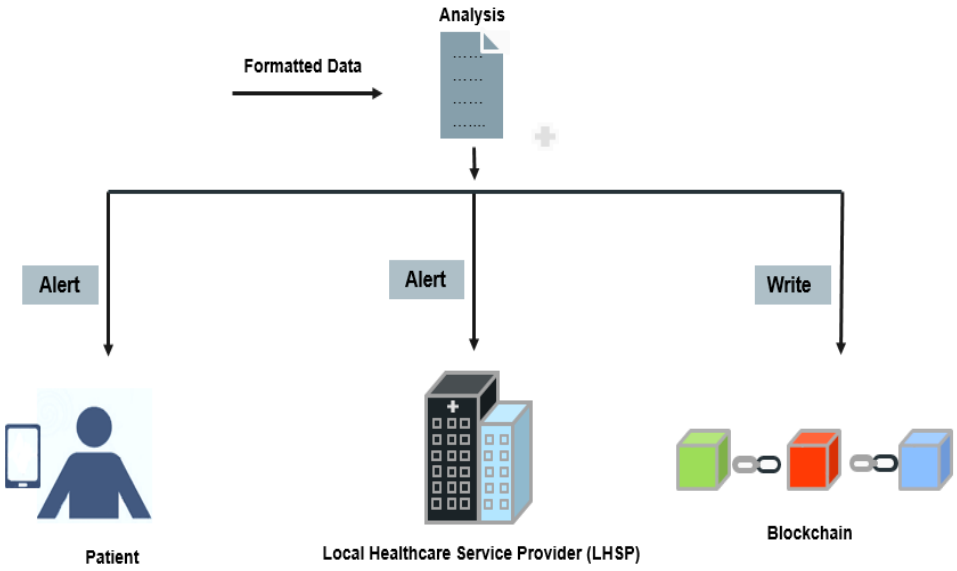

Figure 4.2: Smart contract being initialized sends necessary alerts based on the data and stores the information on the blockchain network permanently

Along with the necessary alerts, patient data will be sent to the healthcare provider. This will help the healthcare provider to estimate primarily that whether the patient condition got worse or not. It will allow the patients to be treated as soon as possible in cases of emergency.

4.3 Summary of Proposed Model

Our proposed model is a deployable smart-contract implemented on blockchain to ensure more security to handling sensitive patient data while triggering necessary emergency alerts and taking additional useful information of the patient in need of medical attention more conveniently.

Chapter 5

Implementation

5.1 Implementation Environment

For Ethereum, we used the Remix Browser IDE To write our Solidity code. Because it comes with a number of features and provides a thorough development experience, the remix is the ideal option for building smart contracts. It also allows us to experiment without having to use Ether. We created our smart contract code for Ethereum by visiting *https://remix.ethereum.org*.

For Hyperledger Fabric, the operating system we were using was Ubuntu (version 20.04.1 LTS). Fabric required git 2.9+, npm v5.x, Docker Engine 17.03+, and Docker-Compose 1.8+ before it could be installed. Fabric was finally installed using the official github repository (*HyperLedger.github.io*). We used JavaScript language and VScode for the coding part.

5.2 Implementation Procedure

For implementing the proposed system first, we try to show that smart contracts can be used to create the backbone of the system as a proof-of-concept. The smart contract is created using solidity, programming language used in Ethereum with a .sol extension file.

In this system the smart device will firstly call ***HealthCareMonitor*** smart contract, which will handle the data given by the smart device. ***HealthCareMonitor*** smart contract will create other smart contracts for specify medical devices and their data. For instance, when the Oximeter provides data, the smart device will call ***HealthCareMonitor*** smart contract. Which calls object of ***HealthCareMonitor*** and function ***OxygenMonitor()*** is called.

