T.C.

ALTINBAS UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

BLOCKCHAIN IN HEALTHCARE: SMART CONTRACTS TO IMPROVE DENTAL HEALTHCARE FOR CHILDREN IN MIXED DENTITION PERIOD

Wildan Mohammed Araby AL-RUBAYE

Master of Science

Supervisor

Asst. Prof. Dr.Sefer KURNAZ

Istanbul, 2021

BLOCKCHAIN IN HEALTHCARE: SMART CONTRACTS TO IMPROVE DENTAL HEALTHCARE FOR CHILDREN IN MIXED DENTITION PERIOD

by

Wildan Mohammed Araby Al-Rubaye

Electrical and Computer Engineering

Submitted to the Institute of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled "BLOCKCHAIN IN HEALTHCARE: SMART CONTRACTS TO IMPROVE DENTAL HEALTHCARE FOR CHILDREN IN MIXED DENTITION PERIOD" prepared and presented by "Wildan Mohammed Araby AL-RUBAYE" was accepted as a Master of Science Thesis in Electrical and Computer Engineering.

____

Asst. Prof. Dr. Sefer KURNAZ
Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Sefer KURNAZ	School of Engineering and Natural Sciences, Altinbas University	____________________
Asst. Prof. Dr. Oğuz KARAN	School of Engineering and Natural Sciences, Altinbas University	____________________
Asst. Prof. Dr. Zeynep ALTAN	Faculty of Engineering and Architecture, Beykent University	____________________

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies:
____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Wildan Mohammed Araby Al-Rubaye

Signature

DEDICATION

I dedicate this thesis to God Almighty my creator, my strong pillar, my source of inspiration, wisdom, knowledge and understanding.

This thesis work is also dedicated to my father "Mohammed", my sister "Ban" and my brother "Ali", who have been a constant source of support and encouragement during the challenges of graduate school and life. I am truly thankful for having them in my life. I also dedicate it to my mother's soul.

I also dedicate it to my best friends "May", "Maryam" and "Sazan", who encourage and support me, thank you very much.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Sefer KURNAZ, I wish to thank my committee members.

I also appreciate all the support I received from my family and friends.

ABSTRACT

BLOCKCHAIN IN HEALTHCARE: SMART CONTRACTS TO IMPROVE DENTAL HEALTHCARE FOR CHILDREN IN MIXED DENTITION PERIOD

AL-RUBAYE, Wildan Mohammed Araby,

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Sefer KURNAZ

Date: 06 /2021

Pages: 102

Privacy preserving is a matter of great importance in many sensitive fields including financial and healthcare areas. As the blockchain technology has achieved an obvious success in privacy protection at the financial level when the Bitcoin has been introduced in 2008, there are considerable attempts and later on successful projects to introduce the blockchain technology to the healthcare fields, where the patient privacy and access control to the personal information is at high priority.

The aim of our thesis mainly considers the introduction of the smart contracts that essentially depend on the technology of blockchain to act as a method for preserving patients' privacies along with improving the reality of dental healthcare providing services.

The main focus will be on the treatment plan of the patient by building a smart contract system between multiple partners. Firstly, the problem will be presented that is why there is a need for implementing blockchain system of smart contracts to improve dental healthcare providing? And why we need to provide digital organizing of treatment plan? Secondly, The Blockchain is discussed as a relatively new technology. Thirdly, the design to solve the problem will be proposed and the system of smart contracts will be built, with explanation of the way by which the system will meet the requirement to give the solutions. Finally, there will be a discussion of how proposed design will be implemented.

By using “the five types models” our proposed smart contract system is built combining the advantages of the blockchain technology with the smart contracts benefits to provide decentralized, tamper resistant, and privacy preserving properties. The programming language that we have used is Solidity along with proof of authority algorithm on Ethereum blockchain.

The proposed blockchain-based smart contract system can provide high level of privacy preservation in the field of patient records in addition to the digitalization of healthcare data.

Keywords: Blockchain, Smart contracts, Healthcare, Solidity

TABLE OF CONTENTS

Pages

LIST OF TABLES

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

HER	:	Electronic Healthcare Records
HGD	:	Healthcare Data Gateway
ICS	:	Indicator-Centric Schema
APMs	:	Alternative Payment Models
MedRec	:	Medical Record
HIE	:	Health Information Exchange
NIST	:	National Institute of Standards and Technology
ISO/TC	:	International Organization for Standardization /Technical Committees
DLT	:	Distributed Ledger Technology
PoW	:	Proof of Work
PoS	:	Proof of Stake
DPoW	:	Delayed Proof of Work
DPoS	:	Delegated Proof of Stake
PBFT	:	Practical Byzantine Fault Tolerant
DBFT	:	Delegated Byzantine Fault Tolerance
PoA	:	Proof of Authority
UNIX	:	Uniplexed Information and Computing System
PoET	:	Proof of Elapsed Time
SGX	:	Software Guard Extension

CPU	:	Central Processing Unit
EOA	:	Externally-Owned Accounts
ID	:	Identity Document
EVM	:	Ethereum Virtual Machine
BFT	:	Byzantine Fault Tolerance
PII	:	Personally Identifiable Information
API	:	Application Programming Interface
NTT	:	Nippon Telegraph and Telephone Public Corporation
SHA-3	:	Secure Hash Algorithm 3
NG	:	Next Generation
TTP	:	Trusted Third Party
HIPAA	:	The Health Insurance Portability and Accountability Act
ONC	:	The Office of the National Coordinator
IBM	:	International Business Machines
IoT	:	Internet of Things
DPS	:	Data Preservation System
BlocHIE	:	A Blockchain-Based Platform for Healthcare Information Exchange
PHD	:	Personal Healthcare Data
EMR	:	Electronic Medical Records
FHIR	:	Fast Healthcare Interoperability Resources
HPE	:	Health Professions Education

DDBMS	:	Distributed Database Management Systems
OBC	:	Open Blockchain
VM	:	Virtual Machine
API	:	Application Programming Interface
GUI	:	Graphical User Interface
ABI	:	Application Binary Interface
DApp	:	Decentralized Application
DFS	:	Distributed File System
IDE	:	Integrated Development Environment
CMC	:	Contract Managing Contracts
ALC	:	Application Logic Contracts
URL	:	Uniform Resource Locator
HTML	:	Hyper Text Markup Language

1. INTRODUCTION

Blockchain technology has potentially changed the healthcare system, prioritizing the patient by preserving its privacy, boosting the security, and providing interoperable data record. Blockchain application in healthcare has great advantages and significant benefits in improving health records management and sharing, enhancing the insurance claim, and providing an advanced health data ledger. These advantaged and benefits lead to introduction of blockchain technologies in dental healthcare as it is progressing and developing field of healthcare.

By coupling these advantages of blockchain with the smart contract system we introduce a suggestion with main goal is focusing on the improvement of dental healthcare providing, targeting the pedodontic patients at the primary school age.

1.1 RESEARCH PROBLEM:

The research problem can be discussed by three questions:

1.1.1 Why the Primary School Age?

Oral health problems can be presented as dental pain which consequently leads to tooth loss that can affect the child appearance, life quality, and nutrition with its direct developmental and growth effects on the child. Dental caries and diseases of the gum are widespread conditions, affecting approximately 80% of school aged children in some areas.

Schools constitute an active environment to promote oral health as there are over 1 billion children worldwide. Thus, Oral health behaviors can be established throughout the school age. In addition to that, Children from age of 6 years till age of 12 years are in mixed dentition period and they are in need to deliver organized dental health care.

The cost for treating dental caries could simply consume the country's healthcare providing budget for children. However, the neglect of this issue will cost more with its financial, social and personal impacts[1].

1.1.2 What Are the Main Problems Facing Dental Healthcare Providing?

The main problems facing dental healthcare providing can be concluded by: The First problem is that the dentist in the primary healthcare center should go to the primary schools at the beginning of the study year to examine each child and fill a diagnosis casesheet to be able to make a treatment plan for each child. This process takes much time and effort.

The second one is that the dentist should write a letter for the parents to inform them with their child's dental problems and they will deliver it by the primary school. After they receive the letter, they should bring their child to the primary healthcare center to have the child treated. With this complicated process the parents usually do not bring their child to the healthcare center unless there is pain and this is when there is irreversible phase of dental problem so the treatment will be more invasive with more cost.

1.1.3 How Can the Privacy and Personal Information Be Preserved?

As the patients' personal information and healthcare data are considered as sensitive and of great importance, they should be preserved from any tampering attempts.

1.2 SUGGESTED SOLUTIONS:

The child should be examined medically and dentally before joining the primary school, the data of the child will be uploaded to a secured system of blockchain technology, so that the permission of accessing to these data is given to the parents and the dentist they choose to treat their child. The dentist can schedule a treatment plan with follow up visits. By this stage the problem of communication between the dentist and the parents has been resolved. In addition to that, the waste of time and effort will also be resolved.

The third problem can be solved by the suggested system that is built by the blockchain technology. By its unique characteristics, blockchain can provide this protection to the private personal information as it will be explained in the next chapters.

1.3 PREVIOUS STUDIES

In recent years, Blockchain technology applications in health care have been emerged with a great focus on its benefits in: improving the management of Electronic Healthcare Records EHR, enhancing the process of insurance claim, accelerating the biomedical researches, and providing advanced healthcare ledgers [2]. In this literature review, we will show a review of blockchain applications in health care according to the mentioned benefits.

In category of improving management of HER, Yue X et.al[3] suggested an application known as Healthcare Data Gateway (HGD) relied on the technology of blockchain making patients' data able to be owned, controlled and shared facilely and safely ensuring their privacy. A simple unified Indicator-Centric Schema ICS is used for that purpose. While Paul Snow et al[4] described the way by which Factom creates a distributed, independent protocol to isolate the Bitcoin's blockchain from the Bitcoin's cryptocurrency with a goal of making Bitcoin’s blockchain applicable in a wide range of uses providing a distributed, immutable ledger. Another well-known example is the Guardtime[5] which is an Estonian company that can provide a system of blockchain for protection of one million healthcare records and providing security and improvement to EHR management.

For enhancing the health insurance claim process, Yip K.[6] Seeked to find the way that blockchain technology would effectively enable or help Alternative Payment Models (APMs) by essential understanding of encryption.

While Attili .S et al [7] claimed that Blockchain technology is able to give support for a modern generation of transactional Apps and regulated business operations by creating the trust, transparency, along with accountability which are fundamental to new commerce.

Several authors also suggested accelerating clinical researches with assistance of blockchain including: Azaria A. et al [8] proposed MedRec: decentralized system for record management to deal with EHRs by using blockchain technology. It can manage authentication, privacy, accountability, Data sharing, and dealing with sensitive information. With incentivizing the medical concerned (the authorities of public health, the researchers, etc.) in order to take part in the blockchain mining for providing them with numerous anonymized data. So MedRec could enable the development of data-economics, providing huge data to enhance researchers along with involving patients and providers in the option of metadata release.

In addition to that, many studies proposed using blockchain technology as a ledger for various types of health care–related data storage. Alevtina Dubovitskaya et al. had suggested, in his paper [9], a framework in order to manage and share the EHR data for patients with cancer. By cooperation with the hospital of Stony Brook University, the framework was applied in a prototype which guarantees the security, privacy, availability, in addition to a precise EHRs access control. The suggested framework can minimize the required time for EHRs sharing improvement, enhance decisions of healthcare, and provide a cost reduction.

From the benefits and applications that are mentioned above, we conclude that health care has become one of significant fast growing application areas in blockchain technology. As well as, the blockchain technology has been introduced to become the underlying infrastructures of the Health Information Exchange (HIE) and act as fundamentals for the health transactions among the healthcare providers, patients, payers, and the relevant parties.

1.4 THESIS OUTLINE

- Chapter one: Introduction, Research Problem, Suggested Solutions, and Previous Studies.
- Chapter two: Technical Back Ground
- Chapter three: Blockchain Technology in Healthcare
- Chapter four: Methodology and Implementation
- Chapter five: Results
- Chapter six: Conclusion and Future Advice

2. TECHNICAL BACKGROUND

2.1 OVERVIEW

Blockchain, as it is defined by NIST, internal report 8202 [9], is a digital ledger that temper evident and resistant implemented and applied in a distributed manner with no central authority, Oxford dictionary has defined the blockchain technology as “A system in which a record of transactions made in bitcoin or another cryptocurrency are maintained across several computers that are linked in a peer-to-peer network” which limited the scope of blockchain within the cryptocurrency[10] . Another definition by Z. Zheng et al. described this technology as "a public ledger, in which all committed transactions are stored in a chain of blocks. This chain continuously grows when new blocks are appended" [11]. While Vitalik Buterin, the Ethereum's founder gave a more comprehensive definition when defines the blockchain as "a magic computer that anyone can upload programs to and leave the programs to self-execute, where the current and all previous states of every program are always publicly visible, and which carries a very strong crypto-economically secured guarantee that programs running on the chain will continue to execute in exactly the way that the blockchain protocol specifies" [12] . ISO/TC 307 technical committees regarded blockchain as "a shared, immutable ledger that can record transactions across different industries, thus enhancing transparency and reducing transaction costs. It is a digital platform that records and verifies transactions in a transparent and secure way, removing the need for middlemen and increasing trust through its highly transparent nature" [13] . Recently, Georgios Dimitropoulos have defined and described the blockchain technology as "a relatively new technology, which relies on previous innovations, primarily Distributed Ledger Technology (DLT) and cryptography. A blockchain is a digital database, which takes shape as a blocks sequence in the form of chains. Drawing on DLT, the ledger is not centrally managed, but rather “distributed,” shared among all participants of the network. In addition, the ledger transactions records between parties in a permanent and secure manner by the means of cryptography. DLT has made possible the connection of blocks of information as an online distributed database. Cryptography is used in blockchain for linking the blocks in a view to make it difficult for exchanging the transaction data" [14].From the definitions above we can conclude that the blockchain technology is "ore technology" and it has been applicable in different fields and sectors. Actually, Security of sensitive data is the major and important issue in different fields, mainly the health care system which requires the privacy preserving technology to protect the patients' privacy and their personal data. Also in financial operations the privacy protection and security are the top requirements of any technology to be used in this field. Which gives blockchain technology a unique position at these fields.

Essentially, the core idea for using the systems of the decentralization basically provides a distributed system that is fault tolerant, where the authorities are distributed without a need for trust on centralized system. That can guarantee important characteristics to the system including transparency, trust, data integrity, etc. In order to achieve interoperability and provide publicly accessible infrastructure, the need for blockchain have become imminent. Blockchain technology enables distributed software infrastructures and the construction of decentralized applications for the untrusted participants.

The blockchain technology is different from most other designs of information systems due to its unique features and properties which are:

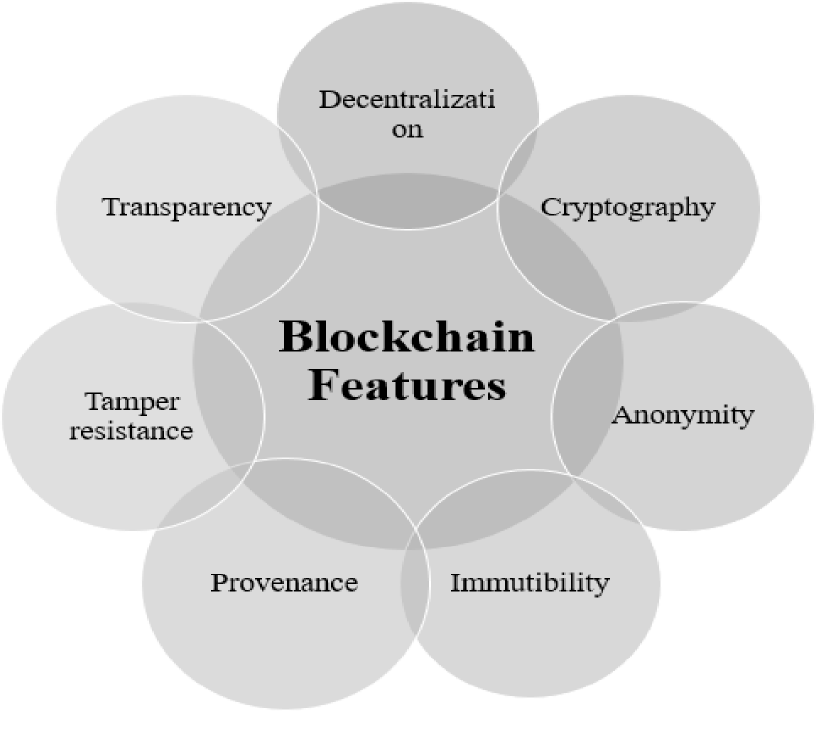


Figure 2.1: Blockchain Features

1. **Decentralization**: It is considered as a basic property of the blockchain that means; each node is found within the blockchain network as distributed and it has its accessibility to the data without central authority, this is implemented by consensus algorithms. The underlying idea of this system usage is for providing the distributed system which is fault-tolerant so that the authority would be distributed with no dependence on centralized systems. Therefore; it guarantees multiple properties such as transparency, trust, and integrity of data. By this property the blockchain technology provides a publicly accessible infrastructure in addition to achieving the interoperability. It ensures constructing the applications depending on the decentralization and building infrastructures of a distributed software related to the huge number of users. There is a main problem facing the centralized systems which is: their tendency to fail at single point and the system will be unable to provide the accurate transparency. The well-known use cases of the decentralized platforms are the Ethereum and Bitcoin [15].
2. **Cryptography**: this property refers to that the blockchain technology can develop protocols that prevent third parties from getting access to the private data and information within the network. Blockchain technology can make the cryptography by multiple different methods including:
    a. Asymmetric cryptography: by the public key that is able to be distributed widely and by the private key that is only identified through its owners, they can be constructed as pairs so that each public key has its consequent private one. This cryptography considered as a fundamental element in blockchain technology and it is mainly the underlying cryptography for wallets and transaction.
    b. Cryptographic hashing: It is directly responsible of production the unique and the essential properties of blockchain which is immutability. Hashing refers to getting input string of random lengths to produce the output string of a constant length. That ensures providing three important features to the network: deterministic, irreversible, collision resistant.
    c. Merkle trees: it is a hash tree that is utilizing the cryptographic hash function to save hash output. It is responsible of the tamper resistance property by detecting any tampering in the transactions within the block because any alteration in the transaction results in that the merkle root would be completely different from the original one [16].

3. **Anonymity**: it is considered as one of the essential characteristics of public type of Blockchains. Anonymity is described as that the subject cannot be identified within a set of subjects [17]. This provides an efficient property to the blockchain for hiding the users' identities and preserve them private [18].
4. **Immutability**: Any record on the blockchain network can be permanently stored, and it will not be altered or deleted unless someone can control on 51% or more of the node in the same time.
5. **Provenance**: every registered transaction's origin can be tracked in the blockchain.
6. **Tampering resistance**: It is the resistance to the intentional tampering of any structure through the costumers and the competencies with getting entity access, such as: a product, a system, or any logic objects and physical objects. Tamper-resistant blockchains lead to the fact that any data transaction within the blockchain is resistant to the tampering in the process of block production [9].
7. **Transparency**: it refers to that the records of the blockchain system which is available for all nodes, it is transparent for data update, so that the blockchain network could be trusted. Which means it cannot be faulted; since it needs high computing ability to change the blockchain network completely.

However, the Bitcoin had emerged in 2008 by Satoshi Nakamoto[19], the concept of distributed data system had early suggested since 1977 by introducing a system for distributed database [20]. Furthermore, the concept of chaining the data blocks immutably by the cryptographic hash function had firstly introduced at Stanford by Ralph Merkle in the 1979 [21], when Merkle explained a method by which information could be linked in a tree structure, nowadays it is called a Merkle hash tree. Then Haber and Stornetta, in 1990 [22] had implemented those concepts to the time stamped records. Those previous attempts, however, cannot involve the whole elements and techniques of blockchain technology.

Dwork and Naor[23], in 1993 proposed proof of computation in an attempt to combat the junk mails. Their concept and the designed underlying proof of work, however, could be introduced in its early simple state in 1978 at Merkle’s Puzzles[24] , Bitcoin considered as the first that used PoW for the mining and achieving consensus.

Many elements of blockchains had been described in the vault system of David Chaum, and in 1982[25], including detailed specifications. Chaum also described the distributed computer system design that could be built, trusted and maintained by participants which are mutually suspicious. It is considered as a system for storing the public records that could protect the user's privacy by its physical security. However, Chaum's work in 1982 went unnoticed, this was because he never published it in any conference or journal.

In order to validate enable the blockchain's private transactions, there are engineers [26],[27] have explored the trusted execution environments implementations, to continue with a method that was fundamentally established in Chaum's vaults.

Recently, what is known as a hybrid blockchains which has been emerged and it combined the defenses against Sybil attacks by Byzantine fault tolerant machine replication. Also, Hyperledger, have emerged which is the project involving Fabric system of private blockchain, and the Ethereum platform, which is a public platform of blockchains. Meanwhile, the researchers focused their efforts to analyze and model the individuals' behavior and mining pools in digital currencies based on the blockchain[28].

2.2 CLASSIFICATION OF BLOCKCHAIN:

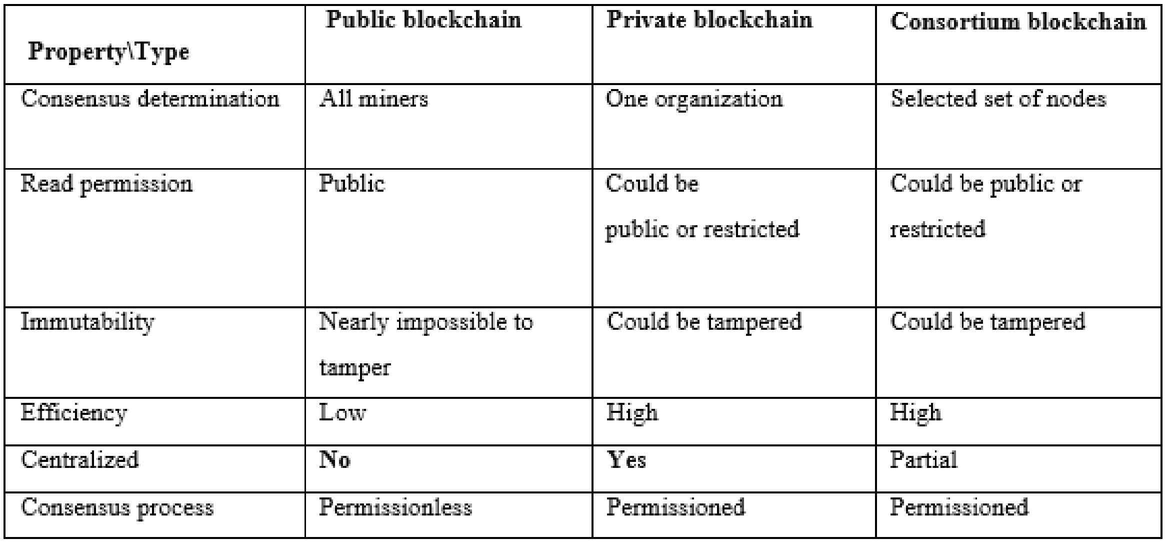
Based on the technological perspective, the blockchain can be classified into three main types: Private, Public, and Consortium blockchains.

1. Private Blockchain: It is a type of blockchain which is built to ensure private and secure data sharing and exchanging among the blockchain network in a single entity or between several entities and the mining has been controlled by a single entity or several participants. This type of blockchain has another name as permissioned blockchains; because strange users will not access the network, until they have received a specific invitation to be validated. This makes the network to tend more towards centralization, while diminishing the essentially blockchain properties of ultimate decentralization, and transparency. The bankchain can be regarded as an implementation of this type of blockchain [29].
2. Public blockchain: it is also named a permission-less blockchain, since it is publicly opened. Everyone is able to access the blockchain network to share and exchange transactions and to contribute in the mining and consensus process of new transaction blocks adding to the

network. It is fully decentralized. Public blockchains often use the PoW protocol, which is a protocol needs entire nodes within blockchain network to have solved the cryptographic puzzles with a force, or Proof of Stake, which is a protocol of block confirmation does not depend on undue computations, for consensus mechanism. The more number of contributors working in this model, the further reduction of the possibility of an attack[30].

3. Consortium blockchain: It is considered a partially decentralized because it works under a group control rather than a single organization control. It is a blockchain where a previously determined nodes have their control on the process of consensus. Instead of giving permission to every user node to take part in the transactions' verification as in public blockchain or allowing only one organization node to be of full control on the consensus process as in private blockchain, some nodes are pre-selected in a consortium blockchain. This allows it to be an ideal candidate for implementation in business collaborations[31].

Table 2.1: Comparison between the three types of blockchain[11]



Property\Type	**Public blockchain**	**Private blockchain**	**Consortium blockchain**
Consensus determination	All miners	One organization	Selected set of nodes
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**	**Yes**	Partial
Consensus process	Permissionless	Permissioned	Permissioned

2.3 BLOCKCHAIN STRUCTURE:

In general, the core structure components of blockchain system can be recognized as the following:

1. Block: it is the fundamental component of the blockchain. Blocks contain the basic data for transactions. They also have block headers that confirm the validity of the block and contain

metadata that characterize the blockchain. The metadata of the blocks includes the following[32]:

a. Version field: it describes the existing version of the blocks.

b. Previous block header hash: references the parent block of previous blocks.

c. The merkle root: includes the transactions cryptographic hash.

d. Nonce and nbits: the number of process repetitions.

2. Chains: they are sequences of blocks in a determined manner.
3. Nodes: they are the essential components of blockchain and regarded as its basic infrastructural units. Nodes represented by any sort of devices such as computers, smart phones or any other servers. On the network of blockchain, all nodes can be connected with each other. The data can be exchanged continuously through the nodes allowing them to be updated. Each full node contains a blockchain copy while the light ones does not have blockchain copy; therefore, the light node should be connected to the full node to be interactive within the network [33].
4. Transactions: they are the core data structure of the blockchain and represented by data, information, records…etc. The transactions are generated by users or by the smart contract for indicating the transferring token between senders and receivers[34].
5. Miners: they are specified nodes which are responsible of performing the process of block verification and creating new blocks. Each miner has its own block that slightly differs from others. The miners' purpose is to make every new block in the network able to be explained as collective decision at the last minute history. Miners helps in creating consensus between the users on the transactions order[32].
6. Consensus: it is sets of rules and protocols that apply the operations of blockchain. These protocols ensures that each node agreement on specific order[32]. According to the decentralization concept, once the block is created and added on the network, the existing nodes would have an option of adding this new block to their ledger copy or ignoring it. The consensus is applied to enable the network majority to have an agreement on single update state so that it can provide a protection and security to the blockchain by preventing dishonesty and tamper attacks. An ideal mechanism of consensus implemented on the blockchain should ensure robust transactions that provide main essential characteristics: the liveness and the Persistence. Persistence property ensures consistency of system responses related to the transactions state. Liveness means that each node or process eventually be able to agree on a

decisions and values. It determines that it can be done during a period to have reached the agreements. The combination of the liveness and the persistence properties guarantees the robustness of the transaction so that only the authentic transaction can be confirmed and approved to be permanent[35].

2.4 CONSENSUS ALGORITHMS:

The workflow basis of the network is by the consensus protocols. It is considering the ways or methods by which the agreement is done for adding new blocks to the blockchain. This approach to achieve the agreement within the network is regarded as important and sophisticated duty. The new transactions could be recorded and added on the network when the added block has been confirmed by the whole nodes within the blockchain. It must consider when the blocks are confirmed, they could not be altered or deleted. The structure of the blockchain is planned to become valid within the trustless and the unreliable networks by tampering individuals. Many different approaches are developed to be consensus algorithm. Those algorithms have been increased in numbers exponentially in harmony with blockchain development[36].

