A Peer-to-Peer Blockchain-based Approach for Blood Donation Community

Authors

Minhaz Kamal (180041231)

Chowdhury Mohammad Abdullah (180041239)

Fairuz Shaiara (180041240)

Supervisor

Dr. Md Azam Hossain

Assistant Professor

Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT)

A thesis submitted to the Department of CSE

in partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT)

Organization of the Islamic Cooperation (OIC)

Gazipur, Bangladesh

May 2023

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Minhaz Kamal, Chowdhury Mohammad Abdullah and Fairuz Shaiara under the supervision of Dr. Md. Azam Hossain, Assistant Professor, Department of Computer Science and Engineering, Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Minhaz Kamal	Chowdhury Mohammad Abdullah	Fairuz Shaiara
Student ID: 180041231	Student ID: 180041239	Student ID: 180041240

Supervisor:

Dr. Md Azam Hossain
Assistant Professor,
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Abstract

Patients undergoing a wide variety of medical operations and treatments benefit greatly from the availability of blood transfusion services, which are an essential component of the healthcare system. These services play a vital part in guaranteeing the high standard of care that these patients get. The majority of this blood comes from people who have voluntarily given it. There is currently no way for blood donor management systems to provide a credible audit trail or traceability. As a consequence of this, there is a substantial possibility that patients will receive a transfusion of blood from untrustworthy sources. In this article, we present a proposal for a system that would be built on Ethereum with the intention of establishing a network of blood donors that is decentralized, transparent, traceable, and safe. The platform makes use of smart contracts to make peer-to-peer interactions easier to coordinate. To further motivate donors to donate blood on a more consistent basis, the system also provides rewards in the form of tokens that can be redeemed for prizes.

Acknowledgement

It is a blessed occasion for us to submit our thesis work because it will mark the completion of our Bachelor of Science program. We would like to begin by expressing our sincere appreciation to Almighty Allah for His bounties showered upon us, which made it possible for us to successfully finish our thesis research project. We wouldn't be in our current situation if it weren't for Allah's benevolence.

We want to thank Dr. Md. Azam Hossain, Assistant Professor, Department of Computer Science and Engineering, Islamic University of Technology, for serving as our mentor and advisor. His inspiration, advice, and ideas for our thesis were priceless. Without his assistance and appropriate direction, our thesis would not follow the right course in the field of study. We were able to complete our thesis work properly because of the important advice, time, and input that he offered throughout the process, starting with the introduction of the thesis subject and moving on to the selection of the study field and the proposal, revision, and implementation. We appreciate his consistent, enthusiastic leadership and insightful counsel.

For his wise counsel and contributions, we also acknowledge Dr. Abu Raihan Mostofa Kamal, Professor, Department of Computer Science and Engineering, Islamic University of Technology.

We would like to express our gratitude to each and every one of the esteemed jury members on our thesis committee for their insightful remarks and helpful critique of our study. They undoubtedly assisted us in improving the research study.

Last but not least, we would like to extend our profound appreciation to everyone of the faculty member at Islamic University of Technology's Department of Computer Science and Engineering. They offered a helping pair of eyes and ears when issues arose, which contributed to the pleasantness of our working environment.

Contents

Abstract i

Acknowledgement ii

1 Introduction **1**

1.1 Background Study 1

1.2 Problem Statement 3

2 Basics of Blockchain **5**

2.1 What is Blockchain? 5

2.2 Why the name blockchain? 6

2.3 Properties of blockchain 7

2.3.1 Immutability 7

2.3.2 Traceability and data provenance 9

2.3.3 Data security and privacy 10

2.4 The Consensus Protocols 10

2.4.1 Proof-of-work (PoW) 11

2.4.2 Proof-of-Stake (PoS) 12

2.4.3 Delegated proof-of-stake (DPoS) 13

2.4.4 Practical Byzantine Fault Tolerance (PBFT) 14

2.4.5 Ripple 14

2.5 Types of Blockchain 15

2.5.1 Public Blockchain 17

2.5.2 Private Blockchain 17

2.5.3 Consortium Blockchain 17
2.5.4 Hybrid Blockchain 17

3 The Merge: Ethereum 2.0 **19**
3.1 PoW vs PoS 19
3.2 Ethereum vs Ethereum 2.0 20
3.2.1 Consensus Mechanism 20
3.2.2 Sharding 21
3.2.3 Beacon Chain 21
3.3 Objectives of the Merge 21
3.3.1 Security 22
3.3.2 Scalability 22
3.3.3 Sustainability 22

4 Related Works **23**
4.1 Paper Title:Blockchain traceability in healthcare:Blood donation supply chain 23
4.1.1 Introduction 24
4.1.2 Proposed System 24
4.1.3 Result Analysis and Drawbacks 25
4.2 Paper Title: Blockchain-based management of blood donation 26
4.2.1 Introduction 26
4.2.2 Proposed System 27
4.2.3 Result Analysis and Drawbacks 27
4.3 Paper Title: Bloodchain: A blood donation network managed by blockchain technologies 29
4.3.1 Introduction 29
4.3.2 Proposed System 29
4.3.3 Result Analysis and Drawbacks 30

5 Our Proposition **32**
5.1 Proposed System 32

5.2 Interaction of Entities Through the System 35

6 Implementation and Evaluation **36**

6.1 Smart Contract Implementation 36

6.1.1 Simulating the Private Network 41

6.2 Smart Contracts Analysis with Securify2 43

6.3 Using Ethereum 2.0 45

6.4 System Evaluation 46

6.4.1 Gas Consumption Analysis 46

6.4.2 Throughput Analysis 48

6.5 Challenges 49

7 Conclusion **51**

7.1 Summary 51

7.2 Future Work 51

Bibliography 52

List of Figures

2.1 Cryptograhic Links between blocks 6
2.2 Actual Structure of a Block 8
2.3 Types of Blockchain with Ven Diagram 16

3.1 Power Consumption by Ethereum over the years. [1] 20

5.1 System architecture. 33
5.2 Sequence diagram of the system. 35

6.1 Custom definition of genesis block. 42
6.2 Docker file for building images. 42
6.3 Number of issues detected per smart contracts. 45
6.4 Comparative Analysis of Gas Consumption 47
6.5 Comparison of Throughput Analysis 48

List of Tables

2.1 Comparison Between Different Types of Consensus Protocols [2] 10
2.2 Comparison Between Different Types of Blockchain 16

6.1 Basic information about User 37
6.2 Data Structure of User 37
6.3 Basic information about Patient 38
6.4 Basic information about Blood Donation Center 38
6.5 Data Structure of Request Format 38

List of Algorithms

1 Algorithm for User Registration 39
2 Algorithm for Posting Request 39
3 Algorithm for Responding to Requests 40
4 Algorithm for Information Sharing and Tokenization 41

Chapter 1

Introduction

1.1 Background Study

The topic of blood transfusion is one of the most delicate as well as important concerns in the area of medicine. It is required in a range of settings, including the treatment of soft tissue injuries (for example, significant burn damage, tissue puncture, etc.), as well as medical procedures and operations (for example, C-section, organ transplant, etc.) that offer a danger of excessive blood loss. In addition, it is essential in the treatment of injuries to the gastrointestinal tract (for example, severe burn damage, tissue puncture, etc.). The treatment of numerous medical problems such as blood cancer, leukemia, and other conditions should also be taken into mind, since this is an additional significant aspect to take into account. In addition, there are others who suffer from conditions such as anemia or thalassemia [3] and have an ongoing requirement for blood transfusions [4].

The number of people who need blood transfusions is increasing at a rate that is much higher than the number of people who are willing to donate blood. Only three percent of those who are eligible to donate blood actually do so, which is not enough to meet the demand when it suddenly spikes, as reported by the "American Red Cross" [5]. It is quite unlikely that the position will undergo significant shifts in the relatively near future, with the exception of becoming more challenging. This is mostly due to the fact that the population of the globe as a whole is getting older, and research has shown that an aging population

is more likely to be a consumer than a provider [6]. In addition, the "American Cancer Society" projects that there will be 1.9 million newly diagnosed cases of cancer in 2022, with around 609,360 people succumbing to the disease in the United States [7]. Every newly diagnosed patient is going to require a significant quantity of blood transfusion and component transfusion at various points throughout the course of their treatment. It has been hypothesized that leukemia and thalassemia will turn out to have comparable outcomes [8, 3].The simultaneous rise in the need for blood component-specific transfusions (such as red blood cells, plasma, and platelets), and the concurrent fall in the number of people willing to donate blood, are both expected to result from the combination of an aging population and chronic diseases. According to [6], this suggests that there will be a shortage of blood donations in the future. As a result, it is absolutely necessary to put forward innovative strategies in order to counteract this deteriorating trend. The process of making contributions ought to be made less complicated, participative, and in some way beneficial for contributors.

Finding a donor whose blood group is compatible is the first step in the blood donation process. After that, it is necessary to conduct additional research into the donors' past and the current state of their health. It is because there are issues over the patient's safety as well as the donor's risk. For instance, a patient will not want to receive blood from a donor who is dependent on potentially dangerous intoxicants or who is contaminated with STDs [9]. In order to speed up the process overall, this needs to go through a variety of blood searching, matching, and screening procedures. In addition, there are no reliable digital solutions for managing blood donation management that are currently available. Various organizations make an effort to recruit blood donors and give blood bags to patients based on the criteria of each individual organization. It is feasible to create a community of peers who are helpful to one another and interact with one another while providing assistance to one another.

A private blood donation system that is based on blockchain technology is what we propose as a solution to the problems described above. A database and a distributed ledger that is powered by the blockchain network will be used in conjunction with one another to organize the essential data. It will assure the traceability of blood donors across the entire community and will lessen the likelihood of obtaining blood that has been polluted. In the

event that there was an anomaly, the system is able to conduct an investigation into the matter within a few seconds and collect evidence. In addition, following every successful contribution, the system will store tokens as an incentive for donors to make additional contributions. Users will be able to receive benefits from well-known health organizations if they have these tokens in their possession. In the end, the system will use smart contracts to fully automate the process of collecting blood donations.

1.2 Problem Statement

A blockchain is a decentralized, public ledger that is incorruptible and employs encryption to secure record keeping. The data that is held in a blockchain may be conceptualized as transactions that have taken place between various entities. These transactions are recorded and kept track of within a block, which can be conceptualized as a collection of ledgers. A string of these blocks can be connected to one another through the utilization of cryptographic hashlinks. If the information contained within a block is altered in any way, the links between blocks and the rest of the chain are essentially severed, and the break will extend all the way to the conclusion of the chain. In addition to this, each node in the network that makes up the blockchain has a copy of the chain instance that is being used. This indicates that a suitable adjustment needs to be made in each and every copy of the document. Immutability, provenance, and traceability are all provided by the blockchain thanks to the combination of cryptographic hash links and the consensus protocol. Because of this, the implementation of blockchain technology into our system might make it possible to track down potentially contaminated blood transfusions in a matter of seconds. When compared to a blockchain network, the speed at which such in-depth investigations may be completed in a centralized or cloud-based system is significantly slower, and there is no guarantee that they will produce the desired results.