As parameter, the smart device will pass the oximeter data. The function will create the object of ***OxygenMonitor()*** contract and pass the data as parameter to the ***check()*** function. This function will analyze the given data and return a string based on the data.There are some other function called ***HeartRateMonitor*** and ***GlucoseMonitor*** which will work same as ***OxygenMonitor()*** according to their given data pattern.The execution flow of the whole system are shown in Figure 5.1

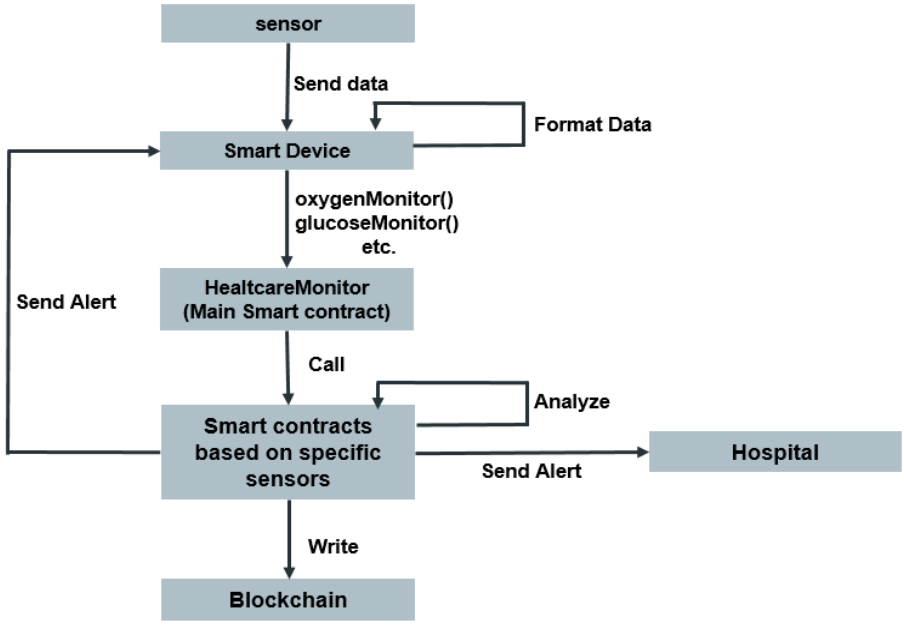

Figure 5.1: Execution flow of the system

In the case of deploying a similar smart contract in hyperledger fabric, The ledger has mainly the world state and the blockchain.The blockchain is made up of hash-linked blocks. The data kept on the ledger is sequential and tamper-resistant because each block header comprises a hash of the current block and a hash value of the previous block. Due to the reduced quantity of planned data in this paper's implementation, all data is still saved in the global ledger rather than the medical cloud. To fulfill the function of searching for cases by keywords, the ledger employs the Couch DB database as the state database (world state). The medical data that will be processed in chaincode as key-value pairs. For addressing specialized business logic, Fabric 2.0x now supports Couch DB as a database. We employ the Raft consensus, which is officially suggested by Fabric, for the implementation. Other consensus techniques may become available for smart contract deployment in the future. The Hyperledger Fabric removes mining and incentive mechanisms by requiring each member of the consortium blockchain to deploy their nodes in order to offer data security and user service.

For the sake of simplicity, we set up a network called ***test-network*** using scripts from

the ***fabric-samples*** repository, which is designed to help you learn about Fabric by allowing us to operate nodes on your local workstation. The implementation uses Javascript for the smart contract. The world ledger stores all the data. As an example, the implementation uses two organizations (***org1, org2***), three nodes, including two peers (***peer0.org1.example.com*** and ***peer0.org2.example.com***) and one ordereing service (***orderer.example.com***) and a channel ***mychannel***.The network components are shown in table 5.1

Table 5.1: Network deployment of Blockchain in Hyperledger Fabric

Organization	org1, org2
Peers	peer0.org1.example.com, peer0.org2.example.com
Orderer	orderer.example.com
Channel	mychannel
Chaincode	Fabcar

The deployment of the blockchain follows the simplest form because this paper is merely a demonstration of the implementation. Depending on the complexity of smart contracts and business logic, the number of organizations, nodes, and channels can be raised. In the deployed smart contract of the hyperledger fabric we declared a key and a structure for patient data. As the smart contract can not be changed after deployment, it's impossible to change the key and the patient data structure, thus ensuring the key-value pairs can always be used to process medical data.The function ***writenewData()*** of the smart contract is used to pass vital patient data, which calls the corresponding functions (***heartRateMonitor()***, ***GlucoseMonitor()*** and ***OxygenMonitor()***) to the medical data to analyze the data and store the status based on that. Here, from the healthcare provider side a simple query function of the smart contract can be called to see the patient data and status.

