Mohammed V University - Rabat
National School of Computer Science
and Systems Analysis

Major :

Ingénierie en Data Science and IOT

End Year Project

Subject :

Generic Patient-Centered Blockchain Based Electronic Health Record (EHR) Management System

---

Supervised by :

Pr. Jamal EL HACHMI

Defensed By :

BOUZID Abdelfattah

Defensed 19/01/2024 in front of the Jury Composed of :

Mr. Abdellatif KOBBANE

Mr. Jamal EL HACHMI

Academic Year 2024-2025

Dedication

“

We dedicate this humble work with sincerity and pride : To our dear parents, our families, our teachers, our friends and to all the people who have supported us along our journey.

It is creditable to your love, your encouragement, the trust you have given us that we are able to accomplish this work today. I hope you will find in this work the testimony of our deep gratitude. No dedication can express our feelings towards you for your moral support.

Thank you for all the moments we spend with you and for the help you have offered us.

Thank you...

”

BOUZID Abdelfattah

Acknowledgements

Before any development on this academic experience, it seems appropriate to start by thanking those who taught us a lot during this semester, and even those who were kind enough to make this project a very fruitful experience. Regarding this work, I would like to express my deep gratitude to my supervised **Pr.Jamal EL HACHMI** for bringing all the necessary help to achieve the project goals .

Please find here the expression of our most profound respect.

Abstract

The current challenges in accessing healthcare services and sharing electronic healthcare records (EHR) due to confidentiality concerns. It emphasizes the need for a secure and reliable infrastructure to facilitate the exchange and storage of medical data efficiently,this research introduces a Patient-Centered Blockchain-Based EHR Management (PCEHRM) system using Ethereum blockchain and IPFS that complements Ethereum by providing decentralized file storage for patient health records, including large files. This ensures data availability and resilience through distribution across a node network, reducing the risk of data loss and enhancing accessibility. This system allows patients to manage their records across multiple stakeholders, ensuring patient privacy and control without centralized infrastructure. The proposed system incorporates an Ethereum smart contract for secure access control and demonstrates its functionality through a web-based interface. The study concludes that the proposed strategy is both efficient and practical, addressing the challenges associated with secure and decentralized access to healthcare data.

Keywords : patient-centered ; blockchain ; privacy ; health record ;Patient-Centered Electronic Health Record Management (PCEHRM) ; InterPlanetary File System(IPFS).

RÉSUMÉ

Les défis actuels liés à l'accès aux services de santé et au partage des dossiers électroniques de santé (DSE) en raison de préoccupations concernant la confidentialité soulignent la nécessité d'une infrastructure sécurisée et fiable pour faciliter l'échange et le stockage efficaces des données médicales. Cette recherche présente un système de gestion des dossiers de santé électroniques axé sur le patient et basé sur la blockchain (PCEHRM) utilisant la blockchain Ethereum et IPFS, qui complète Ethereum en fournissant un stockage de fichiers décentralisé pour les dossiers de santé des patients, y compris les fichiers volumineux. Cela garantit la disponibilité et la résilience des données grâce à une distribution sur un réseau de nœuds, réduisant le risque de perte de données et améliorant l'accessibilité. Ce système permet aux patients de gérer leurs dossiers avec plusieurs parties prenantes, assurant la confidentialité et le contrôle du patient sans infrastructure centralisée. Le système proposé intègre un contrat intelligent Ethereum pour un contrôle d'accès sécurisé et démontre sa fonctionnalité via une interface web. L'étude conclut que la stratégie proposée est à la fois efficace et pratique, répondant aux défis liés à l'accès sécurisé et décentralisé aux données de santé.

Mot clé : blockchain ; confidentialité ; dossier médical ; Gestion des dossiers médicaux électroniques centrée et basée sur le patient (PCEHRM) ; Système de fichiers interplanétaires (IPFS).

List of Acronyms

IOT	*Internet of Things*
IPFS	*InterPlanetary File System*
NFT	*Non-fungible Token*
EHR	*Electronic healthcare record*
HR	*Health records*
EMR	*Electronic medical records*
PHRs	*Personal health records*
P2P	*Peer to peer*
PCEHRM	*Patient Centered Electronic Health Record Management.*

List of Figures

Table of Contents

General Introduction

Health information, such as electronic health records (EHRs) and clinical images, is presently stored in centralized cloud databases, presenting security challenges due to cyberattacks and compromising EHR privacy. Inconsistent standards among stakeholders hinder the efficient sharing of health information. The inadvertent deletion of patient records from hospital databases is a noteworthy concern, underscoring the necessity for robust access controls in new systems. Patients currently have limited control over their health records managed by service providers, and the escalating volume of healthcare data raises apprehensions regarding security and scalability.
Regarding my contribution, I aimed to design a compatible architecture tailored to Moroccan use cases, considering the healthcare ecosystem, the status of various health systems, and the evolution of information technology in Morocco. Additionally, I incorporated these considerations into the implementation phase.

The remaining chapters of this thesis are organized as follows :
"Project Context" provides a detailed overview of the projet context and the fundamentals of blockchain technology, with a particular focus on the blockchain architecture,components and functionalities.A comparative study of various blockchain technology is also highlighted and a various application areas are mentioned in this chapter .
"State of the Art" we survey the state of the art that focuses on background information about blockchain technology and Blockchain used in the E-Health like EHR (Eelectronic Health Record), literature review and detail of the theoretical and practical concepts.
"Proposed Model"presents a overview of my conceptual scenario of the proposed system and describe the proposed approach as a set of software used to prove the concept of my work that is based on Blockchain technology by describing the details of the main functionalities.
"Realisation" this chapter presents the implementation of my proposed architecture by describing the different stages of setting up a local private blockchain and how to deploy a smart contract with an example of dApp(decentralized application) that manage a patients and doctors records a through blockchain network.

Chapitre 1

CONTEXT PROJECT

1.1 Introduction

Electronic health records (EHRs) are a cornerstone of e-health, enabling the collection, storage, and exchange of patient health information in an electronic format. EHRs offer a range of benefits for both patients and healthcare providers, contributing to improved patient care, reduced medical errors, enhanced communication, and better population health management. This chapter presents the general context as well as the main objectives of the project, a presentation of specifications will be in the first place, and then a Blockchain technology global overview is overcoming in the second part .

1.2 Specifications

In traditional healthcare data management using centralized cloud databases, various patient information, including electronic health records (EHRs) and clinical images, is consolidated. However, this approach is susceptible to cyberattacks, compromising data security. Challenges in standardization hinder efficient health information sharing, and the deletion of a patient's EHR from a central database can lead to permanent data loss. To address these issues, the proposed project focuses on developing a tamper-proof Generic Patient-Centered Blockchain-Based EHR Management system. This system aims to enhance security, scalability, and patient control over health records by leveraging blockchain technology for secure storage and sharing of EHR data [20] .

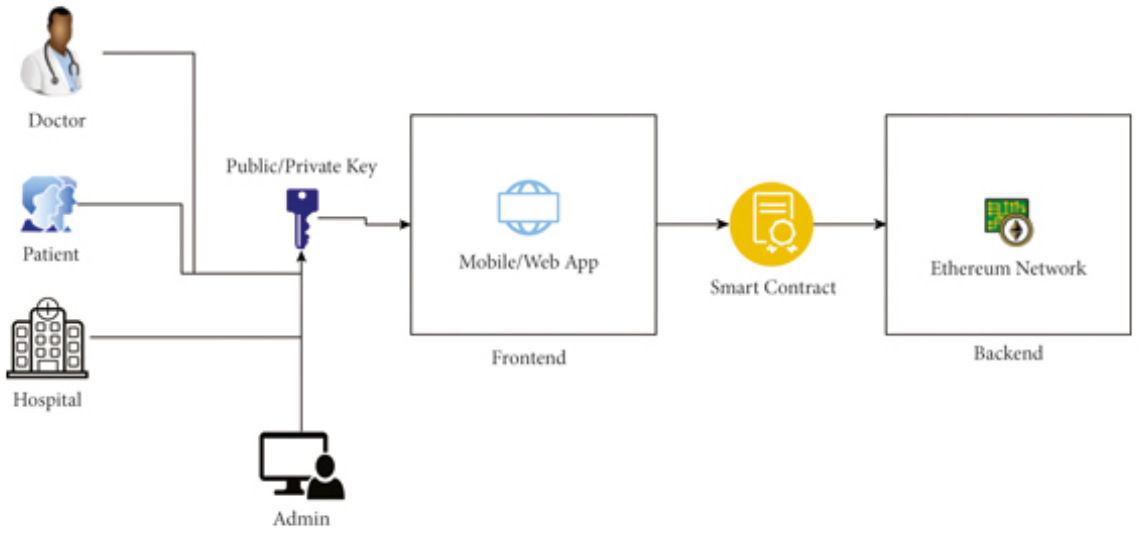


Fig. 1.1 : Overview of the current system.

First, this project mainly has to propose a simple implementation of Generic Patient-Centered Blockchain Based EHR Management to guarantee secure storage of EHR data and secure sharing of this data with Blockchain systems in terms of functionality and robustness. The tasks will be divided into two stages :

- Study the state of the art around Generic Patient-Centered Blockchain Based EHR Management in MOROCCO.
- Design and implementation of a simple Generic Patient-Centered Blockchain Based

EHR Management.