There are three distinct categories that can be applied to blockchain: public blockchain, private blockchain, and consortium blockchain. In a public blockchain, any new user who demonstrates that they have the necessary identification can join the network. On the

other hand, in a private blockchain, only a specific group of entities can join. A consortium blockchain, on the other hand, is focused with multiparty cooperation within a group of corporations or organizations. The proposed infrastructure makes use of a private blockchain and is built on top of Ethereum's distributed ledger system.

Smart contacts are an additional component of the blockchain network that have the potential to support automated transactions in a way that is secure, irreversible, and traceable. It is nothing more than a fragment of computer code that, when run, carries out a predetermined set of logical operations written in a predetermined programming language. In order to carry out mathematical operations while inside of a smart contract, an appropriate quantity of gas must first be spent. As a consequence of this, smart contracts have to be as easy to understand as is practically possible in order to bring down the price of gas use.

Chapter 2

Basics of Blockchain

According to [10], blockchain is a technology that is still in its infancy but is already being heralded as a game-changer for our increasingly digital world. Blockchain, in contrast to other prominent fields such as machine learning or deep learning, is the product of an elaborate collection of numerous definitions, laws, and components. Therefore, it is required to have at least some basic background knowledge about the technology in order to appreciate its applications in the field of DF.

We will examine the fundamentals of blockchain technology as well as its characteristics in the following subsections. In addition to this, we explain various types of blockchains, common consensus protocols, and the applications of each.

2.1 What is Blockchain?

According to [2], blockchain can be described as a decentralized, append-only, and immutable log that stores records by utilizing encryption. The cryptographic method contributes to the record's safety and traceability. The distributed component contributes to the verification of transactions, which ensures that no malicious or inaccurate data may be added to the network. Immutability is a property that can be derived from the dual combination of these two elements [11].

When a suggestion was made for the time-stamping of digital documents, this is considered to be the beginning of blockchain technology [12]. To summarize, the author used the

avalanche effect of hash functions (i.e., a tiny variation in input manifests a massive change in the output of hash function) [13] to detect any slight modifications that were done to the document. Specifically, the author wanted to know if the material had been altered in any way. The hash value is saved after each time a modification is made, and as a result, a series of hash values is generated that represents the many versions of the document at various times in the past. However, the actual implementation of the blockchain technology was originally shown in the now-famous white paper that was written by the creator of Bitcoin [14]. In later years, this network served as a reference point during the process of further developing and improving the technology.

2.2 Why the name blockchain?

The fact that the data that is intended to be stored is seen as transactions between entities is one of the key properties of a blockchain network [11]. A specific number of these transactions will be aggregated together in a block, and after being validated by a consensus protocol (which will be detailed in section 2.4), they will be attached to a structure that resembles a chain. A particular block will be consolidated during this verification process by looking at the hash values, and a chain-like structure will be created as a result of a gradual hash link between the blocks. Because of this, the network of blocks has been given the name "Blockchain." A part of such a chain can be seen in Fig. 2.1, which can be found here.

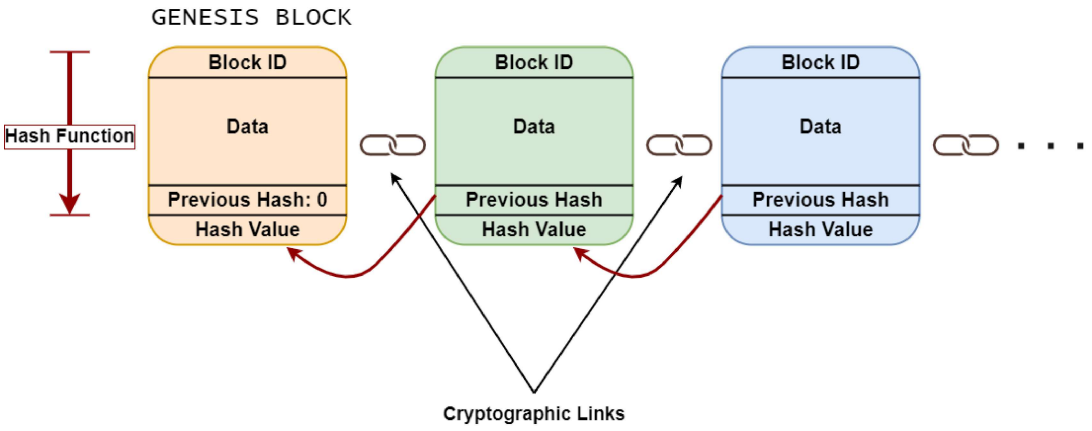

Figure 2.1: Cryptograhic Links between blocks

The essence of blockchain is described as being dispersed, as was just said. To act in a dispersed manner implies to carry out a single mission using multiple people at the same time. On the other hand, decentralization in a network refers to the presence of various points of coordination within the system. According to [15], a blockchain is a distributed network in which every node will keep a copy of the chain. Using a specific "consensus protocol" for verification of new blocks, all the participants in the network will agree upon one single chain and resolve any conflicts before a new block is added to the chain.

In the next sub-sections, the primary attributes of Blockchain will be described. These attributes are what are driving an increase in the usage of this technology and what make it advantageous in a variety of fields, including the scientific community, the financial sector, supply chains, and so on.

2.3 Properties of blockchain

It is possible for Blockchain to provide a variety of distinct security qualities even at its most fundamental level, which differentiates it from other technologies that are currently on the market [11]. This subsection discusses the characteristics of blockchain and explains how those characteristics can be utilized in DF.

2.3.1 Immutability

What this means is that the blockchain is a tamper obvious and tamper proof ledger. To get a better grasp on it, you need to first get the idea of a block straight in your head. Referring to Fig. 2.2, In general, a block can be broken down into the following fields: a block ID, which is a number that is used only once to identify a block; the data, which can be a list of transactions or any smart contracts (i.e. a piece of code designed for automating processes); the hash value from the block that came before it; the hash value of the block that came after it; and the Nonce, which is a number that is only used once. There may be additional fields added in the future based on need, but these are the crucial ones for comprehending the blockchain technology. There may be additional fields added in the future based on need, but these are

the crucial ones for comprehending the blockchain technology [11]. All of these fields are then input into the hash function, which results in the block's hash value being generated.

The very first building brick that makes up a chain is referred to as the Genesis Block [16], always having the previous hash value of zero. It stores all of the required information that is associated with the configuration of the network. After that point, the hash value of the preceding block will be stored in any new block that is anchored to the network. As a result, the hash link that was shown previously in Fig. 2.1 was created. The hash value of one block will be placed in the subsequent block in a sequential fashion, gradually building the chain-like structure that was previously described. The immutability of the chain is due to the sequential nature of the references made to the preceding hash value.

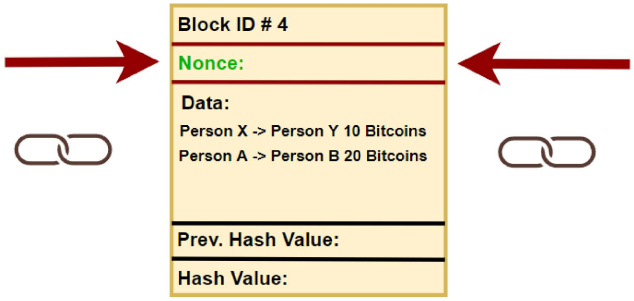

Figure 2.2: Actual Structure of a Block

This is primarily due to the fact that the avalanche effect, as was discussed earlier, will lead any effort to modify any transaction within a block to result in a significant change to the block's hash value. Now, the block that comes immediately after this one will discover that the old hash value it stored does not match the current hash value of the block that came before it. The hash link that connects two blocks is essentially broken as a result of this mismatch. The remainder of the chain is thrown away as a result of this fracture. Therefore, in order for an adversary to successfully change any transaction, the remainder of the chain will first need to be altered one block at a time [17].

Because of the decentralized nature of blockchain technology, each participant in the network will keep a copy of the chain that is identical to the original at all times. Any "consensus protocol" can be used to verify a change before it can be implemented in a block, and this

validation is required before the change can be accepted. Attacking only one chain is not enough to compromise the network's safety because the exact same modification needs to be mirrored on the entirety of or at least the vast majority of the participating chains in the network. Because of this, it is now nearly impossible for a threat actor to amass such a large amount of computational resource in order to launch an assault on a blockchain network. Even if it were conceivable, the cost would be so great that it would eventually outweigh the potential utility of the hack. Because there are cryptographic hash links between two blocks in the network and a consensus procedure to agree on a particular, this challenge has been established for the attacker to contend with. The immutability of the blockchain is a direct consequence of the confluence of these two factors.

In continuation, one more nugget of wisdom that can be added is the observation that, as more blocks are added to the network, it becomes more difficult to alter the chain's previous links. Because more time has passed, the data will become more imprinted in the network, making it more difficult for an adversary to modify that block. This points to the use of a consensus mechanism in conjunction with a hash function as the primary form of protection against fraudulent alterations.

2.3.2 Traceability and data provenance

In the course of the discussion that was held on 2.3.1, a question that may have occurred to you was: if the chain is immutable, then how can it facilitate update or delete operation? If these operations are not performed, the technology will never be suitable for the purpose of providing information that is current and accurate. This is where the audit-trail capability of blockchain networks and their inherent traceability become relevant to the discussion. When an existing record is deleted or updated, a new block containing the updated information will be anchored to the chain without the original record being overwritten or deleted. This new record will then be considered the current or updated information after it has been anchored to the chain. This approach guarantees fundamentally both traceability and audit trails, as well as the provenance of the data. This property of blockchain is really what makes it revolutionary and useful in DF and many other domains as well as other areas of study and

application.

2.3.3 Data security and privacy

The protection of personal information and privacy has become an increasingly pressing issue in light of the proliferation of online criminal activity. Encryption and decryption mechanisms are utilized by blockchain networks through the utilization of public key and private key pairs. It ensures that only the correct receiver will be able to access a certain type of information that is solely intended for that receiver. In other words, it restricts access to the information. The importance of multiparty collaboration across different jurisdictional borders was highlighted earlier in the **??** section. Because of the administrative hierarchy and the importance of the system, a role-based approach is required in order to guarantee access control of the data and a privilege control mechanism within the system. In this regard, permissioned blockchains are available in the form of ready-made frameworks (this topic will be further explored in 2.3). Therefore, by employing encryption and a role-based access control system, blockchain technology has the potential to solve the problems of data privacy and security [18].

2.4 The Consensus Protocols

Table 2.1: Comparison Between Different Types of Consensus Protocols [2]

Attribute	PoW	PoS	DPoS	PBFT	Ripple
Finality	Probabilistic	Probabilistic	Probabilistic	Absolute	Absolute
Faulty Nodes : Good Nodes	1 : 2	1 : 2	1 : 2	1 : 3	1 : 5
Power Requirement	High	Low	Low	Negligible	Negligible
Scalability	High	High	High	Low	High
Genre of Blockchain	Permissionless	Permissionless	Permissionless	Permissioned	Permissioned
Example	Bitcoin	Peercoin	Bitshares	Hyperledger Fabric	Ripple

It is necessary for all participants in a distributed system to arrive at a decision that is acceptable to all of them in order for the system to function properly [19]. The same can

be said for blockchain technology. As was mentioned earlier, each node (participant) in the blockchain network keeps a copy of the chain on their local machine. It is necessary for the remaining nodes in the chain to reach a consensus on any new block that is to be anchored to the chain. To put it another way, they need to ensure that the newly added block does not contain any malicious code. There are many different kinds of protocols being used by the various blockchain networks. A few of them will be covered in greater detail in the subsections that will follow.