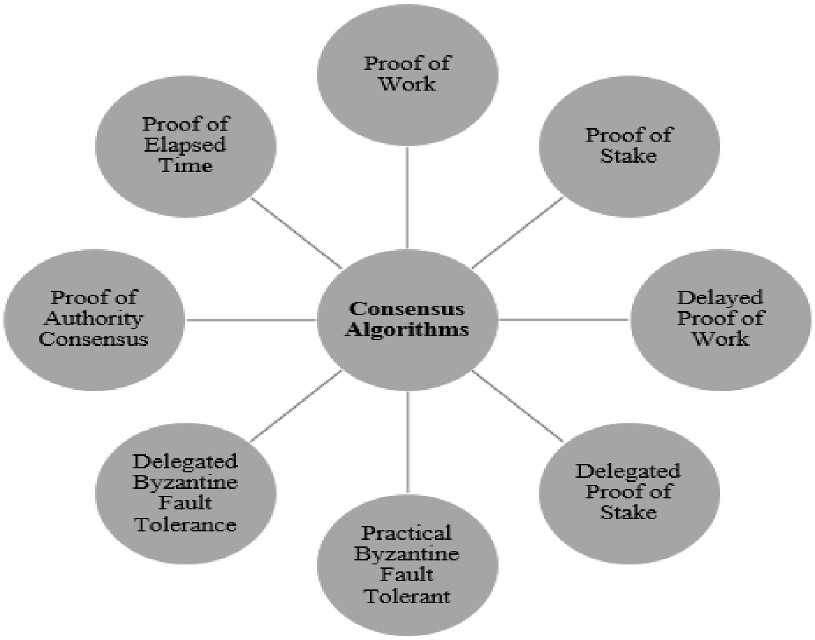


Figure 2.2: Consensus Algorithms

2.4.1 Proof of Work algorithm:

This algorithm has been used by Bitcoin platform. Its main idea is to do the allocation and specialization of the accounting right and reward by the power of the hashing related to the competitive process of the existing nodes in the existing nodes within the network. PoW depends on the workload and considered it as the safeguard. Whenever a new block has been created, this block will be linked with next blocks in such a way resulting in the chain length would be proportional to the quantity of the workload. Every node within the network trusts the longest chain. As a result, if someone starts to tamper within the blockchain network, he will in need of controlling more than 50% of the whole hashing power to confirm its ability of becoming the first user to create the latest block, accordingly mastering the longest chain. Consequently the tampering process gains cannot be greater than its cost. Therefore; PoW could ensure the safety of the blockchain network [37].

On another hand PoW is best fitting for the networks which require scalability. Substantially, the permissionless blockchain networks utilize PoW consensus since they own the participating nodes authenticity, therefore; the network size will obviously increase. This brings some disadvantages and it needs each network node to do investment of large amounts in purchasing tools that are included in the process of mining. Accordingly, this consensus is more subjective to tampering attacks because of its nature of openness[38].

This consensus has the advantage of being difficult to carry out any service denial through the flooding of the blockchain network with bad blocks and the network is open to everyone with a hardware for solving the mathematical problem. However, its disadvantages are that it is computationally intensive by its design, power consuming, hardware arms race and its potential for 51 % attack if obtaining the huge computational power[9].

2.4.2 Proof of Stake algorithm:

PoS protocol has been firstly suggested to control the disadvantage of the increased power consuming of the PoW in the Bitcoin platform. It is a consensus that is mainly utilized by Ethereum. PoS has the property of energy-saving in comparison with PoW protocol. The Miners of the PoS should confirm the owning of the currency quantity. As it has been thought that the users having the large amount of currencies could have a low chance of attacking the blockchain network. The choice that depends on the account balance could be unfair since the one richest user would be linked to be as a dominant within the blockchain network. Therefore, several options suggested regarding the size of the stake to make a decision which one would be able to do forging to the following blocks [39]. The stake can be directly proportional with the opportunity of being a validator of the blocks. The validator of the block will be chosen in a random approach to get the consensus and it is not predetermined. All of the nodes within the network that can produce a valid block can get a reward, while if the produced blocks are not involved within the existing chains, subsequently they would miss some of their stake[29].

In comparison with PoW protocol, the PoS conserves energy, therefore it is considered to be more efficient. On another hand, as the mining cost is very low and may reach nearly zero, the attacks might be a frequent consequence[39].

2.4.3 Delayed Proof of Work algorithm:

Since PoW algorithm consumes large amounts of energy, DPoW consensus protocol had been suggested to benefits from the compute-intensive security of PoW consensus so that it can protect and secure the blockchain networks which use a consensus protocol with Conservative energy consumption. As a result, DPoW consensus algorithm is considered as the hybrid protocol that allow the blockchain networks to protect and secure itself by using the PoW blockchain mining power. Within DPoW consensus, there are some notary nodes, that are chosen by the users, have been responsible for generating new blocks. Each of 64 notary nodes can make the transactions validation and create blocks in round-robin designs without involving the calculation of the mining proof that is compute-intensive and consuming huge energies. In addition, the security of the blockchain network can be confirmed through an approach which includes that latest block hash created on DPoW consensus based network will be joined to PoW based network. The hash of the blocks in the DPoW consensus would be agreed by 52% nodes (33 nodes) prior to be sent to the PoW based network [40]. The protocol of the DPoW has been used by the Komodo platform that utilizes the bitcoin blockchain platform to regain from invalid transactions[41].

DPoW consensus protocol solves the PoW issue of consuming high energy and its prime number which compromises the network security and protection when there is interruption or lost in the communication between the DPoW. Relying on positions and connections of the network, there are some nodes are able to receive block A by the miner and the others can receive block B from other miners. So the nodes that received block A can make a verification of the block validity along with appending that block to their chains and the nodes that received block B would append it on their chain after verifying it. However, when the nodes that are owning block A in their chain would receive block B, the nodes would append the block B in their secondary chain since both of the blocks have the similar parent block. This problem within the blockchain network when the chain separates into two divisions is defined as forking. In a response to solve the forking issue, the rule of the longest chain would be applied.

2.4.4 Delegated Proof of Stake algorithm:

Satoshi Nakamoto, when he was constructing the bitcoin, wanted the entire users to use the CPU for doing the mining. Therefore, all nodes could be matched by hashing power and every node has its own chance of participation in making a decision within their network. According to a technological development and the appreciation to the bitcoin, the designed machines for the mining process have been invented. As a result, the users which have owned large amounts of mining machines will be able to allocate the hashing power and there is a rare chance for the ordinary miner to create a block.

BitShares is considered as a use case for DPoS[42]. In the networks with DPoS, every node has the right to select the witnesses according to the stakes it has. Comprehensively within the blockchain, the higher N witnesses, that would have the accounting right, has gained majority of the votes. Defining of N of witnesses is concluded as at 50 percent or more of stakeholders which are voted should have a belief that there is a sufficient decentralization. The blockchain that are using DPoS algorithm have more efficiency and they are considerably less power consuming than PoW and PoS consensus algorithms [38].

2.4.5 Practical Byzantine fault tolerant algorithm:

The term of Byzantine was used because of the problem is similar to the concept of the Byzantine army when the generals are trying to coordinate their army to be able for attacking Rome by only the use of the messengers in case of one general may become a traitor[43].

PBFT, In 1999, has been published by Castro and Liskov [44], that produced the optimized first PBFT for using practically. PBFT is considered as a replicate protocol that can afford a random fault. Its main concept is represented by that replicas group can constitutes a static group which in turn produces the services. Castro and Liskov used specific optimizations to this essential operational mode in an attempt for improving the latency and throughput of the system[45].

Hyperledger Fabric and Iroha, Hydrachain, Oracle, and BigchainDB are considered as examples of The PBFT. The transaction of PBFT is considered superior in comparison with that of PoW. However, PBFT is in need of an authority for the election of leaders and the supporting nodes, resulting in significant reduction in the decentralization. In a permissioned network owning only determined group of supporting nodes, PBFT carries out the removing process of the problem related to the intensive computing and the protocols based on the capacity. PBFT, as it is utilized by a permissionless blockchain with an open sharing to be a supporting node, the communication will be a problem. The increased number of message transfers in PBFT can result in an increase in the energy consumption because of the overhead of network and communication. In addition, PBFT can experience a Sybil attacks when multiple faulty identities are produced by a single identity despite the presence of authority certificate, as a result it can control the substantial fraction of the permissionless blockchain[40].

2.4.6 Delegated Byzantine Fault Tolerance algorithm:

The DBFT consensus algorithm had been introduced by Da Hong Fei and Erik Zhang for their Neo blockchain in 2017[29]. The significance of DBFT consensus is concluded by its ability to overcome the PBFT consensus drawbacks. It can achieve consensus within the blockchain networks at a little number of peers, as in permissioned networks, in a more quickly manner. In addition to that, the DBFT algorithm is able to reach consensus in permissionless networks just within seconds[46].

The DBFT consensus algorithm has followed the PBFT rules, however, there is no necessity of the whole to participate in the voting for adding a new block. Few nodes are used as representative for other ones and according to multiple rules, they would follow the protocol steps as in PBFT protocol. It is important to mention that DBFT has low chances of delay than PBFT, however l, when there is a limiting of the voters number, it may have threaten the blockchain network decentralization[36].

The DBFT algorithm possesses some significant limitations in comparison with other consensus algorithms. However, the anonymity has played a significant role on permissionless blockchain, the anonymity could not be fulfilled with DBFT, since all of the participants should have their real identities to be declared so that they would be selected as consensus nodes. Furthermore, there will be a centralization within the network due to the fact that the entire network can be under the control of a small part of consensus nodes. Another disadvantage is that there is a speed limit of the creation of new blocks. As previously mentioned, few seconds up to fifteen to twenty seconds are required for every process of block production, while in case of BitShares, only 2 seconds are required by utilizing the DPoS consensus [46].

2.4.7 Proof of Authority Consensus algorithm:

It is consisted of several consensus protocols for private blockchain network that has its prominence due to its high performance as it is compared to the BFT algorithms. The increased performance can be attributed to the fact that the message exchanges are lighter. The PoA consensus has been initially introduced to a core part of the ecosystem in the Ethereum platform to the permissioned blockchains and it has been applied to the Aura and Clique clients.

The Consensus of PoA algorithms is depending on a schema of mining rotation, that approach has been used to make a distribution of the responsibility of blocks between the authorities in a fairly manner. The time can be separated into steps and every step has the authority in order to be a mining leader.

The PoA has two implementations that differently work: both of them have their own first round when the current leader has suggested a new block in a process known as the block proposal. After that Aura needs more rounds that means the acceptance of the block, while Clique does not necessarily require that.

2.4.7.1 Aura:

Aura is considered as an Authority Round PoA algorithm that is applied in Parity (the Rust-based Ethereum client). Any block can be rejected if it has been sent by an authority that is not considered as a current leader. In case of there is no existing transaction available, the empty blocks should be sent, therefore the leader is usually expected to send blocks. However, in case of there is no agreement by the authorities to suggested blocks throughout the acceptance process of the blocks, the vote is initiated for making a decision about the current leader's risk in order to dismiss it. The authority has been able to vote and consider the current leader as harmful and risky if it has suggested more blocks than expected, it has not suggested any block, or it has suggested various blocks for various authorities.

The voting mechanism can be realized by the smart contract concept, and most of votes are needed for effectively dismissing the existing leader within the legitimate set of authorities. When this occurs, the entire blocks that are suggested by that leader will be discarded. In addition, the leader's misbehavior may be resulted from the benign fault such as the asynchrony of the network and the crashing of the software or may be resulted from Byzantine faults as in case of the leader would be subverted and has a harmful behavior.

2.4.7.2 Clique:

It is PoA consensus protocol that is applied by GoLang-based client on the Ethereum, Geth. This algorithm can proceed in a determined period by a specified committed blocks sequence so that special transition blocks will be broadcasted when a new period starts. It can specify the set of authorities that can be used by new authorities requiring to synchronize as the current blockchain snapshot. While Aura consensus is essentially dependent on UNIX time, Clique is computing the current steps with the associated leaders by applying a formulated way which is combining the authorities' amount with that of the block. Along with the current leader, the majority of the authorities are permitted to suggest blocks in every step. Therefore, in order to avoid this, a single Byzantine authority would evoke the blockchain network though imposition of a blocks sheer numbers, every authority within the blockchain has a permission to suggest a new block every (N/2 + 1) blocks and N referred to the trusted nodes. In the same manner, if any of those authorities' act harmfully as in case of suggesting blocks when they have no permissions, they have been voted out. This leads to that the voting against authorities could be casted by each step and if the most of voting has been reached, then the authority would have been eliminated from the legitimate authorities list.

Forks may be happened when more number of authorities can suggest a block throughout steps. However, the limited fork possibility because every authority, which is non-leader, is suggesting a block can delay it randomly, therefore the leader blocks have more tendency to be as the first to be received by the whole authorities. In case of fork to be happened, the protocol of GHOST can be utilized, which is essentially dependent on an approach of block scoring: the forks will be solved as the blocks of the leaders have reached higher scores [47].

2.4.8 Proof of elapsed time:

The concept of PoET consensus algorithm has been proposed by Intel for blockchain construction with the basic idea that is focused on that every node can generate a randomized number to specify the period of time it has to wait prior to be allowed to generate a block. The randomized numbers generation is depending on specific distribution determined by the system in advance. Therefore, when new blocks are submitted to the system, the Software Guard Extension (SGX) would help the node to create the new block and generate a proof of waiting time. This proof has the ability of easily verifying and confirming by the rest of nodes with SGX technology. In comparison with other blockchain consensus protocols, this consensus algorithm has two main advantages: Efficiency since PoET does not need participating nodes to perform any expensive workload of computation prior to creating the new block. And the second advantage is the fairness, since PoET can achieve the goal of one CPU to one vote which was traditionally suggested by Nakamoto in Bitcoin, but was not truly achieved before[48].

PoET consensus uses an approach, to overcome the disadvantage of PoW consensus of requiring a high computational energy to solve complicated problems. This approach suggested by Intel’s SGX technology that provides a mechanism of protecting the selected code and data from being disclosed or modified. In other to control the generation of new blocks, PoET makes each user must wait for some time prior to be allowed to generate a new block. This waiting time requires to follow a distribution of probability which is determined by the consensus. More briefly, there are two measures that are utilized by the PoET to confirm that a user should wait for such a time. Firstly, each user, when generating a block, requires to create a proof for its waiting activity with the help of SGX hardware that is submitted together with the block. Secondly, statistical tests should be performed to ensure whether the waiting times of the user truly follow a specific probability distribution[48].

2.5 BLOCKCHAIN PLATFORMS

2.5.1 Bitcoin:

Bitcoin, which is published by Satoshi Nakamoto in 2009[19] , is generally considered as a virtual, anonymous and decentralized currency that is not controlled by governments or being centralized by a legal entity, also it cannot be exchanged into gold or others commodities[49]. The virtual property of Bitcoin leads to that it generally does not have any physical forms. As a result, the Bitcoin presenting is probably that can be as a saved file on a PC, as an online service, or included in a digital wallet. [49].

Nakamoto suggested a system with a server of peer to peer distributed timestamp that plays the role of a generator for the chronological orders computational proof of the transactions. The electronic coin can be defined as a digital signatures chain. The transactions are defined as they are a hash set of the previous transaction that are signed digitally combined with a next owner of the public key. While the transaction signing is done by the private key, the transaction verification is done by the public key. The public keys are preserved within the wallets, that would be applied on the software, in the hardware, or online[50].

Bitcoin has worked on PoW algorithm to achieve consensus. The Bitcoins issue of taking place by the mining process. To replicate this process, the whole elements would available to the public by a software with an open source through which users can voluntarily make their own PC to be available to the network of Bitcoin for solving a complex mathematical issues. Every computer that can solve the mathematical problems, as a result, is capable of creating blocks and is rewarded to own Bitcoins. The entire number of the created Bitcoins by the mining is limited because the system of Bitcoin is programmed in such a way that results in developing the blocks' the specific time to have less amount of Bitcoins.

In fact, the increase of Bitcoin and the Bitcoins creation are done automatically and this is controlled through the system itself resulting in that there is no need for any central authority intervention on the Bitcoins. The supply and demand are controlling the Bitcoin, without an intervention of any government as in case of printing money, due to the limited Bitcoin number, alongside with the Bitcoin conversion, this is leading to the increase in the volatility of Bitcoins prices. Bitcoin carried out on a permissionless open sourced blockchain network. Therefore, the blockchain of the Bitcoin can be considered as a typical example of open permissionless type of blockchains so that any person have the ability of leaving or joining the network with no need of having any pre-approved requests by any centralized authorities [49].

The transactions are able to transfer the coins in Bitcoin network. These transactions can own more than one input and output. Every output can determine the Bitcoin number that has been sent and the script of the programme. On the other hand, every input can provide a reference for the output of previous transactions and the redeem scripts as a consequent signature that is responsible of the output spending conditions. The transactions do not be able to send more Bitcoin than that are being within the inputs, and there is not any rule to control the way in which the Bitcoin are split among the outputs. In addition, the transaction signers are able to select to contain the fee that is subtracted of the existing Bitcoin to be spend.

The transactions that form the public ledger are serialized in blocks and called the Blockchain. So the blockchain of Bitcoin can be updated at ten minutes' rate and each update corresponds to the recent blocks of the new transactions. The block can be regarded valid, when it is linked with the previous block cryptographically and it is capable of solving a computationally complicated problem. While a fixed amount of Bitcoin has been rewarded for the blocks by the miners, so that the entire transaction fees are paid through the transactions of the block. The chain of the blocks which are responsible of solving the computationally complicated puzzles would be agreed via the network to be similar to the original network. The block position within these chains is considered as it is in its height. As a result, to the lottery puzzle solving process, every time there are more than one block to be possibly found, resulting in the chain tip uncertainty. Therefore, the transactions can be considered as prove when the blocks that contain them will achieved the six blocks depth from the chain tip as minimum.

The limitations of Bitcoin can be felt by the users increasingly and determined as a delayed processing of transaction and increased fees of the transaction. The users pay nearly 0.3 up to 0.7 US dollars for each transaction as the amount transferred independent. The transaction sending costs would be continuously increased according to the increased spaces competition in the Blockchain and the monetary policy of the protocol decreases continuously the resent coin mining that can reward the miners in order to provide the network security.

Increasing the maximum block size can simply be a short term repair to the increasing capacity of Bitcoin, thus permitting more numbers of transactions to be performed within the block. However, there are existed multiple suggestions, none of them can be adopted actively since the community acceptance cannot be reached if blocks sizes do not have been increased dynamically and incrementally or if artificial caps are needed.

Therefore, there are two directions should be focused on to make an improvement in the scalability: firstly, redesigning of the Blockchain protocol underlying for the purpose of supporting larger numbers of transactions each second. Secondly, facilitating off-chain transactions when the transactions have just committed to the Blockchain network if there is a need for an adjudicator [51].

2.5.2 Ethereum:

Ethereum, has been introduced as a new platform for the blockchain in 2013 by Vitalik Buterin. He suggested the intent of Ethereum for merging together and make improvement on scripting of the concepts, the Altcoins protocol along with on-chain Meta protocol, allowing the creation random consensus based applications by the developers. Those consensuses have properties of standardization, ease of scalability, interoperability, and feature-completeness. Ethereum allows every user to write his smart contract and build his applications in a decentralized manner, giving entire users of this platform the opportunity of creating their randomized criteria of the ownership, state transition functions in addition to the transaction formats[52].

On Ethereum platform, the state consists of what is known as Account and every one of those accounts contains an address of twenty bytes. The term of the state transition refers to transferring the information and values so that the account is consisting of 4 parts: The Nonce Counter that is applied for ensuring that each time there is only once transaction is processed, other parts are the Ether balance, the storage, and the Contract Code.

On the platform of Ethereum, the account types can be classified as: Externally Owned Accounts that can be controlled through the private keys and Contract Accounts that can be controlled through the contract code they possess and they have just been be activated by the EOA. Within the Ethereum platform, the transaction refers to a package of signed data that can store a message that the EOA would have sent it. The account creation and the messages calls processes are considered to be the transaction types. The consensus algorithms that are involved in these processes are mainly three types: PoW, PoS, and PoA [54].

Ethereum is a platform based on the transactions state. It starts with a state of creation and increasingly execute the transaction in order to make a modification on it to reach its final state. This state may include several data types. There may also be state changes that are valid or invalid. The state changes that are invalid could be a result of invalid account balance modifying as in case of increased account balance of the receiver without simultaneously reducing in the account balance of the sender. There are multiple transactions have been combined within the block by using the Merkel tree and the blocks can be chained by utilizing of the cryptographic hash. Thus the block considered to be the journal that records the transactions series combining them with the identifiers and the latest blocks[54].

An essential problem of the multiple agent systems and the distributed computing is concluded in achieving the reliability in the whole system in case that there is some faulty processes which usually requires the processes to have agreement multiple data value that is needed throughout the computation. [55]

2.5.3 Hyperledger:

In December 2015, Linux Foundation founded an open-sourced blockchain platform known a Hyperledger. Linux Foundation planned through this platform for creating suitable environment by which the software society developers and the companies of the software can meet and make a coordination for building frameworks based on the blockchain. Therefore, Hyperledger is considered as open source for enterprise projects of blockchain to be developed through all stages of commercialization and development[55]. The main aim of hyperledger is to achieve an improvement of the reliability and enhance these systems performances[54]. The frameworks of the Hyperledger can be divided into five types according to the hyperledger project of the blockchain, which are:

2.5.3.1 Burrow:

It is the private blockchain node that executes the smart contracts. Burrow hyperledger has been constructed to be executed with smart contracts based applications on the multichain environments. The nodes give services of executing the contracts to many connected blockchain networks that are compatible and able to run multiple domains[55].

Hyperledger Burrow can provide private smart contract interpreters to the modular clients of the blockchain that is partly developed for specifications of EVM. More specifically, it is considered as a private or permissioned machine based on smart contracts. It has firstly developed and suggested by Monax to the hyperledger in 2017.

Hyperledger Burrow can provide a blockchain design that is strongly deterministic and focusing on smart contract concept. The members can have the advantages of the access control layer by using the smart contract system. Additionally, Burrow contains four components: the consensus engine that can maintain the network stack among the nodes and the ordering transaction that is used later by the application engine[56].

2.5.3.2 Hyperledger Fabric

It is a platform designed for creating solutions to the distributed ledger, through its modular architecture, it can deliver high degree of flexibility, confidentiality, scalability, and resiliency in such a way to enable the development of the solutions to be ready for any adaptation by the industry.

Fabric permits the services of the membership and consensus to make plugging. It has the property of hosting the smart contracts by what is known as chaincode that contains the system business rules. Every chaincode can define its persistent entries in the state and the hash chain of the blockchain is computed over the transactions that are executed and the persistent state. Fabric is designed to be accommodated with the existed complexity across the whole economy and to give support for various pluggable components[56],[57].

Fabric can be considered as a blockchain platform that is extensible and responsible of Carry out the distributed applications. It can support multiple types of consensus protocols, therefore, Fabric can be implemented to various trusted models and use cases.

It can run the general programming languages based applications with no need to depend on specific native cryptocurrencies. This is on the opposite case of the majority of blockchain platforms that are performing smart contracts, that either they need their codes to have written in specified languages or they are deepening on cryptocurrencies.

In addition to that, Fabric can use a membership of portable notion for the private model that can be combined to the management of industry standard identity. In order to enhance its property of flexibility, the Fabric platform has a novel architectural and reconstruction approach for the method of blockchain coping with non-determinism, performance attacks, and exhaustion of the resource.

Fabric is considerably capable of creating the channels that are responsible of enabling a set of users to create their independent transactions ledger. This is specially of great importance for the networks in which there is some users might compete and do not want every transaction to be known to every user within the network, such in case of a special price offers for some but not all users[56].

2.5.3.3 Hyperledger Indy

It is considered as a type of the distributed Hyperledger. Indy gives libraries, tools, other components that are reusable in order to create and use separate digital identities that are founded on blockchain networks or its distributed ledgers.

Those digital identities have the property of interoperability through applications, administrational domains, and other organization silo. This is involving competitors, friends, and also oppositionists which are dependent on a shared trust source [56].

The consensus protocol that is used is RBFT which do not have enough efficiency when there is an encountered fault. The recent BFTs are dependent on the primary for ordering the requests replica, therefore, if the primary replica has been a malicious node, it would possibly result in requests delays of the transaction that is causing various issues. The new approach of BFTs that is called redundant BFT has been inspired by plentum BFT. This approach carries out various plentum protocol instances in order to prove that the system is being fault tolerant. Through performing that it will be easy to detect the malicious node[54].

The main important properties of Indy are: Firstly, self-sovereignty that enables it to store the artifacts of the identity on the ledger with ownership has be distributed and those forms of artifacts may involve the proof of existence, the public key, or the cryptographic accumulator that are responsible of revocation, therefore, no user except the true owner will be able to change or remove an identity. Secondly, privacy preservation since each identity owner will be able to operate without creating a breadcrumb or correlation risk. Finally, Verifiable claim that can be similar to familiar credentials such as passports, birth certificates, driving license, etc., however, these may be allocated and converted powerfully through the use of what is known as zeroknowledge proofs that ensure the selective disclosure of the only required data by the specific contexts.

The advantages of Indy hyperledger can be concluded as: combining of self-sovereignty, verifiable claims, and the privacy that is highly powerful and resulting in many potential advantages. The bulk troves of sensitive information may be vanished or became useless and the hacking costs has been able to be transformed, due to the decrease in the PII that are related to every partner in the business. The competing demands that are responsible of the privacy preserving in addition to the meeting regulators are considered to be satisfying. The users and the organizations can take the advantage from the highly rich interactions and the secure system. Also the identity ecosystems can get the free markets dynamism and innovation.

However the advancing cryptography, API of Indy Hyperledger is easy and straightforward since the API contains nearly fifty callable functions along with idiomatic wrappers related to different programming languages[56].

2.5.3.4 Sawtooth

Sawtooth has been designed with a modular transaction processing platform and a modular consensus interface for accommodation to the different threat models, smart contract languages, and deployment scenarios. Sawtooth has the conventional architecture of blockchain and a chain of transaction blocks that is linked cryptographically[58].

In sawtooth hype, even during a new business case branching, specific distributed ledger characteristics should be preserved. The distributed hyperledger, in the enterprise deployment, should not end into records of replicated database.

Users require the feature of autonomy and they are able to manage their nodes. That interaction among the participants who are dealing with their own nodes, can provide the integrity to the blockchain network.

To achieve that integrity, blockchain network have to reach three main requirements: providing the security against the harmful users within the network, management of relatively large population of users, and ensuring the dynamic population management. Different consensus protocols that are conventionally used for replicated databases that cannot be compatible with that network. Therefore, Sawtooth used PoET for achieving truly decentralized blockchain. So that the applications are administratively and physically distributed on the position of large amounts of participants within the consensus process. PoET can also provide the security against harmful participants and it has a design that facilitates the running of the nodes arrival and departure within wide networks. Additionally, users are being able to start with a constrained consensus and turn after that into PoET consensus that provide the dynamic, secure and scalable properties that are required by a production networks[56].

2.5.3.5 Iroha:

It is considered as a framework of the blockchain network that has a simple and easy design in order to incorporate within the projects of infrastructure where there is a need for a technology of a distributed ledger.

Iroha has joined the Sawtooth and Fabric hyperledgers to become the third platform framework of Hyperledger of a distributed ledger in 2016. It was initially created and developed by Soramitsu in Japan, then it has suggested for the project of Hyperledger. The main properties of the Iroha are summarized in that it has an easy structure, a domain-driven C++ modern design, it has enhancement to the application development of the mobile, and having new BFT protocol based on the chain that is known as Sumeragi which is strongly inspired by the consensus protocol of B-Chain.

Iroha has had a various approach that differs from Sawtooth and Fabric frameworks by giving characteristics that can provide facilities in creation of end users applications[56].

Fintech, the relatively new technology, have the ability of disruption the current financial services sector through the provision of an effective platform framework for those services. Iroha project has been created and founded by Linux Foundation to be one of the Hyperledger projects. This system has been developed to participate the data among the non-trusted parties which can use a permissioned blockchain network. The important unique features that are related to IROHA and differentiated it from other frameworks are summarized by that the user has no right of storing the whole history of data and the users will only be capable of querying the data in case they have authentication and the permission.

Iroha Hyperledger has the main goal of producing C++ libraries for the Hyperledger project. The provided libraries are based on the Summeragi consensus, API server, SHA-3 hashing, Javascript libraries, etc. [54].

2.5.4 Multichain:

It is considered as a technology of an open source feature that allows the implementation of the private blockchain and provides transactions handling with reduced costs. MultiChain mainly facilitates specific private Blockchain establishing by the users. It provides solutions for the issues associated with privacy, openness, and mining, by the integrated user permissions management[59].

This platform has a special design to communicate on the private blockchains. To overcome the deployment problem of the network, it would provide the portable package with the required privacy preserving and controlling. It enables the network to have deployed and configured by admins instead of the developer. Since the MultiChain platform has compatibility with Bitcoin, so it would be able to import the held assets of Bitcoin to MultiChain[60] .

The core aim of Multichain is to make sure that the activity of blockchain can be only visible to selected users, introduce the control over which transactions have a permission, and enable mining to occur in a secure manner without proof of work and its related costs[61].

In the MultiChain platform, the privileges can be revoked and rewarded by applying the network transactions that contain specific metadata. All privileges are automatically received by the miner of the genesis block, and the privileges contain the administrator rights to be able to manage the other users' privileges. The administrator can reward the privileges for the participants whose outputs include addresses of these participants and the metadata that denotes the privileges conferred. Once the mining privileges and the administration of other participants have been charged, additional constraints are introduced in order to enable a small number of the administrators to make a difference and vote, then those votes will have recorded in an independent transaction by each administrator, when there is a sufficient consensus has been applied that is reached by the change. The earlier few blocks in the chain by which an administrator can bypass the process of voting. The next MultiChain versions can suggest super administrators who are able to revoke and assign the privileges. Because the privileges modifications are included within the transactions metadata, this results in propagation to the entire network nodes for creating a consensus considering the state of current play. Therefore, due to network decentralization, various nodes can deliver the transactions permissions at separated periods, prior or following to rest of the transactions. In case of the payment transaction validation depends on the change of the privilege that was broadcasted, the difference can be considered as critical, with some of the nodes accept the payment and the other nodes reject it.