Chapter 6

System analysis

6.1 Comparison to traditional systems

Blockchain is a very new technology, and applications like ours differ significantly from existing systems that perform comparable services. We compare the properties of our proposed blockchain-based system to those of a remote patient monitoring system that depends only on more traditional communication and data storage methods like cloud computing and relational databases.

Privacy

Transmissions in a typical system are encrypted to prevent any identifying information that might be traceable back to the end users. However, with our system, anonymous addresses will secure patients' identities, thus no linkages between patients and their data will be possible.

Immutability

Databases are subject to both unintentional and malicious manipulation in traditional systems. Verified blocks are irreversible and resistant to all sorts of modification in our system.

Availability

Manual backups are required, and redundancies must be implemented to provide

service in the event of typical system failures. However, because all nodes have a copy of the ledger with every recorded transaction, we enable greater fault tolerance and service availability. Algorithms like PBFT can handle failures of one or more nodes.

Transparency

Patients in traditional systems have no direct control over their data and are unable to link remote transactions to their records. Patients, on the other hand, can link remote monitoring actions to their medical records in our system while keeping security and control.

Traceability

Health records and logs can be altered, and identification in a typical system may not be guaranteed. Blockchain transactions, on the other hand, can be tracked back to their source with assured immutability and are signed by the validators.

Speed

Transactions in a typical system are only restricted by network transmission speeds.A minor delay may occur in our system, depending on how long it takes to verify a block.

6.2 Security analysis

We presume that all cloud and IP protocols are encrypted for the purpose of simplicity, and we don't add any more. Authentication must be included within the smart device for any parties who may be utilising the data (i.e. Patients have the right to view but not alter their own data, while healthcare providers have the right to edit their patients' smart contract criteria). To make a block valid, the proposed consortium blockchain requires a majority of signatures from consortium members, preventing one entity from manipulating the ledger. Only authorized parties have access to the blockchain's viewing capabilities (patients, caregivers,etc). Second, no sensitive information about patients is maintained directly on the blockchain. The blockchain ledger that stores the transactions also acts as a separate layer of protection for both the patient and the healthcare provider.

The Privacy Rule of the Health Insurance Portability and Accountability Act of 1996, or HIPAA, governs the electronic transmission of data. The most evident feature of the law that applies to this system is that data that cannot be identified as belonging to a specific patient

is not covered by HIPAA. The blockchain data, as envisioned, merely stores transaction information, not sensitive health data.

Furthermore, because the account addresses anonymize the patients, the information cannot be easily traced to a specific person, making it HIPAA-compliant. Personal data can only be disclosed to individuals on request or to the Department of Health and Human Services (HHS) in circumstances of inquiry or enforcement action, according to the Privacy Rule. Another advantage of our technology is that it provides HHS with authenticated and immutable records of a patient's monitoring for dispute resolution and investigation. Because we apply reasonable safeguards to secure and enable tiered access to PHI, our system is HIPAA compliant.

Chapter 7

Conclusion and Future Work

Nowadays IoT technology is used in a variety of fields, including agriculture, healthcare, and smart cities. In the domain of healthcare, IoT is used for applications such as frequent monitoring of a patient's health, medicine traceability, and so on. However, IoT has a number of security vulnerabilities that can be addressed by combining IoT with Blockchain. Along with healthcare, blockchain technology assures that patients' sensitive health-related records are protected from alteration and leaking.

In this paper, we attempted to identify possible methods for IoT and Blockchain technology to be integrated into the Healthcare sector in order to improve overall performance and strengthen the current Healthcare sector. Here, we presented proof of concept of integration Blockchain in healthcare, where we utilized blockchain network and its smart contracts to provide a way for secure remote patient monitoring. Here we explored what will happen if we deploy the contracts on different types of blockchain networks.

As we did not implement a decentralized application that calls the contracts, we in the future plan to do so to strengthen our position in integrating Iot blockchain in the healthcare sector.

References

[1] E. W. Kathleen E. Wegrzyn, "Types of blockchain: Public, private, or something in between."

[2] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," *Journal of medical systems*, vol. 42, no. 7, pp. 1–7, 2018.

[3] J. Zhang and M. Wu, "Blockchain use in iot for privacy-preserving anti-pandemic home quarantine," *Electronics*, vol. 9, no. 10, p. 1746, 2020.