2.4.1 Proof-of-work (PoW)

This protocol is implemented in both Bitcoin and Ethereum, two of the most well-known blockchain networks [14] [20]. For the block's hashlinks, the SHA256 cryptographic hashing algorithm is utilized. Within these Proof-of-Work-based networks, newly made transactions are initially placed in a "mempool" queue for processing [21]. A predetermined number of transactions are drawn from this pool, accumulated in a block, and then verified by a "miner" before being added to the chain. The miners are awarded mining fees as a form of compensation for the resources that they put into completing this computationally difficult task [22].

Two guiding principles are used in PoW. The first one states that a block will only be considered valid if the generated hash value of the block is lower than a specific threshold that is referred to as the "target." If this threshold is not met, the block will be thrown away. The miners will select a group of transactions from the mempool to include in a block, and then they will begin computing the hash values. The objective of the miner is to produce a hash value that is lower than the target value. This can be accomplished by modifying the value of the nonce, which was discussed earlier in the subsection 2.3.1. The nonce value is essentially the only variable field of a block, as can be seen in Fig. 2.2. The miners will keep adjusting the value until the hash value falls below the target; once it does, the block will be considered valid and will be added to the chain.This block is also known as the "candidate block" because it contains a hash that is considered to be valid [23] [24].

As was previously mentioned, every node in the network stores the same copy of the

chain, which is also referred to as the ledger in some contexts. Additionally, it is possible that two or more miners mined the same block at the same time. This indicates that there is the potential for multiple candidate blocks to emerge at the same time. The second stipulation of Proof-of-Work is that the winning candidate block must be the one that contains the chain that is the longest, and that block must be accepted by all of the nodes that are part of the network. The remaining candidate blocks will be eliminated from consideration. The network does this to ensure that all of the nodes that are being added are legitimate cites for [24].

In this case, the miners will need to spend a significant amount of computational resources in order to generate a valid hash. Therefore, in order to make a modified block valid, an attacker will need to go through the entire Proof-of-Work process again. The "amount of work" that a miner was required to complete is the "same amount of work" that an attacker needs to "re-do" in order for a change to be considered valid.

The primary problem with PoW is that it requires a lot of energy. In addition to this, there is a possibility of centralization due to the fact that a node with the highest hashing power will always have a better chance of winning more. As a consequence of this, a node that possesses 51% of the hashing power will almost certainly succeed each time it attempts to create a new chain. This completely contradicts the fundamental ideas behind blockchain technology itself. As a result of these factors, proof-of-stake (PoS) is currently under consideration by the community as a potential replacement for proof-of-work [25].

2.4.2 Proof-of-Stake (PoS)

The blockchain network known as Ppcoin was the first to suggest using this protocol [26]. There are nodes Forgers or validators who authenticate a block rather than Miners [27]. Miners are no longer necessary. To begin, in order to become eligible for being a forger, a node is required to make a security deposit in the form of coin-age; this deposit is referred to as the stake. To be more specific, a unique transaction with the name "textit-coinstake" takes place here. During this transaction, the owner of the cryptocurrency pays himself in order to earn the privilege of generating blocks and validating transactions by adding them to the

chain [26]. The likelihood of a node becoming a validator is increased when the stake at that node is increased. After being selected as a validator, it is only able to validate transactions whose total is less than or equal to the amount of the safety deposit that the validator initially provided. Now, if the validator somehow verifies fraudulent transactions, then as a penalty, the total amount of transaction that the validator has validated will be deducted from the deposit. This will happen only if the validator somehow verifies fraudulent transactions.

Because of this, the cost of computing for PoS is significantly lower when compared to the cost of computing for PoW; as a result, there is less waste of energy and cost overhead [28]. Because of this, Ethereum is already working toward the adoption of this new protocol because of the benefits it offers [25].

It's possible to make the case that PoS is affected by the same issue that PoW was, and that argument has some merit. If one node in the network has deposited a certain percentage of the total coins in the ecosystem, say 51% of them, then that node will have an advantage over all of the other nodes in the network. In contrast to PoW, however, it is not nearly as useful in everyday life. This is due to the fact that the current market capitalization of cryptocurrencies is so high that it is practically impossible to imagine anyone owning 51% of the market. When compared to PoW, this indicates that PoS has a more decentralized structure than PoW does [29].

2.4.3 Delegated proof-of-stake (DPoS)

This protocol is almost the same as PoS, but the difference is that it follows a form of digital democracy to elect a validator. Nodes will deposit certain amounts against which they will get some voting power. A node can transfer it's voting power to another node which can vote on behalf of it. After voting, a delegated validator is elected to verify the transactions [30].

A scenario may arise where a delegated node can not perform the task of validation. For this case, a set of backup delegates may also be needed. This totally relies on the use case and implementation.

2.4.4 Practical Byzantine Fault Tolerance (PBFT)

PBFT is a highly practical and low complexity consensus protocol for distributed systems [31]. The algorithm has five phases. Their chronological ordering: *request, pre-prepare, prepare, commit and reply.* The network consists of a single primary node and all other nodes are called secondary nodes. The main condition of PBFT is that the total number of faulty nodes must be less than or equal to one-third of the total number of nodes in the system.

It starts through the client by submitting a request message to the primary node. This message is then broadcasts it to all other secondary nodes as a pre-prepare message. Every node except the faulty ones accept this pre-prepare message only if it is valid. The validity of any message is judged by its sequence number, signatures and other useful metadata associated with it. Upon receiving the pre-prepare message, all the nodes follow up through sending a prepare message to all other nodes. Now this prepare message will be received the Receiving nodes only if it is valid. A node is prepared based on some conditions after which it sends out a commit message. Now a node will carry out the client requst only if it receives $f+1$ commit messages, where f is the number of faulty nodes. Finally, sending a reply message from it towards the client will end the process.

Now to ensure the validity of the reply message, the client waits for $f+1$ reply messages. This is because at most f number of faulty nodes are accepted in the network which means at least $f+1$ nodes need to be non-faulty. The client finally receives a valid response from the network. PBFT is generally used in specific types of blockchain mainly permissioned blockchain that is discussed in 2.5.

2.4.5 Ripple

Ripple Protocol consensus algorithm (RPCA) was primarily introduced to alleviate the problem of low-latency in creating a valid transaction that can be found in other consensus algorithms dealing with the Byzantine General's Problem [32]. The first thing to clarify is that Ripple is merely an algorithm that runs in all the servers in a Ripple network to arrive at consensus. Every server from the same network will maintain the same copy of the ledger

containing all the valid transactions. The most recent copy of the ledger created at any time is called the Last Closed Ledger (LCL). The ultimate goal of the consensus is to be able to create a ledger that will be accepted by all the servers in the network.

The working mechanism of Ripple can be broken down in this manner where the servers continuously receive transactions from other servers on the network. These transactions are collected in a pool of transactions called the Candidate Set. Simultaneously, the servers also receive proposals from other servers which are essentially the transactions that are considered suitable to be added in the ledger. Now these proposals are matched with the ones contained in the Unique Node List (UNL) of each server. If any of the incoming proposal is not in the UNL then it is simple discarded. But if it is in the UNL, then those proposals are matched with the ones in the candidate set. If there is a match then that transaction receives a vote. The process of matching and giving a vote to the transactions is continued until the timer expires and the first round of consensus is finished.

The transactions that received more than or equal to 50% votes are elected to the next round and the whole process is repeated. In each successive round the percentage threshold increases until it reaches 80% or above. At that point the consensus have been reached and all the transactions having 80% or above votes are posted into the ledger to form the LCL. With that, the consensus algorithm finishes its job and the LCL is then adopted by all the servers in the network.

The RPCA does not take any probabilistic approach unlike PoW or PoS and hence it is classified as an absolute-finality protocol.

A comparison is given for the above mentioned protocols in Table 2.1. The types of blockchain will be further discussed in Section 2.5

2.5 Types of Blockchain

So far the general idea and the working mechanism of blockchain has been presented briefly in this paper. This section will contain a brief description about two major types of blockchain: Permissioned and Permissionless. The Permissionless blockchain is decentralized and dis-

Table 2.2: Comparison Between Different Types of Blockchain

Property	Public Blockchain	Consortium Blockchain	Private Blockchain
Consensus Determination	All miners	Selected set of nodes	One organization
Read Permission	Public	Could be public or restricted	Could be public or restricted
Immutability	Nearly impossible to tamper	Could be tampered	Could be tampered
Efficiency	Low	High	High
Centralized	No	Partial	Yes
Consensus Process	Permissionless	Permissioned	Permissioned

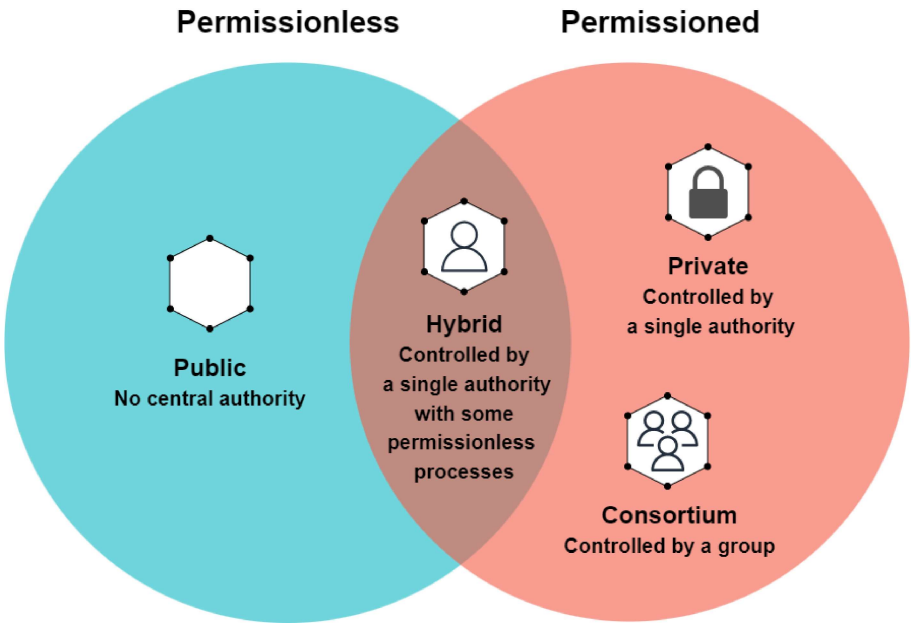

Figure 2.3: Types of Blockchain with Ven Diagram

tributed in nature where any node can join anytime pseudo-anonymously. However, Persmissioned blockchain has central authority and adding any new nodes is policy dependent. Permissionless blockchains are more secured, robust but demand resources for validating a transaction. Permissioned blockchains on the other hand has a central point of control due and more efficient due to the restricted nature.

Based on the working principles, four different varieties of blockchain networks can be found. They are discussed in the following subsection.