Those variations can be fixed when the transaction has been proved on the network, repairing the ultimate ordering. Each node can follow the rule of the replaying the transactions in blockchain order, therefore every transaction included in the block should be validated in accordance of the user permissions state that is directly preceding it. In case of the block transaction does not have the permission, regarding this rule, the whole block will be considered as invalid and the valid block miners should be at the allowed list when they performed the entire privilege change that is determined within the transaction of that block[61].

2.5.5 Litecoin:

Litecoin has been founded in 2011. It is competitive platform which leads to the Bitcoin designing with the main designing aim of Litecoin is to provide a quick process of small value transactions. As stated by the litecoin founder, Charles Lee, this platform has regarded to be a silver in comparison with bitcoin platform which was considered as a gold. The idea of Litecoin is to create a cryptocurrency which can process the payments more quickly than Bitcoin. The variation between Litecoin and Bitcoin is that for mining of Bitcoin fast computing power and heavy processing is required, while the mining of Litecoin can be done by an ordinary desktop computer with lesser processing power compared to Bitcoin. Also there are About 84 million Litecoins in circulation when compared with 21 million Bitcoins. In addition, the processing time of Litecoin transaction is about 2.5 min in a comparison with the Bitcoin processing time which is about 10 minutes [62].

However, Litecoin and Bitcoin have similarities at technological level, Litecoin uses different hashing algorithm than Bitcoin known as Scrypt. Litecoin miners are granted with 25 new coins per block and the amount gets halved approximately every four years. Some updates as lightening network have first been applied to Litecoin and then used by Bitcoin. Lightening network basically means that small transactions can be dealt outside the blockchain. This is making the payments to be faster and transaction fees to be low[63].

2.6 SMART CONTRACTS:

In 1994, Nick Szabo, has introduced the smart contract as a protocol of a transaction that is computerized with main aims of their designing to provide the common contractual condition with the satisfaction, decrease accidental and malicious exceptions, and to reduce the trusted needs of the intermediaries. In addition, there are related economic aims include: reducing the costs of the enforcement and arbitration along with lowering fraud loss. In his well-known example, Szabo made the smart contracts analogy to the vending machines which is taken in coins through an uncomplicated mechanism, dispensing the product and changes in accordance with the shown prices[64].

Szabo has believed that the smart contracts are effectively functional more than the paper-based contracts due to the clear logic, enforcement and verification of cryptographic algorithms. However, the smart contracts core idea has not get well understood till the introduction of the blockchain technology, when the public and append-only the consensus mechanism and Distributed Ledger Technology (DLT) make implementation of smart contract concept to be in its intended position[65].

The blockchain has two essential operations: reading of blocks content and appending of new blocks. When an adversary would try to modify or remove some blocks, it requires considerably expensive computations, which makes the process nonviable. In blockchain, the currencies can be transferred by any user to others and can take part in the transactions verification process. When the Bitcoin has been developed and grown, the blockchain became popularly known that it is able to use for developing solutions based on the decentralization in various fields. Because of the Bitcoin architecture can support only scripts for the currency transactions confirmation and validation, and these scripts were insufficient to support other applications. Therefore, the protocol of smart contract has been introduced to blockchain.

On the blockchain network, it is unable to the smart contract to be modified or altered. It can be clearly verified, self-enforced, observed, and relied on the blockchain accessing methods, in order to achieve the privacy. In spite of the blockchain's scripts can just allow transactions validation and confirmation, Bitcoin platform has been regarded as the ear implementation of the smart contracts within the network of the blockchain. Later, in 2015, Vitalik Buterin has introduced the platform of Ethereum on blockchain focusing on the decentralized payment system along with the language of Turing complete, which enhanced the smart contracts development on a blockchain[66].

There are three main features of the smart contracts:

1- Autonomy: that means after the smart contracts execution and launching, the contract and its initiating agent will not be able to include in other contract.

2- self-sufficient: as the contracts will be self-sufficient by its ability to marshal the resources that can raise the funds through providing the service, and spend it when required.

3- Decentralization: since the smart contracts do not found in the centralized servers, they are regarded as decentralized. On contrast, they can be self-executed and distributed throughout the nodes of the blockchain network.

The smart contracts concept has significant implementation on the blockchain. Since they are regarded as the blockchain activators that constitute the foundation of the programmable systems of Blockchain. Furthermore, these programmable and automation properties of smart contracts enable the encapsulation of the complicated behaviors of nodes in distributed blockchain networks, which facilitate promoting of the blockchain technology applications. Consequently, it is facilitating the process of building various sorts of decentralized autonomous organization programmable and decentralized autonomous society[67].

The entire process of smart contracts consists of four consequent stages as the following steps:

1- Smart contracts creation: Several parties included first in the negotiation on the rights, obligations, and prohibitions of the contracts. After the discussions and negotiations, the agreement can be reached. Consultant and layers can facilitate the initial contractual agreement among the parties. Software engineers, then, transfer the agreement which is written by a natural language into smart contracts that are written by programming language. In the same manner of the software development of the computer, the conversion process of the smart contract is consisting of a suggested design, an implementations and a validation. The smart contracts creation is a linked and connective process that is involved with multiple iterations and negotiations. At the same time, it also includes different parties such as engineers of software, lawyers, and stakeholders.

2- The smart contracts deployment: when the smart contracts validation process has been done, they would be deployed on the blockchains platform. Smart contract that is saved on the network cannot be changed or altered because blockchain characteristics of immutability. The modifications or corrections require a new contract creation. Through the blockchain networks, the multiple parties after the deployment of the smart contracts on blockchains can have access to the contracts. Furthermore, the included parties' digital assets of the smart contracts will be secured by the process of freezing their related wallets. While parties themselves can be recognized through their digital wallets.

3- Smart contract execution: Following to the smart contracts deployment process the contractual clauses would be observed and estimated. When the contractual conditions get reached, the contractual process will be consequently executed. Because the smart contracts are built up of many declarative statements with logic connections. Therefore, when achieving the condition, its statements can be executed. As consequences to that a transaction would be validated and executed by miners within the blockchains. After that the updated states and the committed transactions have been kept within the blockchain network.

4- The completion of the Smart contract: after executing the smart contracts has, the parties' states within the contract show be updated. Consequently, the transactions of the execution process of the smart contracts and their updated states will be saved in blockchains. After that, the digital assets can be transferred among smart contract partners. As a result, the involved parties' digital assets will be available to access and the smart contracts will be eventually accomplished.

It is significant to note that throughout the stages of smart contracts deployment, execution and completion, there will be sequences of transactions is corresponding to every smart contract statement that would be executed and saved on the network. As a result, these three stages require to record data on the blockchain network[68].

When compared with conventional contracts, the smart contracts system has the following advantages:

• Reducing risks: because of the immutability property of blockchain, smart contracts cannot be changed or modified as they have been published. In addition, all of the transactions which are duplicated, stored, and accessed to the blockchain network will be traceable and auditable. Therefore, any harmful behavior as in case of the mitigated financial frauds.

• Cutting down of the service and administration costs: as blockchain network ensures the entire system trusting through the mechanisms of the distributed consensus without a need for a mediators or a central broker. The stored smart contracts in the blockchain network can automatically be triggered in a decentralized manner. As a result, the administration and services costs can be significantly reducing because of the intervention from the third party.

• Improving the efficiency of business process: by smart contracts of the blockchain, the intermediary dependence has been eliminated and as a result it will improve the business process efficiency. Consequently, the turnaround time will be effectively reduced[68].

2.6.1 Smart contracts programming languages:

The selection of smart contracts programming language for writing a smart contract is essentially depending on the platforms of the smart contracts. However, there are generally significant requirements. The most important desirable features of smart contracts programming language that affect the contract and language designs.

1- Reasoning: the language behavior model must allow to determine modality characteristics and facilitate the confirmation process of their fulfillment. The underlying calculus model and system are focusing at this feature.

2- Safety: abstractions of the language have to hold the integrity property. Rigorous semantics provide this feature.

3- Expressivity: fundamentally, the language must be expressive in order to be effectively in harmony with a various range of use cases.

4- Readability: a contract behavior representative language should be intuitive so that it will be simple to inspect and write with[69].

The programming languages that are used for the purpose of smart contracts writing are classified into three main types:

1- Low-level languages: Those languages have been designed for direct execution through the underlying execution environment. The concepts and principles of computational model, formal semantics, logic for reasoning about programs, metering, and typing have been usually introduced on that level. Additionally, smart contracts are stored on the blockchain network, mostly in low-level bytecode that enables imposing the suitability considerations. The low level languages are: BITCOIN-SCRIPT, EVM, and MICHELSON.

2- High-level languages: those languages are with the core idea of making the smart contracts writing process to be easier for developers through readability and to be safer high-level syntactic constructs are enhanced by a system type which can provide machine services abstractions. The safety aspect appears from that point and refers to the ability of those languages to guarantee and ensure the abstractions integrity and abstractions introduced through the programmer by using the language’s definitional facilities. In the safe languages, the abstractions can be used abstractly, on the other hand, in an unsafe language they cannot. Therefore, in order to comprehensively understand the way by which a program may misbehave, it is essential to keep in mind the entire details of the all sorts of low-level languages such as the data structures layout in memory and the order in which the compiler can allocate them. The semantics of both low and high level languages should be considered. The high level languages include: SOLIDITY, FLINT, and LIQUIDITY.

3- Intermediate-level languages: those languages that represent a compromise between a high-level and low-level target languages. Fundamentally, they have been designed in such a way to simplify the program verification or the static analysis, depending on the system type, the computation model, reasoning, semantics, etc. In addition, the intermediate level languages permit making a compilation unification that provides a language that will be compiled for various platforms. SCILLA is an example of such languages[69].

2.7 BLOCKCHAIN LIMITATIONS:

However, the blockchain has great potential and benefits, it has also a few challenges and limitations, which constraint the wide usage of blockchain.

There are three significant limitations that may influence the blockchain implementation in different fields, they include the following points:

1- Scalability: With the recent increased transactions amount, the network of blockchains have been overgrown. Every node within the blockchain network should store the entire transactions to be able to make them validated within the network so that they inspect whether the transaction origin is currently spent. In addition, because of time intervals and the block size original restriction that are used for generation of new blocks, the network will process 7 transactions only per second as in Bitcoin, that are not be able to achieve the requirement for processing a huge numbers of transactions actually. In the same way, when blocks capacities are quite low, there will be a delay of numerous little transactions because of the miners' preference of the transactions with a high transaction cost.

There are many approaches suggested for solving the scalability issue within the blockchain network that have been categorized as two types:

a. Storage optimizing of blockchain: Because the nodes are difficult to operate the entire ledger copy, the Bruce that are suggested novel schemes of cryptocurrency, by which the older records of the transaction can be eliminated by the network. The databases which are known as the account tree is used for holding a balance within the entire non empty addresses. In addition, lightweighted clients can assist in fixing that issue. The VerSum, a novel scheme, was suggested for providing alternative method to allow the lightweighted client to be existed. The

VerSum scheme can provide lightweighted client permission for outsourcing the computations with high costs over a large input. It confirms the result of the computation as correct by comparing the multiple results of the server.

b. Redesigning of the blockchain: as to Bitcoin-NG (the Bitcoin next generation) was suggested by its main concept of is the decoupling of the conventional block into two entries: the microblock and the key block for leader election in order to achieve the transactions storage. This protocol can divide the time into periods. In every one, the miners should hash in order to create the key block. When this key block has been created, the node became as a leader that is capable of generating the microblocks. The Bitcoin-NG is also having extension to the strategy of the longest chain when the microblocks have no weight to carry. By that approach, the blockchain will be redesigned and reconstructed along with the existing tradeoff between the size of the block and the network security would have been addressed[39].

2- Selfish Mining: On the blockchain network, the transactions are usually regarded as immutable when applying the consensus protocol and make a verification of the blocks in the chain. However, most of the attacks can be done after having control on more than fifty percent of the miners within the blockchain network. By that way, the whole writing of the blocks on the chain would subject to hacking, and the incorrect block would have been presented. Additionally, making the control over multiple computing capabilities of the whole usual nodes allow the attackers to review the nearly entire transactions history through the forking and providing a faked history.

On the other hand, in the majority of attacks, it is considered that the attacker who has less than fifty percent of the entire computation power to be still very dangerous. In particular, the selfish mining strategy and its related expansions can cause the attacks. In the selfish mining process, the attacker which is originally a miner, places the blocks that have done the mining on the private branches rather than broadcasts them. After that, these private branches are manifested publicly only when they will be more than that of the public chain. Then, the long private chain would substitute existing public chains[70].

3- Security and privacy: The blockchain security problems, are depending upon fundamental implementation of the software and the hardware securities, in addition to the protocols and messages that are required for it to be effectively functional. However, the consensus protocols that are used, when regarded to be a method for ensuring the trust and fairness of untrusted systems, it can provide targets for the probable attack. Furthermore, with all of the public transactions, the potential leakage of the privacy is increased. As a result of the immutability feature of the blockchain, any of the illegal blocks can be persistent for the lifetime on the blockchain network.

Also the customers' trust is regarded as a driving force for the growth blockchain technology especially at the financial and banking level. The transactions within the blockchain network that are available to the publicly raise the privacy issue of the systems based on the blockchain. Therefore, banks and financial institutions are required to be implementing at private blockchain. The same issue is implemented where the privacy is essentially required as in the medical records when the privacy preservation is of great significance[71],[70].

4- Lack of skill set: Blockchain technology is considered as an evolving and tough technology to be understood. Also there is a difficulty in understanding the concepts of blockchain to provide support for that. This requires highly skilled and efficient personnel for handling the complexities [71].

3. THE BLOCKCHAIN TECHNOLOGY IN HEALTHCARE:

Blockchain technology has its popularity after publication of the white paper of Satoshi Nakamoto in 2008[19]. After that it has introduced in different domains beyond its uses in financial and banking fields. As the blockchain technology is considered as a privacy preserving network that enables it to be introduced to the healthcare fields with a significant improvement making at the different levels. Healthcare is regarded as an essential domain at the information technology level because it has potentially evolved at the electronic health records, the pharmaceutical supply chain, remote patient monitoring, the biomedical researches, the healthcare insurance claims, and the population health management.

The huge amount of healthcare data records of those sources lead to quality problems at the processes of analysis, diagnosis and risk prediction because of increased numbers of the cybercrime attacks. Healthcare records have significant impacts at both patients and healthcare provider institutions levels because patient data sharing amongst multiple healthcare providers by EHR can enhance the accuracy of diagnosis and analysis. However, the health data storage can become the single failure point by being aimed by the attacks that leads to denial of services or ransomware attacks.

From that point of view, data security has its great significance in medical records applications and has its potential role in protection of sensitive medical data. Those medical data involve important patient information, which cannot be shared with the untrusted parties due to the safety problems and the privacy issues along with the possibility of misusing the patient’s data. The sensitive data includes the patient information of its medical records that comprise his medical history and the time-specific information of the healthcare services that are provided. The emerging technology of blockchain based smart contracts is promising to gives the solutions that support the security of patient information and data while these data being accessed and shared through multiple institutions. Sharing of the healthcare data depends on blockchain technology to eliminate the restrictions that cause a separation of independent healthcare providers and can make healthcare data universally shareable. Therefore, the integration of blockchain with the healthcare system can effectively contribute to human health and improve the healthcare services qualities[72].

The blockchain technology has unique characteristics that can effectively be applicable at the healthcare field with significant improvements at the services providing and privacy protection. Since the blockchain is decentralized, this make the blockchain is the backbone of the healthcare management systems where the stakeholders at the healthcare system can get access control of the same data with no dependence of any central authority role on healthcare data. The feature of immutability in the blockchain has an essential role on data protection. As the stored data on the network could not be tampered, this results improving the security of the stored data on the network. In addition to that, the blockchain, through using cryptographic keys, can provide protection to the identity and privacy of the stored data. The patient data ownership and controlled data usage can be achieved in blockchain technology by using cryptographic protocols that ensure prevention and detection of misusing patient data by other stakeholders. The stored data on the blockchain network is replicated at several nodes which can ensure the data availability and the system is guaranteed to be resilient and robust against any security attack, data loss, or tempering attempts. Furthermore, the blockchain openness and transparency create a trustful environment for distributed healthcare applications which facilitate the healthcare stakeholders' acceptance process. The verification of stored healthcare records on the network is a requirement for the healthcare system, therefore, the blockchain provides verification of the integrity and validity of the stored records on the network[73].

With the growing of blockchain, different organizations and industries are adopting this technology. Healthcare field has represented an important area for application of blockchain where there are multiple use cases have successfully been implemented.

As the platforms of blockchain technology have been introduced with features of decentralization, transparency, and authentication that can provide a consensus driven method in order to facilitate the multiple entities interaction by using a shared ledger, the technology of blockchain has the power of revolutionizing the system of healthcare. It is improving the data exchanging and the transparency among clinical trials and research systems through providing the doctors, the patients, the researchers, along with other professionals in the healthcare system with an organized method to control the process of exchanging the sensitive and permissioned data. The healthcare organizations that are included within the blockchain consortium have the ability of sharing and exchanging the information in the system. Blockchain technology can provide the healthcare organizations with significant opportunities to deliver effective diagnoses and consecutively treatments by increased sharing of data, and efficient safe clinical trials by tracking process of the research method [74].

3.1 THE NEED FOR BLOCKCHAIN IN HEALTHCARE:

Healthcare system is considered as a fertile field where the blockchain has great potential role. The focus in healthcare system is mainly in the data management that takes the advantages of the potential contenting various systems and increases the healthcare records accuracy. The Blockchain technology has been effectively implemented in different areas of healthcare as in supporting the medication prescribing, pregnancy, management of supply chains, and any management of sensitive data. Blockchain supports data sharing, access control, and audit trail management of medical records. In addition, there are multiple healthcare fields that can benefit from the introduction of blockchain including provider credential, the medical contracting, billing, clinical trials, exchanging the medical record, and the anti-counterfeiting medications. By blockchain the healthcare services have been transformed in an attempt to provide a patient centric approach. In the blockchain based healthcare systems, there will be able to provide enhanced reliability and security of patients’ data due to the patients' ability of controlling their healthcare records. These systems can also facilitate the consolidation of patients’ data and enable the medical records exchange across multiple institutions. The storage process of patients' healthcare data has great importance in fields of healthcare since the information sensitivity is considered as a desirable target of cyber-attack. Therefore, the security of all sensitive data becomes challenging. Additionally, exchanging and accessing control to the patients’ medical records can be a use case which has benefited of blockchain. As the technology of blockchain is considered as resistant to the attacks and failures by its different access control methods, it can provide secured healthcare data frameworks [75].

3.2 PLANNING THE APPROPRIATE BLOCKCHAIN FOR THE HEALTHCARE USE:

As the medical information is considered as a personal information, the appropriate blockchain type can be used is the private (permissioned) blockchain. According to the proposed decision model by Würst and Gervais [76], the blockchain technology can makes sense in when there are various parties which cannot trust each other and they require interactions and data exchanging, but they would not have a desire to include a TTP. The proposed model introduces important aspects which are considerably necessary for deciding if a specific scenario is requiring the blockchain to be used. Related to the storing issues, the factors that are required to be considered are:

1. The existed need for storing data.
2. The multiple write access
3. The availability of TTP and its online status.

At the beginning, we should determine the need for storing data. Subsequently, it is required to specify if there will be a need to write access of different involved parties. Therefore, if there is only one writer, then there is not any need for blockchain and the alternative solutions of traditional databases will be regarded. It should also observe that if the TTPs are available, they are online, and they have full trust, so the need for blockchain will be diminished.

The Würst and Gervais suggested decision model can effectively help to determine the need for specific type of blockchain that should be used as: firstly, if the TTP is always online, it would be responsible of write operations and it functions as a verifier for the state transitions. Secondly, if TTP is in offline state, it will be functioned as a permissioned blockchain certified authority, where the whole entities within the system are identified. Thirdly, if there is no trust between the writers, using the permissioned type of blockchain makes sense. Finally, if the writers are not fixed and they are unknown to the participants, as in case of many cryptocurrencies including Bitcoin, a permissionless blockchain is the appropriate solution[76].

The infrastructure of the medical data is largely depending on the third parties and in multiple cases those TTP will not be fully trusted. Therefore, the blockchain that depends on a consensus without a centralized authority, can be considered as a possible solution for this issue[75].

3.3 HEALTHCARE CHALLENGES AND THE BLOCKCHAIN PROPOSED SOLUTIONS:

The healthcare system is considered as a complex ecosystem of different stakeholders and complicated interactions that are sensitive. This results in challenges of data privacy and security with operative inefficiency. However, the trusted access and the ownership of the medical and administrative data is essentially critical, therefore the process should be as simple as possible with effective cost reduction.

In the field of healthcare, new approaches are aiming to introduce the distributed ledger and decentralization of blockchain to be the solution for the critical issues of security, record universality, interoperability, etc.

By the blockchain technology, the network gives the users permission to exchange valuable data throughout a distributed ledger that users possess and its contents are in synchronization. The accountability and the cost efficiency are derived by the following supportive concepts [7]:

1. The cryptographic concept which guarantees ledger contents integrity without violating HIPAA requirements that are caused a limitation of the healthcare record utilization and a delay in its real time use.
2. The consensus concept by which the most of chain nodes can verify the validity of transactions.
3. The smart contract concept that is responsible of authorization and notarizing each transaction.

The most important healthcare challenges can be concluded with their proposed blockchain solutions as following:

1. Access control: The traditional method to achieve the access control is generally assuming there is a trust among the data owners and the storing entities. Those entities usually represent the servers that have a full trust to define and enforce the access control policies [77]. By using the blockchain technology all data records will be ensured to become secured, without probability of any threatening by bad actors. This is due the blockchain features of decentralization and cryptographic storage of data.
2. The data provenance: which means the historical data recording along with its origins. In the healthcare field the provenance can produce the transparency and auditability of EHR, and obtain the trust within the EHR systems[77]. By using blockchain technology the ownership can be altered only by the owners according to specific algorithm protocol. The origins of data are also traceable since all data sources can be ensured, leading to the increase of the reusability of the verified data. Therefore, blockchain is considered suitable to be used in management of critical healthcare data such as patient consent records[2].

3. The real time availability of patient’s data helps in providing a continuous monitoring for patients with high risk that gives notification to the care teams in order to coordinate the treatment plan in the case of critical situation. Therefore, the blockchain technology provides access to real-time data and guarantees the data continuous availability so that it facilitates the improving process of the healthcare treatment at emergency situations. Accessing to the real-time data provides a resource for early disease diagnosis and subsequently improved medical treatment and early intervention so that the patient’s life is saved[78].
4. Interoperability: it is defined as a coordinated connection of multiple information systems, devices, or applications throughout organized boundaries in order to have accessing, exchanging and cooperative usage of data between the different stakeholders, to achieve the purpose of optimizing the individual and populations health[77]. Since the blockchain technology constitutes an open source software for healthcare, the interoperability challenges that are faced by healthcare data can be eliminated since the blockchain technology provides different services to health care providers (institutions), care givers (doctors) and medical researchers. However, a comprehensive set of data is needed in order to understand the nature of disease, its diagnosis, development medications, and for providing a remote health monitoring to the patients. Blockchain provides shared data ledger from different sources including data from patients wearing sensors, medical images, and mobile application[78].
5. Data sensitivity and protection: Due to the health records are considered as sensitive information such patient identity, treatment plans, medications, and patient details, that sensitive information need have protection efficiently. Therefore, the blockchain technology is a distributed architecture with fault-tolerance and disaster recovery features. So if there has been a single point of failure the data would have been recovered from any server[78].
6. The accuracy of Medical data: the patient’s healthcare information is often distributed among several facilities, healthcare institutions, and companies of insurance. For obtaining an accurate medical history of a patient, the whole parts of the required data for combining and unifying. That could be performed through the storage of all patients’ healthcare data such as prescriptions history, symptoms, payment information, treatment plans, facilities acquired, and other information, on the blockchain network which can always maintain updated, tamper resistant, and traceable records which makes the healthcare providers capable of providing efficient and appropriate strategies of treatment for their patients. As a result, by the blockchain

technology, the healthcare professionals can have a comprehensive view of the medical history of their patients[79].

3.4 BLOCKCHAIN APPLICATIONS IN HEALTHCARE FIELD:

3.4.1 Healthcare Records Management:

The field of healthcare experienced an obvious shift to the EHR systems with their designs are aiming for combining electronic medical records (EMR) and paper-based records. Many systems are used for storing clinical records and laboratory results as multiple components. These systems have been introduced for enhancing the patients’ safety aspects through increasing information access and preventing errors. The main purposes of those systems is to give solutions for the paper based medical records problems and provide effective approaches for transforming the healthcare sector status[80].

Electronic medical record management systems have been used effectively for sharing patients' records among various institutions. However, there are still challenges to access distributed patient data by multiple EMRs due to the regional limitation of the existing EHRs or their relation to an affiliated medical institution[81]. Depending on the published report of the Office of the National Coordinator (ONC) of the Health Information Technology[82] there is an obstruction to access the patient records is a difficulty in finding the addresses of the providers. There are many projects provide approaches for overcoming that issues[81].

Gaby G. Dagher et al[83]. Proposed a framework based on the blockchain to provide a secure, interoperable, and effective access of healthcare records by the patients, providers, and third parties, in addition to the preservation of the patients’ privacy when dealing with their sensitive information. The framework, Ancile, had used an Ethereum based blockchain along with the smart contracts to enhance the data security, and employ a technique of cryptography to make the system more secure. Ancile can have interaction with needs of the providers, the patients, and the third parties and for understanding the way by which the framework can address prolonged healthcare concerns of privacy and security.

Tareq Ahram et al 14 introduced Healthchain, a healthcare application, which is constructed depending on the blockchain technology utilizing the IBM platform. Their concept can be transferred to multiple industrial fields such as: finance, government, and manufacturing as the features of security, efficiency, and scalability are of great importance. In this research the Blockchain vital role was indicated in transforming digitization process of applications and industries through providing the trusted secure framework, producing an active value chain, and creating a tight integration on other technologies as in case of IoT, cloud, and computing. The HealthChain is considered as one of many examples that can show the transformative capability of Blockchain[84].

Hongyu Li et al, in their paper[85], proposed a Data Preservation System (DPS) based on blockchain technology to manage the healthcare data. In an attempt to build reliable solutions for storage that guarantee the verifiability and primitiveness of the stored data along with the privacy preservation for users, they utilized the blockchain technology framework. Their essential idea is to provide the proof of primitiveness data concept and to utilize data structure and decentralization property of the blockchain for developing DPS. Through the suggested DPS, the participants will be able to keep their important data secure, and verify their data originality if there are suspected tampering attempts. They also utilize a wise strategy for data storage and various cryptographic algorithms for ensuring the privacy of the users such as the adversaries cannot be able to read the plain texts if there is an attempt of steeling the data. The researchers applied a DPS that is based on the Ethereum platform and the evaluation result of its performance can show its efficiency and effectiveness.

Shan Jiang et al. suggested BlocHIE[86], a platform based on blockchain for exchanging the healthcare information. BlocHIE includes two healthcare data types that are considered: the electronic healthcare records and the personal medical data. In their paper, Shan Jiang et al, analyze the various requirements for healthcare data sharing from various sources. Related to their analysis, they architected the BlocHIE of two types of loose coupled Blockchain: PHD Chain of Personal Healthcare Data and EMR Chain of Electronic Medical Records. Within the EMR Chain, they combine the on chain storage and off chain verification techniques in order to improve the privacy and authentic ability. In addition, they propose two algorithms for improving the system and the users' fairness. As a result, to their implementation and evaluation, they indicate the proposed BlocHIE as practical and effective.

FHIRChain 17, proposed by Zhang et al., it includes a blockchain-based App that explains the blockchain potentials to effectively enhance the sharing process of the healthcare data along with maintaining the security of the data source. The FHIR-Chain would also have been extended to deal with interoperability issues of healthcare data, as in case of coordination with other stakeholders and insurance companies in order to provide an easy secure accessing of the patients' medical information.

In their paper, Zhang et al., provide contributions of implementing the blockchains to the clinically data share and exchange as they presented FHIRChain for meeting the ONC requirements through following the HL7 standards of FHIR. Also they proposed the decentralized FHIR Chain based application by utilizing the digital health identity in order to provide authentication of the users in a study case for remote cancer care[87].