1.3 Blockchain Overview

Blockchain is the process that allows transactions to be verified by a group of unreliable actors.it is made up of blocks, each block contain the record of all the exchanges made between users at a given time. These different blocks provide the history of all transactions since it's creation and allow everyone to check the accuracy of the data exchanged. A simple analogy for understanding blockchain technology is a Google Doc. When we create a document and share it with a group of people, the document is distributed instead of copied or transferred. This creates a decentralized distribution chain that gives everyone access to the document at the same time. No one is locked out awaiting changes from another party, while all modifications to the doc are being recorded in realtime, making changes completely transparent. Blockchain referred to as Distributed Ledger Technology, means that makes the history of any digital asset unalterable and transparent through the use of decentralization and cryptographic hashing [1] .

1.4 History of Blockchain

Blockchain is a new technology, it already boasts a rich and interesting history. The following is a brief timeline of some of the most important and notable events in the development of blockchain.

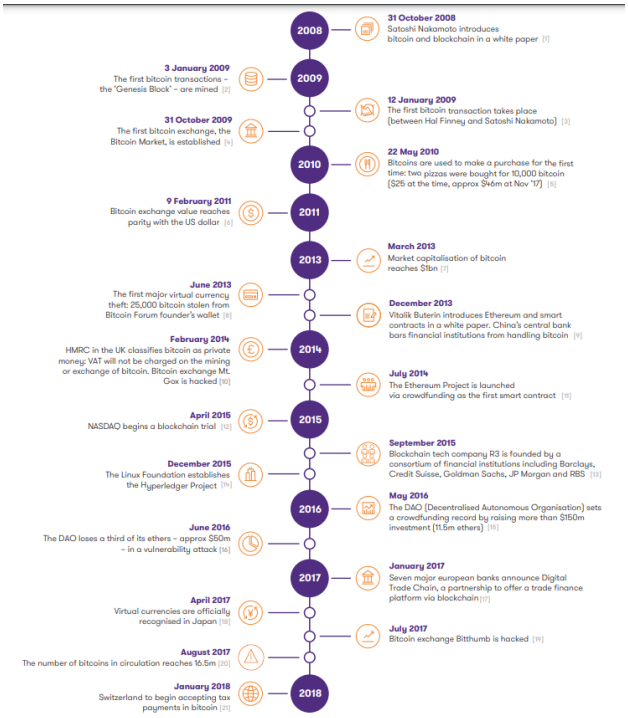


Fig. 1.2 : The blockchain timeline From inception to maturity

Blockchain technology is a form of distributed ledger technology.it's a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain. Each block in the chain contains a number of transactions,

and every time a new transaction occurs on the blockchain, a record of that transaction is added to every participant's ledger. The decentralised database managed by multiple participants is known as Distributed Ledger Technology. Blockchain is a type of Distributed Ledger Technology in which transactions are recorded with cryptographic signature called a hash [2] .

1.5 Distributed Ledger Technology

Distributed ledger technology refers specifically to the technological infrastructure and protocols that allow the simultaneous access, validation and updating of records that characterizes distributed ledgers. It operates on a computer network spread over multiple entities or locations. DLT uses cryptography to secure store data, cryptographic signatures and keys to allow access only to authorized users. A distributed ledger's nodes process and validate each item, resulting in a record of each item and agreement on its validity. Static data, such as a registry, and dynamic data, such as financial transactions, can both be recorded in a distributed ledger. Blockchain is a well-known example of a distributed ledger technology

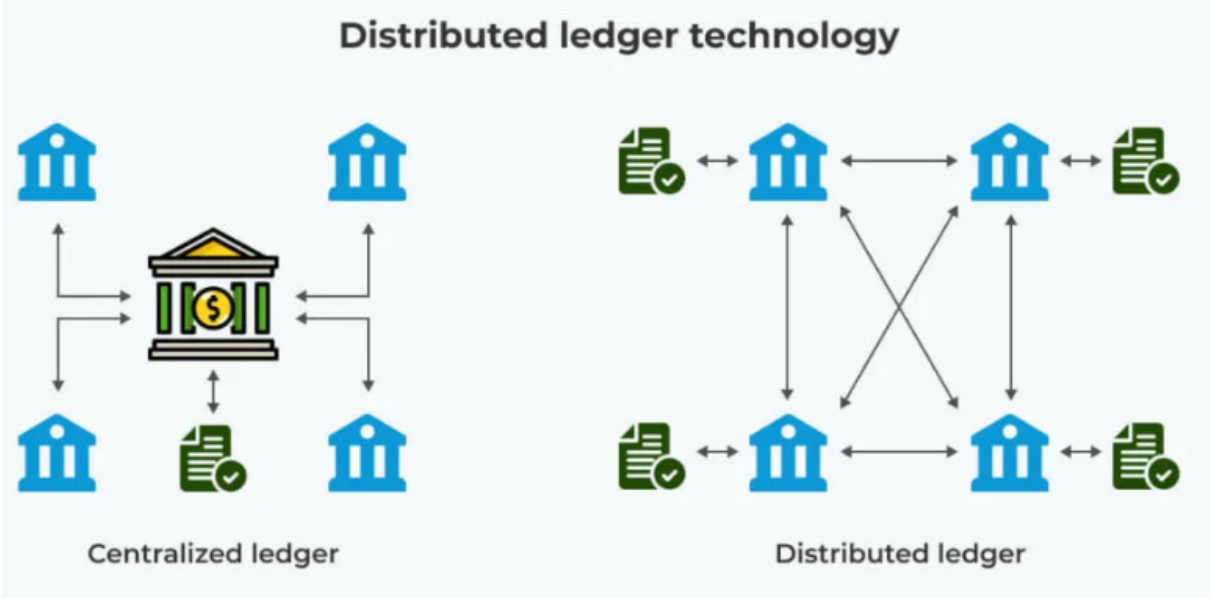


Fig. 1.3 : Distributed ledger technology

5Properties of Distributed Ledger Technology The concept of the ledger in distributed ledger technology (DLT) came into existence before Bitcoin and blockchain technology. The first DLT systems appeared in 1982, and the first mention of the term "blockchain" was in 1991. Unclear terminology and fuzzy boundaries over the time between blockchain and DLT have resulted in 'DLT' evolving into an umbrella term used to designate a variety of loosely related concepts in the blockchain.
Distributed Ledger Technology system needs to be capable of ensuring the following properties, either in the existing system or with minimal changes to the system [3] .

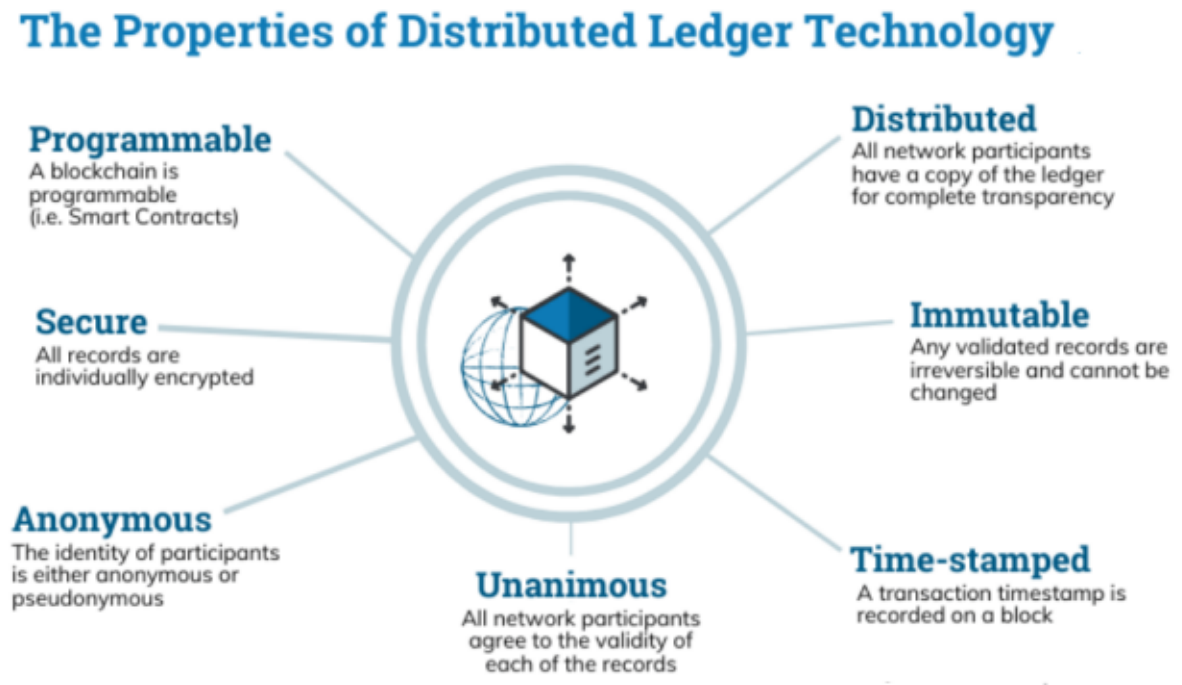


FIG. 1.4 : Distributed ledger technology

Distributed ledger benefits

Because it eliminates the need for a central authority or mediator, distributed ledger technology has the potential to speed up transactions. Similarly, DLT has the potential to lower transaction costs. Running the highly decentralized verification process and distributing copies of the ledger take substantial computing resources, which has been shown to hurt the performance of DLT in certain networking environments compared to centralized ledgers [4] .

The technology has already demonstrated its ability in to bring benefits to users,including the following :

- Increased visibility into and transparency of data contributed to the ledger.
- Lower operational costs thanks to the elimination of a central authority.
- Faster transaction speeds because there's no lag in updates to ledgers.
- Greatly reduced risks of fraudulent activity, tampering and manipulation.
- Increased reliability and resiliency because there's no longer a central system that creates the potential for a single point of failure.
- Significantly higher levels of security

1.6 Blockchain infrastructures

According to Melanie Swan, founder of the Blockchain Science Institute, blockchain technology has experienced 4 phase [5] :

- **The blockchain 1.0 :** Phase of multi-technology portfolio innovation represented by Bitcoin.
- **The blockchain 2.0 :** Phase represented by Ethereum, which is transferred by digital

assets.