2.5.1 Public Blockchain

In simple words, networks like bitcoin, etherium are public blockchain where anyone can take part anytime. These can be characterized as Permissionless blockchain. Few common consensus protocols used in these networks are PoW (2.4.1), PoS (2.4.2) and DPoS (2.4.3).

2.5.2 Private Blockchain

Permissioned blockchains, also known as private blockchains, are characterized by the presence of a central authority that is responsible for selecting particular nodes. A well-known example of a private blockchain network is Hyperledger, which is an umbrella project that makes use of the PBFT consensus protocol (2.4.4). Ripple (2.4.5), which is also an example of a private blockchain, is the cryptocurrency that is utilized for business-to-business currency exchanges.

2.5.3 Consortium Blockchain

It is exactly the same as a private blockchain, except that rather of having a single ruling node, there is a collection of such nodes (i.e. several organizations) that adds greater decentralization to the network. Due to the necessity of collaboration between various organizations, the policies, planning, designing, and implementation of such networks are exceedingly complex. One such network that exists in the real world is called the Global Shipping Business Network Consortium. Through the use of blockchain technology, this network intends to make the process of collaboration in the maritime and shipping industries more open and safe.

2.5.4 Hybrid Blockchain

These kinds of networks make use of aspects that are present in both permissioned and permissionless blockchains. Even if confirmation of transactions is still necessary, the network is controlled by a single node or entity. A hybrid blockchain like IBM Food Trust is shown here as an example. The Venn diagram seen in fig. 2.3 illustrates how the features of hybrid blockchains intersect with those of other types of blockchains. The following

table 2.2, provides a summary chart that compares and contrasts the various blockchain implementations.

In the next chapter (chapter 3), the document explains the transition from Proof-of-Work to Proof-of-Stake. A comparative analysis of the added benefits and different research challenges are given in details by Ethereum 2.0. Then in Chapter 4, we analyse the related works in details to gain insight how others have solved the research challenges that we identified. Then in Chapter 5, we proposed our system in details followed by the implementation and performance evaluation in Chapter 6. Finally Chapter 7 draws conclusion by summarizing the whole work and identifying the future scopes to enhance the proposed work.

Chapter 3

The Merge: Ethereum 2.0

Proof-of-stake (PoS) is an alternative validation method that was proposed and supported by Vitalik Buterin, the founder of Ethereum. This transition would result in a reduction in the amount of energy required for validation. Stakeholders in PoS are required to verify new transactions by placing the Ethereum they currently possess into a smart contract as collateral. This is in contrast to PoW, which requires users to search randomly for the nonce. Stakers are selected using an algorithm called RANDAO, and they risk being punished if they are unable to fulfill their duties. In the past, the Ethereum network relied on a process known as proof-of-work, which required users to solve difficult problems that required a significant amount of energy. As a result, the Ethereum network used a significant amount of electricity. However, the network has shifted to using a different method known as proof-of-stake, which does not require this kind of problem-solving and is much more energy-efficient. This method was developed to improve the network's performance. As a direct consequence of this, there has been a 99.98% reduction in the amount of electricity that the Ethereum network consumes (Fig. 3.1). This major improvement is nothing short of a revolution. This is why the community addresses it as the advent of Ethereum 2.0.

3.1 PoW vs PoS

In Proof of Work, miners compete against one another to solve a difficult algorithm in order to add new data blocks to the chain. This process uses a significant amount of electricity

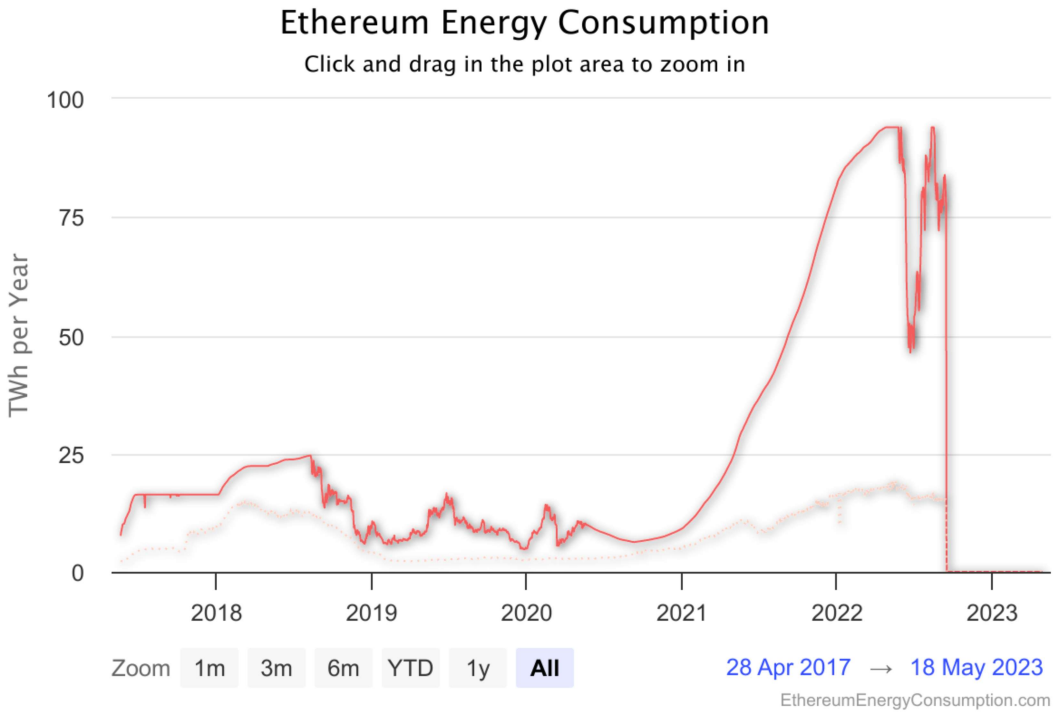

Figure 3.1: Power Consumption by Ethereum over the years. [1]

because miners employ powerful computing devices in their work. On the other hand, Proof-of-Stake (PoS) does not require mining; instead, users lock their funds on the blockchain as stakers and participate in adding new blocks without relying on computing power. Adding new blocks is an integral part of PoS. When a block is successfully added, successful stakers are rewarded with additional tokens when the block is added.

3.2 Ethereum vs Ethereum 2.0

3.2.1 Consensus Mechanism

In Ethereum 2.0, the Proof of Stake (PoS) mechanism is utilized. With this mechanism, validators are chosen at random to hash new blocks, and in order to participate, one must stake at least 32 ETH. It is known as "slashing," and it occurs when a validator's staked funds are reduced because they were caught authenticating transactions that were not legitimate.

Because users no longer need expensive rigs to mine new tokens, this PoS mechanism is more sustainable and allows for greater decentralization. This results in increased user participation and decentralization.

3.2.2 Sharding

Sharding is a method for increasing the processing speed of Ethereum 2.0. This is accomplished by distributing data across a total of 64 distinct chains, which are referred to as "shards." Each shard functions in the same way as the previous chain, but the workload is reduced because the information is distributed across multiple databases. This makes it more efficient, and it can process up to 100,000 transactions per second, whereas the original Ethereum could only process 15 transactions per second. It's the equivalent of adding a number of new lanes to an already congested highway, which makes the flow of traffic more orderly and expedient.

3.2.3 Beacon Chain

Beacon Chain is the name of the blockchain that Ethereum 2.0 uses to connect its 64 individual chains into a single, interconnected network. This central component is in charge of managing all of the sharded chains and enables transactions across the entire network. The Beacon Chain is in charge of randomly selecting validators to add new data blocks, monitoring their activity, and punishing them if they try to exploit or cheat the system. In addition, it is responsible for selecting validators to add new data blocks. It is essential to use random selection to ensure that no one participant has an advantage over the others in the process of being chosen as a validator.

3.3 Objectives of the Merge

The great merge is nothing sudden rather it was part of the great roadmap of Ethereum toward becoming the first truly global computer.

3.3.1 Security

The primary focus of Ethereum is on enhancing the level of protection afforded to developers and investors. The transition from a Proof of Work (PoW) consensus model to a Proof of Stake (PoS) model has two primary goals: the first is to reduce the likelihood of being targeted by cybercriminals, and the second is to stop any one entity from gaining the majority of control in a decentralized network. In addition, it can mitigate the dangers posed by the 51% attack threats that have plagued other blockchains in recent years.

3.3.2 Scalability

On Ethereum's blockchain, there is a significant number of decentralized finance applications and non-fungible tokens (NFTs), which results in an ongoing high volume of transactional activity. In spite of this, the Ethereum blockchain was only capable of processing 15 transactions per second because of the limitations of the older Proof-of-Work consensus model that it used. Users were forced to pay high transaction fees as a result, and their funds transfers were also significantly slowed down. These problems are solved by the more recent Proof-of-Stake consensus model, which also allows for faster transaction processing with lower fees. Additionally, it eliminates the need for mining, which makes Ethereum more sustainable.

3.3.3 Sustainability

Mining in Ethereum's older consensus model, known as Proof-of-Work (PoW), required a significant amount of electricity and could not be maintained indefinitely. The more recent PoS model does away with the requirement for mining, making Ethereum more sustainable and friendly to the environment. In order to make a name for themselves in the blockchain industry, new blockchain projects should make becoming environmentally friendly a top priority as soon as possible.

Chapter 4

Related Works

In recent years, there has been a major uptick in interest about the application of blockchain technology to blood donation processes; scholars and organizations are investigating the possibilities that such an integration may present. Traceability, transparency, and safety are just a few of the issues that might arise throughout the process of giving blood. Blockchain, which is a distributed and immutable ledger, offers a number of one-of-a-kind characteristics that can help address these problems. This section gives an overview of the previous research and related efforts that have studied the potential use of blockchain technology in blood donation systems. It attempts to assess the current state of research, highlight major findings, and explain the gaps and opportunities for further inquiry in this potentially fruitful topic. The current corpus of knowledge can provide us with insights into the potential of blockchain technology to alter the way blood donations are managed, recorded, and distributed, ultimately leading to improvements in the efficacy of blood transfusion services.

4.1 Paper Title:Blockchain traceability in healthcare:Blood donation supply chain

Published in:

- 2021 23rd International Conference on Advanced Communication Technology (ICACT)

- Conference held in PyeongChang, Korea (South)

Authors:

- Sadri, Samin and
- Shahzad, Aamir
- Zhang, Kaiwen

4.1.1 Introduction

The significance of blood donations to the provision of medical care is examined in this section, along with the dangers associated with an inefficient supply chain. It demonstrates how the use of Blockchain technology can help address these challenges by allowing for tracing and verifying information at each stage of the supply chain. The paper discusses the findings of a study that makes use of smart contract solutions to confirm research findings and investigates issues of privacy, safety, and provenance within the BDSC ecosystem. In Section II, an overview of previous research on this subject is presented. Subsequent sections, Sections III-V, go into greater depth about technological aspects, system modeling, programming codes utilized for validation purposes, and measurable results, which are further examined in Section V. In the final part of the chapter, Section VI draws to a close by describing the future contributions that will be made to improve upon current practices within this field's landscape more generally speaking. These improvements can be made from the perspective of regulatory agencies or healthcare providers themselves, both of which stand to gain a great deal from the insights that are gained through research such as the one that is presented here today.