While Kai Fan et al suggested MedBlock system that successfully introduces solutions to the issues of the large scale data management and exchange within the EMR systems. The system of MedBlock, based on blockchain technology, provide patients with easy access to their EMRs of different hospitals without causing the last data to be fragmented into variant data-bases. The data exchanging and sharing through the blockchain network can provide the hospitals and the healthcare institutions a comprehensive medical history of the patients prior to consulting. Kai Fan et al suggested effective blockchain based scheme of sharing and privacy preservation that ensures the privacy included within its data through using the integration of the encryption technology and the protocol of access control. That method of making the semi trusted third parties have no data access guarantees that no agencies could have accessed to medical data of the patients. This suggested design can retrieve encrypted data location in a quick manner and improve the system efficiency[88].

3.4.2 Medical researches and clinical trials improvement:

The blockchain technology hold an important implementation in medical researches and education. It has also great effects in the clinical trials by eliminating the data falsification and exclusion of the undesired results of the clinical researches. Blockchain can facilitate the process of giving the permission to the patients so that their information can be used in clinical trials due to the data encoded anonymization. In addition, the blockchain immutability character certifies the integrity of data that is allocated by the network for the clinical studies. The blockchain transparency and publicity features can facilitate the replication of the research from the records based on the blockchain. Those reasons result in that the blockchain technology is expected to make a revolutionary effect in biomedical research area. Additionally, Blockchain has potentially revolutionized the process of peer review for publications of clinical researches according to its properties of decentralization, transparency , and immutability[73].

Benchoufi et al, in their paper[89], constructed a workflow consent by the use of the Blockchain technologies benefiting from its characteristics of transparency and traceability. They designed a concept based on the Proof of Concept consisted of time stamp of every step in the consent collections of the patient by the Blockchain; this results in historicizing and archiving the consent by the cryptographic validity in a secure, transparent, and unfalsified manner.

The technology of blockchain has its potential impact on the applications of Health Profession Education (HPE) as introduced by Funk et al.[90], in their paper, they made an implementation for blockchain using in building a HPE system that id based on competency and values, and it can put credential services with no need for depending on third parties. These possibilities can make the core idea of a HPE system that is built on the blockchain technology promising and by providing an essential infrastructure in order to facilitate the education and training of the health care professionals.

In a similar way, Nugent et al. introduced their paper[91] and they extended their concept by the use of the smart contract, code, and data that can remain at specified addresses on the network, and the execution can become valid in cryptographic manner through the network in order to show the way by which trust in clinical trials will be implemented and the manipulation of the data can be minimized or even diminished. They presented the smart contracts of the blockchain to be able to supply the technological solutions for the manipulation issues, by serving as a trusful administrator and to provide immutable records for trial history. They presented the approach of using the smart contracts on Ethereum platform in order to provide improvement of data transparency for clinical trials.

3.4.3 The Supply Chain and the Pharmaceutical Management:

The supply chain is considered as important implications that utilizes the blockchain technology for the pharmaceutical management, especially at the fields of medications and pharmaceutical industries. Because the substandard and counterfeit medications delivery can result in significant consequences on the patients' health, and this becomes as a common issue to be faced by the pharmaceutical industry. The Blockchain technology can be introduced due to its ability of giving solutions to that issue[92].

The blockchain network can be used for recording the pharmaceuticals movement and for the authentications through the use of supply-chain so that the entire manufacturing items can be determined by their recognizable codes and the blockchain network will have the ability of checking the codes and testing the product authenticity[2]. The blockchain technology as a public ledger can make everyone within the network to be capable of verifying if the number of the identification in a certain medication is authentic or not. This can considered as a huge step in the battle that is faced due to the counterfeit medications availability [93].

As example, Gem Health Network [94] , has been included in this field and it is based on the Blockchain utilizing the Ethereum platform. It provides access of multiple medical providers to the same data. That allowed development of new type of healthcare applications that solves important operational problems and unlocks wasted resources. It also represented an ecosystem of healthcare that is combining the individuals, businesses, and the experts. In addition, it improves patient centralized care along with addressing the operational problems and the efficiency issues. This can be effective in limiting the medical errors resulting from the outdated data; thus preventing the medical problems in their early phase. Also, it permits the involved healthcare professionals to have tracked the correspondence among patients and their physicians that have previously happened. As a consequence, the whole treatment for the patient will be done in a transparently, creating the confidence between all medical stakeholders.

4. METHODOLOGY AND IMPLEMENTATION

4.1 RATIONALE FOR USE BLOCKCHAIN TECHNOLOGY OVER DISTRIBUTED DATABASE MANAGEMENT SYSTEMS

In order to demonstrate the reasons behind introducing the blockchain as a feasible distributed ledger for improving the healthcare applications, we will briefly describe the core benefits and the significant advantages of blockchain when compared with traditionally distributed database management systems (DDBMS)[95].

First of the key benefits of the blockchain is the decentralization. Because of the centralized management of the DDBMSs; therefore, the users are feeling as they operate a centralized database, however the substantial machine could be distributed physically. As the blockchain is regarded as a management system, with the feature of peer to peer and the decentralization property, in which all nodes can work independently according to a predetermined protocols[96],[97],[98].

Therefore, the blockchain technology has become suitable for the healthcare applications where the healthcare stakeholders require collaboration with each other's independently with no control of a central authorities or an intermediary.

The other benefit of the blockchain technology is that its property of the immutable audit trail. DDBMSs give permissions of creating, reading, updating, and deleting functions similar to most of the database systems, whereas within the blockchain, there will be only creating and reading function. Therefore, it is very complicated to alter the recorded data. As a result, blockchain has been appropriate where there is a requirement of immutable ledger for recording the critical data.

The third benefit of blockchains is the provenance of the data. As the digital assets ownership on DDBMS could have been adjusted by the administrator of the system, while in case of using blockchain, the ownership could be altered and modified by the owners themselves only, according to the protocols of cryptography.45 In addition, the assets origins are also traceable since the records and the data could be approved, maximizing the x confirmed transactions and data sources or [99]. This makes the blockchain appropriate for using it in the management of the critical digital assets as in the patients consent records.

The fourth advantage of blockchain is the availability and robustness. However, both of the blockchain and the DDBMs are dependent on the distributed technology and this makes it unable to experience any single failure point, DDBMs is too costly to meet the high-level redundancy of data that blockchain did. This results in every node has an entire history of the recorded data. Blockchain is appropriately used where the data preserving and the records continuous availability are of great importance as in case of electronic health records.

The final advantage of the blockchain technology is its capability of improving the privacy and security by its cryptographic protocols. As in case of Bitcoin, where the blockchain used the 256-bit (SHA- 56) with cryptographic hash functions that defined by the U.S FIPS 186-4, that is published on NIST[100]. Also the SHA-256 can be utilized to create the addresses of the users for improving of both anonymity and privacy so that every user can be introduced by its hash value rather than its real identity. In addition to that, Bitcoin blockchain has used an asymmetric cryptographic algorithm that is called 256-bit Elliptic Curve Digital Signature Algorithm [101], so it can generate and confirm the high level of security of private and public keys to be as a digital signature, and thus ensuring the digital assets' ownership, as in the patient record.

4.2 RATIONALE FOR USE ETHEREUM BLOCKCHAIN PLATFORM

In order to present the rationale for using Ethereum platform, we will make a comparison among several blockchain platforms types by explaining determined criteria regarding the usability, flexibility, and potential performance. A brief explanation of these criteria that is involved in the comparison of the blockchain platforms will be included below. However, it may not be the major concern of the users, if the platform is with an open or closed source, but it would be considered. Although the blockchain platforms that have been involved are with open sources so this will not be included in the comparative criteria for each platform. It is considered that the platforms being open source to be suitable for adoption and innovation, as anyone will be able to download the platform in other to use and modify[102].

The platforms that are included in the comparison are:

1. Ethereum[103]: it is considered as a platform of blockchain that has the Turing complete contract language which permits the developing of the smart contracts. On Ethereum, the smart contract is working on top of the customized built network such as in smart contracts of the

Bitcoin. The ethereum platform, by its facilities of developing the smart contracts, permits sophisticated use cases as in case of insurance and financial exchanges to become executed on the distributed platform [104].

2. IBM Open Blockchain (OBC): IBM,in 2016, made their Open Blockchain (OBC) as an open sourced project to become part of the new project of Linux Foundation [105]. The code of IBM constitutes the core portion of this project and the Hyperledger Fabric. OBC is focusing on how to allow the organizations to deal with the enterprise technology in order to produce applications that can provide solutions for the common business problems in an effective approach. The IBM with OBC has an important goal that is to improve the workflows of the business by using the blockchain platform for trade, issue, management and service assets. Additionally, it suggests for automation of the business processes through the deployment of business laws as the blockchain's smart contracts which will be validated by each stakeholder trustfully. Other applications may involve an asset depository, a manufactured supply chains, and a communication platforms[102] .

3. Intel Sawtooth Lake platform: The experimental contributions of Intel to the project of Hyperledger, which is Sawtooth-Lake, has been planned in a design for being as a distributed ledger platform that is versatile and highly modular[106]. The Sawtooth Lake has similarities with Bitcoin but with two main differences that is concluded by having an extensible transaction types and a unique consensus algorithm.

4. Eris: it is the platform that is targeting the applications of the enterprise. The point of concern is the capabilities that are based on the permissions determined the smart contracts creating and transacting users on the network, validating the transactions [107]. Eris platform is considered as a modular platform that utilities Docker in order to provide permissions for multiple components to be exchanged by the compatible ones. The default components may be Eris's key signing daemon, Ethereum VM, mint-client of talking to Terndermint, Tendermint Socket Protocol of consensus, and the Solidity compiler[108].

5. BlockStream Sidechain Elements: It is an interesting blockchain innovation[109]that introduces the sidechains which are basically standalone blockchains and they have the ability to be desecrated into multiple blockchains. That will permit the assets to be transferred

between two networks of blockchain. Additionally, it provides a modularity degree through using elements. These are considered to be unique characteristics that could be arbitrarily added and combined with the sidechain. The elements are still developing and they include the Confidential Transactions, where the only involved one in the transaction is able to see the transferred amounts [110], and the value of Signature Covers that can invalidate the transaction’s signature as its related input is being spent, to allow for faster validation of the transactions. There are other potential use cases of side-chains including: blockchains creation with advanced privacy features and with advanced smart contracts, or used an applications base for giving control for banks to master their digital currencies giving the control for the companies to master reward schemes of the users[111].

The following is a brief explanation for the criteria used in the comparison[102]:

1. The usability: The first point of comparison is degree of difficulty in learning and using a platform. The indicators of the usability are the number of various methods that will be available to interact with the platforms and platform-specific knowledge levels which are required for developing a platform (if there is a requirement of a platform-specific language for programming smart contracts). Therefore, the platform that requires less specified knowledge is regarded to be favorable.

2. The support and documentation: The significant comparative aspects are the documentation qualities and quantities along with the developer resources of the platform. Those involve the considerations of a platform related designs, tutorials, features, technical details of implementation, and the examples. As the documentation is essential to enable the users to have accessed to the platform. Therefore, the more documentation is regarded as favorable, especially in case of the documentation is aiming on several types of users as application user, developer, and miner.

3. The development: the development history of the platform, from releasing notes or Github commit history of the repository, and the community working sizes to maintain and develop that platform are considered. The longer history general produces platforms that can be developed better, and the bigger communities give indications of future sustainability of the development.

4. The limitations and flexibility: despite the various types blockchain platforms are available and they are aimed to be used for certain tasks, this pays attention to the platforms with general purpose. Flexible platforms are able to be used for multiple purposes. Therefore, they can be more useful, profitable and can inspire innovations. Therefore, any probable restrictions or increasing in the variety of applications, the platform could be used for is observed.

5. Security: at every platform security, the user anonymity and the security of the blockchain itself are concerned. Since the blockchains, in general, is used for transacting some values as monetary, securing those transactions constitutes a significant issue of the users.

6. Currency: the currency that is used by the platform is also considered such in case of real world currency and Bitcoins, and the way by which the users obtain it by mining or buying it with currencies.

7. Scalability: an investigation should be made, when it is possible, to find out how the platform can scale with network sizes and the transactions numbers per second which are required for validation. From what has been observed by the current issues of Bitcoin, the scalability will be essential to each blockchain platform which can adopted.

8. The consensus and the incentive Mechanisms: the most important features of the blockchain's design are the consensus cost (computational power, time, or energy), the consensus difficulty, and whether the difficulty can be altered or modified are considered carefully. If an incentive is placed for encouraging the user to make the consensus decisions would also be considered, however not the entire consensus protocols need incentive mechanisms.

Table 4.1: comparison between platforms

Platforms\criteria	Ethereum platform	Open-Blockchain platform of IBM	Intel Sawtooth Lake platform	Eris platform	Block Stream Sidechain Elements platform
Usability	Yes	Yes	No	No	No
Support and Documentation	Yes	Yes	Yes	Yes	Yes
Development	Yes	Yes	No	Yes	Yes
Limitations and flexibility	Yes	No	Yes	Yes	Yes
Consensus and incentive mechanisms	Yes	Yes	Yes	Yes	Yes
Scalability	Yes	Yes	Yes	Yes	Yes
Currency	Yes	Yes	Yes	Yes	Yes
Security	Yes	Yes	No	No	Yes

4.3 RATIONALE OF USING SOLIDITY PROGRAMMING LANGUAGE

Solidity is considered as a high-level, human-readable codes that can break them down into specified instructions that can be easily understood by machines. The important advantages of using Solidity are[112]:

1. Solidity can provide object-oriented programming features in contracts involving multiple level inheritance characteristics.
2. Solidity developed for Contracts can maintain multiple members of variables for represent and arrangements.

3. Multiple kinds of supporting roles can be carried in Solidity throughout the expedite Application Binary Interface (ABI).

4. The solidity development provides a reliable and secure process for various platforms included in the agreement or the settlement between two members.

5. By using the contracts fundraising, solidity can provide solutions for different issues like the third party expenses and can reduce the data management cost.

6. Solidity poses a similar syntax to C++ and JavaScript that can facilitate the learning of basics of Blockchain developing for ones with respective skills. Furthermore, the same source code of Solidity could be written in C++.

7. Solidity can also be used for programming smart contracts on other networks, rather than Ethereum, such as Monax and its Hyperledger, Tendermint, Burrow Blockchain, Counterparty, and Zeppelin by Digital Currency Group.

8. Solidity can also support multiple type-safe functions by facilitating ABI.

4.4 HOW USERS INTERACT WITH BLOCKCHAIN AND SMART CONTRACTS SYSTEM

In this thesis, there will be three types of users as it is demonstrated in figure 4.1: dentists, patients and school. We will explain in details each type of users and the way by which they can interact with blockchain and smart contracts systems as following:

1. Dentists: the dentist must be medical professional, has a certification, and has a license and authorized to practice dental procedures. The dentist will have the ability to access and change the case sheets of his patients in order to provide a suitable treatment plan and avoid medical mishaps, also he is able to do verification of the patients account identities and he is able to see what other dentists had done to the case sheets of the patient. Therefore, the dentist can control his patient’s safety and identify himself by a secure way on access of patient data within the network in such a way that there is not any unauthorized entity that will try to access them.

2. Patients: the patient must be a private person that seeks dental healthcare service of one of the healthcare professionals. The patient will be able to see what case sheet he has and to identify

himself in a secure way on the access of the personal information within the blockchain network. He can also share information with the dentist and school on the blockchain network.

3. School: its main duty is to observe and supervise the treatment process and it can ensure the commitment of the patients to the treatment plan.

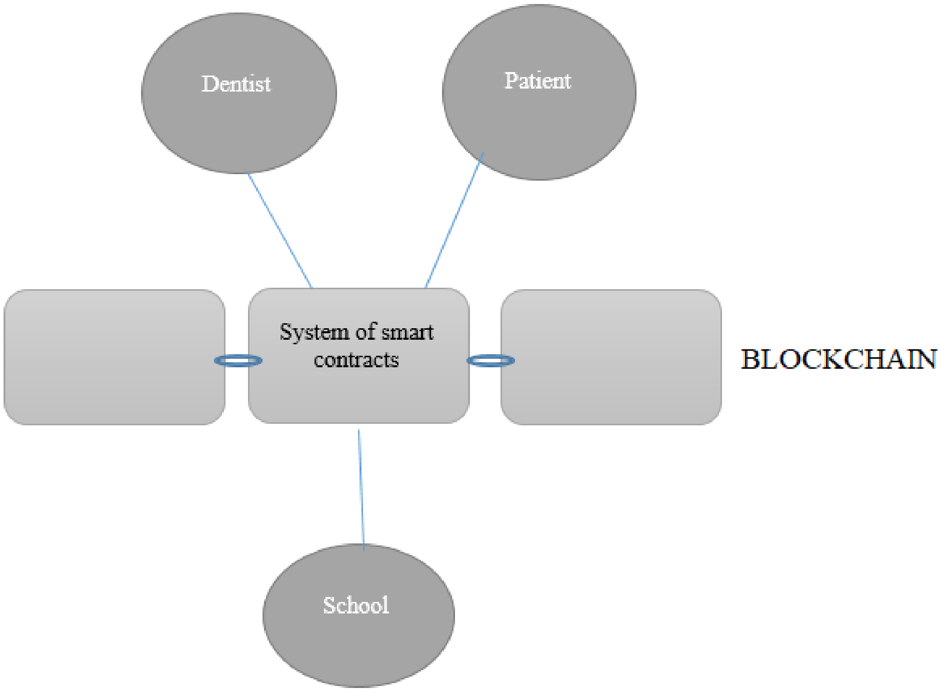


Figure 4.1: show the different type of users in our blockchain

4.5 INFRASTRUCTURE OF THE BLOCKCHAIN AND THE SMART CONTRACTS SYSTEM

The infrastructure of the blockchain technology and the smart contracts system can be summarized in four levels: the top level is the software system level, then it he container level, the component level and final one is the contract level as demonstrated in figure 4.2

As it is related to our work, the "Smart contracts" component will be explained in details.

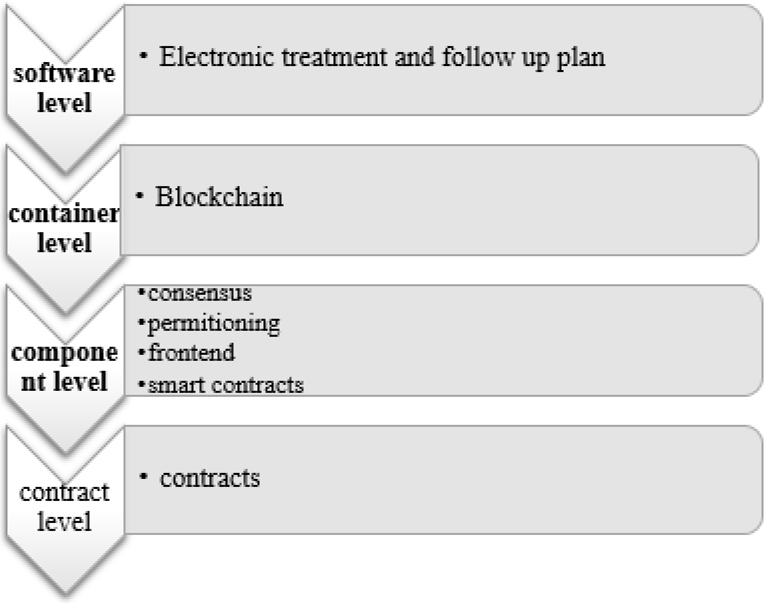


Figure 4.2: infrastructure of blockchain and smart contracts system

4.6 THE SYSTEM DESIGN AND DEVELOPMENT:

The implementation framework is a decentralized application (DApp) which can support a permissioned blockchain by the back end distributed file system (DFS). Ethereum is used for implementing the smart contracts system of the healthcare blockchain network. The fundamental smart contracts constituents are modifiers, events, functions, and state variables that is written by the Solidity, the high level programming language. Remix IDE and a private PoA ethereum blockchain have been built in order to deploy the smart contract. In the process of the smart contract creation, the three stages that are included are: the writing, the compiling, and the deploying, though the utility of the Solidity programming language.

4.7 THE SMART CONTRACTS SYSTEM:

For achieving the better work, a non-trivial DApp, requires more than one smart contract. There is not any method for writing scalable and secure smart contracts back end without distribution of the logic and data upon several contracts. It will be difficult to determine exactly the approach of doing that. Therefore, we are going to classify the smart contracts into types instead considering each contact according to its functions, we are considering the contracts according to what they are.

However, there are various methods for the classification of the contracts, we will use what is known "the five types model". It is an easy model by which the contracts can be categorized into five essential types[113]:

1. The database contracts: Those have just been used for data storage. The unique required logic will be the function which permits the other contracts to update, write, and get the data. These contracts provide a simple method to check the caller permissions regardless the permission type.

2. The contract managing contracts (CMCs): Those contracts has a unique purpose that is only management of other types of contracts. Their essential duty is to track the entire contracts components within the system, facilitate the modular design, and manage the communication among those components of the contract. Separating this functionality from the logic of normal business should consider the good practice, and has several advantages to the system, we will explain them later.

3. The controller contracts: Those contracts are working on the storage of the contracts. By their flexible system, both of the controller and database contracts could be substituted by contracts within the system which are capable of sharing one public API (however this is not usually required). Controller contracts have the ability to be advanced, and they are able to do batching, read and write, or reading from and writing too many databases rather than one system.

4. The utility contracts: Those contracts are capable of performing a specified duty, and they may be called by the other system contracts with no restriction. Utility contract may be the contract which can hash the string by the use of several algorithms and can offer random numbers. They do not usually require large storage spaces, and they possess several or no dependency.

5. The application logic contracts (ALCs): they include an application specified code. In general, in case of the contract uses the controller along with the other contracts for performing application specified duties, it can be considered to be ALC.

In our smart contracts system, we have eleven contracts that we can presented them according to the five models as follows:

1. *ManageContractsEnabled*: is the base class for contracts in our smart contracts system.

2. *InfomationManagerEnabled*: is the base class for contracts that is inherited from *ManageContractsEnabled*, it only allows *InformationManager* contract to call the contracts in our system.

3. *ManageContracts*: they are the contract that can manage all contracts in the smart contracts system and all contract should be connected to them or inherited from *ManageContractsEnabled*.

4. *InformationManager*: this contract is inherited from *ManageContractsEnabled*, it is an application contract that the user can interact with. It is also responsible of all permissions that are given to each user.

5. *ContractProvider*: this can interface to get contracts from *ManageContracts*.

6. *PermissionsDB*: this contract is also inherited from *ManageContractsEnabled*, it is considered as a database contract that is used for storing all permissions.

7. *Permissions*: this is controller contract that is inherited from *InfomationManagerEnabled*.

8. *PatientInformationDB:* it is inherited from *ManageContractsEnabled.* It is a database contract that is used for storing all information of patients such as case sheet and the dentist that is responsible of the case.

9. *Patient:* it is an application contract that is inherited from *InfomationManagerEnabled* contract, it is used for dealing with patients' requests such as retrieving the casesheet or changing the dentist.

10. *Dentist:* it is an application contract that is inherited from *InfomationManagerEnabled* contract, it is used for dealing with dentists' requests such as adding patient or adding casesheet, etc.

11. *School*: it is an application contract that is inherited from *InfomationManagerEnabled* contract. It is used for dealing with the schools' requests. This contract only to approve the casesheet and it

is only called when the school want to ensure if the patient (student) is committed to the follow-up.

4.8 TESTING AND DEPLOYING THE SMART CONTRACTS

In order to test our smart contracts and deploy them, we require private blockchain network. Here, a question is raised, why we require a private blockchain network? The answer of this question can be briefed as the privacy. Since there is no need to keep the data in public blockchain, we want to store the data in private and secure place, also we need to know how to access the blockchain network, who is handling it and who is interacting with it. All these reasons lead to build a private blockchain network.

Now, we want to build a private ethereum blockchain network, there are two consensus algorithms provided by this blockchain: PoW and PoA algorithms. We choose PoA algorithm according to the comparison demonstrated on table 4.2.

Table 4.2: a comparison between PoW and PoA algorithms

Property/Algorithm	PoW	PoA
Speed	Slow	Fair
Node Id	Open	Permissioned
Transaction scalability	Low	High
Power consuming	High	Low
Mechanism	Solve math problem	Reputation and identity
Attack	Vulnerable	Safe

4.8.1 Build a private PoA Ethereum Blockchain Network

To build a private PoA Ethereum Blockchain Network, we need to follow the steps as shown in figure 4.3.

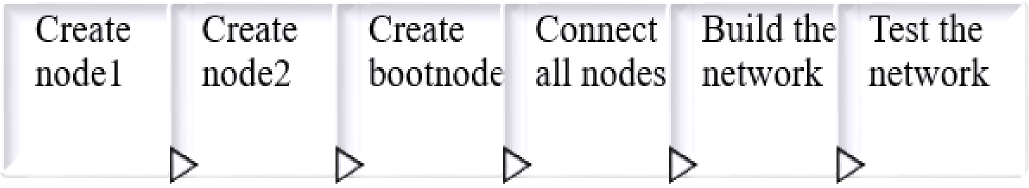


Figure 4.3: steps of building a private PoA Ethereum Blockchain Network

4.8.1.1 Prerequisites to build a private PoA Ethereum Blockchain Network

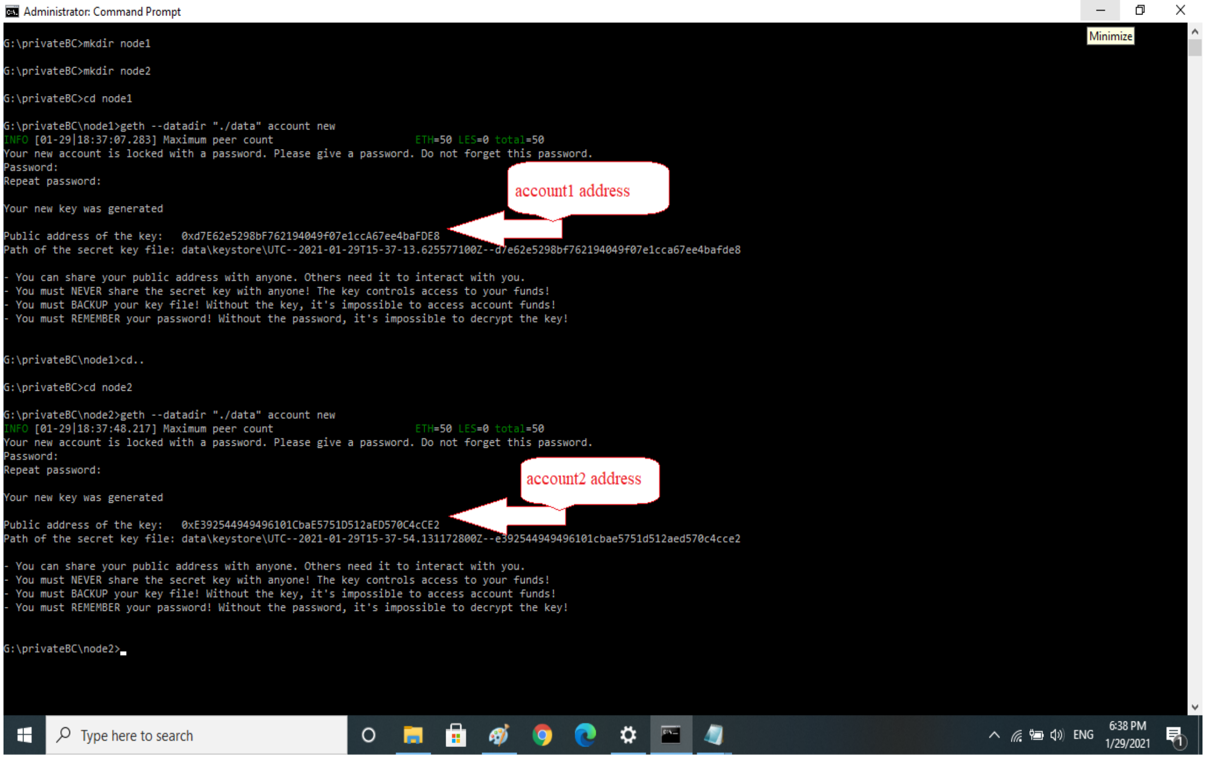
For building a private PoA Ethereum Blockchain Network, there are some prerequisites:

1. Computer with operating system (we will use windows 10)
2. Install Geth (Download it from this link https://geth.ethereum.org/downloads/)
3. Internet connection.

In our computer we will create a folder named it privateBC, and we will also create two folders within this private BC, that are named node1 and node2.