[4] A. S. Yeole and D. R. Kalbande, "Use of internet of things (iot) in healthcare: A survey," in *Proceedings of the ACM Symposium on Women in Research 2016*, pp. 71–76, 2016.

[5] K. Christodoulou, P. Christodoulou, Z. Zinonos, E. G. Carayannis, and S. A. Chatzichristofis, "Health information exchange with blockchain amid covid-19-like pandemics," in *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 412–417, IEEE, 2020.

[6] M. Wohrer and U. Zdun, "Smart contracts: security patterns in the ethereum ecosystem and solidity," in *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, pp. 2–8, IEEE, 2018.

[7] M. Antwi, A. Adnane, F. Ahmad, R. Hussain, M. H. ur Rehman, and C. A. Kerrache, "The case of hyperledger fabric as a blockchain solution for healthcare applications," *Blockchain: Research and Applications*, p. 100012, 2021.

[8] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[9] C.-H. Huang and K.-W. Cheng, "Rfid technology combined with iot application in medical nursing system," *Bulletin of Networking, Computing, Systems, and Software*, vol. 3, no. 1, pp. 20–24, 2014.

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[11] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," *arXiv preprint arXiv:1906.11078*, 2019.

[12] M. Swan, *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.

[13] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, 2014.

[14] M. Salimitari and M. Chatterjee, "A survey on consensus protocols in blockchain for iot networks," *arXiv preprint arXiv:1809.05613*, 2018.

[15] W. Wang, D. Hoang, Z. Xiong, D. Niyato, P. Wang, P. Hu, and Y. Wen, "A survey on consensus mechanisms and mining management in blockchain networks, 1–33," *Preprint*, 2018.

[16] M. Salimitari and M. Chatterjee, "A survey on consensus protocols in blockchain for iot networks," *arXiv preprint arXiv:1809.05613*, 2018.

[17] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, "Proofs of space," in *Annual Cryptology Conference*, pp. 585–605, Springer, 2015.

[18] N. Szabo, "Formalizing and securing relationships on public networks," *First monday*, 1997.

[19] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system. manubot," *Tech. Rep.*, 2019.

[20] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[21] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020.

[22] C. Sillaber and B. Waltl, "Life cycle of smart contracts in blockchain ecosystems," *Datenschutz und Datensicherheit-DuD*, vol. 41, no. 8, pp. 497–500, 2017.

[23] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013.

[24] A. Douglas, R. Holloway, J. Lohr, E. Morgan, and K. Harfoush, "Blockchains for constrained edge devices," *Blockchain: Research and Applications*, vol. 1, no. 1-2, p. 100004, 2020.

[25] J. Li, J. Cheng, N. Xiong, L. Zhan, and Y. Zhang, "A distributed privacy preservation approach for big data in public health emergencies using smart contract and sgx," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 65, no. 1, pp. 723–741, 2020.

[26] Y. Huang, J. Wu, and C. Long, "Drugledger: A practical blockchain system for drug traceability and regulation," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1137–1144, IEEE, 2018.

[27] V. Ahmadi, S. Benjelloun, M. El Kik, T. Sharma, H. Chi, and W. Zhou, "Drug governance: Iot-based blockchain implementation in the pharmaceutical supply chain," in *2020 Sixth International Conference on Mobile and Secure Services (MobiSecServ)*, pp. 1–8, IEEE, 2020.

[28] J.-H. Tseng, Y.-C. Liao, B. Chong, and S.-w. Liao, "Governance on the drug supply chain via gcoin blockchain," *International journal of environmental research and public health*, vol. 15, no. 6, p. 1055, 2018.

[29] K. Häyrinen, K. Saranto, and P. Nykänen, "Definition, structure, content, use and impacts of electronic health records: a review of the research literature," *International journal of medical informatics*, vol. 77, no. 5, pp. 291–304, 2008.

[30] 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)*, pp. 25–30, IEEE, 2016.

[31] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *2016 IEEE 18th international conference on e-health networking, applications and services (Healthcom)*, pp. 1–3, IEEE, 2016.

[32] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, 2014.

[33] T. Chen, Z. Li, H. Zhou, J. Chen, X. Luo, X. Li, and X. Zhang, "Towards saving money in using smart contracts," in *2018 IEEE/ACM 40th International Conference on Software Engineering: New Ideas and Emerging Technologies Results (ICSE-NIER)*, pp. 81–84, IEEE, 2018.