- **The blockchain 3.0 :** Phase Describe the attempts to fix the current problems in the blockchain industry – specifically, issues regarding scalability, interoperability, and privacy.
- **The blockchain 4.0 :** New generation of blockchain technology. Deliver blockchain as a business-usable environment for developing and running applications,bringing the technology fully mainstream.

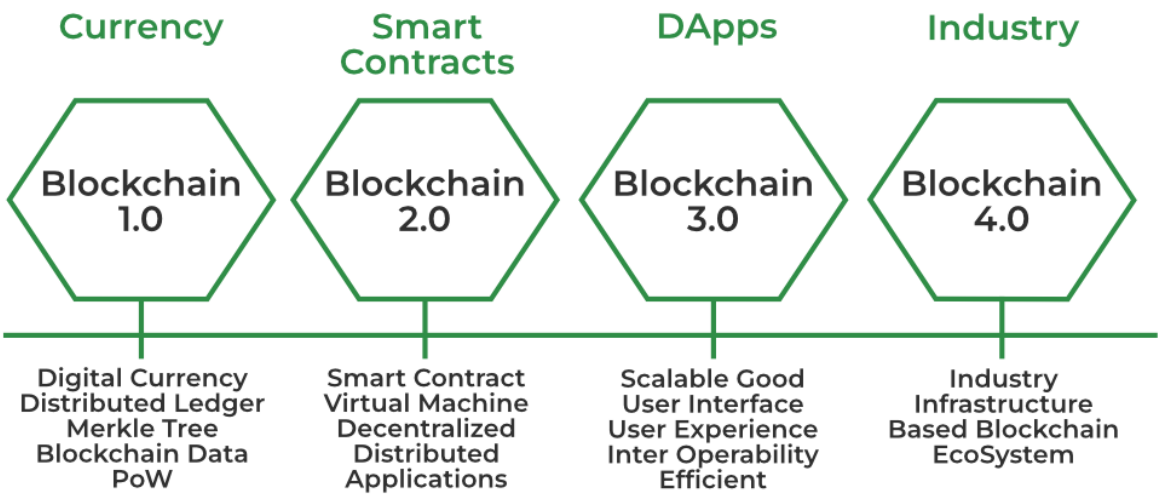


Fig. 1.5 : Phases of Evolution of Blockchain

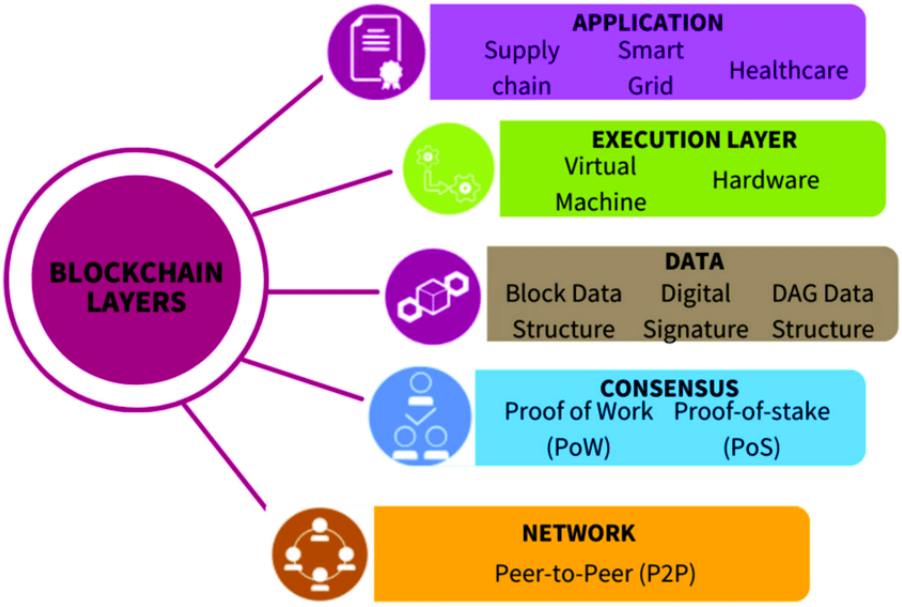


Fig. 1.6 : Blockchain Layers .

The blockchain platform can be divided into five levels :

Network Layer

The first layer in the multi-layered Distributed Ledger Technology (DLT) stack is the network layer, functioning as a decentralized peer-to-peer network. Participants are categorized into lightweight and full nodes, with full nodes storing a complete ledger, handling mining and validation, while lightweight nodes supplement by storing block headers and issuing transactions. This layer plays a crucial role in peer discovery, transactions, and block propagation, impacting DLT performance based on factors like blockchain size, peer discovery speed, and network delays. The network layer is fundamental for the efficient operation of the DLT system.

Consensus Layer

The consensus layer is crucial in DLT systems, aiming to achieve agreement among all nodes. Common consensus algorithms include proof-based methods like Proof of Work (PoW), as seen in Bitcoin, Ethereum, and Dogecoin, but it involves significant computing resource waste. Proof of Stake (PoS) and Proof of Authority (PoA) are alternative proof-based algorithms. Practical Byzantine Fault Tolerance (PBFT) is another algorithm, designed to handle malicious replicas. PBFT operates in three phases - pre-prepare, prepare, and commit - ensuring ordered requests and safety, relying on a fault tolerance formula (n−1)/3. Hyperledger Fabric is an example implementing PBFT consensus.

Data Layer

The data layer of the blockchain encompasses the foundational technology, including data block and chain structure, hash functions, Merkle tree, asymmetric public key data encryption, and timestamp technology. Despite the complexity, the primary function is data storage, with data blocks forming chains accessible by any full node. The Merkle tree structure ensures data security by recording transactions, generating hashes stored as a Merkle root, making tampering extremely difficult due to the decentralized nature of blockchain. However, the energy-demanding and slow processing nature of the Merkle tree structure is a notable drawback.

Execution Layer

The execution layer of the blockchain incorporates runtime environments like virtual machines (VMs), containers, and compilers on nodes. It facilitates the implementation of smart contracts, essential for establishing trust. Smart contracts operate on local VMs in each network node, ensuring mutual consent among non-trusting parties by collecting self-executing computer instructions. However, a drawback is the inefficient use of computing resources, particularly due to aborted transactions.

Application Layer

The application layer, atop the blockchain network, connects decentralized applications with the technology. Beyond cryptocurrencies, it includes smart contracts and diverse applications. Smart contracts, crucial in cryptocurrency, enable, verify, and enforce contract execution. In the IoT realm, the application layer extends blockchain use to smart cars, healthcare, farming, and cities [18] .

In blockchain, each block's header contains the hash value of the previous block's information, allowing users to verify integrity by comparing calculated and stored hash values. Hash functions also generate public-private key pairs. Hash pointers, incorporating data and password hashes, serve to verify data integrity in addition to retrieving information. The blockchain is essentially a list of hash pointers, connected by hash values, verifying data integrity and detecting changes in block information. This structure ensures the tamper-proof nature of the information within each block.

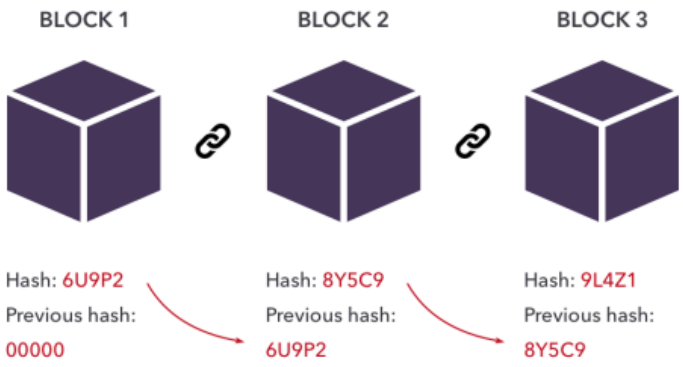


Fig. 1.7 : blockchain linked blocks.

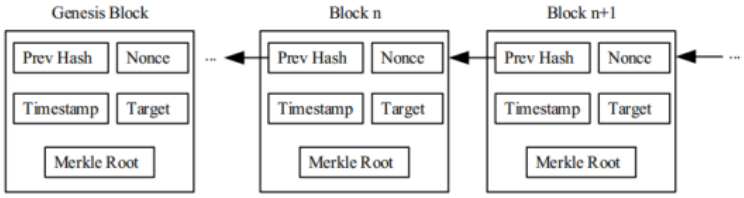


Fig. 1.8 : blockchain structure.

The blocks in blockchain hold all the data information of the whole network, mainly composed of the block header containing metadata and the block body containing all transaction data. The block header encapsulates the previous block hash, the current block's difficulty target, and the current block solution random number, Merkle root, and timestamp. The block body contains a list of transactions for storing transaction information[19] .

- **Prev Hash :** The block hash is a key segment of the blockchain. This field is the hash value of the data information of the previous block, and all the blocks on the chain are sequentially connected.
- **Nonce :** The header information of each data block contains a random number, and the initial value is 0. The node running the bitcoin mining machine continuously performs a SHA256 operation on the overall data of the block.
- **Timestamp :** The blockchain technique requires that the node must have a timestamp in the current data block header to indicate the write time of the block data. The blocks on the main chain are arranged in chronological order. The timestamp can be used as a proof of the existence of block data, helping to form a blockchain database that is not tamperable and unforgeable .
- **Merkle Root :** The Merkle Tree, devised by cryptographer Merkle, ensures efficient verification of extensive data integrity. In its standard structure, it incorporates the block's transaction database, the root hash of the block header, and branching elements culminating in the root hash. The Merkle tree operation entails grouping block data, inserting new hash values, and constructing the tree until the final root hash serves as the Merkle root in the block header. Bitcoin adopts a double SHA256 hash function, employing two SHA256 operations on the original data to achieve uniform storage and identification.
- **Transaction list :** The transaction list contains a lot of details of the transaction record, including the time of each transaction, transaction number, bitcoin amount,payer and other information. In the data block, each bitcoin is written and received together, so each bitcoin can be traced back.