4.8.1.2 Creation of two nodes with ethereum accounts

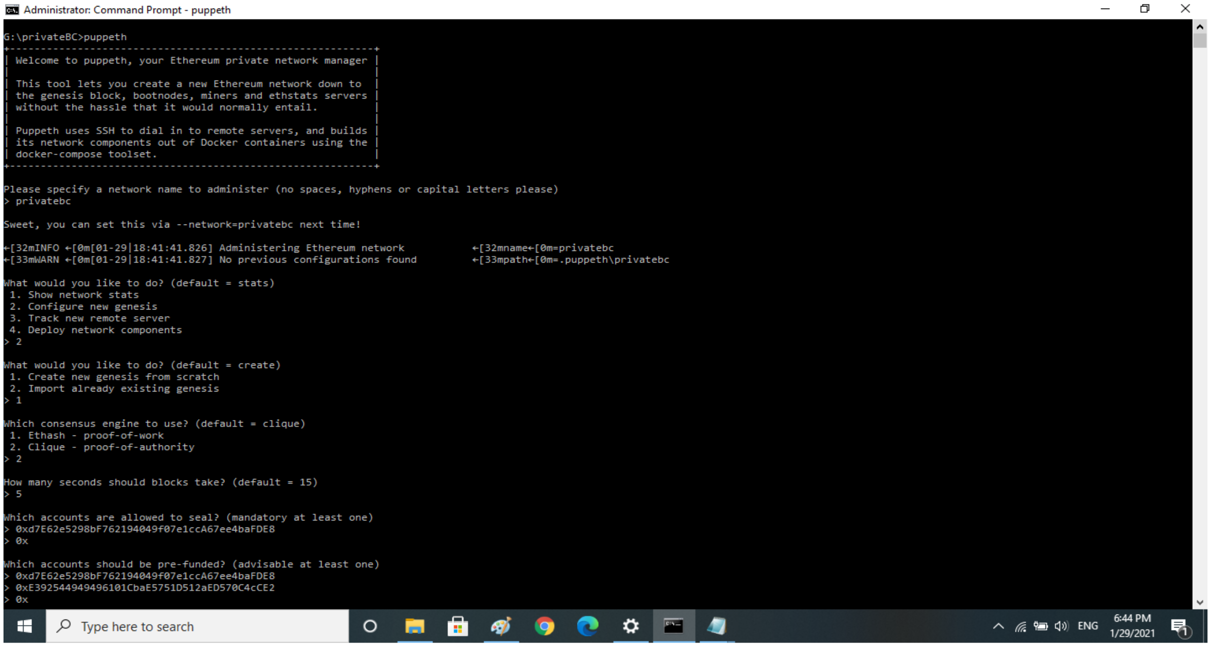
After we created a folder named "privatBC" and inside it built two folders named "node1 and node2", by using command prompt terminal (as administrator), move to node1 and type a command which allows us to create an ethereum account for first node (node1), also we do the same thing with node2. See figure 4.4.



```
G:\privateBC>mkdir node1

G:\privateBC>mkdir node2

G:\privateBC>cd node1

G:\privateBC\node1>geth --datadir "./data" account new
INFO [01-29|18:37:07.283] Maximum peer count                       ETH=50 LES=0 total=50
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key:   0xd7E62e5298bF762194049f07e1ccA67ee4baFDE8
Path of the secret key file: data\keystore\UTC--2021-01-29T15-37-13.625577100Z--d7e62e5298bf762194049f07e1cca67ee4bafde8

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!


G:\privateBC\node1>cd..

G:\privateBC>cd node2

G:\privateBC\node2>geth --datadir "./data" account new
INFO [01-29|18:37:48.217] Maximum peer count                       ETH=50 LES=0 total=50
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key:   0xE392544949496101CbaE5751D512aED570C4cCE2
Path of the secret key file: data\keystore\UTC--2021-01-29T15-37-54.131172800Z--e392544949496101cbae5751d512aed570c4cce2

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!


G:\privateBC\node2>
```

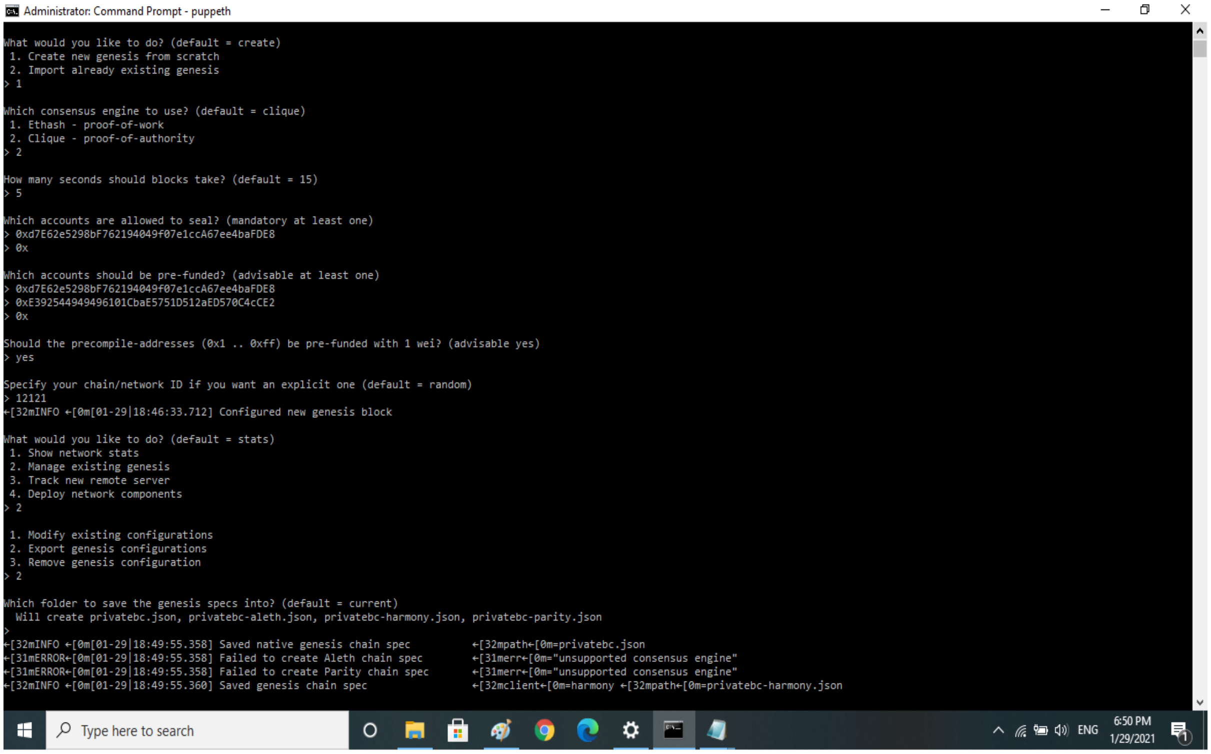
Figure 4.4: Create node1 and node2

Now, we have two nodes each one has its ethereum account. After creating these two nodes, we should generate the genesis block (it is the private blockchain network the first block that contains all the primary information about the private network. Again by using the terminal, we enter to privateBC folder and use "puppeth" tool (puppeth is a tool that aids us in creating a private ethereum blockchain network and it is available in Geth), we do that by type puppeth in terminal. After that there will be some questions as shown in figures 4.5 and 4.6, then we will generate the genesis block named privatebc.json, see figure 4.7.



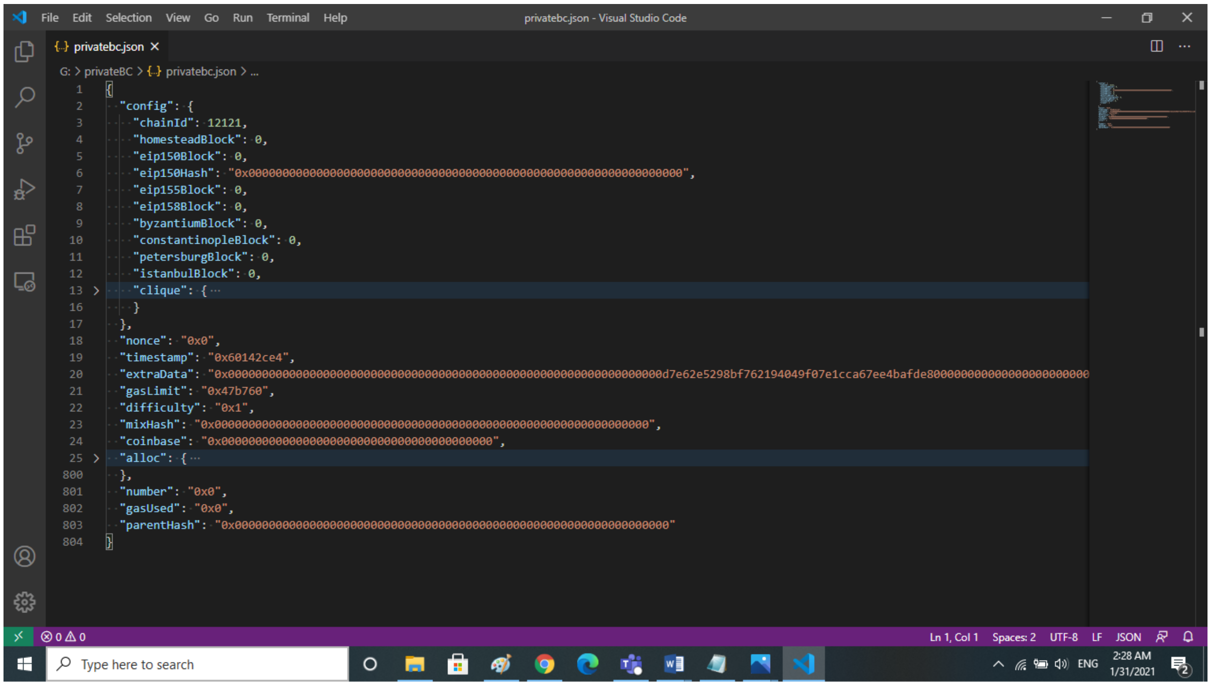
```
G:\privateBC>puppeth
+-----------------------------------------------------------+
| Welcome to puppeth, your Ethereum private network manager |
|                                                           |
| This tool lets you create a new Ethereum network down to  |
| the genesis block, bootnodes, miners and ethstats servers |
| without the hassle that it would normally entail.         |
|                                                           |
| Puppeth uses SSH to dial in to remote servers, and builds |
| its network components out of Docker containers using the |
| docker-compose toolset.                                   |
+-----------------------------------------------------------+

Please specify a network name to administer (no spaces, hyphens or capital letters please)
> privatebc

Sweet, you can set this via --network=privatebc next time!

←[32mINFO ←[0m[01-29|18:41:41.826] Administering Ethereum network           ←[32mname←[0m=privatebc
←[33mWARN ←[0m[01-29|18:41:41.827] No previous configurations found         ←[33mpath←[0m=.puppeth\privatebc

What would you like to do? (default = stats)
 1. Show network stats
 2. Configure new genesis
 3. Track new remote server
 4. Deploy network components
> 2

What would you like to do? (default = create)
 1. Create new genesis from scratch
 2. Import already existing genesis
> 1

Which consensus engine to use? (default = clique)
 1. Ethash - proof-of-work
 2. Clique - proof-of-authority
> 2

How many seconds should blocks take? (default = 15)
> 5

Which accounts are allowed to seal? (mandatory at least one)
> 0xd7E62e5298bF762194049f07e1ccA67ee4baFDE8
> 0x

Which accounts should be pre-funded? (advisable at least one)
> 0xd7E62e5298bF762194049f07e1ccA67ee4baFDE8
> 0xE392544949496101CbaE5751D512aED570C4cCE2
> 0x
```

Figure 4.5: Create the genesis block part A



```
What would you like to do? (default = create)
 1. Create new genesis from scratch
 2. Import already existing genesis
> 1

Which consensus engine to use? (default = clique)
 1. Ethash - proof-of-work
 2. Clique - proof-of-authority
> 2

How many seconds should blocks take? (default = 15)
> 5

Which accounts are allowed to seal? (mandatory at least one)
> 0xd7E62e5298bF762194049f07e1ccA67ee4baFDE8
> 0x

Which accounts should be pre-funded? (advisable at least one)
> 0xd7E62e5298bF762194049f07e1ccA67ee4baFDE8
> 0xE392544949496101CbaE5751D512aED570C4cCE2
> 0x

Should the precompile-addresses (0x1 .. 0xff) be pre-funded with 1 wei? (advisable yes)
> yes

Specify your chain/network ID if you want an explicit one (default = random)
> 12121
←[32mINFO ←[0m[01-29|18:46:33.712] Configured new genesis block

What would you like to do? (default = stats)
 1. Show network stats
 2. Manage existing genesis
 3. Track new remote server
 4. Deploy network components
> 2

 1. Modify existing configurations
 2. Export genesis configurations
 3. Remove genesis configuration
> 2

Which folder to save the genesis specs into? (default = current)
  Will create privatebc.json, privatebc-aleth.json, privatebc-harmony.json, privatebc-parity.json
>
←[32mINFO ←[0m[01-29|18:49:55.358] Saved native genesis chain spec          ←[32mpath←[0m=privatebc.json
←[31mERROR←[0m[01-29|18:49:55.358] Failed to create Aleth chain spec        ←[31merr←[0m="unsupported consensus engine"
←[31mERROR←[0m[01-29|18:49:55.358] Failed to create Parity chain spec       ←[31merr←[0m="unsupported consensus engine"
←[32mINFO ←[0m[01-29|18:49:55.360] Saved genesis chain spec                 ←[32mclient←[0m=harmony ←[32mpath←[0m=privatebc-harmony.json
```

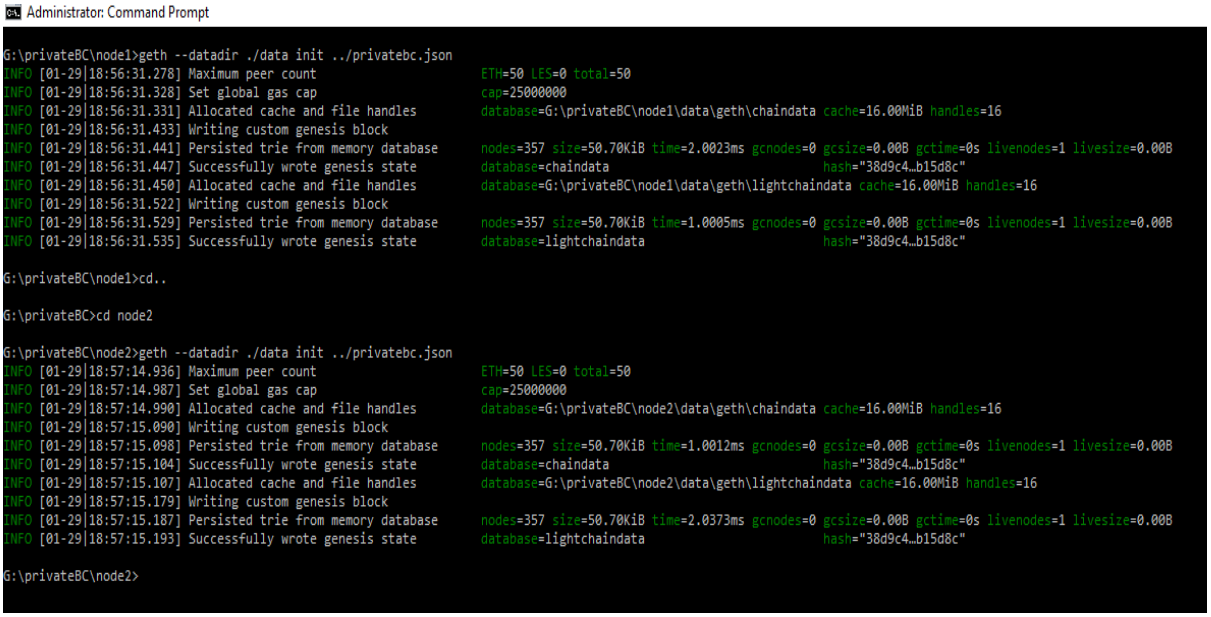
Figure 4.6: Create the genesis block part B



```json
{
  "config": {
    "chainId": 12121,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip150Hash": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "clique": {···
    }
  },
  "nonce": "0x0",
  "timestamp": "0x60142ce4",
  "extraData": "0x0000000000000000000000000000000000000000000000000000000000000000d7e62e5298bf762194049f07e1cca67ee4bafde800000000000000000000
  "gasLimit": "0x47b760",
  "difficulty": "0x1",
  "mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "coinbase": "0x0000000000000000000000000000000000000000",
  "alloc": {···
  },
  "number": "0x0",
  "gasUsed": "0x0",
  "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000"
}
```

Figure 4.7: The genesis block

Now, we have a genesis block, what we want to do is to connect node1 and node2 by using the genesis block see figure 4.8.



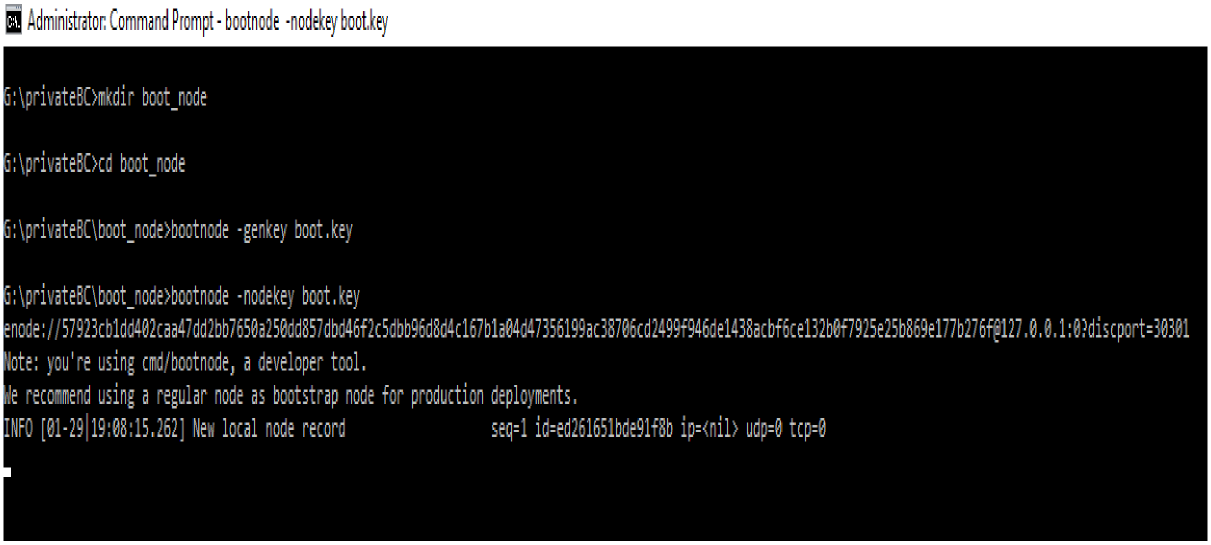
```
G:\privateBC\node1>geth --datadir ./data init ../privatebc.json
INFO [01-29|18:56:31.278] Maximum peer count                      ETH=50 LES=0 total=50
INFO [01-29|18:56:31.328] Set global gas cap                      cap=25000000
INFO [01-29|18:56:31.331] Allocated cache and file handles        database=G:\privateBC\node1\data\geth\chaindata cache=16.00MiB handles=16
INFO [01-29|18:56:31.433] Writing custom genesis block
INFO [01-29|18:56:31.441] Persisted trie from memory database     nodes=357 size=50.70KiB time=2.0023ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [01-29|18:56:31.447] Successfully wrote genesis state        database=chaindata                             hash="38d9c4…b15d8c"
INFO [01-29|18:56:31.450] Allocated cache and file handles        database=G:\privateBC\node1\data\geth\lightchaindata cache=16.00MiB handles=16
INFO [01-29|18:56:31.522] Writing custom genesis block
INFO [01-29|18:56:31.529] Persisted trie from memory database     nodes=357 size=50.70KiB time=1.0005ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [01-29|18:56:31.535] Successfully wrote genesis state        database=lightchaindata                        hash="38d9c4…b15d8c"

G:\privateBC\node1>cd..

G:\privateBC>cd node2

G:\privateBC\node2>geth --datadir ./data init ../privatebc.json
INFO [01-29|18:57:14.936] Maximum peer count                      ETH=50 LES=0 total=50
INFO [01-29|18:57:14.987] Set global gas cap                      cap=25000000
INFO [01-29|18:57:14.990] Allocated cache and file handles        database=G:\privateBC\node2\data\geth\chaindata cache=16.00MiB handles=16
INFO [01-29|18:57:15.090] Writing custom genesis block
INFO [01-29|18:57:15.098] Persisted trie from memory database     nodes=357 size=50.70KiB time=1.0012ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [01-29|18:57:15.104] Successfully wrote genesis state        database=chaindata                             hash="38d9c4…b15d8c"
INFO [01-29|18:57:15.107] Allocated cache and file handles        database=G:\privateBC\node2\data\geth\lightchaindata cache=16.00MiB handles=16
INFO [01-29|18:57:15.179] Writing custom genesis block
INFO [01-29|18:57:15.187] Persisted trie from memory database     nodes=357 size=50.70KiB time=2.0373ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [01-29|18:57:15.193] Successfully wrote genesis state        database=lightchaindata                        hash="38d9c4…b15d8c"

G:\privateBC\node2>
```

Figure 4.8: Connect the genesis block with node1 and node2

4.8.1.3 Creation of the bootnode

Bootnode is a bootstrap node in our private network when it starts, the other nodes in the same private network can find each other. At the same time this node would not mine any block (the main advantage of bootnode is to connect all nodes in the network).

When we created the two nodes (node1 and node2), there was no connection between them, however, by using bootnode we will make a connection between these two nodes. To do that we create a folder named "boot_node" inside "privateBC" folder, enter to this folder by using terminal, and type commands as shown in figure 4.9, in this figure we can see there is an enode generated (enode is used to describe an ethereum node) and this enode is used to connect the node1 and node2 with the bootnode to start the network.

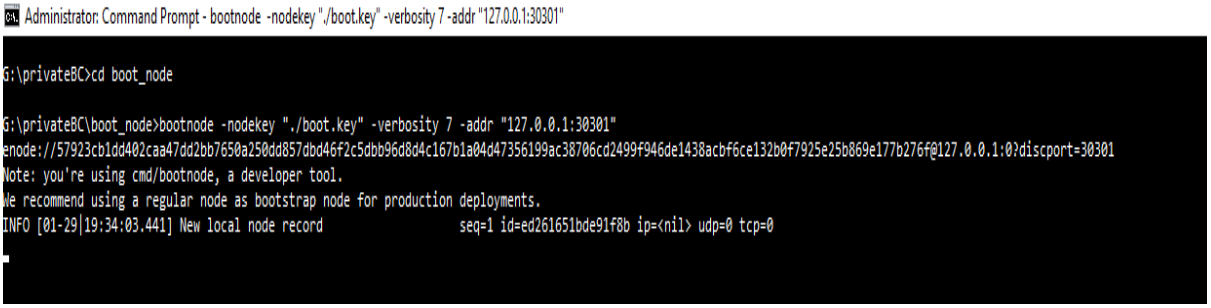


```
G:\privateBC>mkdir boot_node

G:\privateBC>cd boot_node

G:\privateBC\boot_node>bootnode -genkey boot.key

G:\privateBC\boot_node>bootnode -nodekey boot.key
enode://57923cb1dd402caa47dd2bb7650a250dd857dbd46f2c5dbb96d8d4c167b1a04d47356199ac38706cd2499f946de1438acbf6ce132b0f7925e25b869e177b276f@127.0.0.1:0?discport=30301
Note: you're using cmd/bootnode, a developer tool.
We recommend using a regular node as bootstrap node for production deployments.
INFO [01-29|19:08:15.262] New local node record                    seq=1 id=ed261651bde91f8b ip=<nil> udp=0 tcp=0
```

Figure 4.9: Create bootnode

4.8.1.4 Connecting all nodes

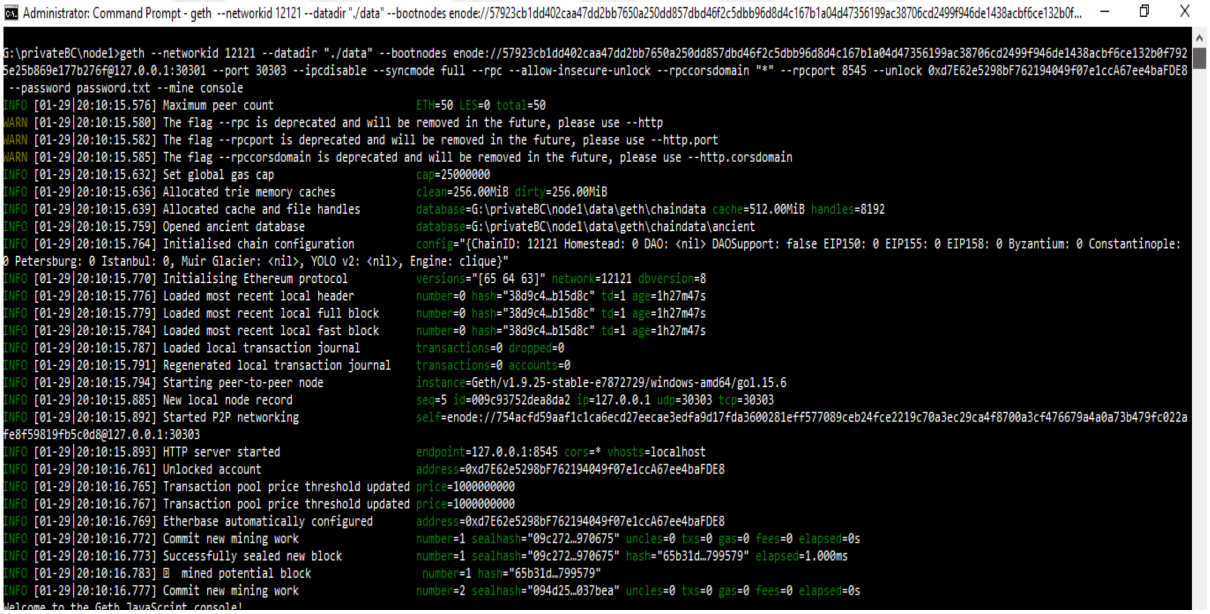
To complete our private blockchain network, we should connect node 1 and node2 with boot node. This is done by opening three terminals, the first terminal to run bootnode, we do that by typing commands as shown in figure 4.10:



```
Administrator: Command Prompt - bootnode -nodekey "./boot.key" -verbosity 7 -addr "127.0.0.1:30301"

G:\privateBC>cd boot_node

G:\privateBC\boot_node>bootnode -nodekey "./boot.key" -verbosity 7 -addr "127.0.0.1:30301"
enode://57923cb1dd402caa47dd2bb7650a250dd857dbd46f2c5dbb96d8d4c167b1a04d47356199ac38706cd2499f946de1438acbf6ce132b0f7925e25b869e177b276f@127.0.0.1:0?discport=30301
Note: you're using cmd/bootnode, a developer tool.
We recommend using a regular node as bootstrap node for production deployments.
INFO [01-29|19:34:03.441] New local node record                    seq=1 id=ed261651bde91f8b ip=<nil> udp=0 tcp=0
```