[34] N. Grech, M. Kong, A. Jurisevic, L. Brent, B. Scholz, and Y. Smaragdakis, "Madmax: Surviving out-of-gas conditions in ethereum smart contracts," *Proceedings of the ACM on Programming Languages*, vol. 2, no. OOPSLA, pp. 1–27, 2018.

[35] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020.

[36] T. Chen, X. Li, Y. Wang, J. Chen, Z. Li, X. Luo, M. H. Au, and X. Zhang, "An adaptive gas cost mechanism for ethereum to defend against under-priced dos attacks," in *International conference on information security practice and experience*, pp. 3–24, Springer, 2017.

[37] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 254–269, 2016.

[38] https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatis.html#hyperledger-fabric. Accessed: 2010-09-30.

[39] H. Jin, Y. Luo, P. Li, and J. Mathew, "A review of secure and privacy-preserving medical data sharing," *IEEE Access*, vol. 7, pp. 61656–61669, 2019.

[40] J. K. O'herrin, N. Fost, and K. A. Kudsk, "Health insurance portability accountability act (hipaa) regulations: effect on medical record research," *Annals of surgery*, vol. 239, no. 6, p. 772, 2004.

Appendix A

Codes for Appendix

```
pragma solidity ^0.4.0;

contract OxygenMonitor {

    uint maxlevel=100;
    uint minlevel=90;

    // normal  range for oxygen level is 95-100
    function check(uint oxylevel ) public constant returns (string
       memory){


        if(oxylevel>=95 )
        {
            return('Oxygen level is okay');
        }
        else if(oxylevel<95 && oxylevel>90)
        {
            return('Oxygen level is slighty low');
        }
        else if(oxylevel<=90 )
        {
            return('Oxygen level is too low');
        }

    }
}


contract HeartRateMonitor {

    uint min=60;
```

```
32      uint max=100;
33
34      // normal  range for heart rate is 60-100
35      function check(uint bpm) public constant returns (string memory){
36
37          if(bpm < min||bpm > max){
38              if(bpm <= min){
39
40                  return ('heart rate is low');
41              }
42              else if(bpm>max && bpm<max+20){
43
44                  return ('heart rate is high');
45              }
46               else if( max+20 <= bpm ){
47
48                  return ('heart rate is too high');
49              }
50          }
51          else{
52
53              return ('heart rate is in normal range');
54          }
55      }
56  }
57
58  contract GlucoseMonitor {
59
60      uint min=80;
61      uint max=200;
62
63      // normal  range for blood glucose Level is 80-140mg/dl
64      function check(uint glucoseLevel) public  returns (string memory)
            {
65
66          if(glucoseLevel < min||glucoseLevel > max-60){
67              if(glucoseLevel <= min)
68              {
69                  return ('Glucose level is low');
70              }
71              else if(glucoseLevel>140 && glucoseLevel<max)
72              {
73                   return ('Glucose level is in high');
74              }
75              else if(glucoseLevel>=max)
76              {
77                   return ('Glucose level is Too high');
```

```
78              }
79
80          }
81          else{
82
83              return ('Glucose level is in normal range');
84          }
85      }
86  }
87
88  contract HealthCareMonitor{
89      uint[] lastbpm;
90      uint[] lastglucoseLevel;
91      uint[] lastoxygenLevel;
92      uint i;
93      uint j;
94      uint k;
95
96      function heartRateMonitor(uint bpm)public  returns (string memory
            ){
97
98          HeartRateMonitor hrm = new HeartRateMonitor();
99
100         lastbpm.push(bpm);
101
102         return hrm.check(bpm);
103     }
104
105     function glucoseMonitor(uint glucoseLevel)public  returns (string
            memory){
106
107         GlucoseMonitor gm = new GlucoseMonitor();
108         lastglucoseLevel.push(glucoseLevel);
109
110         return gm.check(glucoseLevel);
111     }
112
113     function oxygenMonitor(uint oxygenLevel)public  returns (string
            memory){
114
115         OxygenMonitor oxm = new  OxygenMonitor();
116         lastoxygenLevel.push(oxygenLevel);
117
118         return oxm.check(oxygenLevel);
119     }
120
121     function patientstate()public view returns (uint[] memory , uint
```

```
        [] memory , uint[] memory ,string memory){
122
123         return(lastbpm,lastglucoseLevel,lastoxygenLevel, "string
                check");
124     }
125
126     function glucose()public view returns (uint[]  ,string memory){
127
128         return(lastglucoseLevel, "glucose");
129     }
130
131
132 }
```