4.1.2 Proposed System

This article examines how blockchain technology can be used to make the traditional supply chain for blood donations into a system that is decentralized, automated, and secure. In order to accomplish this goal, the study makes use of blockchain's inherent decentralization

as well as distributed ledger solutions. The text also exposes trust and safety vulnerabilities in the blood donation supply chain, including instances when people received contaminated blood from donors with HIV tests that were positive. Specifically, the language cites situations where people received infected blood from donors. The process flow of information at a typical Blood Control Center is defined in detail using a variety of functions, such as approveDonation(), separateBlood(), packBloodUnit(), and so on, which are carried out as elements of BDSC (blood donation supply chain). These functions are conducted as a part of the blood donation supply chain.

4.1.3 Result Analysis and Drawbacks

Participants in the blood donation supply chain are able to carry out their operations and track information at each stage with the assistance of the BDSC system, which makes use of a private blockchain network. The dashboard displays the number of donors, centers, distributors, and hospitals that have participated in many surgeries. Also displayed is the total number of end-users. While preserving everyone's right to personal secrecy, smart contracts outline the roles that each participant is expected to play and provide assistance in monitoring the flow of information throughout the process. The findings indicate strong levels of trust and safety across the entire process, from the point of origin to the end user, without disclosing any privacy problems for the individuals involved in each level of operation. These participants' actions were determined either by role-based smart contracts or by the system itself. Transactions are automatically discarded if they are found to be infected during laboratory testing; however, they will be completed overall if negative results occur. This is done so that an end-user can receive blood safely through this study's proposed Blockchain solution using Ethereum technology. This solution fares well against the risks associated with existing supplies of blood circulation management systems that are currently available worldwide today.

4.2 Paper Title: Blockchain-based management of blood donation

Published in:

- IEEE Access (2021) published by Institute of Electrical and Electronics Engineers (IEEE)

Authors:

- Hawashin, Diana
- Mahboobeh, Dunia Amin J.
- Salah, Khaled
- Jayaraman, Raja
- Yaqoob, Ibrar
- Debe, Mazin
- Ellahham, Samer

4.2.1 Introduction

The obstacles of managing blood donations are discussed, along with the ways in which blockchain technology can assist in finding solutions to these problems. It says that identifying the source information of donated blood in a trusted manner across the supply chain is crucial to assure authenticity and reduce hazards associated with transporting infective donated blood or counterfeit products. These risks can be reduced by tracing the source information of donated blood throughout the supply chain. A solution that is fully decentralized, auditable, traceable, transparent, private, secure, and trustworthy in the management of blood donation processes is proposed in this paper. This solution is built on Ethereum's private blockchain. The proposed system incorporates both smart contracts and algorithms in order to implement features and specify rules pertaining to the management of blood

donations. In addition, the system makes use of InterPlanetary File System (IPFS) in order to circumvent storage limits.

4.2.2 Proposed System

This article discusses a potential system that would use blockchain technology to manage blood donations. The foundation of the system is the private Ethereum network, which offers anonymity and secrecy to authorized individuals who are granted access to particular activities included within smart contracts. Smart contracts can be divided into two categories: production and consumption. These two categories each have their own unique set of permitted actors. IPFS is an example of a decentralized storage system that may be used to store huge files in a secure manner while still maintaining the hash values of those files on the ledger. Utilizing components of blockchain technology such as distributed ledgers, smart contracts, and application programming interfaces (APIs), among other things, the suggested solution has as its overarching objective the improvement of data security and transparency within blood management procedures.

This article discusses a potential method for managing blood donations that is based on blockchain technology. It provides an explanation of the sequence diagrams that describe the interactions between the various actors, intelligent contracts, and storage systems. While the consuming smart contract manages the ordering and distribution of blood units, the production smart contract is responsible for the collection, transportation, and creation of blood units. Both of these include a number of processes that are carried out by a variety of different entities, including as phlebotomists, technicians, doctors, and so on. These steps are represented through events that are triggered in each step of the process until the procedure is completed or until patients receive the necessary blood components.

4.2.3 Result Analysis and Drawbacks

The functioning of smart contracts for a proposed blockchain-based blood donation management system was tested and validated using Remix IDE, as is described in the text. Functions such as Collectwholebloodunit, Createbloodunit, BloodunitRequested, BloodUnitPrescrip-

tion, and BloodUnitTransfusion were put through their paces during the tests. Figures 6-10 illustrate the successful executions that were carried out. The inputs that were used during testing were not actual data but rather assumptions that were made specifically for the purpose of testing. Table 1 contains a list of the actors' addresses that are included in the smart contract.

This part of the article examines the potential use of a private Ethereum blockchain to track donated blood units as they are being transported and distributed. The method that has been proposed is adaptable to several other kinds of supply chains by making a few changes in the code that governs smart contracts. The primary distinction between donated blood and other products is that it must be stored at a particular temperature; nonetheless, tracking down its source can be accomplished in a manner analogous to that of other products by scanning the one-of-a-kind identification tags that are affixed to them. This system can be further customized by the use of straightforward alterations to accommodate various applications, such as the payments or fund transfers required in certain supply chains.

The article also provides an overview of the security evaluation conducted on the smart contracts that were designed for a blockchain-based blood donation management system. In order to test and validate the code against potential vulnerabilities and cyberattacks, the program known as Oyente was utilized. According to the findings, the code did not include any flaws, which indicates that it is safe for use on a decentralized network. Comparisons are made between the suggested solution and other, already-existing, non-blockchain-based solutions. The proposed approach is shown to have advantages in terms of decentralization, security features, tracing capabilities, and accountability assurance over the other options. In addition, contrasts are drawn between other blockchain platforms, such as Ethereum and Hyperledger Fabric, with reference to the programming languages utilized by each of these platforms. a platform's simplicity of use, the expense of its infrastructure, the consensus process it uses (proof-of-work versus non-fee payment), the storage alternatives it offers (on-chain versus off-chain), and so on and so forth.

4.3 Paper Title: Bloodchain: A blood donation network managed by blockchain technologies

Published in:

- Network journal published by Multidisciplinary Digital Publishing Institute (MDPI)

Authors:

- Le, Hai Trieu
- Nguyen, Tran Thanh Lam
- Nguyen, Tuan Anh
- Ha, Xuan Son
- Duong-Trung, Nghia

4.3.1 Introduction

The application of blockchain technology to a variety of industries, including Healthcare, is one of the topics covered in this article. In particular, the absence of effective blood supply chains and a lack of openness about the quality of blood and its origin have led to the development of a suggested blockchain-based system called BloodChain. This method intends to enable traceability along the blood donation supply chain while simultaneously protecting the privacy of all parties participating in the process. The study provides an overview of the structure of its approach and emphasizes the possible advantages of enhancing health outcomes through maintaining a stable blood supply and demand even in extraordinary circumstances.

4.3.2 Proposed System

A protocol known as BloodChain is proposed in this paper as a means of managing blood donation supply chains through the utilization of blockchain technology. The technology

is able to identify issues with the blood's origin and quality, which is particularly helpful for urgent requests. Donors, medical personnel, and transportation personnel are examples of stakeholders that participate in the process and transmit and receive data through the ledger component of the Hyperledger Fabric architecture. Donors are able to monitor the information that the ledger stores about them while at the same time retaining the privacy protection mechanisms that are regulated by smart contracts. Authorized users are able to acquire release information without sensitive data being published needlessly, which could violate privacy concerns. In order to improve transparency across all parties involved in the management of this complex network of transactions related to healthcare delivery systems such as those found within hospitals or clinics dealing with patient care needs including transfusions using donated blood samples collected voluntarily by individuals who wish to donate them for medical purposes, all queries are logged in the ledger. This allows stakeholders to know who accessed their data, when it was accessed, where it was accessed from, etc., which improves the quality of care patients receive.

4.3.3 Result Analysis and Drawbacks

The results of the requests sent to the create data function were measured; with one worker being initialized in the network, the initial number of requests is 1000 per second. The number of requests is continuously sent to the system within 2 min and gradually increases to 10,000 requests per second. From the results it can be observed that the number of successfully executed requests is relatively high compared to the failed requests; most of the successful requests are above 12,000 requests per second, while the number of failed requests only ranges from 2–3 requests. The article measures and evaluates the latency corresponding to three functions of data initialization, query, and update. The latency of the data initialization function of the blood samples, in most of the three latency levels are stable. Specifically, the maximum latency index fluctuates in the range of 500 s when the number of requests increases from 1000 to 10,000. The maximum latency index fluctuates quite strongly from 1.41 s/request to 4.64 s/request. However, the system's processing time gradually decreased and remained stable when the number of requests increased from 1000 to 10,000. The highest

value is 4.65 s for a total of 2000 requests.

In summary, this section discusses a research that was conducted to examine the effectiveness of BloodChain, a blood donation system that is based on blockchain technology. The authors subjected their system to a number of tests to determine how well it functioned in a variety of environments, and they discovered that, on the whole, it performed satisfactorily. They also explored various security and privacy risks that are associated with employing blockchain technology in healthcare settings, and they presented an approach for controlling access restriction based on user traits. This is one of the first studies to concentrate explicitly on the application of blockchain technology for blood management, as opposed to merely addressing generic problems in healthcare systems more generally.

Chapter 5

Our Proposition

5.1 Proposed System

The drawbacks of the conventional methods are addressed by the implementation of the proposed system, which incorporates both blockchain technology and a database into its workings. Because the system deals with sensitive information pertaining to patients' health, the usage of private Ethereum is necessary to secure patient anonymity. Additionally, as a consequence of this, gas payments associated with the execution of smart contracts that are applicable on the public Ethereum chain will no longer be required when paid for using real ether. In addition, it provides increased throughput while simultaneously lowering the amount of latency, which, from the point of view of the user, results in a seamless integration process for the blockchain. Figure 4 provides a high-level representation of the architecture of the proposed BloodComm system. The components of the architecture are broken down and discussed in this section.