1.7 Types of blockchains

A blockchain is a data structure that makes it possible to create a digital ledger of data and share it among a network of independent parties.
There are many different types of blockchains :

Public blockchains

Public blockchains, such as Bitcoin, are large distributed networks that are run through a native token. They're open for anyone to participate at any level and have open-source code that their community maintains

Permissioned blockchains

Permissioned blockchains, such as Ripple, control roles that individuals can play within the network. They're still large and distributed systems that use a native token. Their core code may or may not be open source.

Private blockchains

Private blockchains tend to be smaller and do not utilize a token. Their membership is closely controlled. These types of blockchains are favored by consortiums that have trusted members and trade confidential information.

All three types of blockchains use cryptography to allow each participant on any given network to manage the ledger in a secure way without the need for a central authority to enforce the rules. The removal of central authority from database structure is one of the most important and powerful aspects of blockchains [6] .

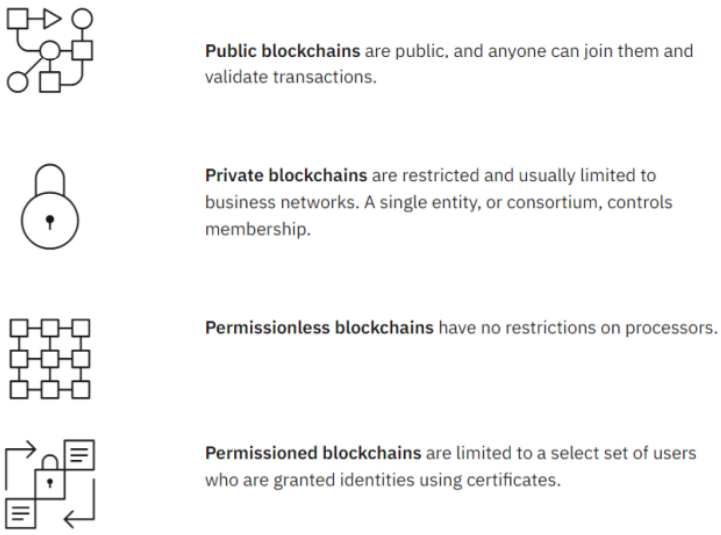


FIG. 1.9 : Types of blockchain.

1.8 The Structure of Blockchains

The main features of blockchain technology can be summarized as follows :

Decentralization

Instead of relying on a single trusted entity, trust is spread across multiple or all participants, depending on the agreed upon consensus algorithm. This does not only mean that multiple copies of a data item are stored on all nodes, but also that the integrity of the data is governed by many de- centralized parties.

Immutability

The immutability property of a blockchain refers to the fact that any data once written on the blockchain cannot be changed. To understand immutability, consider sending email as an example. Once you send an email to a bunch of people, you cannot take it back. In order to find a way around, you'll have to ask all the recipients to delete your email which is pretty tedious. This is how immutability works.
Once the data has been processed, it cannot be altered or changed. In case of thecblockchain, if you try to change the data of one block, you'll have to change the entire blockchain following it as each block stores the hash of its preceding block. Change in

one hash will lead to change in all the following hashes. It is extremely complicated for someone to change all the hashes as it requires a lot of computational power to do so. Hence, the data stored in a blockchain is non-susceptible to alterations or hacker attacks due to immutability[3] .

Limited privacy

All data in the blockchain is publicly visible to all participants. Private or permissioned blockchains limit the range of disclosure. However, they do not cryptographically protect the data. In order to achieve privacy, additional layers,such as zero knowledge proofs or a commitment scheme are required .
The blockchain networks are built from three major components :

- **A cryptographic keypair** public key + private key that is stored in a blockchain wallet) Allow a secure digital identity reference.The keypair helps ensure that Users exchanging data without exposing private details. By signing transactions with private key and also place an "ownership stamp"on it, meaning the transaction can be traced back to you if needed.

- **A decentralized P2P network** Instead of a central authority, a community of users decides whether your transaction is valid and can be added to the blockchain.The community uses mathematical verification to evaluate the history of the individual blocks that are proposed to be added and the "sender" signature validity.Once enough users verify that your transaction is valid, it is processed and recorded on the blockchain.

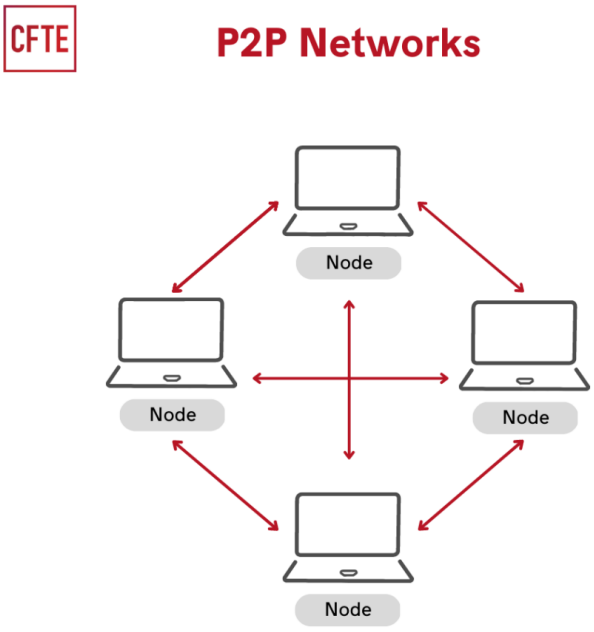


FIG. 1.10 : P2P network.

- **The network servicing protocol** The block, packed with transactional data, digital signatures and a timestamp, is broadcasted to the network's participants. The block verification process requires tremendous computing power. Public blockchains encourage the community to service the network by offering a reward for their effort
– cryptocurrencies such as Bitcoin or Ethereum [7].

Fundamental trust mechanism

Unlike in centralized systems where some administrator manages database and makes the decision of file storage and update. In decentralized systems Fundamental trust mechanism also known as consensus mechanism is used to make the nods agree upon storage of data. There are many consensus mechanisms in existing blockchain technology but four of them are considered major in use [1].

Consensus

Blockchain consensus is a protocols that ensure synchronization between all nodes in the network. Each consensus aims to answer a specific question : how can we ensure the authenticity of each transaction ? Anyone can submit information and decide to store it on a blockchain. It is therefore essential to be able to review this information and decide by consensus whether it is possible to add it to the network or not.

Consensus means that all nodes in the network must agree on an identical version of the blockchain. Somehow, the consensus mechanism of a blockchain is an internal and automatic audit of its network. This protocol is therefore essential and has two functions :
• It allows the blockchain to be updated while ensuring that every block in the chain is valid . The people participating in the validation of the blocks (called the "nodes" of the network) must have an incentive to become involved in the security of the network.
• It prevents a single entity from controlling the entire network and thus guarantees its decentralization.

Proof of Work (PoW)

Proof-of-Work,is the original consensus algorithm in a Blockchain network.
To prove the credibility of data in blocks, this algorithm is used to confirm transactions and produce new blocks to the chain. With PoW, miners compete against each other to complete transactions on the network and get rewarded.
Proof of Work is the mechanism that allows the decentralized network to come to consensus, or agree on things like account balances and the order of transactions.
This mechanism uses the method to solve the puzzle. When a node wants to create a block it must resolve a puzzle. Upon resolving the puzzle successfully a new block is created and broadcasted to other nodes to achieve the consensus.
The process of verifying the transactions in the block to be added, organizing these transactions in a chronological order in the block and announcing the newly mined block to the entire network does not take much energy and time. The energy consuming part is solving the 'hard mathematical problem' to link the new block to the last block in the valid blockchain [3] .

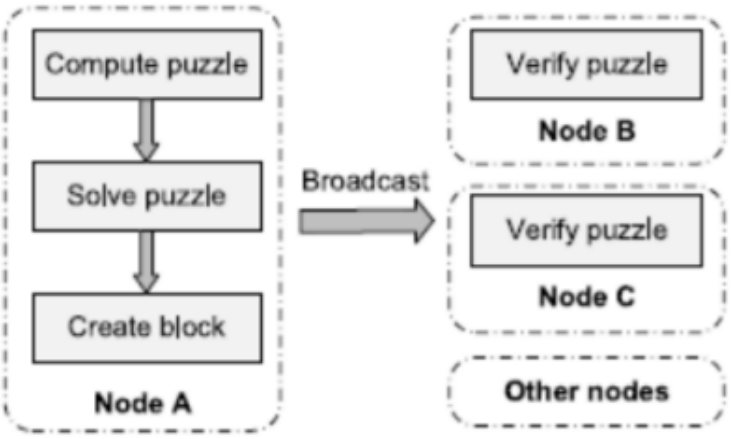


FIG. 1.11 : PoW consensus Mechanism.