Figure 4.10: Start bootnode

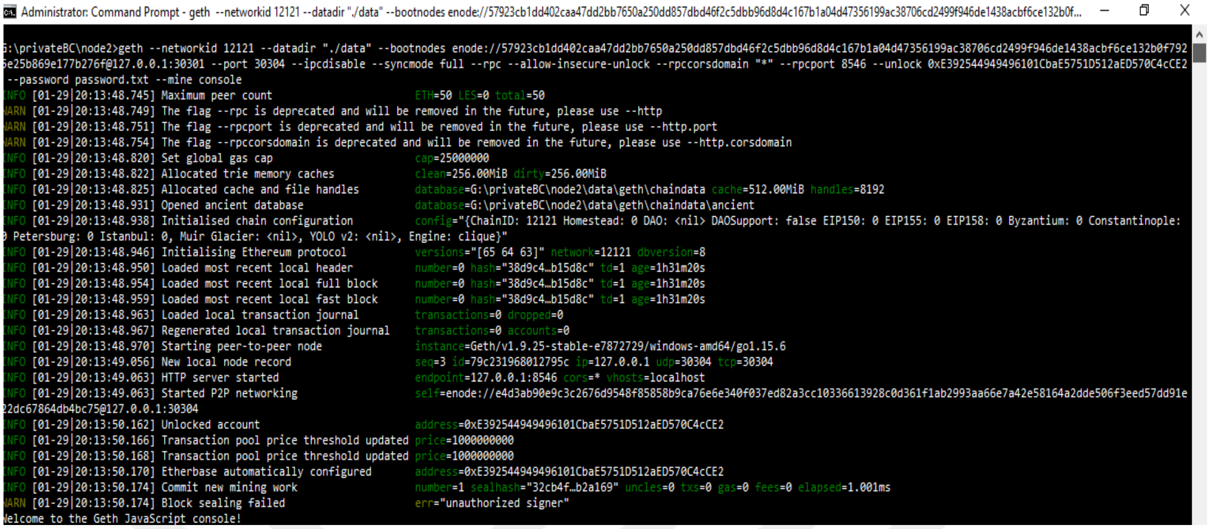
The second terminal is used for connecting node1 with bootnode by typing a command in the terminal as shown in figure 4.11:



```
Administrator: Command Prompt - geth --networkid 12121 --datadir "./data" --bootnodes enode://57923cb1dd402caa47dd2bb7650a250dd857dbd46f2c5dbb96d8d4c167b1a04d47356199ac38706cd2499f946de1438acbf6ce132b0f...

G:\privateBC\node1>geth --networkid 12121 --datadir "./data" --bootnodes enode://57923cb1dd402caa47dd2bb7650a250dd857dbd46f2c5dbb96d8d4c167b1a04d47356199ac38706cd2499f946de1438acbf6ce132b0f792
5e25b869e177b276f@127.0.0.1:30301 --port 30303 --ipcdisable --syncmode full --rpc --allow-insecure-unlock --rpccorsdomain "*" --rpcport 8545 --unlock 0xd7E62e5298bF762194049f07e1ccA67ee4baFDE8
 --password password.txt --mine console
INFO [01-29|20:10:15.576] Maximum peer count                       ETH=50 LES=0 total=50
WARN [01-29|20:10:15.580] The flag --rpc is deprecated and will be removed in the future, please use --http
WARN [01-29|20:10:15.582] The flag --rpcport is deprecated and will be removed in the future, please use --http.port
WARN [01-29|20:10:15.585] The flag --rpccorsdomain is deprecated and will be removed in the future, please use --http.corsdomain
INFO [01-29|20:10:15.632] Set global gas cap                       cap=25000000
INFO [01-29|20:10:15.636] Allocated trie memory caches             clean=256.00MiB dirty=256.00MiB
INFO [01-29|20:10:15.639] Allocated cache and file handles         database=G:\privateBC\node1\data\geth\chaindata cache=512.00MiB handles=8192
INFO [01-29|20:10:15.759] Opened ancient database                  database=G:\privateBC\node1\data\geth\chaindata\ancient
INFO [01-29|20:10:15.764] Initialised chain configuration          config="{ChainID: 12121 Homestead: 0 DAO: <nil> DAOSupport: false EIP150: 0 EIP155: 0 EIP158: 0 Byzantium: 0 Constantinople:
0 Petersburg: 0 Istanbul: 0, Muir Glacier: <nil>, YOLO v2: <nil>, Engine: clique}"
INFO [01-29|20:10:15.770] Initialising Ethereum protocol           versions="[65 64 63]" network=12121 dbversion=8
INFO [01-29|20:10:15.776] Loaded most recent local header          number=0 hash="38d9c4…b15d8c" td=1 age=1h27m47s
INFO [01-29|20:10:15.779] Loaded most recent local full block      number=0 hash="38d9c4…b15d8c" td=1 age=1h27m47s
INFO [01-29|20:10:15.784] Loaded most recent local fast block      number=0 hash="38d9c4…b15d8c" td=1 age=1h27m47s
INFO [01-29|20:10:15.787] Loaded local transaction journal         transactions=0 dropped=0
INFO [01-29|20:10:15.791] Regenerated local transaction journal    transactions=0 accounts=0
INFO [01-29|20:10:15.794] Starting peer-to-peer node               instance=Geth/v1.9.25-stable-e7872729/windows-amd64/go1.15.6
INFO [01-29|20:10:15.885] New local node record                    seq=5 id=009c93752dea8da2 ip=127.0.0.1 udp=30303 tcp=30303
INFO [01-29|20:10:15.892] Started P2P networking                   self=enode://754acfd59aaf1c1ca6ecd27eecae3edfa9d17fda3600281eff577089ceb24fce2219c70a3ec29ca4f8700a3cf476679a4a0a73b479fc022a
fe8f59819fb5c0d8@127.0.0.1:30303
INFO [01-29|20:10:15.893] HTTP server started                      endpoint=127.0.0.1:8545 cors=* vhosts=localhost
INFO [01-29|20:10:16.761] Unlocked account                         address=0xd7E62e5298bF762194049f07e1ccA67ee4baFDE8
INFO [01-29|20:10:16.765] Transaction pool price threshold updated price=1000000000
INFO [01-29|20:10:16.767] Transaction pool price threshold updated price=1000000000
INFO [01-29|20:10:16.769] Etherbase automatically configured       address=0xd7E62e5298bF762194049f07e1ccA67ee4baFDE8
INFO [01-29|20:10:16.772] Commit new mining work                   number=1 sealhash="09c272…970675" uncles=0 txs=0 gas=0 fees=0 elapsed=0s
INFO [01-29|20:10:16.773] Successfully sealed new block            number=1 sealhash="09c272…970675" hash="65b31d…799579" elapsed=1.000ms
INFO [01-29|20:10:16.783] ⍰  mined potential block                  number=1 hash="65b31d…799579"
INFO [01-29|20:10:16.777] Commit new mining work                   number=2 sealhash="094d25…037bea" uncles=0 txs=0 gas=0 fees=0 elapsed=0s
Welcome to the Geth JavaScript console!
```

Figure 4.11: Connect node1 with bootnode

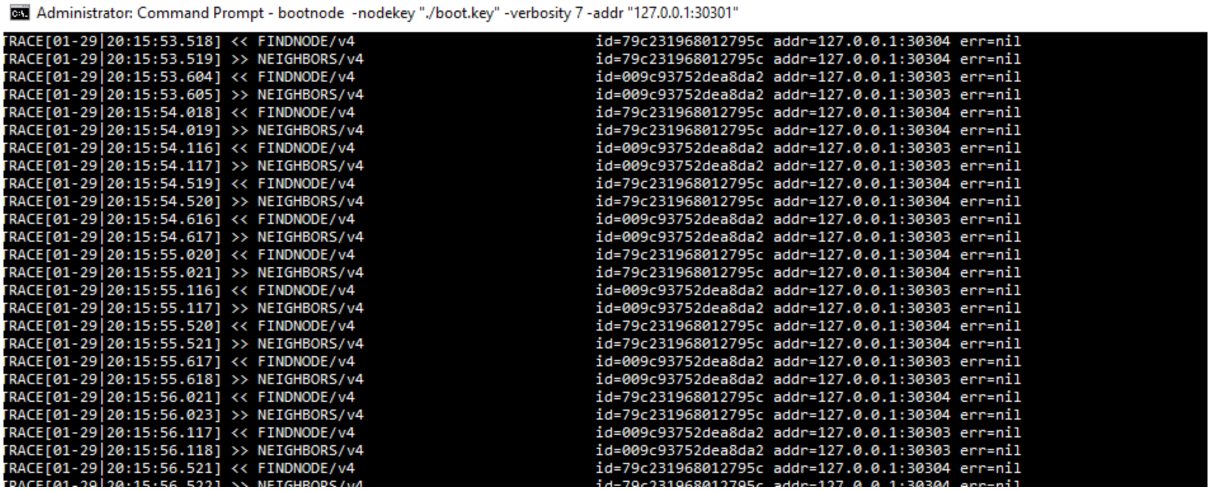
The same thing with the third terminal that is used for connecting node2 with bootnode by typing a command as shown in figure 4.12:



```
Administrator: Command Prompt - geth --networkid 12121 --datadir "./data" --bootnodes enode://57923cb1dd402caa47dd2bb7650a250dd857dbd46f2c5dbb96d8d4c167b1a04d47356199ac38706cd2499f946de1438acbf6ce132b0f...

G:\privateBC\node2>geth --networkid 12121 --datadir "./data" --bootnodes enode://57923cb1dd402caa47dd2bb7650a250dd857dbd46f2c5dbb96d8d4c167b1a04d47356199ac38706cd2499f946de1438acbf6ce132b0f792
5e25b869e177b276f@127.0.0.1:30301 --port 30304 --ipcdisable --syncmode full --rpc --allow-insecure-unlock --rpccorsdomain "*" --rpcport 8546 --unlock 0xE392544949496101CbaE5751D512aED570C4cCE2
 --password password.txt --mine console
INFO [01-29|20:13:48.745] Maximum peer count                       ETH=50 LES=0 total=50
WARN [01-29|20:13:48.749] The flag --rpc is deprecated and will be removed in the future, please use --http
WARN [01-29|20:13:48.751] The flag --rpcport is deprecated and will be removed in the future, please use --http.port
WARN [01-29|20:13:48.754] The flag --rpccorsdomain is deprecated and will be removed in the future, please use --http.corsdomain
INFO [01-29|20:13:48.820] Set global gas cap                       cap=25000000
INFO [01-29|20:13:48.822] Allocated trie memory caches             clean=256.00MiB dirty=256.00MiB
INFO [01-29|20:13:48.825] Allocated cache and file handles         database=G:\privateBC\node2\data\geth\chaindata cache=512.00MiB handles=8192
INFO [01-29|20:13:48.931] Opened ancient database                  database=G:\privateBC\node2\data\geth\chaindata\ancient
INFO [01-29|20:13:48.938] Initialised chain configuration          config="{ChainID: 12121 Homestead: 0 DAO: <nil> DAOSupport: false EIP150: 0 EIP155: 0 EIP158: 0 Byzantium: 0 Constantinople:
0 Petersburg: 0 Istanbul: 0, Muir Glacier: <nil>, YOLO v2: <nil>, Engine: clique}"
INFO [01-29|20:13:48.946] Initialising Ethereum protocol           versions="[65 64 63]" network=12121 dbversion=8
INFO [01-29|20:13:48.950] Loaded most recent local header          number=0 hash="38d9c4…b15d8c" td=1 age=1h31m20s
INFO [01-29|20:13:48.954] Loaded most recent local full block      number=0 hash="38d9c4…b15d8c" td=1 age=1h31m20s
INFO [01-29|20:13:48.959] Loaded most recent local fast block      number=0 hash="38d9c4…b15d8c" td=1 age=1h31m20s
INFO [01-29|20:13:48.963] Loaded local transaction journal         transactions=0 dropped=0
INFO [01-29|20:13:48.967] Regenerated local transaction journal    transactions=0 accounts=0
INFO [01-29|20:13:48.970] Starting peer-to-peer node               instance=Geth/v1.9.25-stable-e7872729/windows-amd64/go1.15.6
INFO [01-29|20:13:49.056] New local node record                    seq=3 id=79c231968012795c ip=127.0.0.1 udp=30304 tcp=30304
INFO [01-29|20:13:49.063] HTTP server started                      endpoint=127.0.0.1:8546 cors=* vhosts=localhost
INFO [01-29|20:13:49.063] Started P2P networking                   self=enode://e4d3ab90e9c3c2676d9548f85858b9ca76e6e340f037ed82a3cc10336613928c0d361f1ab2993aa66e7a42e58164a2dde506f3eed57dd91e
22dc67864db4bc75@127.0.0.1:30304
INFO [01-29|20:13:50.162] Unlocked account                         address=0xE392544949496101CbaE5751D512aED570C4cCE2
INFO [01-29|20:13:50.166] Transaction pool price threshold updated price=1000000000
INFO [01-29|20:13:50.168] Transaction pool price threshold updated price=1000000000
INFO [01-29|20:13:50.170] Etherbase automatically configured       address=0xE392544949496101CbaE5751D512aED570C4cCE2
INFO [01-29|20:13:50.174] Commit new mining work                   number=1 sealhash="32cb4f…b2a169" uncles=0 txs=0 gas=0 fees=0 elapsed=1.001ms
WARN [01-29|20:13:50.174] Block sealing failed                     err="unauthorized signer"
Welcome to the Geth JavaScript console!
```

Figure 4.12: Connect node2 with bootnode

In figures 4.13we can see node1 works on port 30303, node2 works on port 30304, and our private PoA ethereum blockchain network is ready. Therefore, we can use it to deploy our smart contracts by connecting it to remix IDE.



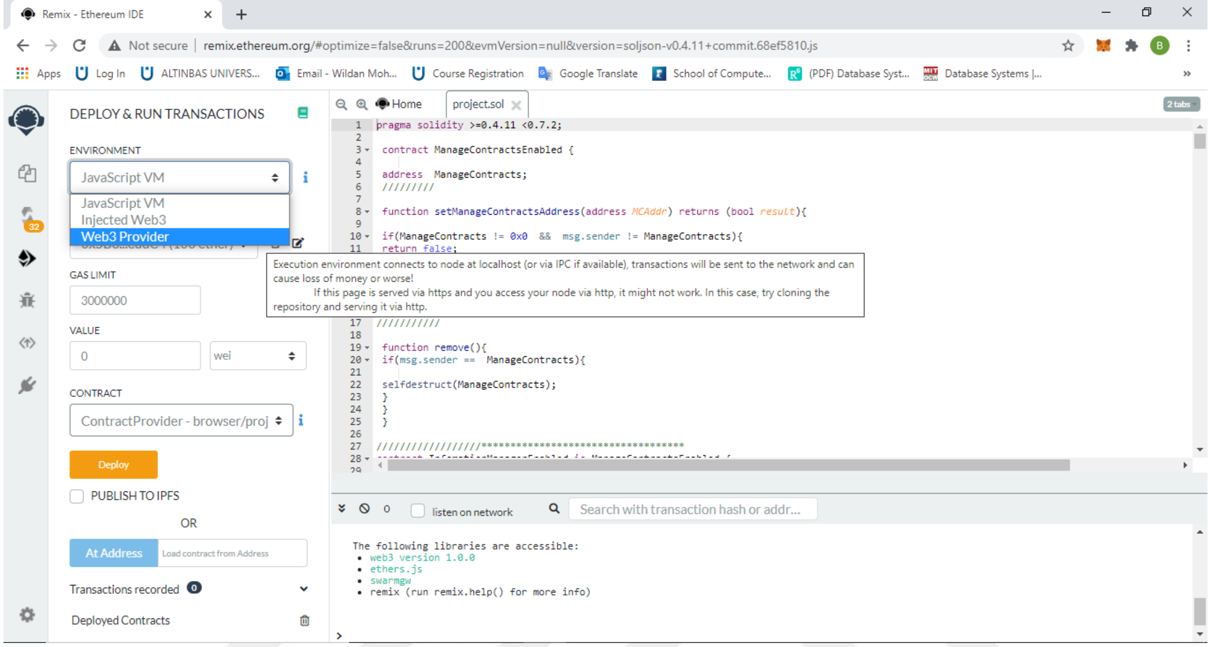
```
Administrator: Command Prompt - bootnode -nodekey "./boot.key" -verbosity 7 -addr "127.0.0.1:30301"
TRACE[01-29|20:15:53.518] << FINDNODE/v4                         id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:53.519] >> NEIGHBORS/v4                        id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:53.604] << FINDNODE/v4                         id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:53.605] >> NEIGHBORS/v4                        id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:54.018] << FINDNODE/v4                         id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:54.019] >> NEIGHBORS/v4                        id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:54.116] << FINDNODE/v4                         id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:54.117] >> NEIGHBORS/v4                        id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:54.519] << FINDNODE/v4                         id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:54.520] >> NEIGHBORS/v4                        id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:54.616] << FINDNODE/v4                         id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:54.617] >> NEIGHBORS/v4                        id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:55.020] << FINDNODE/v4                         id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:55.021] >> NEIGHBORS/v4                        id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:55.116] << FINDNODE/v4                         id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:55.117] >> NEIGHBORS/v4                        id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:55.520] << FINDNODE/v4                         id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:55.521] >> NEIGHBORS/v4                        id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:55.617] << FINDNODE/v4                         id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:55.618] >> NEIGHBORS/v4                        id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:56.021] << FINDNODE/v4                         id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:56.023] >> NEIGHBORS/v4                        id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:56.117] << FINDNODE/v4                         id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:56.118] >> NEIGHBORS/v4                        id=009c93752dea8da2 addr=127.0.0.1:30303 err=nil
TRACE[01-29|20:15:56.521] << FINDNODE/v4                         id=79c231968012795c addr=127.0.0.1:30304 err=nil
TRACE[01-29|20:15:56.522] >> NEIGHBORS/v4                        id=79c231968012795c addr=127.0.0.1:30304 err=nil
```

Figure 4.13: The private PoA ethereum blockchain network

4.8.1.5 Creation and deployment of the smart contract:

In order to deploy the smart contract, we have to connect Remix IDE with the private PoA ethereum blockchain network. To do that, from the left side of Remix IDE we choose

DEPLOY&RUN TRANSACTIONS, then from Environment we select Web3Provider environment, see figure 4.14.



ENVIRONMENT: JavaScript VM
- JavaScript VM
- Injected Web3
- Web3 Provider

GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: ContractProvider - browser/proj

```
pragma solidity >=0.4.11 <0.7.2;

 contract ManageContractsEnabled {

 address  ManageContracts;
 /////////

 function setManageContractsAddress(address MCAddr) returns (bool result){

 if(ManageContracts != 0x0  &&  msg.sender != ManageContracts){
 return false;
```

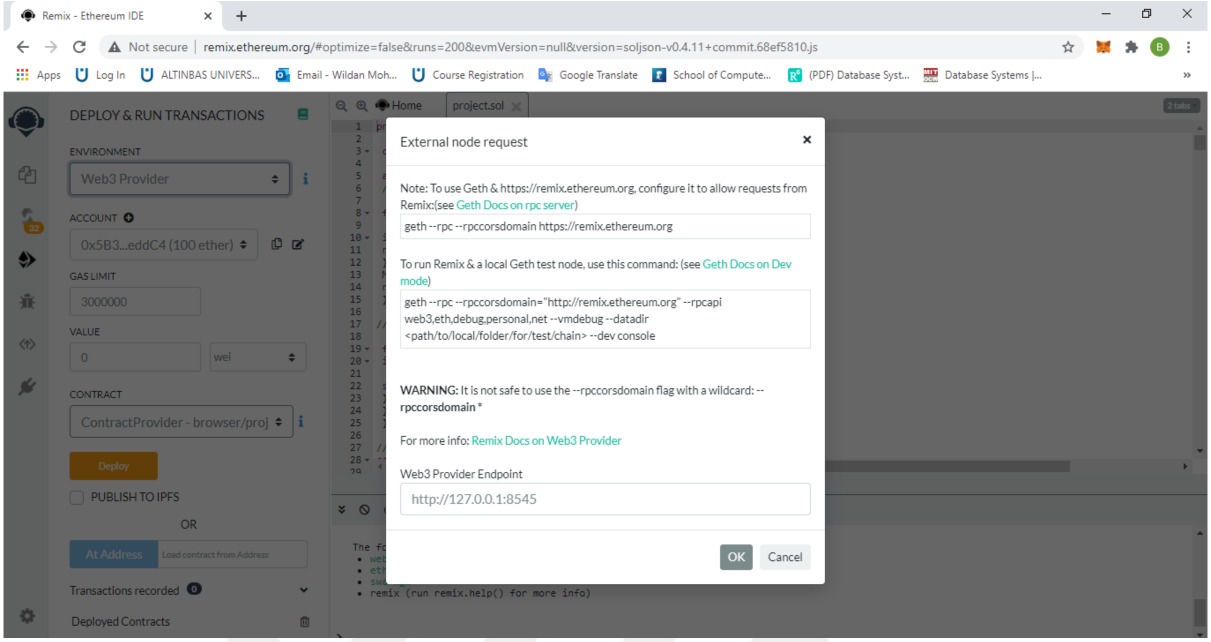
Execution environment connects to node at localhost (or via IPC if available), transactions will be sent to the network and can cause loss of money or worse!
If this page is served via https and you access your node via http, it might not work. In this case, try cloning the repository and serving it via http.

```
///////////

 function remove(){
 if(msg.sender ==  ManageContracts){

 selfdestruct(ManageContracts);
 }
 }
 }

//////////////////*********************************
```

```
The following libraries are accessible:
 • web3 version 1.0.0
 • ethers.js
 • swarmgw
 • remix (run remix.help() for more info)
```

Figure 4.14: Connect Remix IDE with The private PoA ethereum blockchain network

After selecting the environment, we enter the endpoint URL the private PoA ethereum blockchain network in Web3 Provider Endpoint field as it is demonstrated in figure 4.15. When the connection has been completed, the account of ethereum would appear in Remix IDE, see figure 4.16. As a result, we are ready to deploy the smart contracts.



External node request

Note: To use Geth & https://remix.ethereum.org, configure it to allow requests from Remix:(see Geth Docs on rpc server)

```
geth --rpc --rpccorsdomain https://remix.ethereum.org
```

To run Remix & a local Geth test node, use this command: (see Geth Docs on Dev mode)

```
geth --rpc --rpccorsdomain="http://remix.ethereum.org" --rpcapi
web3,eth,debug,personal,net --vmdebug --datadir
<path/to/local/folder/for/test/chain> --dev console
```

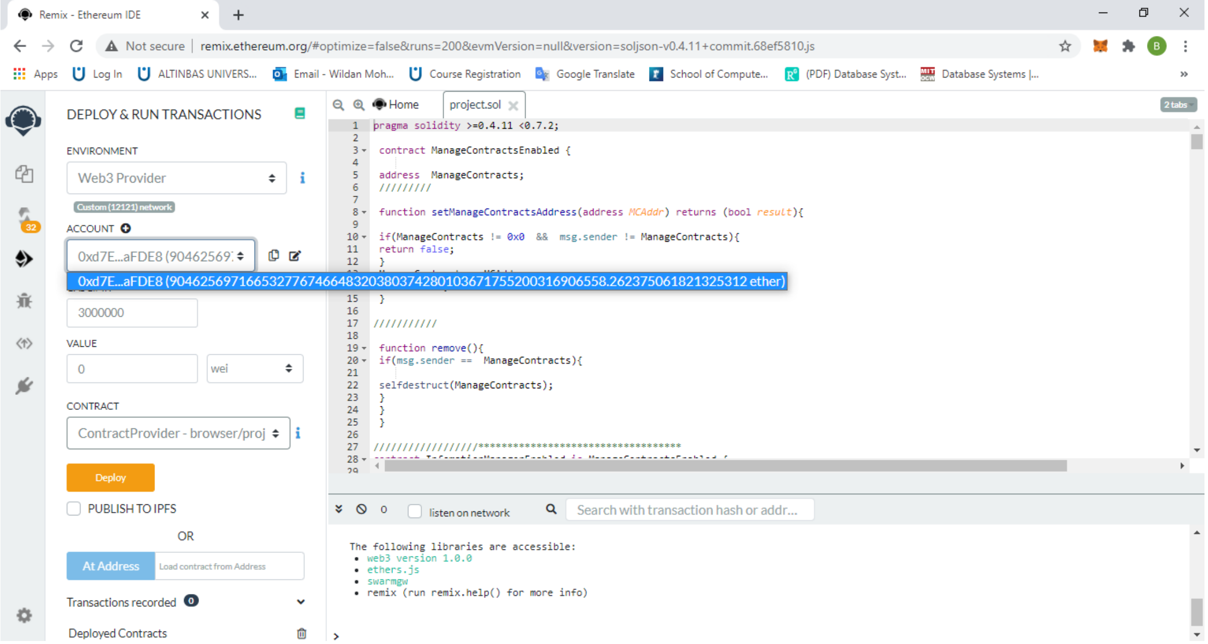
WARNING: It is not safe to use the --rpccorsdomain flag with a wildcard: --**rpccorsdomain ***

For more info: Remix Docs on Web3 Provider

Web3 Provider Endpoint: http://127.0.0.1:8545

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.11+commit.68ef5810.js
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Web3 Provider
ACCOUNT
0x5B3...eddC4 (100 ether)
GAS LIMIT
3000000
VALUE
0
wei
CONTRACT
ContractProvider - browser/proj
Deploy
PUBLISH TO IPFS
OR
At Address
Load contract from Address
Transactions recorded 0
Deployed Contracts
Home
project.sol
OK
Cancel

Figure 4.15: Entering the endpoint URL



ENVIRONMENT: Web3 Provider — Custom (12121) network

ACCOUNT: 0xd7E...aFDE8 (904625697166532776746648320380374280103671755200316906558.262375061821325312 ether)

```
pragma solidity >=0.4.11 <0.7.2;

 contract ManageContractsEnabled {

 address  ManageContracts;
 /////////

 function setManageContractsAddress(address MCAddr) returns (bool result){

 if(ManageContracts != 0x0  &&  msg.sender != ManageContracts){
 return false;
 }
```

```
 }

///////////

 function remove(){
 if(msg.sender ==  ManageContracts){

 selfdestruct(ManageContracts);
 }
 }
 }

/////////////////*********************************
```

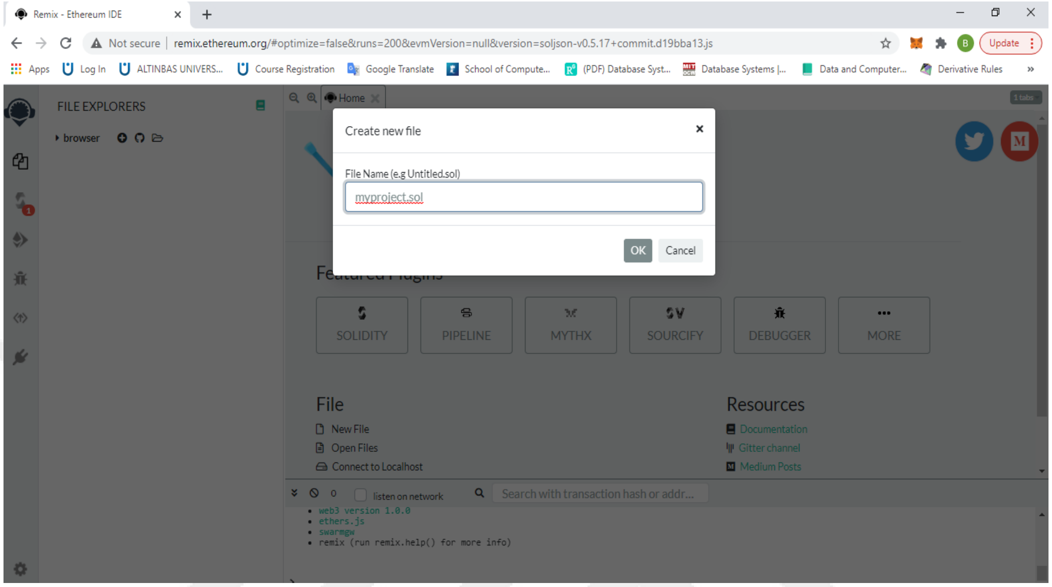
```
The following libraries are accessible:
• web3 version 1.0.0
• ethers.js
• swarmgw
• remix (run remix.help() for more info)
```

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.11+commit.68ef5810.js
DEPLOY & RUN TRANSACTIONS
3000000
VALUE
0
wei
CONTRACT
ContractProvider - browser/proj
Deploy
PUBLISH TO IPFS
OR
At Address
Load contract from Address
Transactions recorded 0
Deployed Contracts
Home
project.sol
2 tabs
listen on network
Search with transaction hash or addr...

Figure 4.16: the ethereum account of the private blockchain

As we mentioned in section 4.7, we have eleven contracts, now we will see how do create them.

By using chrome browser, we will open the Remix IDE from this link http://remix.ethereum.org/ , from FILE EXPLORER in the left side, click on the plus sign (Create New File) to create file type solidity (.sol) and give it a name, this file will contain the codes of all smart contracts , see figure 4.17.



remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js

Create new file

File Name (e.g Untitled.sol): myproject.sol

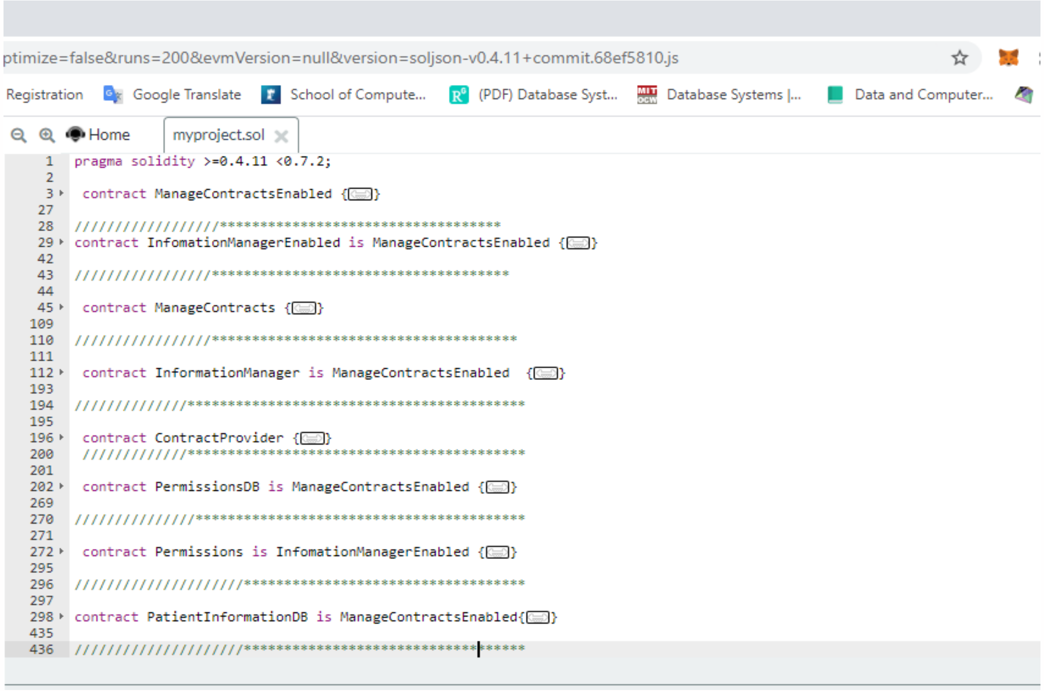
```
• web3 version 1.0.0
• ethers.js
• swarmgw
• remix (run remix.help() for more info)
```

Figure 4.17: Create a new file to contain the smart contracts code

Now, everything is ready to start writing our codes. The first statement is ***pragma solidity >=0.4.11 <0.7.2,*** it means that all contracts in this file will compile with compiler versions from 0.4.11 to 0.7.2. The next statement is ***contract*** keyword (it is like class keyword in *C#* and *Java* programming languages), it is followed by the contract name and inside the contract we will write and define all functions, modifiers, variables, etc. As shown in figure 4.18.