- **Users:** Users will be able to register for the platform by supplying the necessary details, at which point they will be issued a wallet address that will be used to identify them when they are utilizing the blockchain. One individual can play the role of both a blood donor and a recipient, depending on the specifics of their situation. When a user has a need for blood, they have the ability to submit a request by providing the pertinent information (such as the blood group, the quantity needed, the organization where the

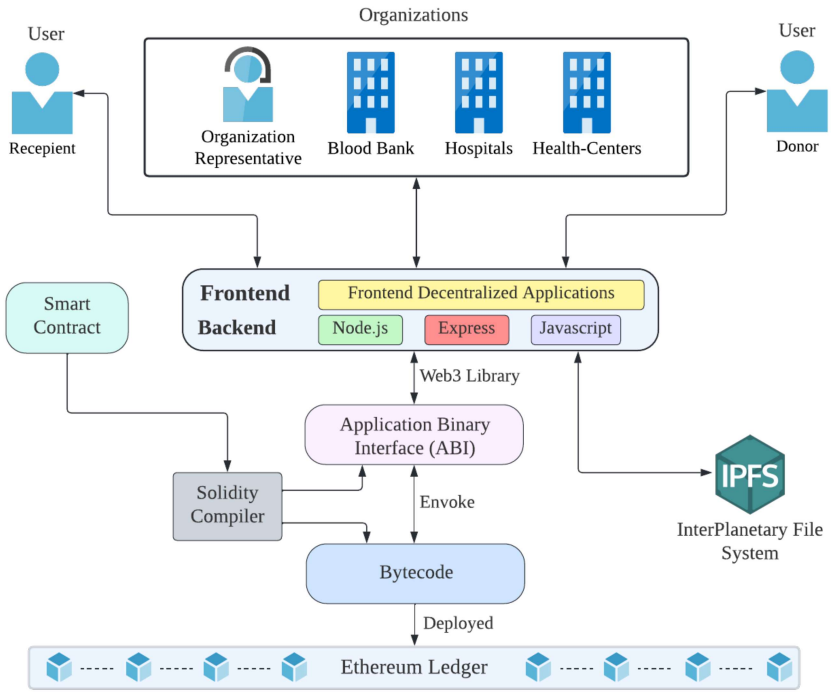

Figure 5.1: System architecture.

transfusion will take place, etc.). Through the use of a request feed, this request will be distributed to the contributors who are the most closely matching the criteria. Users who have the capacity to make donations will have the ability to respond to the request whenever it is most convenient for them. Following the completion of a successful transfusion, the system is designed to give the donor the option of being rewarded with fungible tokens of varying types. The location of both the donors and the recipients will be taken into consideration by the system, which will then present potential donors with the request for a blood group that is geographically nearest to them. Due to the high rate of change caused by the fact that blockchain users are mobile, the coordinates will not be stored on the blockchain.

- **Organizations**: Blood banks, hospitals, and other healthcare facilities will all be included in the organization registration process. These groups make blood transfusions easier to do. In contrast to users, organizations rarely relocate their physical locations. It will be saved on the distributed ledger (blockchain). In the case of a transfusion, an

organization will play the role of a meeting place for the two persons involved.

- **Organization Representatives**: Representatives will engage with our system while acting in the roles of their respective organizations throughout the interaction. After a blood transfusion, an organization will enter into the ledger the scanned copies of the test findings as well as the hash value associated with them. This additional layer of verification ensures that the blood was tested in a reputable medical facility, which means that the hospital might be held accountable in the event that a contaminated blood transfusion was administered.

- **Application:** It is planned to make use of front-end Distributed Applications (DApps), which will be complemented by a suitable back end. The Application Binary Interface, or ABI, will be the medium via which the back end will have communication with the blockchain ledger.

- **IPFS:** In order to alleviate some of the strain placed on the blockchain, some data will be saved in the IPFS. For instance, the results of scanned tests will be included in the IPFS. Their hash values will be recorded on the blockchain in order to facilitate a comparison between them at a later time. The information that will be given to users of the application will have been obtained in a seamless manner from both the IPFS and the blockchain.

- **Smart Contract:** The programming language known as Solidity was utilized in the creation of the smart contract. This section of code will be compiled with the Solidity compiler, which will result in the production of two files: an Application Binary Interface (ABI) and a Bytecode file. While the bytecode is being executed in the ledger, the application binary interface (ABI) is used for interfacing purposes between the application and the ledger.

- **Ethereum Ledger:** The immutable distributed ledger that holds all of the records as well as the bytecodes of the smart contracts that have been deployed.

5.2 Interaction of Entities Through the System

A sequence diagram is depicted in the following figure 5.2 for the purpose of gaining a better grasp of the system's aforementioned actors. We are going to take into consideration a *blood recipient* who is in urgent need of blood. A request for a blood donation would be made by the receiver, who would do so by inputting the necessary information into the internet access point that was provided. After that, this will be sent to the blockchain ledger through APIs, where it will be kept in an irreversible manner. These requests will be distributed to other users by the program, which will do so by integrating data from the ledger and the database. Now, a donor will respond to this request and notify the recipient of their response. At this point, both parties speak with one another and meet at the organization that was previously agreed upon in order to finish the procedure of the transfusion. To show their gratitude to the giver, the recipient may provide the giver with a gift as a form of reward. When seen from the perspective of the blockchain, the logic chain is carried out through the use of smart contracts, and the results are permanently recorded in the distributed ledger.

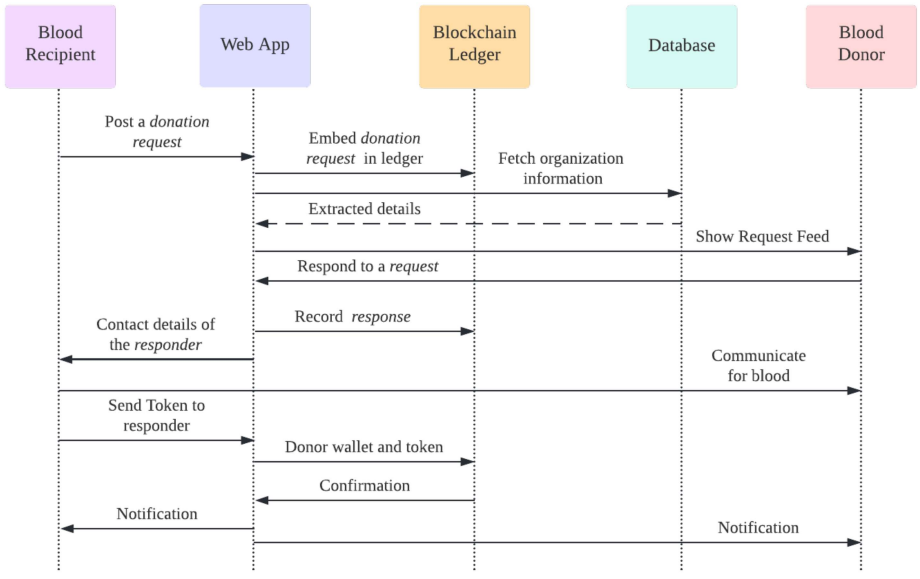

Figure 5.2: Sequence diagram of the system.

Chapter 6

Implementation and Evaluation

6.1 Smart Contract Implementation

The "User," "Request," and "Token" Solidity contracts make up the system implementation. Together, these contracts make it easier for system users to register, publish and answer requests, and handle tokens.

The basis for user management is the "User" contract. Users may register by entering personal information including their name, contact information, blood type, gender, and birth date. For quick retrieval, registered users are saved in arrays and mappings. The contract contains tools for retrieving user data, totaling registered users, and determining a user's legitimacy. Users must be authorized and registered under the "User" contract in order to interact with the "Request" contract.

Users can publish requests and reply to already posted requests using the "Request" contract. Users can post requests by providing information on the patient, including name, contact information, blood type, gender, location, and other characteristics. Along with the address of the requester and the state of the request's fulfillment, this data is kept in the contract. Users can update the completion status of their own requests, search for requests based on their ID or the address of the requester, and react to requests from other users. The "User" contract is a prerequisite for the "Request" contract since it verifies user addresses and makes sure that only authorized users may interact with the contract.

The "Token" contract implements the ERC20 token standard, creating a token called "BloodNode Coin" with the symbol "BDC" and 2 decimal places. It manages token balances for different addresses within the system. The initial total token supply is set to 100,000, and these tokens are assigned to the wallet address of the contract deployer. Users can transfer tokens to other addresses, approve token transfers for delegated use, check token allowances, receive approval, and execute functions in a single call. The "Token" contract provides a means of exchange or value within the system, allowing users to utilize the "BloodNode Coin" tokens for various purposes.

With the use of the "BloodNode Coin" tokens, users may register, submit requests, answer requests, and trade value utilizing the contracts that make up the system. While the "Request" contract enables interactions linked to requests, the "User" contract assures the accuracy and maintenance of user information. The "Token" contract expands the functionality of the system by integrating a token-based economy. This system implementation offers a base for creating platforms and apps that handle token transactions, user registrations, and requests.

Table 6.1: Basic information about User

BASIC_INFORMATION_OF_USER
Name: User Name DOB: Date of Birth Contact: Contact Number Blood Group: 'A+'/'A-'/'B+'/'B-'/'AB+'/'AB-'/'O+'/'O-' Gender: 'male'/'female'/'others'

Table 6.2: Data Structure of User

Prefix	USER
Key	User_ID { id: user_id
Value	BASIC_USER_INFORMATION wallet_address: User's wallet address }

Table 6.3: Basic information about Patient

BASIC_INFORMATION_OF_PATIENT
Name: Patient Name Contact: Patient's Contact Number Blood Group: 'A+'/'A-'/'B+'/'B-'/'AB+'/'AB-'/'O+'/'O-' Gender: 'male'/'female'/'others' Description: Information about the requirements

Table 6.4: Basic information about Blood Donation Center

BASIC_INFORMATION_OF_BLOOD_DONATION_CENTER
Name: Center's Name Contact: Center's Contact Number Address: Center's Address Location: Coordinates of the center

Table 6.5: Data Structure of Request Format

Prefix	REQUEST
Key	Request_ID
	{
	id: request_id
	BASIC_PATIENT_INFORMATION
Value	BASIC_INFORMATION_OF_BLOOD_DONATION_CENTER
	wallet_address: Requester wallet address
	isCompleted: 'yes'/'no'/'ongoing'
	}

Algorithm 1 Algorithm for User Registration

Inputs:

BUI: Basic Information of User;

UWA: User Wallet Address;

Outputs:

```
if addressRegistred(UWA) == True then
    User already Registered!;
else
    registerAddress(UWA);
    user_id = id;
    BASIC_USER_INFORMATION = BUI;
    wallet_address = UWA;
end if
```

Algorithm 2 Algorithm for Posting Request

Inputs:

BUP: Basic Information of Patient;

BUBDC: Basic Information of Blood Donation Center;

RQWA: Requesterś Wallet Address;

Outputs:

```
if addressRegistred(RQWA) == True then
    request_id = id;
    Request.BASIC_Information_of_Patient = BUP;
    Request.BASIC_Information_of_BDC = BUBDC;
    BASIC_USER_INFORMATION = BUI;
else
    User do not have the permission;
end if
```

Algorithm 3 Algorithm for Responding to Requests

Inputs:

RPWA: RespondersẂallet Address;

RL: Request List;

Outputs:

```
if addressRegistred(RPWA) == True then
    for request in RL do
        if applicable(RPWA) == True then
            Responder[request_id] += RPWA;
        else
            Not applicable for this request;
        end if
    end for
else
    User do not have the permission;
end if
```

Algorithm 4 Algorithm for Information Sharing and Tokenization

Inputs:

BURQ: Basic Information of Requester;

RQWA: Requesterś Wallet Address;

RPWA: Responderś Wallet Address;

RPL: Responders List;

Outputs:

```
1: if addressRegistred(RQWA) == True then
2:     for responders in RPL[request_id] do
3:         if donorApplicable(responders) == True then
4:             sharePersonalInformation(responders);
5:             requestHasBeenCompleted(request_id);
6:             if requestCompleted(request_id) == True AND enoughBalance(RQWA) == True
   then
7:                 TransferToken(Amount, responder.RPWA);
8:             else
9:                 Process can not be completed.;
10:            end if
11:        else
12:            Donor is not applicable;
13:        end if
14:    end for
15: else
16:    User do not have the permission;
17: end if
```

6.1.1 Simulating the Private Network

As we are building a private ethereum network, there must be constituent peers who will be performing transactions, validation, and storing copy of the blockchain ledger. In our

experiment, we used docker to simulate peers in the network. The official `geth` image from the docker hub was pulled locally and a custom `genesis.json` (refer to fig. 6.1) was pasted in the base images. Combining together, the full docker file was formed and a docker image was compiled from the new docker image as shown in fig. 6.2. Finally, 25, 50, 75, and 100 docker containers were made to run in `daemon mode` in the local system using this docker file.

```
{
  "config": {
    "chainId": 1214,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "ethash": {}
  },
  "difficulty": "1",
  "gasLimit": "12000000",
  "alloc": {}
}
```

Figure 6.1: Custom definition of genesis block.

```
FROM ethereum/client-go:v1.10.1

ARG ACCOUNT_PASSWORD

COPY genesis.json /tmp

RUN geth init /tmp/genesis.json \
    && rm -f ~/.ethereum/geth/nodekey \
    && echo ${ACCOUNT_PASSWORD} > /tmp/password \
    && geth account new --password /tmp/password \
    && rm -f /tmp/password

ENTRYPOINT ["geth"]
```

Figure 6.2: Docker file for building images.