Appendix B

Codes for Appendix

```
'use strict';

const { Contract } = require('fabric-contract-api');

class FabCar extends Contract{

   //initledger
   //writeData
   //readData
   async initLedger(ctx){
    await ctx.stub.putState("test","*****")
    return "success"
  }

  key1 = "pnew"
  patient1 ={
  "name":"",
  "sugar":[],
  "oxy":[],
  "heartrate":[],
  "heartratestatus":"",
  "sugarstatus":"",
  "oxystatus":""
  }

  async writenewData(ctx,valuename, valueoxy,vaulehrt,valuesugar){
    this.patient1.name=valuename.toString();
    this.patient1.oxy.push(Number(valueoxy));
    this.patient1.sugar.push(Number(valuesugar));
    this.patient1.heartrate.push(Number(vaulehrt));
    this.patient1.heartratestatus= await this.heartRateMonitor(
       vaulehrt);
```

```
32      this.patient1.sugarstatus= await this.GlucoseMonitor(valuesugar);
33      this.patient1.oxystatus= await this.OxygenMonitor(valueoxy);
34
35      console.log("**********",this.patient1);
36      await ctx.stub.putState(this.key1, Buffer.from(JSON.stringify(
            this.patient1)));
37      return Buffer.from(JSON.stringify(this.patient1));
38    }
39
40    async heartRateMonitor(vaulehrt){
41
42      let bpm= await Number(vaulehrt);
43      let min=60;
44      let max=100;
45    // normal  range for heart rate is 60-100
46
47      if( bpm < min||bpm > max){
48        if(bpm <= min){
49          return 'heart rate is low';
50        }
51        else if(bpm>max && bpm<max+20){
52          return 'heart rate is high';
53        }
54        else if( max+20 <= bpm ){
55          return  'heart rate is too high';
56        }
57      }
58      else{
59        return 'heart rate is in normal range';
60      }
61    }
62
63
64    async GlucoseMonitor (valuesugar){
65      let glucoseLevel= await Number(valuesugar);
66      let min=80;
67      let max=200;
68      // normal  range for blood glucose Level is 80-140mg/dl
69
70      if(glucoseLevel < min||glucoseLevel > max-60){
71        if(glucoseLevel <= min){
72          return ('Glucose level is low');
73        }
74        else if(glucoseLevel>140 && glucoseLevel<max){
75          return ('Glucose level is in high');
76        }
77        else if(glucoseLevel>=max){
```

```
78          return ('Glucose level is Too high');
79        }
80      }
81      else{
82         return ('Glucose level is in normal range');
83      }
84    }
85
86
87    async OxygenMonitor (valueoxy){
88      let oxylevel = await Number(valueoxy);
89      let maxlevel=100;
90      let minlevel=90;
91      // normal  range for oxygen level is 95-100
92
93      if(oxylevel>=95 ){
94        return('Oxygen level is okay');
95      }
96      else if(oxylevel<95 && oxylevel>90){
97        return('Oxygen level is slighty low');
98      }
99      else if(oxylevel<=90 ){
100       return('Oxygen level is too low');
101     }
102   }
103
104
105   async queryPatientsByName(ctx, name){
106     let queryString={}
107     queryString.selector={"name":name}
108     let iterator=await ctx.stub.getQueryResult(JSON.stringify(
          queryString))
109     let result= await this.getIteratorData(iterator)
110     return JSON.stringify(result)
111   }
112
113   async getIteratorData(iterator){
114     let resultArray=[]
115
116     while (true){
117       let res = await iterator.next()
118       let resJson= {}
119       if(res.value && res.value.value.toString() ){
120         resJson.key=res.value.key;
121         resJson.value=JSON.parse(res.value.value.toString('utf-8'))
122         resultArray.push(resJson)
123       }
```

```
124
125       if(res.done){
126         await iterator.close();
127         return resultArray
128       }
129     }
130   }
131 }
132
133 module.exports = FabCar;
```

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This thesis was generated on Tuesday 11th January, 2022 at 4:22pm.