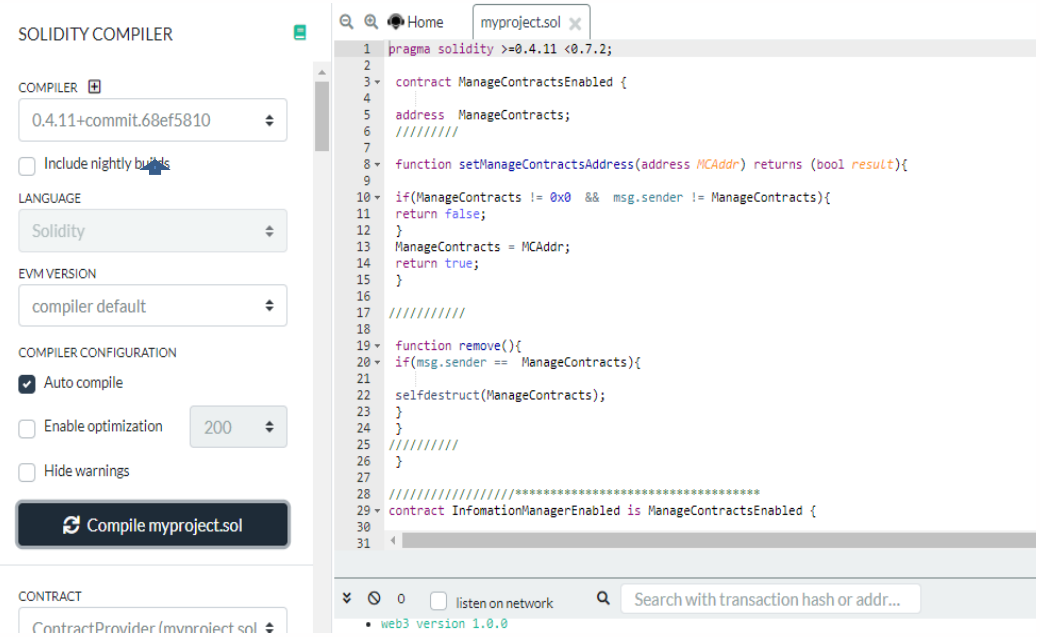
After the smart contracts' codes are completed, the next step is to compile the smart contracts to check if there are any syntax errors, to do that click compile button on the left to the Remix IDE see figure 4.19.

Now, our smart contracts are ready to deploy, from DEPLOY & RUN TRANSACTIONS in the left side we select the contracts we needed to deploy and click on Deploy button to deploy them. 'Manage Contracts ' contract must be deployed first and then the other contracts. As shown in figures 4.20, 4.21



```
pragma solidity >=0.4.11 <0.7.2;

 contract ManageContractsEnabled {↔}

/////////////////*************************************
contract InfomationManagerEnabled is ManageContractsEnabled {↔}

/////////////////************************************

 contract ManageContracts {↔}

/////////////////**************************************

 contract InformationManager is ManageContractsEnabled  {↔}

//////////////*****************************************

 contract ContractProvider {↔}
 ////////////*****************************************

 contract PermissionsDB is ManageContractsEnabled {↔}

///////////////****************************************

 contract Permissions is InfomationManagerEnabled {↔}

///////////////////*************************************

contract PatientInformationDB is ManageContractsEnabled{↔}

///////////////////*************************************
```

Figure 4.18: The code of our smart contracts



COMPILER

0.4.11+commit.68ef5810

☐ Include nightly builds

LANGUAGE

Solidity

EVM VERSION

compiler default

COMPILER CONFIGURATION

☑ Auto compile

☐ Enable optimization 200

☐ Hide warnings

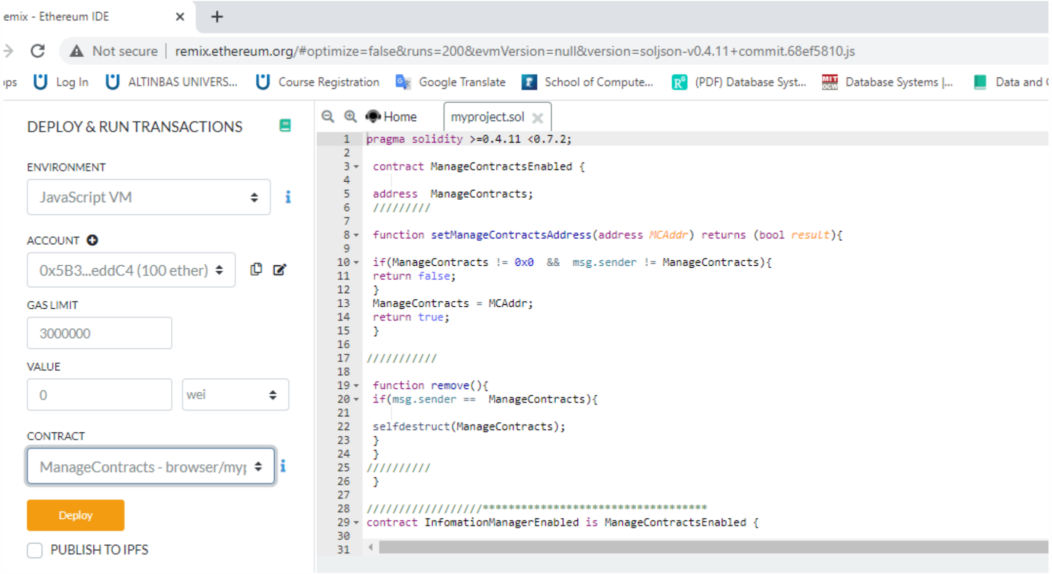
CONTRACT

ContractProvider (myproject.sol

```
pragma solidity >=0.4.11 <0.7.2;

 contract ManageContractsEnabled {

 address  ManageContracts;
 /////////

 function setManageContractsAddress(address MCAddr) returns (bool result){

 if(ManageContracts != 0x0  &&  msg.sender != ManageContracts){
 return false;
 }
 ManageContracts = MCAddr;
 return true;
 }

///////////

 function remove(){
 if(msg.sender ==  ManageContracts){

 selfdestruct(ManageContracts);
 }
 }
//////////
 }

/////////////////************************************
contract InfomationManagerEnabled is ManageContractsEnabled {
```

```
0
• web3 version 1.0.0
```

Figure 4.19: compile the smart contracts



emix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.11+commit.68ef5810.js
ps Log In ALTINBAS UNIVERS... Course Registration Google Translate School of Compute... (PDF) Database Syst... Database Systems |... Data and
DEPLOY & RUN TRANSACTIONS
Home
myproject.sol

ENVIRONMENT: JavaScript VM

ACCOUNT: 0x5B3...eddC4 (100 ether)

GAS LIMIT: 3000000

VALUE: 0 wei

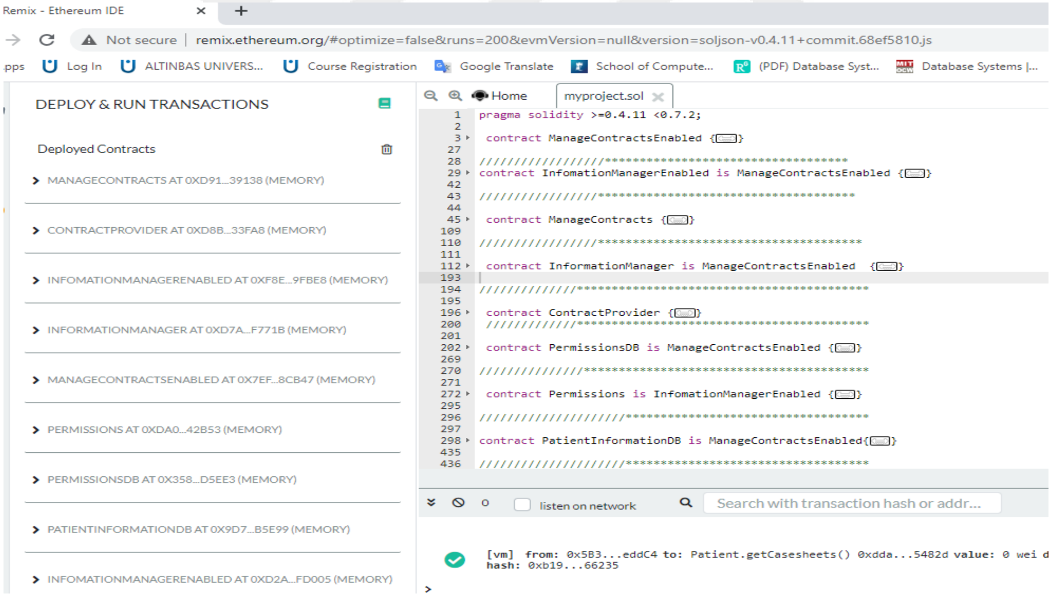
CONTRACT: ManageContracts - browser/my|

Deploy

□ PUBLISH TO IPFS

```
pragma solidity >=0.4.11 <0.7.2;

 contract ManageContractsEnabled {

 address  ManageContracts;
 /////////

 function setManageContractsAddress(address MCAddr) returns (bool result){

 if(ManageContracts != 0x0  &&  msg.sender != ManageContracts){
 return false;
 }
 ManageContracts = MCAddr;
 return true;
 }

///////////

 function remove(){
 if(msg.sender ==  ManageContracts){

 selfdestruct(ManageContracts);
 }
 }
//////////
 }

/////////////////*******************************
contract InfomationManagerEnabled is ManageContractsEnabled {

```

Figure 4.20: Deploy the *ManageContracts* contract



Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.11+commit.68ef5810.js
pps Log In ALTINBAS UNIVERS... Course Registration Google Translate School of Compute... (PDF) Database Syst... Database Systems |...
DEPLOY & RUN TRANSACTIONS
Home
myproject.sol
listen on network
Search with transaction hash or addr...

Deployed Contracts

- MANAGECONTRACTS AT 0XD91...39138 (MEMORY)
- CONTRACTPROVIDER AT 0XD8B...33FA8 (MEMORY)
- INFOMATIONMANAGERENABLED AT 0XF8E...9FBE8 (MEMORY)
- INFORMATIONMANAGER AT 0XD7A...F771B (MEMORY)
- MANAGECONTRACTSENABLED AT 0X7EF...8CB47 (MEMORY)
- PERMISSIONS AT 0XDA0...42B53 (MEMORY)
- PERMISSIONSDB AT 0X358...D5EE3 (MEMORY)
- PATIENTINFORMATIONDB AT 0X9D7...B5E99 (MEMORY)
- INFOMATIONMANAGERENABLED AT 0XD2A...FD005 (MEMORY)

```
pragma solidity >=0.4.11 <0.7.2;

 contract ManageContractsEnabled {↔}

//////////////////*********************************
contract InfomationManagerEnabled is ManageContractsEnabled {↔}

/////////////////***********************************

 contract ManageContracts {↔}

/////////////////************************************

 contract InformationManager is ManageContractsEnabled  {↔}

//////////////**************************************

 contract ContractProvider {↔}
 ////////////**************************************

 contract PermissionsDB is ManageContractsEnabled {↔}

///////////////*************************************

 contract Permissions is InfomationManagerEnabled {↔}

/////////////////////*********************************

contract PatientInformationDB is ManageContractsEnabled{↔}

/////////////////////*********************************
```

```
[vm] from: 0x5B3...eddC4 to: Patient.getCasesheets() 0xdda...5482d value: 0 wei d
hash: 0xb19...66235
```

Figure 4.21: All deployed smart contracts

4.9 THE WEB3.JS API

In previous sections we wrote the smart contracts codes and deploy them on private PoA ethereum blockchain network by using remix IDE, but the remix IDE only helps the developers in testing their own smart contracts, which is inappropriate for end users. For enabling the end users to interact with the smart contracts, we should design a frontend to facilitate the complexity of the smart contracts (backend). To do that, we must use API " that is the software intermediary which can be used as an interface by software components in order to communicate with each other"[114].

The API we will use is web3.js API, it is a collection of JavaScript libraries[115] "plays the role of a bridge which can make a communication bridge among the frontend and the backend "[116], see figure 4.22.

In order to install the web3.js on our computer, we need to install Node.js, we can download it from this link: https://nodejs.org/en/download/, Node.js "it is a server environment of open source nature, that can run on different platforms (Windows, Linux, Unix, Mac OS X, etc.)"[117].

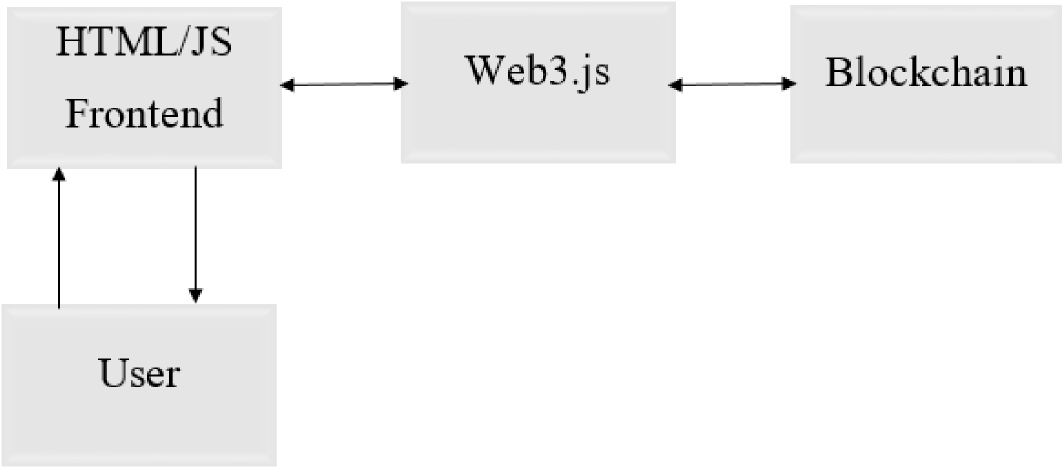


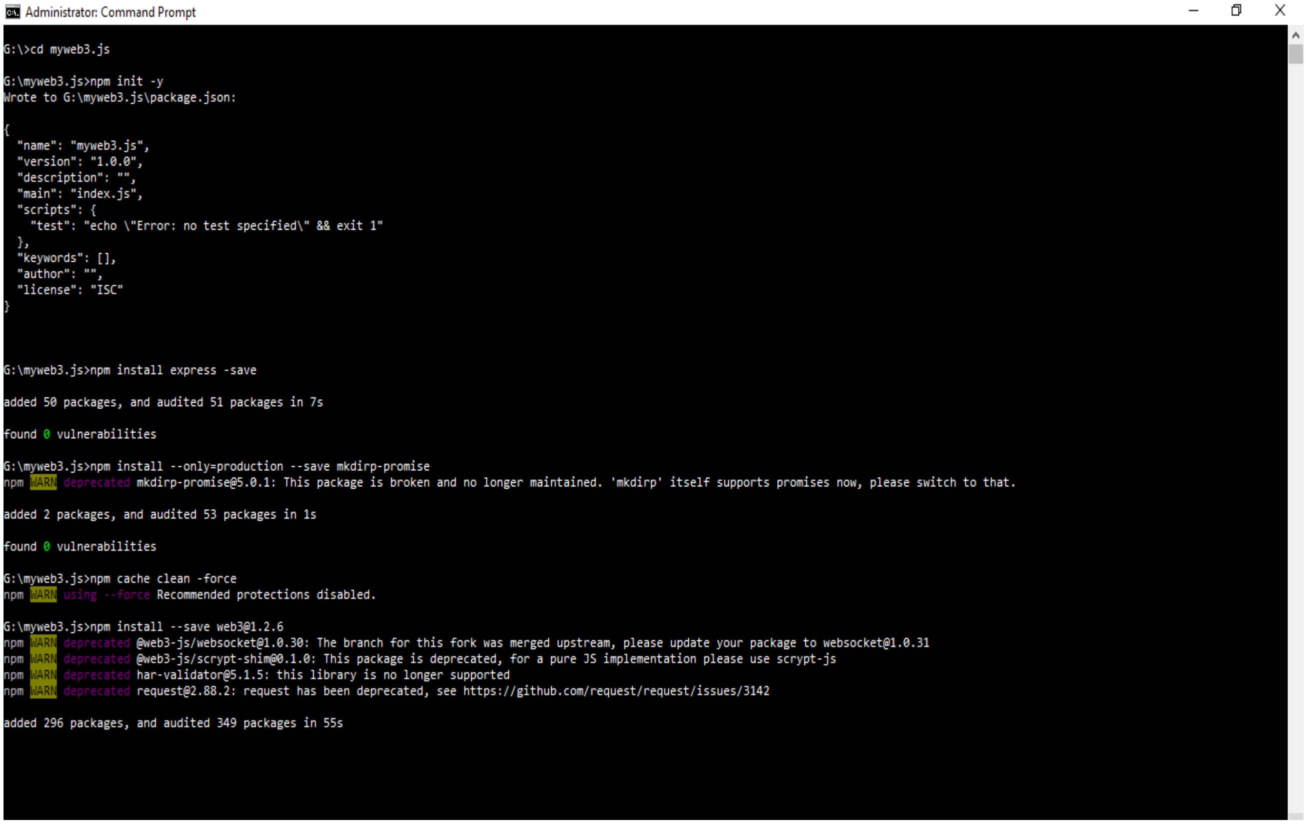
Figure 4.22: web3.js[115]

After installing node.js, we are ready to install web3.js by writing these command in the terminal, see figure 4.23:

1. mkdir myweb3.js
2. cd myweb3.js
3. npm init -y

4. npm install express –save
5. npm install --only=production --save mkdirp-promise
6. npm cache clean –force
7. npm install --save web3@1.2.6

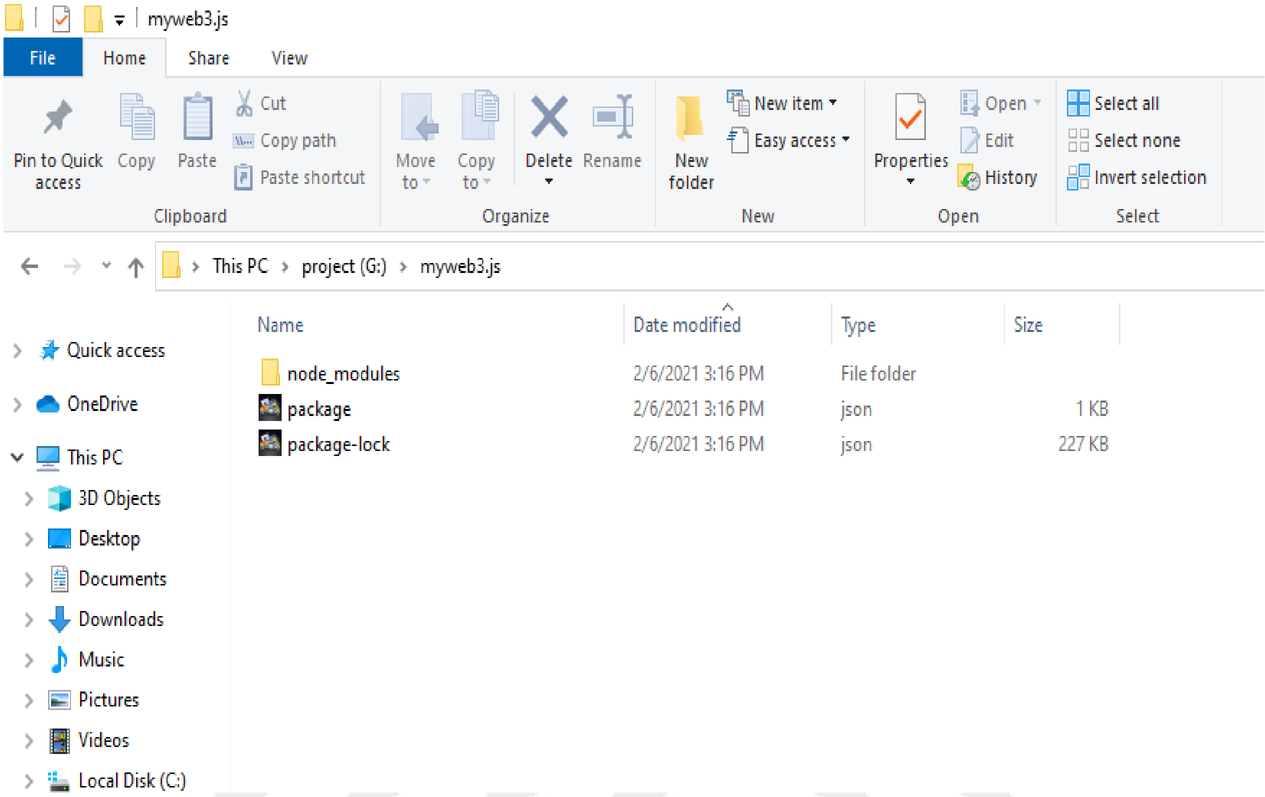
Now, we have myweb3.js folder inside it node modules folder, package.son file and package-lock.json file, all these files together make up the web3.js API, see figure 4.24.



Administrator: Command Prompt

```
G:\>cd myweb3.js

G:\myweb3.js>npm init -y
Wrote to G:\myweb3.js\package.json:

{
  "name": "myweb3.js",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}



G:\myweb3.js>npm install express -save

added 50 packages, and audited 51 packages in 7s

found 0 vulnerabilities

G:\myweb3.js>npm install --only=production --save mkdirp-promise
npm WARN deprecated mkdirp-promise@5.0.1: This package is broken and no longer maintained. 'mkdirp' itself supports promises now, please switch to that.

added 2 packages, and audited 53 packages in 1s

found 0 vulnerabilities

G:\myweb3.js>npm cache clean -force
npm WARN using --force Recommended protections disabled.

G:\myweb3.js>npm install --save web3@1.2.6
npm WARN deprecated @web3-js/websocket@1.0.30: The branch for this fork was merged upstream, please update your package to websocket@1.0.31
npm WARN deprecated @web3-js/scrypt-shim@0.1.0: This package is deprecated, for a pure JS implementation please use scrypt-js
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142

added 296 packages, and audited 349 packages in 55s
```

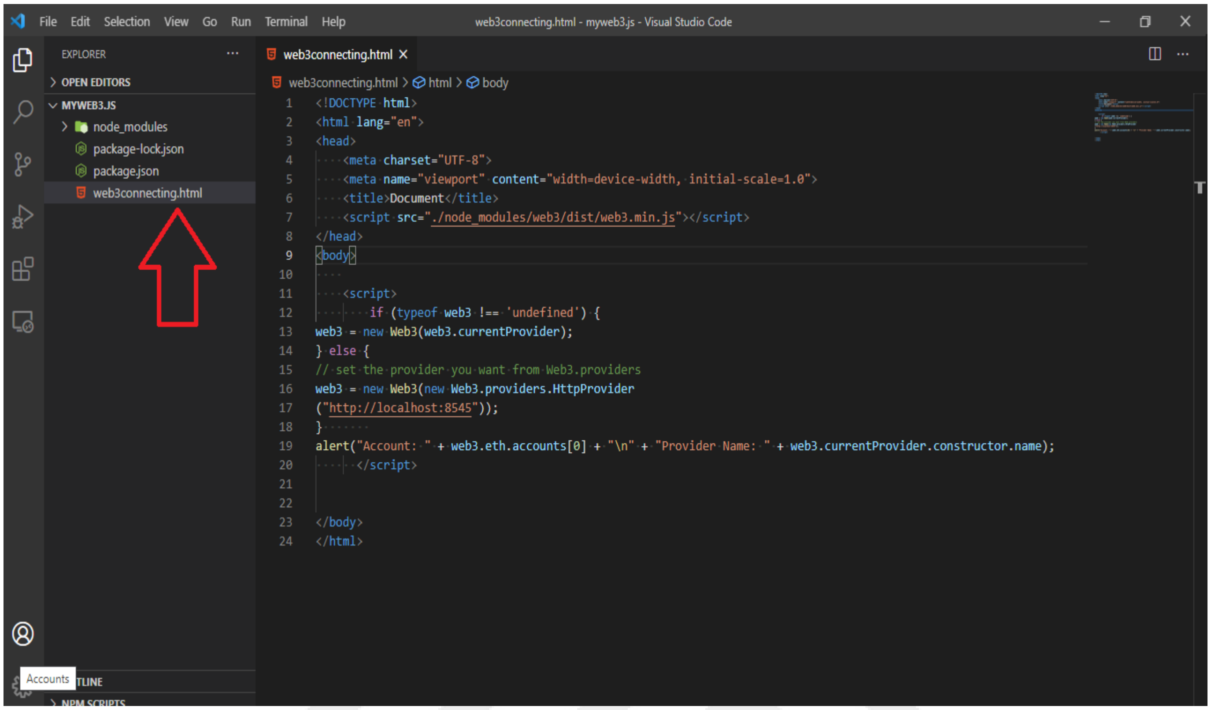
Figure 4.23: Installing web3.js



Name	Date modified	Type	Size
node_modules	2/6/2021 3:16 PM	File folder	
package	2/6/2021 3:16 PM	json	1 KB
package-lock	2/6/2021 3:16 PM	json	227 KB

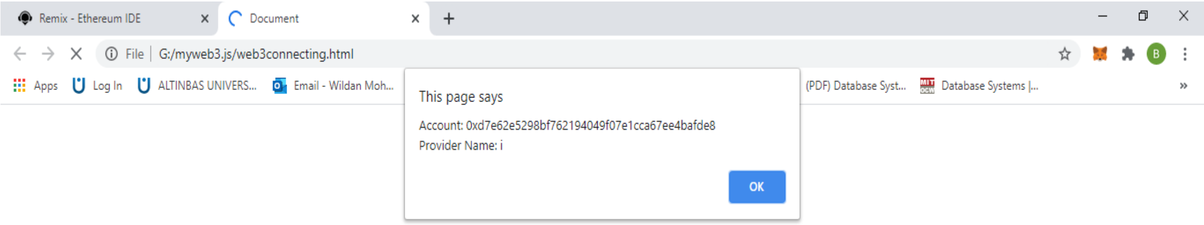
Figure 4.24: myweb3.js folder

To connect the web3.js with our private PoA ethereum blockchain network, we should install the visual studio code on this link https://code.visualstudio.com/download . As shown in figure 4.25, we open the myweb3.js folder in the visual studio code, creating a HTML file named (web3connecting.html) to write the code which enable us to make a connection between the web3.js and the private PoA ethereum blockchain network. Figure 4.26 shows that the connection has been done successfully.