Mining

In the context of blockchain technology, is the process of adding transactions to the large distributed public ledger of existing transactions, known as the blockchain. Blockchain mining involves adding transactions to the existing blockchain ledger of transactions distributed among all users of a blockchain. While mining is mostly associated with bitcoin, other technologies using a blockchain employ mining as well.**Mining** involves creating a hash of a block of transactions that cannot be easily forged, protecting the integrity of the entire blockchain without the need for a central system. Mining is typically done on a dedicated computer, as it requires a fast CPU, as well as higher electricity usage and more heat generated than typical computer operations. The main incentive for mining is that users who choose to use a computer for mining are rewarded for doing so. In the case of bitcoin, it is 25 bitcoins per hash. That is why some hackers use machines they break into to mine bitcoins, getting an unwitting victim to pay for the costs of mining while reaping none of the benefits Proof of Work systems end up requiring massive amounts of energy for the computing power used by **miners** [1] .

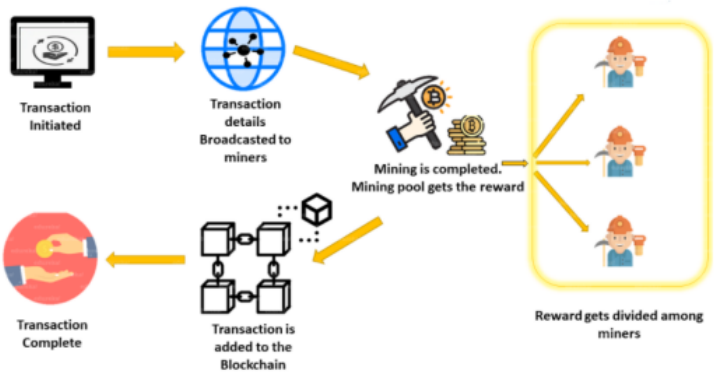


FIG. 1.12 : Minning.

Proof of Stake (PoS)

FIG. 1.13 : Proof of stake.

Proof of Stake is the second most popular consensus mechanism and solves many of the disadvantages found on PoW blockchains like lack of speed, poor scalability, inefficient energy consumption, and high barrier to entry. Examples of current industry-leading PoS blockchains include Polkadot, EOSIO, and Cardano. Ethereum 2.0 model is based on the Proof-of-Stake concept and currently, a lot of applicatioon (Decentralized Application) are running on Ethereum's peer-to-peer network. **The Proof of Stake (PoS)** concept states that a person can mine or validate block transactions according to how many coins they hold. This means that the more coins owned by a miner, the more mining power they have [3].
Proof-of-stake comes with a number of improvements to the proof-of-work system :

- better energy efficiency – you don't need to use lots of energy mining blocks .
- lower barriers to entry, reduced hardware requirements – you don't need elite hardware to stand a chance of creating new blocks .
- stronger immunity to centralization – proof-of-stake should lead to more nodes in the network.
- stronger support for Ethereum network.

Validators Validators are chosen to produce the next block in Proof of Stake blockchains based on their stake. A larger amount staked by a validator could offer them a better probability of producing the next block, despite the fact that random functions are generally used to prevent a front-running consensus. Validators propose blocks, which are then passed on to the rest of the group, who verify and add the accepted block [7] .

FIG. 1.14 : Proof of stake.

1.9 The Blockchain Life Cycle

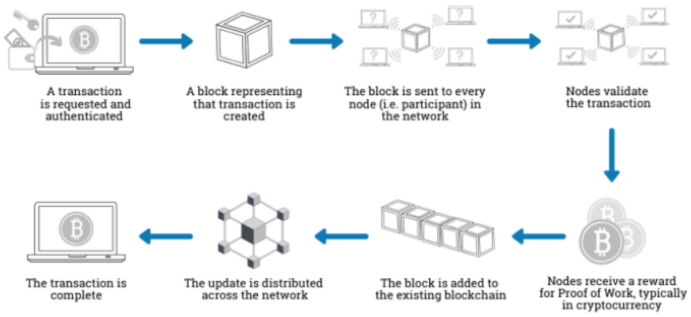


FIG. 1.15 : How Does Blockchain Work.

Authentication

Although the blockchain was created without the need for a central authority, transactions must still be verified.

Cryptographic keys, a string of data (Example a password) that identifies a user and grants access to their account or "wallet" of value on the system, are used to do this. Each user has their own private key and a public key that everyone can see. Using them both creates a secure digital identity to authenticate the user via digital signatures and to 'unlock' the transaction they want to perform.

Authorisation

Before being added to a block in the chain, the transaction must be accepted, or authorised, by both users. A public blockchain uses consensus to decide whether or not to add a transaction to the chain. This means that the transaction must be accepted by a majority of the network's nodes.

The people who own the computers in the network are incentivised to verify transactions through rewards. This process is known as proof of work [8].

1.10 Blockchain Challenges

Blockchain is a storage technology and transmission of information at low cost, secure,transparent, and operating without central control body, this technology is proposed for many application domains,in this section we provide an overview of blockchain challenges :

Scalability

The first major problem in its adoption is scalability. Although the transaction network can process thousands of transctions per second without any failures, for Bitcoin (around

3-7 transactions per second) and Ethereum (around 15-20 transactions). the speed of transaction processing has dropped significantly. Makes blockchain unsustainable in largescale applications. Bitcoin's Lightning Network and Ethereum's Plasma Network can be seen as scaling solutions that promote spontaneous transactions at minimal cost. For mass adoption, blockchai will need to be accelerated to become viable. Verifying trans- actions is a key part of the consensus protocol, as nodes in the blockchain network are expected to verify every transaction in every block. Quantity Transactions within a block and the time between blocks will adjust the required computing power,which directly affects the transaction confirmation time. Therefore, the consensus protocol has a direct impact on the scalability of the blockchain network.

Interoperability

Interoperability is the second biggest problem to solve because it is one of the main reasons why organizations are still not embracing technology. Most blockchains operate in silos and cannot communicate with other peer to peer networks because they cannot send and receive information from another blockchain-based system.In order to overcome this problem, various projects have been carried out to eliminate this problem.the Smart Bridges architecture to bridge the communication gap between networks. The project claims to provide universal transmission and transmission and ensure global interoperability.

Anonymity and data privacy

Privacy is not enforced in the Bitcoin protocol by design. A key feature of Bitcoin is its transparency. In blockchain each transaction can be checked, audited and traced from the system's very first transaction. This is indeed an unheard of newlevel of transparency that doubtlessly helps to build trust. How- ever this transparency has a knock on effect on privacy, even though there is no direct relationship between wallets and individuals, user anonymity seems to be compromised despite the mechanisms that Bitcoin provides, such as pseudonymous and the use of multiple wallets. In this sense, some effort has been made to provide stronger anonymity features in Bitcoin. On the other hand not just open virtual currencies, but many applications based on public blockchain technology require a higher level of privacy in the chain,specifically those that deal with sensitive data.

Legal issues

Blockchain has the ability to cross jurisdictions because the nodes of the blockchain can be located anywhere in the world.
This can lead to many complex jurisdictional issues and the relevant contractual relationship must be carefully considered. The principles of the contract and title to property vary by jurisdiction, so it is important to determine the appropriate applicable. For example, in a typical banking transaction, if the bank is at fault, regardless of the mechanism or location of the transaction, it may take legal action against the bank, and the applicable jurisdiction is likely to be governed by the contract. However, in a decentralized environment, it can be difficult to determine the right set of rules to apply. At the simplest level,

each transaction can fall under the jurisdiction of the location of each node in the network.

Smart contracts

Smart contracts … guarantee a very, very specific set of outcomes. There's never any confusion and there's never any need for litigation. They are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss.
Smart contracts work by following simple "if/when…then…" statements that are written into code on a blockchain. A network of computers executes the actions when predetermined conditions have been met and verified. These actions could include releasing funds to the appropriate parties, registering a vehicle, sending notifications, or issuing a ticket. The blockchain is then updated when the transaction is completed. That means the transaction cannot be changed, and only parties who have been granted permission can see the results [9] .
As shown bellow the code for a smart contract that was written on the Ethereum blockchain :

```
/* Allow another contract to spend some tokens in your behalf */
function approve(address _spender, uint256 _value)
    returns (bool success) {
    allowance[msg.sender][_spender] = _value;
    return true;
}

/* Approve and then comunicate the approved contract in a single tx */
function approveAndCall(address _spender, uint256 _value, bytes _extraData)
    returns (bool success) {
    tokenRecipient spender = tokenRecipient(_spender);
    if (approve(_spender, _value)) {
        spender.receiveApproval(msg.sender, _value, this, _extraData);
        return true;
    }
}

/* A contract attempts to get the coins */
function transferFrom(address _from, address _to, uint256 _value) returns (bool success) {
    if (balanceOf[_from] < _value) throw;                 // Check if the sender has enough
    if (balanceOf[_to] + _value < balanceOf[_to]) throw;  // Check for overflows
    if (_value > allowance[_from][msg.sender]) throw;   // Check allowance
    balanceOf[_from] -= _value;                           // Subtract from the sender
    balanceOf[_to] += _value;                             // Add the same to the recipient
    allowance[_from][msg.sender] -= _value;
    Transfer(_from, _to, _value);
    return true;
}
```

FIG. 1.16 : Smart contract code.

1.11 Blockchain Applications

Blockchain applications are far more than just cryptocurrency. This technology is impacting a variety of sectors in ways that range from how contracts are enforced to making government work more efficiently [11] .

Internet of Things

The Internet of Things (IoT) is the next technology in blockchain applications. IoT has millions of applications and many safety concerns, and an increase in IoT products means

better chances for hackers to steal your data on everything from an Amazon Alexa to a smart thermostat.
Blockchain-infused IoT adds a higher level of security to prevent data breaches by utilizing transparency and virtual incorruptibility of the technology to keep things ”smart.” Below are a few US companies using blockchain to make the Internet of Things safer and smarter.