6.2 Smart Contracts Analysis with Securify2

The smart contracts are coded with Solidity version `>=0.5.12` and compiled with the required Solidity compilers. This is done in order to ensure that they function properly. Once a smart contract has been deployed, unlike other software components, it cannot be changed in any meaningful way until the system is restarted and forced to begin its operations from the beginning once more. This is one of the distinguishing features of the development of smart contracts. Because of this, the need of doing a security assessment on smart contracts cannot be overstated. It is possible that the attack on the DAO in the year 2016 was the first demonstration of the severe consequences of insecure smart contracts [33]. The exploitation of weaknesses in smart contracts is only expected to grow in the future years, resulting in the loss of millions of dollars. A well-known multi-signature wallet known as Parity was targeted in a theft that occurred in July of 2017, and about 30 million USD was taken. After only a few months, another problem in the same wallet caused approximately 280 million dollars to be frozen. Even if the system we have described is not a finance application, there are still potential security risks that might result in severe damage. For instance, an attacker could get control of users' accounts via fuzzing, manipulate tokens, launch denial of service attacks, and so on.

The use of smart contracts in permissionless blockchains raises some security concerns because they make it possible for entities that do not trust one another to interact with one another without relying on trusted third parties. In order to solve this issue within our systems, we have used Securify, which is a fully automated and scalable static security analyzer for Ethereum smart contracts. It is capable of demonstrating that certain contract behaviors are either safe or unsafe in relation to a specific property. The analysis is broken down into two stages: the first is the symbolic analysis of the dependency graph of the contract, and the second is the checking of compliance and violation patterns that are specified in a domain-specific language. Securify has performed security audits on more than 18,000 user-submitted contracts and is frequently utilized by industry professionals for this purpose. Securify currently has 37 vulnerabilities with documentation of their severity level with Low, Medium, High, and Critical [34].

We have in total 3 smart contracts (i.e. `.sol` files). We ran smart contracts for all three and generated their security reports. Since the reports are too long to present here in this document, a fraction is given here.

Listing 6.1: Security analysis of user.sol

```
Name: ShadowedStateVariable
Severity: HIGH
Description: State variables in inherited contract should not
    be named identically to inherited variables
Name: ShadowedLocalVariable
Severity: MEDIUM
Description: Reports local variable declarations that shadow
   declarations from outer scopes.
Name: SolcVersion
Severity: LOW
Description: Avoid complex solidity version pragma statements
   .
Name: TooManyDigits
Severity: INFO
Description: Usage of assembly in Solidity code is
   discouraged.
Name: UninitializedLocal
Severity: INFO
Description: A variable is declared but never initialized.
Name: AssemblyUsage
Severity: INFO
Description: Usage of assembly in Solidity code is
   discouraged.
Name: IncorrectERC721Interface
Severity: MEDIUM
```

```
Description: Incorrect signature for ERC721 interface
   functions.
```

A graph in fig. 6.3 illustrates the overall number of threats per severity level in three analysis reports. Here we can see only one critical level threat is present in the request

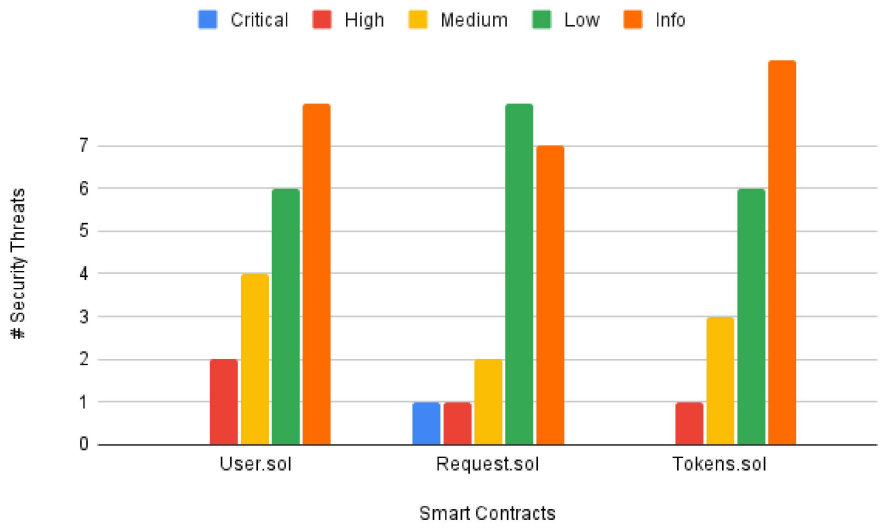

Figure 6.3: Number of issues detected per smart contracts.

6.3 Using Ethereum 2.0

We implemented these steps to build a private Ethereum 2.0 network using Docker. First, we verified that Docker was set up on our computers. We were able to run containers that included the Ethereum 2.0 network nodes because of Docker.

We then established a Docker network to allow container communication. Our own Ethereum 2.0 network was built on top of this network. We then created the network nodes after setting up the network.

The Beacon Node, which was in charge of overseeing the Ethereum 2.0 network and generating fresh blocks, was the first thing we built. We constructed a container, particularly for the Beacon Node using Docker instructions, and we linked it to the Docker network we

had previously established. The administration of the network and block production was handled by this container.

The container for the Validator was then made. In the Ethereum 2.0 network, validators took part by approving and recommending blocks. We built a Docker container just for the Validator and joined it to the Docker network, much like the Beacon Node. This container helped the network's consensus and validation processes.

We created a connection between the Beacon Node and the Validator after both containers had been set up. The Validator container was connected to the Beacon Node container via the Docker network once we had the container IDs for both the Beacon Node and Validator containers. The Validator might interface with the Beacon Node and take part in the Ethereum 2.0 network thanks to this connectivity.

We ran a ping test to ensure that the Beacon Node and the Validator were connected. This test verified that the containers were linked to the private network and could communicate with one another.

We were able to access each container separately once the ping test was successful. We may access the Beacon Node or Validator containers with Docker instructions to carry out particular operations or communicate with the Ethereum 2.0 network. We were able to control the network, carry out orders, and keep tabs on the development of our own Ethereum 2.0 network thanks to this access.

Overall, by following these instructions, we were able to use Docker to successfully build a private Ethereum 2.0 network. We were able to join the Ethereum 2.0 network and carry out numerous tasks relating to block production, validation, and consensus thanks to the Beacon Node and Validator containers connected within the Docker network.

6.4 System Evaluation

6.4.1 Gas Consumption Analysis

The examination of gas consumption based on the supplied data reveals intriguing tendencies among various peer amounts.

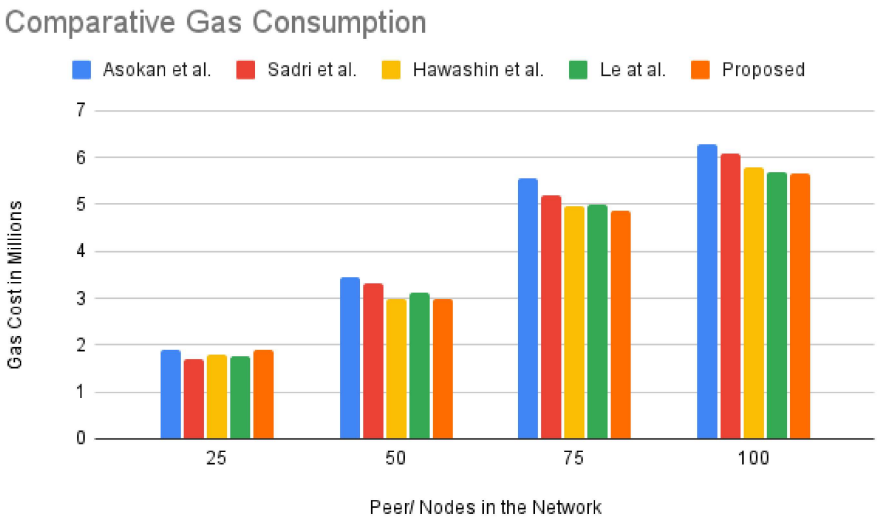

Figure 6.4: Comparative Analysis of Gas Consumption

Gas usage for the 25 peers ranged from 1.7 to 1.9. The gas consumption of 1.9 in both Asokan et al.'s and our method indicates a similar efficiency in this case.

The gas consumption rises when you reach 50 peers, with values between 3 and 3.45. The maximum gas consumption is shown by Asokan et al., whereas the lowest consumption is shown by our method, at 3.

Gas consumption continues to increase for 75 peers, with numbers ranging from 4.88 to 5.55. Asokan et al. once more show the largest gas consumption, but our method continues to have a considerably lower consumption of 4.88.

The gas consumption finally peaks at 100 peers, ranging between 5.65 and 6.3. While Asokan et al. continues to indicate the greatest gas consumption, our method shows a consumption that is somewhat lower at 5.65.

The data as a whole show from 6.4 that our method effectively and efficiently manages gas usage while maintaining competitive gas consumption across various peer amounts. To properly evaluate the effectiveness and acceptability of each strategy in certain circumstances or scenarios, though, more research and considerations may be required.

6.4.2 Throughput Analysis

The throughput analysis data reveals information about how well the system performs when there are different numbers of peers.