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <script src="./node_modules/web3/dist/web3.min.js"></script>
</head>
<body>
    
    <script>
            if (typeof web3 !== 'undefined') {
web3 = new Web3(web3.currentProvider);
} else {
// set the provider you want from Web3.providers
web3 = new Web3(new Web3.providers.HttpProvider
("http://localhost:8545"));
}      
alert("Account: " + web3.eth.accounts[0] + "\n" + "Provider Name: " + web3.currentProvider.constructor.name);
         </script>


</body>
</html>
```

Figure 4.25: Connect the web3.js with the private PoA ethereum blockchain network



This page says

Account: 0xd7e62e5298bf762194049f07e1cca67ee4bafde8
Provider Name: i

Figure 4.26: The connection has been done successfully

5. RESULTS

In this thesis, a high-level programming language, solidity, along with a private PoA Ethereum blockchain network, has been used to build a secure smart contract system for the aim of achieving a secure network of peer to peer property and its main purpose of protection the private personal information. In addition, the web3.js API has been installed for a purpose of making a communication between the backend and frontend.

Our smart contract system has three main types of users: dentist, patient, and school. With the specific requirements of each entity and the method of interactions among them have been described in section 4.4. The smart contract system has five essential types of contracts that are called “the five types model” according to what they are, including: database contract, controller contracts, contract managing contract, application logic contract, and utilizing contract. According to this five type's model, our smart contract system has been built with eleven smart contracts that includes: ManageContractsEnabled, InfomationManagerEnabled, Manage contracts, InformationManager, ContractProvider, PermissionsDB, Permissions, PatientInformationDB, Patient, Dentist, and School, as explained in details in section 4.7.

6. CONCLUSION AND FUTURE ADVICE

6.1 CONCLUSION

However, the technology of blockchain has many implementations in healthcare and there are many papers about its healthcare application and their benefits, introducing this technology to the field of dental healthcare is still limited. Therefore, this thesis has introduced the blockchain technology to a special branch of dental practice, with the main goals of improving the quality of dental healthcare services and facilitating dentist's work.

Blockchains have gained its transparency and visibility since it has enhanced security, robustness and reliability of the distributed system. Multiple fields have experienced progress and improvement by researches depending on that technology, such as financial services, data analysis, and healthcare. The main features of blockchain technology that enable it to be rapidly growing with wide applications are: decentralization, data immutability, transparency, privacy, and distributed ledgers. However, the healthcare data records include confidential information of the patients that makes the blockchain systems sophisticated due to the high risk of a privacy violation.

The main aim of the thesis is introduction of the blockchain technology with the smart contracts system to become the solution for constructing applications when the privacy is priority, focusing mainly on the treatment plans, constructed on a system of smart contracts based on blockchains. The problem has been presented, why there is a need for digitalizing the treatment plans and why blockchain technology has been proposed for providing the solution. Then blockchains is explored as a newly progressing technology with the background has been demonstrated, since it is regarded as comparatively new technology with its system is relatively unfamiliar. After that, the design is suggested and explained in order to provide solutions to the reported problems. Additionally, a smart contracts system is built to confirm the method by which an implementation could be applied and proposed the guidelines by which the blockchain-based system has its design so that the determined requirements can be achieved. Finally, there is a discussion to be explained considering the different designs applicability in the blockchain according to the issue of privacy dealing fields of applications.

6.2 FUTURE ADVICE

With the current deficiency in blockchain application in dental healthcare field, there are great opportunities to introduce this technology in dentistry and developing its benefits to meet the practical needs. The dental healthcare fields are considered as fertile field for future studies and further development through using blockchain technology.

REFERENCES

[1] S. Y. L. Kwan, P. E. Petersen, C. M. Pine, and A. Borutta, “Health-promoting schools: an opportunity for oral health promotion,” *Bull. World Health Organ.*, vol. 83, no. 9, pp. 677–685, 2005, [Online]. Available: https://www.scielosp.org/article/bwho/2005.v83n9/677-685/en/.

[2] A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, and F. Wang, “Secure and Trustable Electronic Medical Records Sharing using Blockchain.,” *AMIA ... Annu. Symp. proceedings. AMIA Symp.*, vol. 2017, no. 6, pp. 650–659, Nov. 2017. doi: 10.1093/jamia/ocx068.

[3] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, “Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control,” *J. Med. Syst.*, vol. 40, no. 10, p. 218, Oct. 2016. doi: 10.1007/s10916-016-0574-6.

[4] C. P. Snow *et al.*, “Factom,” 2018, [Online]. Available: https://www.factom.com/.

[5] O. Williams-Grut, “Estonia is using the technology behind bitcoin to secure 1 million health records,” 2016. https://www.businessinsider.com/guardtime-estonian-health-records-industrial-blockchain-bitcoin-2016-3?r=UK&IR=T.

[6] K. Yip, “Blockchain & Alternative Payment Models,” p. 9, 2016, [Online]. Available: http://www.truevaluemetrics.org/DBpdfs/Technology/Blockchain/15-54-kyip_blockchainapms_080816.pdf.

[7] I. B. M. Global, B. Services, and P. Sector, “IBM Global Business Services Public Sector Team 6710 Rockledge Dr., Bethesda, MD 20817 August 8, 2016,” p. 12, 2016, [Online]. Available: https://www.healthit.gov/sites/default/files/8-31-blockchain-ibm_ideation-challenge_aug8.pdf.

[8] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, “MedRec: Using Blockchain for Medical Data Access and Permission Management,” in *2016 2nd International Conference on Open and Big Data (OBD),* Aug. 2016, pp. 25–30. doi: 10.1109/OBD.2016.11.

[9] D. Yaga, P. Mell, N. Roby, and K. Scarfone, “Blockchain technology overview,” Gaithersburg, MD, Oct. 2018. doi: 10.6028/NIST.IR.8202.

[10] Futurethinkers, “19 Industries The Blockchain Will Disrupt.” https://futurethinkers.org/industries-blockchain-disrupt/.

[11] H. Wang, Z. Zheng, S. Xie, H. N. Dai, and X. Chen, “Blockchain challenges and opportunities: a survey,” *Int. J. Web Grid Serv.*, vol. 14, no. 4, p. 352, 2018. doi: 10.1504/IJWGS.2018.10016848.

[12] V. Buterin, “Visions, Part 1: The Value of Blockchain Technology,” 2015. https://blog.ethereum.org/2015/04/13/visions-part-1-the-value-of-blockchain-technology/.

[13] C. Naden, “BLOCKCHAIN TECHNOLOGY SET TO GROW FURTHER WITH INTERNATIONAL STANDARDS IN PIPELINE,” 2017. https://www.iso.org/news/Ref2188.htm.

[14] G. Dimitropoulos, “The Law of Blockchain,” *SSRN Electron. J.*, no. March, 2020. doi: 10.2139/ssrn.3559970.

[15] F. ul Hassan *et al.*, “Blockchain And The Future of the Internet:A Comprehensive Review,” pp. 1–21, Feb. 2019, [Online]. Available: http://arxiv.org/abs/1904.00733.

[16] V. Lai and K. O’Day, “INTRODUCTION TO CRYPTOGRAPHY IN BLOCKCHAIN TECHNOLOGY,” 2018. https://crushcrypto.com/cryptography-in-blockchain/.

[17] A. Pfitzmann and M. Hansen, “A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management,” *Tech. Univ. Dresden*, pp. 1–98, 2010, [Online]. Available: http://dud.inf.tu-dresden.de/Anon_Terminology.shtml%5Cnhttp://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf.

[18] H. F. Atlam, A. Alenezi, M. O. Alassafi, and G. B. Wills, “Blockchain with Internet of Things: Benefits, Challenges, and Future Directions,” *Int. J. Intell. Syst. Appl.*, vol. 10, no. 6, pp. 40–48, Jun. 2018. doi: 10.5815/ijisa.2018.06.05.

[19] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” p. 9, 2008, [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[20] J. B. Rothnie *et al.*, “Introduction to a system for distributed databases (SDD-1),” *ACM Trans. Database Syst.*, vol. 5, no. 1, pp. 1–17, Mar. 1980. doi: 10.1145/320128.320129.

[21] R. C. Merkle, “SECRECY, AUTHENTICATION, AND PUBLIC KEY SYSTEMS,” *I NFORMAT I Syst. Lab. STANFORD Electron. Lab. Dep. Electr. Eng. STANFORD Univ. STANFORD, CA 94305*, p. 193, 1979.

[22] S. Haber and W. S. Stornetta, “How to Time-Stamp a Digital Document,” in *Advances in Cryptology-CRYPT0’ 90*, vol. 537 LNCS, Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 437–455. doi: 10.1007/3-540-38424-3_32.

[23] C. Dwork and M. Naor, “Pricing via Processing or Combatting Junk Mail,” in *Advances in Cryptology — CRYPTO’ 92*, vol. 740 LNCS, Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 139–147. doi: 10.1007/3-540-48071-4_10.

[24] R. C. Merkle, “Secure communications over insecure channels,” *Commun. ACM*, vol. 21, no. 4, pp. 294–299, Apr. 1978. doi: 10.1145/359460.359473.

[25] D. L. Chaum, “Computer systems established, maintained and trusted by mutually suspicious groups,” 1982.

[26] R. Cheng *et al.*, “Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contracts,” in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, Jun. 2019, pp. 185–200. doi: 10.1109/EuroSP.2019.00023.

[27] M. Brandenburger, C. Cachin, R. Kapitza, and A. Sorniotti, “Blockchain and Trusted Computing: Problems, Pitfalls, and a Solution for Hyperledger Fabric,” p. 13, May 2018, [Online]. Available: http://arxiv.org/abs/1805.08541.

[28] A. T. Sherman, F. Javani, H. Zhang, and E. Golaszewski, “On the Origins and Variations of Blockchain Technologies,” *IEEE Secur. Priv.*, vol. 17, no. 1, pp. 72–77, Jan. 2019. doi: 10.1109/MSEC.2019.2893730.

[29] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, “Everything You Wanted to Know About the Blockchain: Its Promise, Components, Processes, and Problems,” *IEEE Consum. Electron. Mag.*, vol. 7, no. 4, pp. 6–14, Jul. 2018. doi:

10.1109/MCE.2018.2816299.

[30] N. A. A. Farah, “Blockchain Technology : Classification, Opportunities, and Challenges,” *Int. Res. J. Eng. Technol.*, vol. 5, no. 5, pp. 3423–3426, 2018, [Online]. Available: https://www.irjet.net/archives/V5/i5/IRJET-V5I5659.pdf.

[31] S. S. Kolhe, “Blockchain based smart contracts for business process automation,” Eindhoven University of Technology, 2018.

[32] P. Raj, K. Saini, and C. Surianarayanan, *Blockchain Technology and Applications*, First edit. CRC Press, 2020.

[33] L. Franke, M. Schletz, and S. Salomo, “Designing a Blockchain Model for the Paris Agreement’s Carbon Market Mechanism,” *Sustainability*, vol. 12, no. 3, p. 1068, Feb. 2020. doi: 10.3390/su12031068.

[34] W. Wang *et al.*, “A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks,” *IEEE Access*, vol. 7, pp. 22328–22370, 2019. doi: 10.1109/ACCESS.2019.2896108.

[35] R. Zhang, R. Xue, and L. Liu, “Security and Privacy on Blockchain,” *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–34, Jul. 2019. doi: 10.1145/3316481.

[36] S. M. H. Bamakan, A. Motavali, and A. Babaei Bondarti, “A survey of blockchain consensus algorithms performance evaluation criteria,” *Expert Syst. Appl.*, vol. 154, p. 113385, Sep. 2020. doi: 10.1016/j.eswa.2020.113385.

[37] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, “A review on consensus algorithm of blockchain,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2017, vol. 2017-Janua, pp. 2567–2572. doi: 10.1109/SMC.2017.8123011.

[38] T. Ali Syed, A. Alzahrani, S. Jan, M. S. Siddiqui, A. Nadeem, and T. Alghamdi, “A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recommendations,” *IEEE Access*, vol. 7, pp. 176838–176869, 2019. doi: 10.1109/ACCESS.2019.2957660.

[39] 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),* Jun. 2017, pp. 557–564. doi: 10.1109/BigDataCongress.2017.85.

[40] Ismail and Materwala, “A Review of Blockchain Architecture and Consensus Protocols: Use Cases, Challenges, and Solutions,” *Symmetry (Basel).*, vol. 11, no. 10, p. 1198, Sep. 2019. doi: 10.3390/sym11101198.

[41] Daniel, “Blockchain Interoperability: Connecting Isolated Protocols,” 2018. https://komodoplatform.com/blockchain-interoperability/.

[42] F. Schuh and D. Larimer, “BITSHARES 2.0: GENERAL OVERVIEW,” p. 10, 2017, [Online]. Available: https://cryptorating.eu/whitepapers/BitShares/bitshares-general.pdf.

[43] E. Buchman, “Tendermint: Byzantine Fault Tolerance in the Age of Blockchains,” University of Guelph, 2016.

[44] M. Castro and B. Liskov, “Practical Byzantine Fault Tolerance,” *Lab. Comput. Sci. Massachusetts Inst. Technol.*, p. 14, 1999, [Online]. Available: http://pmg.csail.mit.edu/papers/osdi99.pdf.

[45] N. Chondros, K. Kokordelis, and M. Roussopoulos, “On the Practicality of `Practical’ Byzantine Fault Tolerance,” p. 13, Oct. 2011. doi: 10.1007/978-3-642-35170-9_22.

[46] G. CHRISTOFI, “Study of consensus protocols and improvement of the Delegated Byzantine Fault Tolerance (DBFT) algorithm,” university of politecnica, 2019.

[47] S. De Angelis, “Assessing Security and Performances of Consensus algorithms for Permissioned Blockchains,” p. 64, May 2018, [Online]. Available: http://arxiv.org/abs/1805.03490.

[48] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, “On Security Analysis of Proof-of-Elapsed-Time (PoET),” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics),* vol. 10616 LNCS, no. October, pp. 282–297, 2017. doi: 10.1007/978-3-319-69084-1_19.

[49] R. Houben and A. Snyers, “Cryptocurrencies and blockchain: Legal context and implications for financial crime, money laundering and tax evasion,” *Eur. Parliam.*, no. July, pp. 1–101, 2018, [Online]. Available: www.europarl.europa.eu/cmsdata/150761/TAX3 Study on cryptocurrencies and blockchain.pdf.

[50] D. Vujicic, D. Jagodic, and S. Randic, “Blockchain technology, bitcoin, and Ethereum: A brief overview,” in *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, Mar. 2018, vol. 2018-Janua, no. August, pp. 1–6. doi: 10.1109/INFOTEH.2018.8345547.

[51] P. McCorry, M. Möser, S. F. Shahandasti, and F. Hao, “Towards Bitcoin Payment Networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9722, 2016, pp. 57–76. doi: 10.1007/978-3-319-40253-6_4.

[52] V. Buterin, “A next-generation smart contract and decentralized application platform,” *Etherum*, no. January, pp. 1–36, 2014, [Online]. Available: http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf.

[53] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “LSB: A Lightweight Scalable Blockchain for IoT security and anonymity,” *J. Parallel Distrib. Comput.*, vol. 134, pp. 180–197, Dec. 2019. doi: 10.1016/j.jpdc.2019.08.005.

[54] C. Saraf and S. Sabadra, “Blockchain platforms: A compendium,” in *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, May 2018, no. May, pp. 1–6. doi: 10.1109/ICIRD.2018.8376323.

[55] V. Dhillon, D. Metcalf, and M. Hooper, “The Hyperledger Project,” in *Blockchain Enabled Applications*, Berkeley, CA: Apress, 2017, pp. 139–149. doi: 10.1007/978-1-4842-3081-7_10.

[56] S. Aggarwal and N. Kumar, “An Introduction to Hyperledger,” in *Advances in Computers*, 2020. doi: 10.1016/bs.adcom.2020.08.016.

[57] C. Cachin, S. Schubert, and M. Vukolić, “Non-determinism in Byzantine Fault-Tolerant

Replication," *Leibniz Int. Proc. Informatics, LIPIcs*, vol. 70, pp. 24.1-24.16, Mar. 2016. doi: 10.4230/LIPIcs.OPODIS.2016.24.

[58] K. Olson, M. Bowman, J. Mitchell, S. Amundson, D. Middleton, and C. Montgomery, "Sawtooth: An Introduction," no. January, pp. 1–7, 2018, [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2018/01/Hyperledger_Sawtooth_WhitePaper.pdf.

[59] L. Castaldo and V. Cinque, "Blockchain-Based Logging for the Cross-Border Exchange of eHealth Data in Europe," in *Communications in Computer and Information Science*, vol. 821, Springer International Publishing, 2018, pp. 46–56. doi: 10.1007/978-3-319-95189-8_5.

[60] L. Kan, Y. Wei, A. Hafiz Muhammad, W. Siyuan, L. C. Gao, and H. Kai, "A Multiple Blockchains Architecture on Inter-Blockchain Communication," in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, Jul. 2018, pp. 139–145. doi: 10.1109/QRS-C.2018.00037.

[61] G. Greenspan, "MultiChain Private Blockchain - White Paper," *White Pap.*, pp. 1–17, 2015, [Online]. Available: http://www.multichain.com/download/MultiChain-White-Paper.pdf.

[62] J. Bhosale and S. Mavale, "Volatility of select Crypto-currencies : A comparison of Bitcoin , Ethereum and Litecoin," *Annu. Res. J. SCMS, Pune*, vol. 6, no. March, pp. 132–141, 2018, [Online]. Available: https://www.scmspune.ac.in/journal/pdf/current/Paper 10 - Jaysing Bhosale.pdf.

[63] J. Göttfert, "Cointegration among cryptocurrencies," *Master Thesis*, 2019.

[64] N. Szabo, "Smart Contracts," 1994. https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html.

[65] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 11, pp. 2266–2277, Nov. 2019. doi: 10.1109/TSMC.2019.2895123.

[66] V. Y. Kemmoe, W. Stone, J. Kim, D. Kim, and J. Son, "Recent Advances in Smart Contracts: A Technical Overview and State of the Art," *IEEE Access*, vol. 8, pp. 117782–117801, 2020. doi: 10.1109/ACCESS.2020.3005020.

[67] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, "An Overview of Smart Contract: Architecture, Applications, and Future Trends," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, vol. 2018-June, no. Iv, pp. 108–113. doi: 10.1109/IVS.2018.8500488.

[68] Z. Zheng *et al.*, "An overview on smart contracts: Challenges, advances and platforms," *Futur. Gener. Comput. Syst.*, vol. 105, pp. 475–491, Apr. 2020. doi: 10.1016/j.future.2019.12.019.

[69] A. V. Tyurin, I. V. Tyuluandin, V. S. Maltsev, I. A. Kirilenko, and D. A. Berezun, "Overview of the Languages for Safe Smart Contract Programming," *Proc. Inst. Syst. Program. RAS*, vol. 31, no. 3, pp. 157–176, 2019. doi: 10.15514/ISPRAS-2019-31(3)-13.

[70] W. Gao, W. G. Hatcher, and W. Yu, "A Survey of Blockchain: Techniques, Applications, and Challenges," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, Jul. 2018, vol. 2018-July, no. i, pp. 1–11. doi: 10.1109/ICCCN.2018.8487348.

[71] C. Komalavalli, D. Saxena, and C. Laroiya, "Overview of Blockchain Technology Concepts," in *Handbook of Research on Blockchain Technology*, Elsevier, 2020, pp. 349–371. doi: 10.1016/B978-0-12-819816-2.00014-9.

[72] H. M. Hussien, S. M. Yasin, S. N. I. Udzir, A. A. Zaidan, and B. B. Zaidan, "A Systematic Review for Enabling of Develop a Blockchain Technology in Healthcare Application: Taxonomy, Substantially Analysis, Motivations, Challenges, Recommendations and Future Direction.," *J. Med. Syst.*, vol. 43, no. 10, p. 320, Sep. 2019. doi: 10.1007/s10916-019-1445-8.

[73] C. Agbo, Q. Mahmoud, and J. Eklund, "Blockchain Technology in Healthcare: A Systematic Review," *Healthcare*, vol. 7, no. 2, p. 56, Apr. 2019. doi: 10.3390/healthcare7020056.

[74] M. A. Cyran, “Blockchain as a Foundation for Sharing Healthcare Data,” *Blockchain Healthc. Today*, p. 6, Mar. 2018. doi: 10.30953/bhty.v1.13.

[75] M. Hölbl, M. Kompara, A. Kamišalić, and L. Nemec Zlatolas, “A Systematic Review of the Use of Blockchain in Healthcare,” *Symmetry (Basel).*, vol. 10, no. 10, p. 470, Oct. 2018. doi: 10.3390/sym10100470.

[76] K. Wust and A. Gervais, “Do you Need a Blockchain?,” in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, Jun. 2018, pp. 45–54. doi: 10.1109/CVCBT.2018.00011.

[77] A. Hasselgren, K. Kralevska, D. Gligoroski, S. A. Pedersen, and A. Faxvaag, “Blockchain in healthcare and health sciences—A scoping review,” *Int. J. Med. Inform.*, vol. 134, no. November 2019, p. 104040, Feb. 2020. doi: 10.1016/j.ijmedinf.2019.104040.

[78] T. Poongothai, K. Jayarajan, G. Rajeshkumar, and P. S. K. Patra, “BLOCKCHAIN TECHNOLOGY IN HEALTHCARE APPLICATIONS,” *J. Crit. Rev.*, vol. 8, no. 1 (SI), pp. 8701–8706, 2020. doi: 10.31838/jcr.07.19.978.

[79] I. Yaqoob, K. Salah, R. Jayaraman, and Y. Al-Hammadi, “Blockchain for healthcare data management: opportunities, challenges, and future recommendations,” *Neural Comput. Appl.*, no. November, Jan. 2021. doi: 10.1007/s00521-020-05519-w.

[80] A. Shahnaz, U. Qamar, and A. Khalid, “Using Blockchain for Electronic Health Records,” *IEEE Access*, vol. 7, pp. 147782–147795, 2019. doi: 10.1109/ACCESS.2019.2946373.

[81] D. Tith *et al.*, “Application of Blockchain to Maintaining Patient Records in Electronic Health Record for Enhanced Privacy, Scalability, and Availability,” *Healthc. Inform. Res.*, vol. 26, no. 1, p. 3, 2020. doi: 10.4258/hir.2020.26.1.3.

[82] Y. Pylypchuk, C. Johnson, J. Henry, and D. Ciricean, “Variation in Interoperability among U.S. Non-federal Acute Care Hospitals in 2017,” *Off. Natl. Coord. Heal. Inf. Technol.*, vol. 42, no. 4, pp. 1–15, 2018, [Online]. Available: https://www.healthit.gov/sites/default/files/page/2018-11/Interop variation_0.pdf.

[83] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, “Ancile: Privacy-preserving

framework for access control and interoperability of electronic health records using blockchain technology," *Sustain. Cities Soc.*, vol. 39, no. August 2017, pp. 283–297, May 2018. doi: 10.1016/j.scs.2018.02.014.

[84] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba, "Blockchain technology innovations," in *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*, Jun. 2017, no. 2016, pp. 137–141. doi: 10.1109/TEMSCON.2017.7998367.

[85] H. Li, L. Zhu, M. Shen, F. Gao, X. Tao, and S. Liu, "Blockchain-Based Data Preservation System for Medical Data," *J. Med. Syst.*, vol. 42, no. 8, p. 141, Aug. 2018. doi: 10.1007/s10916-018-0997-3.

[86] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, and J. He, "BlocHIE: A BLOCkchain-Based Platform for Healthcare Information Exchange," in *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, Jun. 2018, no. May 2019, pp. 49–56. doi: 10.1109/SMARTCOMP.2018.00073.

[87] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data," *Comput. Struct. Biotechnol. J.*, vol. 16, pp. 267–278, 2018. doi: 10.1016/j.csbj.2018.07.004.

[88] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "MedBlock: Efficient and Secure Medical Data Sharing Via Blockchain," *J. Med. Syst.*, vol. 42, no. 8, p. 136, Aug. 2018. doi: 10.1007/s10916-018-0993-7.

[89] M. Benchoufi, R. Porcher, and P. Ravaud, "Blockchain protocols in clinical trials: Transparency and traceability of consent," *F1000Research*, vol. 6, no. 1, p. 66, Jan. 2017. doi: 10.12688/f1000research.10531.1.

[90] E. Funk, J. Riddell, F. Ankel, and D. Cabrera, "Blockchain Technology," *Acad. Med.*, vol. 93, no. 12, pp. 1791–1794, Dec. 2018. doi: 10.1097/ACM.0000000000002326.

[91] T. Nugent, D. Upton, and M. Cimpoesu, "Improving data transparency in clinical trials using blockchain smart contracts," *F1000Research*, vol. 5, no. October, p. 2541, Oct. 2016. doi: 10.12688/f1000research.9756.1.

[92] F. Angeletti, I. Chatzigiannakis, and A. Vitaletti, “The role of blockchain and IoT in recruiting participants for digital clinical trials,” in *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sep. 2017, pp. 1–5. doi: 10.23919/SOFTCOM.2017.8115590.

[93] I. Radanović and R. Likić, “Opportunities for Use of Blockchain Technology in Medicine,” *Appl. Health Econ. Health Policy*, vol. 16, no. 5, pp. 583–590, Oct. 2018. doi: 10.1007/s40258-018-0412-8.

[94] G. Prisco, “The Blockchain For Healthcare: Gem Launches Gem Health Network With Philips Blockchain Lab,” 2016. https://bitcoinmagazine.com/articles/the-blockchain-for-heathcare-gem-launches-gem-health-network-with-philips-blockchain-lab-1461674938.

[95] T.-T. Kuo, H.-E. Kim, and L. Ohno-Machado, “Blockchain distributed ledger technologies for biomedical and health care applications,” *J. Am. Med. Informatics Assoc.*, vol. 24, no. 6, pp. 1211–1220, Nov. 2017. doi: 10.1093/jamia/ocx068.

[96] S. Meunier, “No TitleBlockchain technology — a very special kind of Distributed Database,” 2016. https://medium.com/@sbmeunier/blockchain-technology-a-very-special-kind-of-distributed-database-e63d00781118.

[97] “Whitepaper: BigchainDB 2.0 The Blockchain Database,” no. May, pp. 1–14, 2018, [Online]. Available: https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf.

[98] L. Martin, “Blockchain vs. relational database: Which is right for your application?,” 2017. https://techbeacon.com/security/blockchain-vs-relational-database-which-right-your-application.

[99] J.-T. Lorenz, B. Münstermann, M. Higginson, P. B. Olesen, N. Bohlken, and V. Ricciardi, “Blockchain in insurance – opportunity or threat?,” *McKinsey Co.*, no. July, pp. 1–9, 2016, [Online]. Available: http://www.mckinsey.com/industries/financial-services/our-insights/blockchain-in-insurance-opportunity-or-threat.

[100] Q. H. Dang, “Secure Hash Standard,” Gaithersburg, MD, Jul. 2015. doi: 10.6028/NIST.FIPS.180-4.

[101] C. F. Kerry and P. D. Gallagher, “Digital Signature Standard (DSS),” John Wiley & Sons, Inc., Gaithersburg, MD, Jul. 2002. doi: 10.6028/NIST.FIPS.186-4.

[102] M. Macdonald, L. Liu-Thorrold, and R. Julien, “The Blockchain: A Comparison of Platforms and Their Uses Beyond Bitcoin,” *Work. Pap.*, no. February, pp. 1–18, 2017. doi: 10.13140/RG.2.2.23274.52164.

[103] Ethereum, “Ethereum Project,” 2016. https://ethereum.org/en/.

[104] S. Omohundro, “Cryptocurrencies, smart contracts, and artificial intelligence,” *AI Matters*, vol. 1, no. 2, pp. 19–21, Dec. 2014. doi: 10.1145/2685328.2685334.

[105] Ethereum, “Frequently Asked Questions,” 2015. https://ethdocs.org/en/latest/frequently-asked-questions/frequently-asked-questions.html.

[106] G. Prisco, “Blockstream Moves Ahead With Sidechain Elements, The First Implementation Of Sidechains,” 2015. https://bitcoinmagazine.com/articles/blockstream-moves-ahead-sidechain-elements-first-implementation-sidechains-1433883105.

[107] S. Davenport and R. Ford, “SGX: the good, the bad and the downright ugly,” *Virus Bulletin*, 2014. https://www.virusbulletin.com/virusbulletin/2014/01/sgx-good-bad-and-downright-ugly.

[108] Eris, “The Blockchain Client for Information Age Organizations,” 2016. https://erisindustries.com/components/erisdb.html.

[109] “IBM Blockchain,” 2016. https://www.ibm.com/blockchain.

[110] G. Maxwell, “Bringing New Elements to Bitcoin with Sidechains,” *2015*. https://diyhpl.us/wiki/transcripts/gmaxwell-sidechains-elements/.

[111] “Signature Covers Value.” https://elementsproject.org/elements/signature-covers-value/.

[112] “Build Exemplary Smart Contracts with Solidity Blockchain Development.” https://www.qsstechnosoft.com/technologies/solidity-blockchain-development.

[113] “The Five Types Model,” 2018. https://github.com/monax/legacy-

docs/blob/master/solidity/solidity_1_the_five_types_model.md.

[114] J. Best, “Application Programming Interface (API),” in *Breaking Digital Gridlock*, Hoboken, New Jersey: John Wiley & Sons, Inc., 2018, pp. 71–86. doi: 10.1002/9781119421900.ch5.

[115] D. Mohanty, *Ethereum for architects and developers with case studies and code samples in solidity*. 2018. doi: 10.1007/9781484240755.

[116] S. Pareek, A. Upadhyay, S. Doulani, S. Tyagi, and A. Varma, “E-Voting Using Ethereum Blockchain,” *Int. J. Res. Trends Innov.*, vol. 3, no. 11, pp. 30–34, 2018.

[117] “Node.js.” https://www.w3schools.com/nodejs/default.asp.