Personal Identity Security

Lot of person complained of identity fraud and theft, with an identity being stolen every two seconds. Fraud on this scale can occur via everything from forged documents to hacking into personal files.
By keeping social security numbers, birth certificates, birth dates and other sensitive information on a decentralized blockchain ledger, the government could see a drastic drop in identity theft claims. Here are a few blockchain-based enterprises at the forefront of identity security.

Healthcare

Blockchain in healthcare have shown the potential to reduce healthcare costs, improve access to information across stakeholders and streamline businesses processes by using an advanced system for collecting and sharing private information could be just what the doctor ordered to make sure that an already bloated sector can trim down exorbitant costs.
Blockchain contracts help patients and doctors securely transfer sensitive medical information. The smart contracts establish the parameters of what data can be shared and even displays details of personalized health plans for each patient.

Government

Blockchain can be in the form of improving government. As mentioned previously, some state governments are already using the technology to secure government documents, and it's can also improve bureaucratic efficiency, accountability and reduce massive financial burdens.
Voting platform that runs on blockchain. The encrypted biometric security system makes it secure to vote on a mobile device from anywhere in the world without fear of hacking or data corruption. West Virginia is one of the first states to use the company's platform to collect votes from eligible service people and travelers abroad during elections .

1.12 Summary

The blockchain technology uses decentralized network architecture to maintain its network. This means that block chaining is not centrally controlled by any corporation or agency but is a decentralized network, which makes it more secure. According to Block Chain Council, the term “Blockchain Technology” usually refers to the transparent, trustless, publicly accessible ledger that allows us to securely and quickly transfer the

ownership of units of value by means of public key encryptions and proof of work methods. The purpose of blockchain is to establish and govern minimum standards, to develop measurements and inform the public if an individual meets or exceeds the minimum standard.

1.13 Conclusion

After presenting project context and analyzing the concept of Blockchain technology and his potential advantages. the next chapter will be about the State of the art(Benchmarking) of using blockchain in E-Health in MOROCCO like Generic Patient-Centered Blockchain Based EHR Management System .

Chapitre 2

A STATE OF THE ART ON SECURING MEDICAL DATA WITH USE CASES FROM MOROCCO

This chapter is dedicated to talk about the state of the art on securing medical data and the global state of e-health in MOROCCO .

2.1 Introduction

The E-Health Transformation in Morocco advocates harnessing digitalization to address socio-economic challenges, With a holistic approach, it emphasizes impacts on public services, economic productivity, and social equity. Healthcare digitization is a focal point, envisioning enhanced patient care through electronic health records and telemedicine.
Mobile health apps empower patient involvement, while a digitization-conducive environment is crucial, requiring infrastructure and regulatory adjustments. Addressing the digital divide, inclusive strategies are promoted.We must stress a participatory approach, involving collaboration among e-health stakeholders, and concludes with a strategic exercise to identify a common cultural foundation for a national e-health strategy in Morocco.

2.2 Digital transformation in Morocco : a godsend for e-health development

Digital transformation will enable Morocco to face its socio-economic challenges, but most particularly to improve the quality of public services, economic productivity and competitiveness, and reduce social and spatial inequalities. For the latter, digitisation can open up new perspectives by allowing disadvantaged populations to access information and social benefits, including healthcare services. Healthcare digitisation opens up new perspectives thanks to the mass and volume of health-related data available. The development of electronic health records to facilitate information sharing and medical monitoring will improve patient care. Telemedicine will make it possible to redraw the health map and will enable a territorial rebalancing in favour of areas with low medical density. Mobile health applications will provide comfort to patients by encouraging their involvement in and commitment to their treatment and health journey [12] .

2.3 An E-health conducive environment

To advance e-health,we must create a conducive digitization environment is essential. This involves enhancing training programs, establishing infrastructure, adapting regulations, and fostering a digital culture. Overcoming the challenge of digital transformation in healthcare requires inclusion of low digital users, particularly vulnerable populations affected by the digital divide. Success in e-health projects hinges on a participatory and patient-focused approach, mobilizing the entire ecosystem around the Ministry of Health. The white paper, based on interviews with key players in Morocco's e-health ecosystem, aims to address strategic, operational, economic, cultural, technical, regulatory, ethical, security, and governance issues, seeking to establish a common culture for a national e-health strategy [12] .
In this project, I aimed to make a basic decentralized app for managing electronic health records. It's designed for different users like patients, doctors, nurses, and an admin doctor who oversees everything.

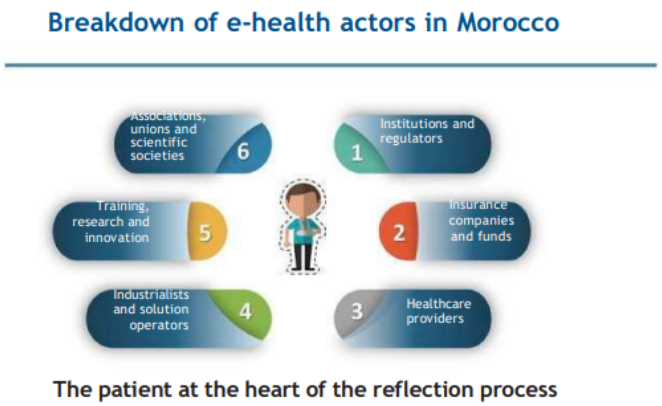


FIG. 2.1 : Breakdown of e-health actors in Morocco.

2.4 Some applications of E-health in MOROCCO .

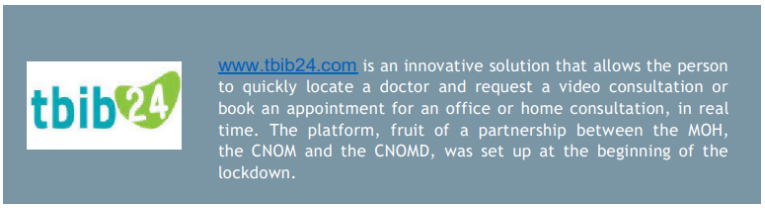


FIG. 2.2 : tbib24 mobile and web app.

www.tbib24.com is an innovative solution that allows the person to quickly locate a doctor and request a video consultation or book an appointment for an office or home consultation, in real time. The platform, fruit of a partnership between the MOH, the CNOM and the CNOMD, was set up at the beginning of the lockdown .

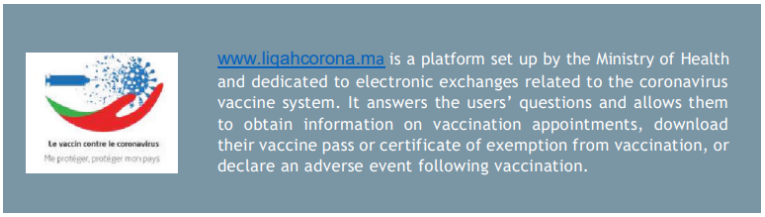


FIG. 2.3 : liqahcorona mobile and web app.

www.liqahcorona.ma is a platform set up by the Ministry of Health and dedicated to electronic exchanges related to the coronavirus vaccine system. It answers the users' questions and allows them to obtain information on vaccination appointments, download their vaccine pass or certificate of exemption from vaccination, or declare an adverse event following vaccination.

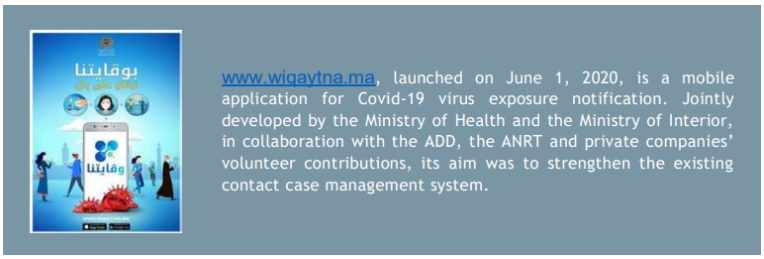


FIG. 2.4 : wiqaytna mobile and web app.

www.wiqaytna.ma launched on June 1, 2020, is a mobile application for Covid-19 virus exposure notification. Jointly developed by the Ministry of Health and the Ministry of Interior, in collaboration with the ADD, the ANRT and private companies' volunteer contributions, its aim was to strengthen the existing contact case management system [12] .

2.5 The link between the investment in Start-up culture and E-Health sector(MOROCCO position in this field) .

2.5.0.1 Morocco's positioning on the regional and continental levels .

Regional positioning. Regionally, Egypt remains the major investment hub in tech start-ups 8 : since 2015, 80 percent of investments in start-ups in the North Africa region were attracted by Egyptian start-ups. In second position comes Tunisia which continues to advance in African rankings thanks to its reform of the Start-up Act 9 legal framework, enacted in 2018. Very attractive levels of support are encouraging foreign investors to take more interest in Tunisian start-ups.

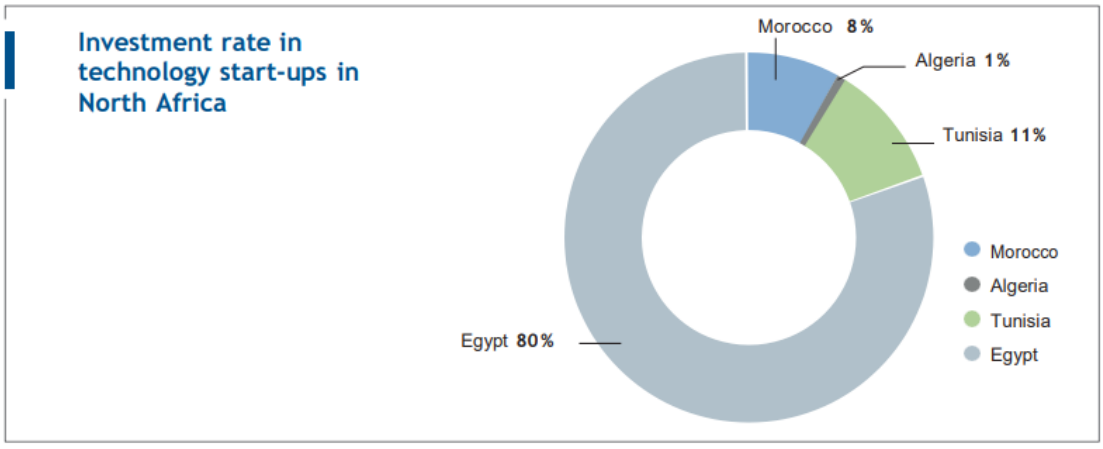


FIG. 2.5 : Investment rate in technology start-ups in North Africa .