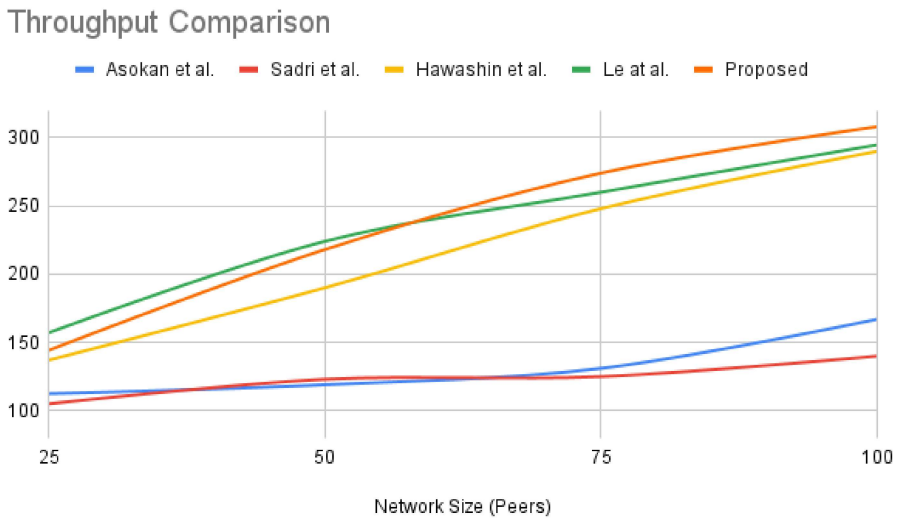

Figure 6.5: Comparison of Throughput Analysis

Asokan et al.'s throughput for 25 peers was 112.5, the highest of the examined techniques. Our method and that of Sadri et al. both showed somewhat lower throughputs of 105 and 144.14, respectively. Higher throughputs of 137 and 157, respectively, demonstrated by Hawashin et al. and Le et al., showed how well they were able to manage the workload.

Sadri et al. showed an improvement in throughput with a value of 123 when the number of peers reached 50, whereas our technique also showed a significant rise to 218. Throughputs of 119 and 224 were attained by Asokan et al. and Le at al., respectively, demonstrating their capacity to manage the increased workload. The strategy used by Hawashin et al. had the greatest throughput of all methods at 190, demonstrating its effectiveness in managing a bigger peer network.

Hawashin et al. showed a considerable increase in throughput, reaching 248 for 75 peers. Le et al. also showed enhanced performance with 260 throughputs. Sadri et al. consistently achieved a throughput of 125, but our method showed a significant increase to 274. Asokan

et al. were able to handle a greater number of peers because of their throughput of 131.

Last, but not least, Hawashin et al. showed the greatest throughput of 290 with 100 peers, demonstrating its effectiveness in controlling a bigger network. Le at al. came in second with a throughput of 294.7, whereas our method came in first with a throughput of 308, demonstrating its capacity to manage more peers. Indicating their capacity to manage the increased workload, Sadri et al. attained a throughput of 140 whereas Asokan et al. demonstrated a throughput of 167.

Overall, the data show from 6.5 that different techniques and peer amounts have variable throughput performances. Each strategy demonstrates its merits in various circumstances, demonstrating how crucial it is to take into account certain needs and network sizes when assessing the system's performance.

6.5 Challenges

We face a number of difficulties while working with the Ethereum source code, which must be overcome for effective development and deployment.

- **Choosing the correct client implementation:** Geth, Parity, and OpenEthereum are just a few of the many client implementations that Ethereum provides. When choosing the best client implementation, we must carefully take into account elements like performance, stability, community support, and individual project needs. We must evaluate and select the implementation that best suits the requirements of our project because each one has unique features, advantages, and considerations.

- **Smart Contracts unit testing and security analysis:** The foundation of decentralized apps on Ethereum is smart contracts. We must carry out thorough unit testing to make sure they are secure and reliable. Ensure that Smart Contracts behave properly, this entails developing extensive unit tests that cover a variety of scenarios, edge situations, and input validations. To find and address any vulnerabilities and attack vectors, we also need to do rigorous security analysis, including code reviews, audits, and vulnerability assessments.

- **Gas consumption optimization:** On the Ethereum network, the computational power necessary to execute transactions and Smart Contracts is measured in terms of gas. Costs of transactions and scalability are directly impacted by gas use. We must use effective coding techniques, such as limiting storage operations, avoiding pricey calculations, and optimizing data structures, to reduce gas consumption. We can make sure that our application stays within gas constraints and continues to be cost-effective for end users by carefully addressing gas expenses during development.

In order to overcome these obstacles, we must make use of our technological know-how, follow industry standards, and use exhaustive testing techniques. We must keep up with the most recent developments and community debates in the Ethereum ecosystem. In this approach, we may take advantage of advancements, tools, and methods that boost the effectiveness, security, and performance of our development processes.

Chapter 7

Conclusion

7.1 Summary

After conducting an exhaustive investigation of the current corpus of scholarly literature, we discovered substantial flaws in the works that have previously been published in terms of their understanding of the problem statement at hand, their implementation, their in-depth security study of the smart contracts, and their reproducibility. As a result, we suggest a system that will, on the whole, overcome all of these deficiencies and contribute to the area of application-based blockchain research.

7.2 Future Work

We want to carry out a thorough review of our system in upcoming activities. In order to evaluate the system's performance, scalability, and stability, a sizable number of users, transactions, and interactions will be used during testing. We may learn a lot about how our system functions in actual situations and spot any possible bottlenecks or opportunities for development by carrying out such extensive assessments.

We also intend to concentrate on integrating IoT devices for automatic record-keeping. The seamless collection and recording of data from diverse sensors and devices will be made possible by the integration of IoT devices into our system. By supplying timely and

precise data inputs that may be used for decision-making, analytics, and other functions, this integration will improve the system's capabilities. By integrating IoT devices into the record-keeping process, we can automate data management, assure the accuracy and consistency of recorded information, and speed record-keeping.

In today's digital world, protecting user privacy is a vital problem. De-identification of users will thus be given priority in order to protect their privacy. De-identification methods are designed to eliminate or encrypt personally identifying information (PII) from user data while still enabling analysis and insight-generating uses for the information. We can preserve user privacy while still using their data for other things like research, analytics, and system upgrades by putting in place reliable de-identification procedures. This strategy guarantees adherence to privacy laws and increases user confidence in our system.

In conclusion, our future work will involve undertaking extensive assessments of system performance, including IoT devices for automatic record keeping, and putting into practice efficient de-identification methods to protect user privacy. These initiatives will improve our system's usability, scalability, and privacy while also boosting its overall efficacy and user happiness.

Bibliography

[1] "Ethereum energy consumption index," Dec 2022.

[2] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, pp. 557–564, 2017.

[3] F. A. Sayani and J. L. Kwiatkowski, "Increasing prevalence of thalassemia in america: Implications for primary care," *Annals of Medicine*, vol. 47, no. 7, pp. 592–604, 2015. PMID: 26541064.

[4] M. S. H. Jiisun, R. A. Rupa, M. H. Chowdhury, H. Mushrofa, and M. R. Hoque, "Blood Donation Systems in Bangladesh: Problems and Remedy," *International Journal of Business and Management*, vol. 14, pp. 145–145, July 2021.

[5] American Red Cross, "US Blood Supply Facts." Accessed: Sep. 15, 2022.

[6] E. M. d. Oliveira and I. A. Reis, "What are the perspectives for blood donations and blood component transfusion worldwide? a systematic review of time series studies," *Sao Paulo Medical Journal*, vol. 138, pp. 54–59, 2020.

[7] American Cancer Society, "Cancer facts amp; figures 2022." Accessed: Sep. 15, 2022.

[8] The Leukemia amp; Lymphoma Society, "Lymphoma Survival Rate Blood Cancer Survival Rates." Accessed: Sep. 15, 2022.

[9] B. H. Shaz, "Chapter 66 - transfusion transmitted diseases," in *Transfusion Medicine and Hemostasis* (C. D. Hillyer, B. H. Shaz, J. C. Zimring, and T. C. Abshire, eds.), pp. 361–371, San Diego: Academic Press, 2009.

[10] N. Kube, "Daniel drescher: Blockchain basics: a non-technical introduction in 25 steps," 2018.

[11] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," *CoRR*, vol. abs/1906.11078, 2019.

[12] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," in *Conference on the Theory and Application of Cryptography*, pp. 437–455, Springer, 1990.

[13] P. Patil, P. Narayankar, D. Narayan, and S. M. Meena, "A comprehensive evaluation of cryptographic algorithms: Des, 3des, aes, rsa and blowfish," *Procedia Computer Science*, vol. 78, pp. 617–624, 2016.

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

[15] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business & Information Systems Engineering*, vol. 59, no. 3, pp. 183–187, 2017.

[16] R. S. Bhadoria, Y. Arora, and K. Gautam, "Blockchain hands on for developing genesis block," in *Advanced applications of blockchain technology*, pp. 269–278, Springer, 2020.

[17] F. Hofmann, S. Wurster, E. Ron, and M. Böhmecke-Schwafert, "The immutability concept of blockchains and benefits of early standardization," in *2017 ITU Kaleidoscope: Challenges for a Data-Driven Society (ITU K)*, pp. 1–8, IEEE, 2017.

[18] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Comput. Surv.*, vol. 52, jul 2019.

[19] N. T. Nguyen, "Consensus system for solving conflicts in distributed systems," *Information Sciences*, vol. 147, no. 1, pp. 91–122, 2002.

[20] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[21] M. Saad, L. Njilla, C. Kamhoua, J. Kim, D. Nyang, and A. Mohaisen, "Mempool optimization for defending against ddos attacks in pow-based blockchain systems," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 285–292, 2019.

[22] S. Balsamo, I. Malakhov, A. Marin, and I. Mitrani, "Transaction confirmation in proof-of-work blockchains: auctions, delays and droppings," in *2022 20th Mediterranean Communication and Computer Networking Conference (MedComNet)*, pp. 140–149, 2022.

[23] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "Sok: Consensus in the age of blockchains," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, AFT '19, (New York, NY, USA), p. 183–198, Association for Computing Machinery, 2019.

[24] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, (New York, NY, USA), p. 3–16, Association for Computing Machinery, 2016.

[25] R. Zhang and W. K. V. Chan, "Evaluation of energy consumption in block-chains with proof of work and proof of stake," *Journal of Physics: Conference Series*, vol. 1584, p. 012023, jul 2020.

[26] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, no. 1, 2012.

[27] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology* (J. Garcia-Alfaro, G. Navarro-Arribas, H. Hartenstein, and J. Herrera-Joancomartí, eds.), (Cham), pp. 297–315, Springer International Publishing, 2017.

[28] F. Saleh, "Blockchain without Waste: Proof-of-Stake," *The Review of Financial Studies*, vol. 34, pp. 1156–1190, 07 2020.

[29] P. R. Nair and D. R. Dorai, "Evaluation of performance and security of proof of work and proof of stake using blockchain," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 279–283, 2021.

[30] F. Schuh and D. Larimer, "Bitshares 2.0: general overview," *accessed June-2017.[Online]. Available: http://docs. bitshares. org/downloads/bitshares-general. pdf*, 2017.

[31] M. Castro, B. Liskov, *et al.*, "Practical byzantine fault tolerance," in *OsDI*, vol. 99, pp. 173–186, 1999.

[32] B. Chase and E. MacBrough, "Analysis of the XRP Ledger Consensus Protocol," *arXiv preprint arXiv:1802.07242*, 2018.

[33] M. del Castillo, "The dao attacked: Code issue leads to $60 million ether theft," *Saatavissa (viitattu 13.2. 2017)*, vol. 3, 2016.

[34] P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Bünzli, and M. Vechev, "Securify: Practical security analysis of smart contracts," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, (New York, NY, USA), p. 67–82, Association for Computing Machinery, 2018.