Continental positioning. According to the Global Start-up Ecosystem Research Center 10 which publishes the Global Start-up Ecosystem Index on a continuous, dynamic and interactive basis, Morocco ranks 10th among African countries and 3rd in North

Africa, after Tunisia and Egypt.

FIG. 2.6 : Top 10 african start-ups.

International positioning. Specialised platforms and media place Morocco in the 95th position in world rankings (in red on the map). After two consecutive years of significant decline in global rankings, Morocco runs the risk of dropping entirely out of the world's top 100 in 2023. However, given its highly skilled human capital, Morocco can improve its ranking.

The Moroccan e-health ecosystem

The e-health (HealthTech) sector is linked to the digital, start-up, innovation and investment fields in general. One cannot dissociate e-health from the general context of these concepts. It would therefore be wise to look at the studies and analyses that ho-

listically address these aspects. In Morocco, the health technology environment is very modest, with only a few outstanding experiences stemming from isolated initiatives.

The HealthTech sector in Morocco can be subdivided as follows :

- **Biotechnology :** companies that develop new drugs, bio-diagnostics, cosmetics or new therapies ;

- **Medtech-Medical Technologies :** companies that develop new medical equipment ;

- **Digital HealthTech :** companies that develop services or software in the medical field based on digital technology.Despite signs of progress buoyed by Covid-19, which prompted the creation of health technology start-ups, HealthTech in Morocco remains underdeveloped, and the overall ecosystem of start-ups generally remains below its potential when compared to a target benchmark [12] :

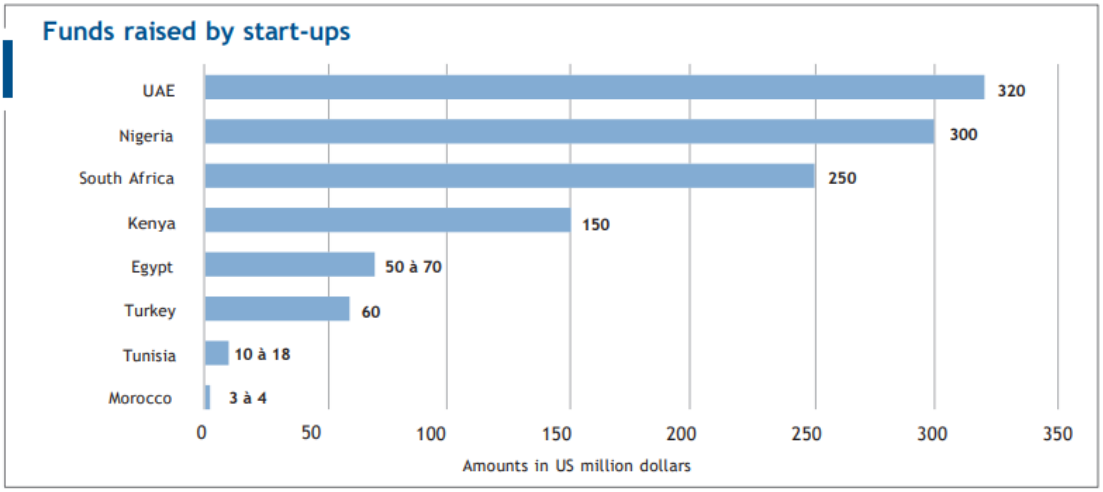


FIG. 2.7 : Funds raised by start-up from the web site start-up.ma.

2.6 Uses cases of e-health in morocco .

The e-health sector in Morocco is growing rapidly, with an increasing number of applications and services available to patients, healthcare professionals and institutions [12] .

The main use cases for e-health in Morocco are as follows :

1. **Telemedicine :**
   enables patients to consult a healthcare professional remotly using communication tools such as videoconferencing or telephone. This solution is particularly useful for patients who live in rural areas or have difficulty traveling .

2. **Online appointment booking :**

allows patients to book an appointment with a healthcare professional without having to travel in person. It's a convenient, time saving solution.

3. **The electronic medical record (EMR) :**
   is a computerized record containing a patient's medical information,it enables healthcare professionals to share a patient's medical information, thus facilitating care coordination.

4. **Remote monitoring :**
   enables healthcare professionals to remotely monitor patients with chronic conditions such as diabetes or hypertension,this reduces the need for face-to-face visits and improves the quality of care.

5. **Continuing training :**
   enables healthcare professionals to train and keep up to date with the latest knowledge,this is an important factor in guaranteeing quality of care.

6. **Electronic Health Records (EHRs) :**
   Electronic health records (EHRs) are a cornerstone of e-health, enabling the collection, storage, and exchange of patient health information in an electronic format. EHRs offer a range of benefits for both patients and healthcare providers, contributing to improved patient care, reduced medical errors, enhanced communication, and better population health management.

EHRs for Improved Patient Care

EHRs provide healthcare providers with a comprehensive and up-to-date view of a patient's medical history, encompassing diagnoses, treatments, medications, allergies, immunizations, laboratory and test results, radiology images, and other relevant health information. This holistic view of a patient's health enables providers to make informed decisions, deliver personalized care, and track patient progress over time [12] .

Here are some specific examples of how blockchain can be used to improve EHRs :

1. **Secure storage of EHR data :** A blockchain can be used to create a secure and tamper-proof repository for EHR data,this would ensure that patient data is protected from unauthorized access and modification.

2. **Audit trail of EHR access :** A blockchain can be used to create an audit trail of all access to EHR data,this would make it possible to track who has accessed patient data and when.

3. **Secure sharing of EHR data :** A blockchain can be used to create a secure and trusted platform for sharing EHR data between healthcare providers,this would

make it easier for providers to access patient data from different systems and improve coordination of care.

4. **Automated updating of EHRs :** A blockchain can be used to automate the process of updating EHRs,this would save time and money for healthcare providers.

5. **Improved interoperability of EHRs :** A blockchain can be used to create a standard format for EHR data,this would make it easier for providers to access and use patient data from different systems.

These are just a few examples of the many ways in which blockchain can be used to improve EHRs

2.7 Conclusion

To sum up, Morocco is making big strides in using technology for healthcare. They're using things like video calls with doctors and electronic health records. Also, mobile apps are being used to get patients more involved and offer personalized care. Morocco is actively creating an environment that supports using technology in healthcare, showing how serious they are about improving health services. They're also working on making sure everyone, even those who are not very familiar with technology, can benefit. By working together and involving many people, Morocco is leading the way in creating a plan for using technology in healthcare across the whole country.

Chapitre 3

Proposed Model

3.1 Introduction

In this chapter I will describe the main goal of the approach depending on an article as a set of software and projects used to demonstrate the concept of this work,I will present the details of the principal functionalities of the proposed system .

3.2 Framework Components

In this project, as shown in Figure 3.1, it established a permissioned infrastructure in an Ethereum Blockchain framework that allows complete control of records by patients while ensuring privacy, durability, and security. This is carried out by maintaining health information primarily on the blockchain as hashes, whereas the original bulk quantities of data are stored off-chain in IPFS to guarantee effectiveness and scalability .

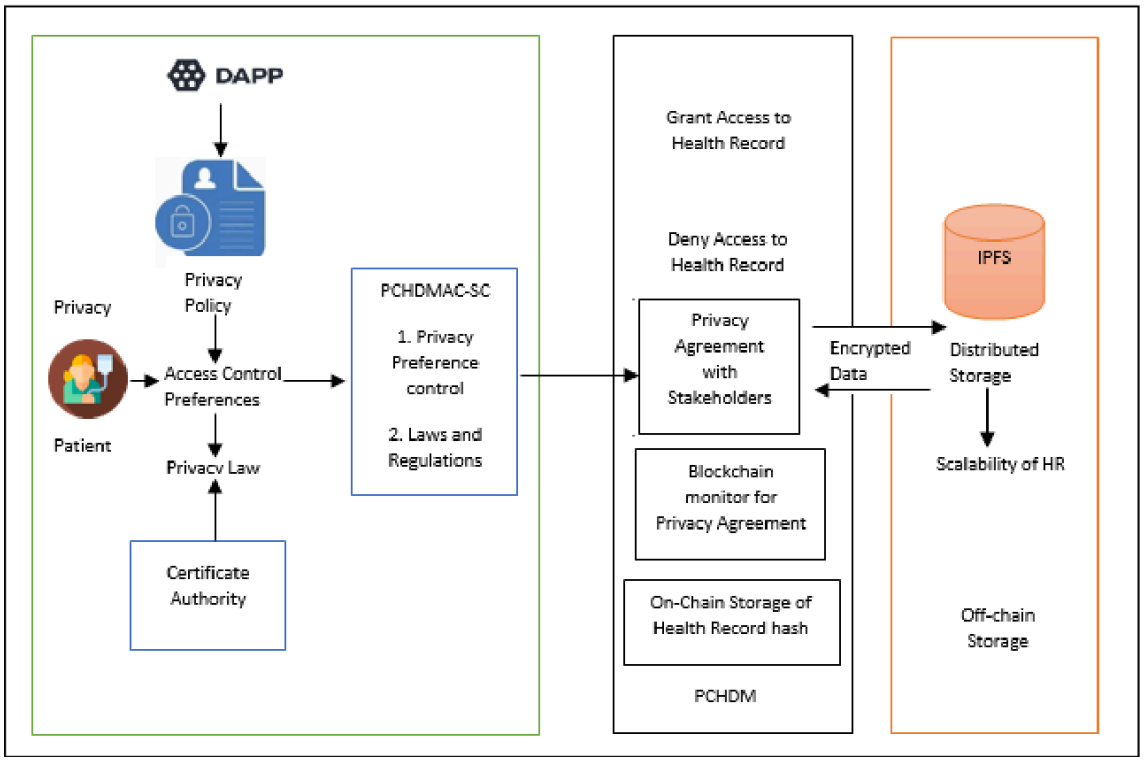


Fig. 3.1 : Framework of the proposed system .

A Patient-Centric Healthcare Data Management Access Control-Smart Contract is a protocol for smart contract chain code. This system employs role-based access control for recognized stakeholders without utilizing incentive mining. Unique role-based IDs are generated upon stakeholder registration, and each party receives public and private key pairs for health information storage and transfer. Doctors create a patient's health record, encrypted and stored indelibly in IPFS, with the record hash preserved in the Ethereum blockchain. Access to update the record is controlled, allowing or blocking based on granted or revoked access. Patients can selectively share records with relevant stakeholders, ensuring a patient-centric view. Session expiration and smart contracts enhance data privacy, and the framework exhibits improved scalability and interoperability compared to existing systems [13].

3.2.1 Ethereum Blockchain

Ethereum, modeled after the Bitcoin architecture, introduces a framework for programmable smart contracts (SC). These software programs define rules for contractual agreements, reducing the costs associated with centralized databases. Utilizing the Ethereum virtual machine within the blockchain network, SC operations are executed, with transaction costs determined by gas values. Gas is acquired using digital currency and covers the computational expenses of SC execution. Ethereum includes two primary account types : contract accounts governed by contract code and externally owned accounts (EOAs) controlled by private keys. Ethereum transactions involve various factors such as account nonce, sender signature, recipient address, gas limit, ether values, gas price, and medical data endpoints. The associated state database, structured as a Merkle-Patricia tree similar to IPFS, enhances security and robustness for off-chain and on-chain storage of medical records. The proposed system, implemented using Ethereum's smart contract architecture, establishes a clear and secure access control mechanism, preventing unauthorized access without patient consent [14] .

3.2.2 Distributed InterPlanFile System (IPFS)

Within the P2P IPFS system, a cryptographic hash serves as a distinct fingerprint for each file. In this regard, the hash address is employed to make the contents immutable. In the IPFS file storage structure, Merkle DAGs combined Merkle trees with DAGs. The fundamental functionality of IPFS to access health information may be achieved by contentbased addressing rather than location-based addressing. By harnessing the IPFS structure,lowered bandwidth costs can be achieved, file download speeds can be improved, and a substantial amount of data may be transmitted without duplication, which can free up storage space. Additionally, IPFS is an immutable storage solution since the hash value of an IPFS file cannot be modified [15] .

3.2.3 A Background of the Proposed System .

Our system structure, as displayed in Figure 3.2, has the organization use three peer nodes, with one acting as a validating peer node and the others as an ordering node for registering stakeholders. Multiple peers can access the same database in this system, which also uses IPFS for distributed data storage. Multiple peers can be added to various locations on different machines to test the system's scalability. This framework, which contains its own ledger and smart contract copies, provides access to ledgers for smart contracts .

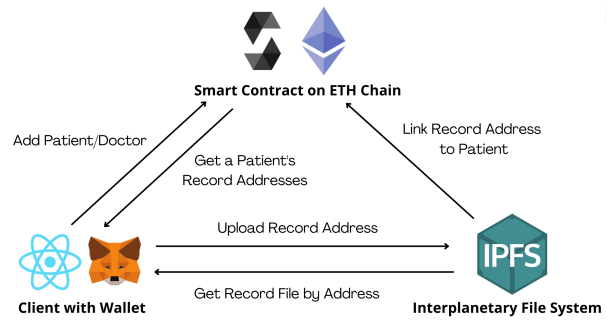


Fig. 3.2 : Patient Centered Electronic Health Record Management.

Peer nodes are linked to the application, which then uses smart to update the ledger. Figure 3.3 shows the data structure of the blockchain ledger after the integration of PCEHRM data fields, as it is intended to record only the information that patients wish to provide in a transaction. Peernode1, Peernode2, and Peernode3 are the three peer nodes in the organization, and each one has a copy of the ledger and a smart contract . In this sense, the patient's profile, address and location, diagnoses, doctor recommendations, next review notes, physician's names, medication, scan and test reports, and hospital ID are all included in the healthcare records.
The following stakeholders make up the PCEHRM :

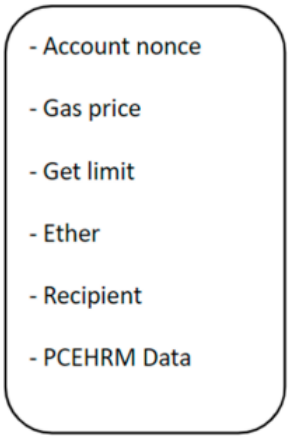


Fig. 3.3 : The data structure of the blockchain ledger of PCEHRM .

3.2.4 Record Owner

Patients retain their medical data, making them the owner of the record. To store the data, patients are required to acknowledge and sign an agreement on PCEHRM-SC in the Ethereum blockchain. The chain networks permit patients to specify access rights to their healthcare information, defined by each PCEHRM-SC within its context. Table 3 further describes the patient roles in detail.

Table 3. Patient roles.

Patient	Grant-Revoke-Commit, Read Record Revoke permission from Doctor/Service Providers. Permission to Doctor to Read/write of their her. Able to search for Doctor/Labs.

3.2.5 Data Uploader

The doctors/lab technicians may upload their patients' medical data to Data Uploaders. The primary responsibilities of the data uploaders include submitting the concerned individual's encrypted clinical data to the IPFS community and validating the preliminary transaction at the blockchain. This is further illustrated in Table 4.

Table 4. Doctors/lab roles.

Doctor/Labs	-Create/Read/Write on Permission for EHR -Search for Doctor in the network
	-Read/Write on Permission for EHR. -Search for Labs the network.

3.2.6 Data Users

Data users are defined as the parties that require patients' medical or clinical records for further action, namely, hospitals, researchers, insurance companies, and doctors. In this context, the role-based access control approach in PCEHRM-SC specifies the mechanism for patients to grant access rights to data users .

3.2.6.1 Data Encryption .

This framework ensures privacy and integrity in blockchain data through cryptographic methods. In the patient-doctor interaction, the doctor initiates access permission to retrieve specific information from IPFS, creating a patient-centric view without revealing all details. A session key (SK) encrypts and stores this view in IPFS for a limited period, facilitating secure record updating. Patients are notified after revisions, and the SK is automatically erased upon committing to their health record, prioritizing privacy. Smart chain code on the backend preserves the hash value securely on the Ethereum blockchain, with the ledger notifying patients of successful record creation or updates. This approach safeguards patient data, offering a privacy-focused and secure health record management system.

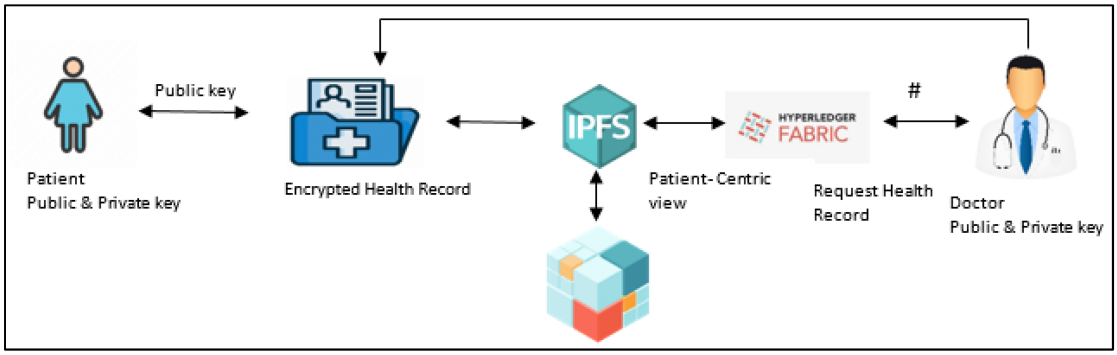


Fig. 3.4 : Collaboration diagram between patient and doctor in PCEHRM.

3.2.6.2 PCEHRM-SC

In the proposed health chain architecture, the role-based access process is patient-controlled, initiated when doctors seek permission to access patient health records on IPFS. Patients can allow or revoke access for authorized users, including doctors, insurance agents, pharmacists, researchers, and laboratory technicians. With patient consent, doctors can create, view, and modify records before the patient commits updates. The

patient-centric view is session-specific, accessible to stakeholders whose ownership and object ID match the patient's. Access regulations, privacy agreements, and control mechanisms are enforced by smart contracts on the Ethereum blockchain, specifying distinct stakeholder profiles and access rights.
Moreover, the solution classifies privacy attributes into three levels : Level 1 grants sole access to patients, Level 2 allows approved stakeholders access to medical records, and Level 3 permits patient caregivers emergency access to health records. This approach ensures a patient-centric, secure, and regulated system for health record management.

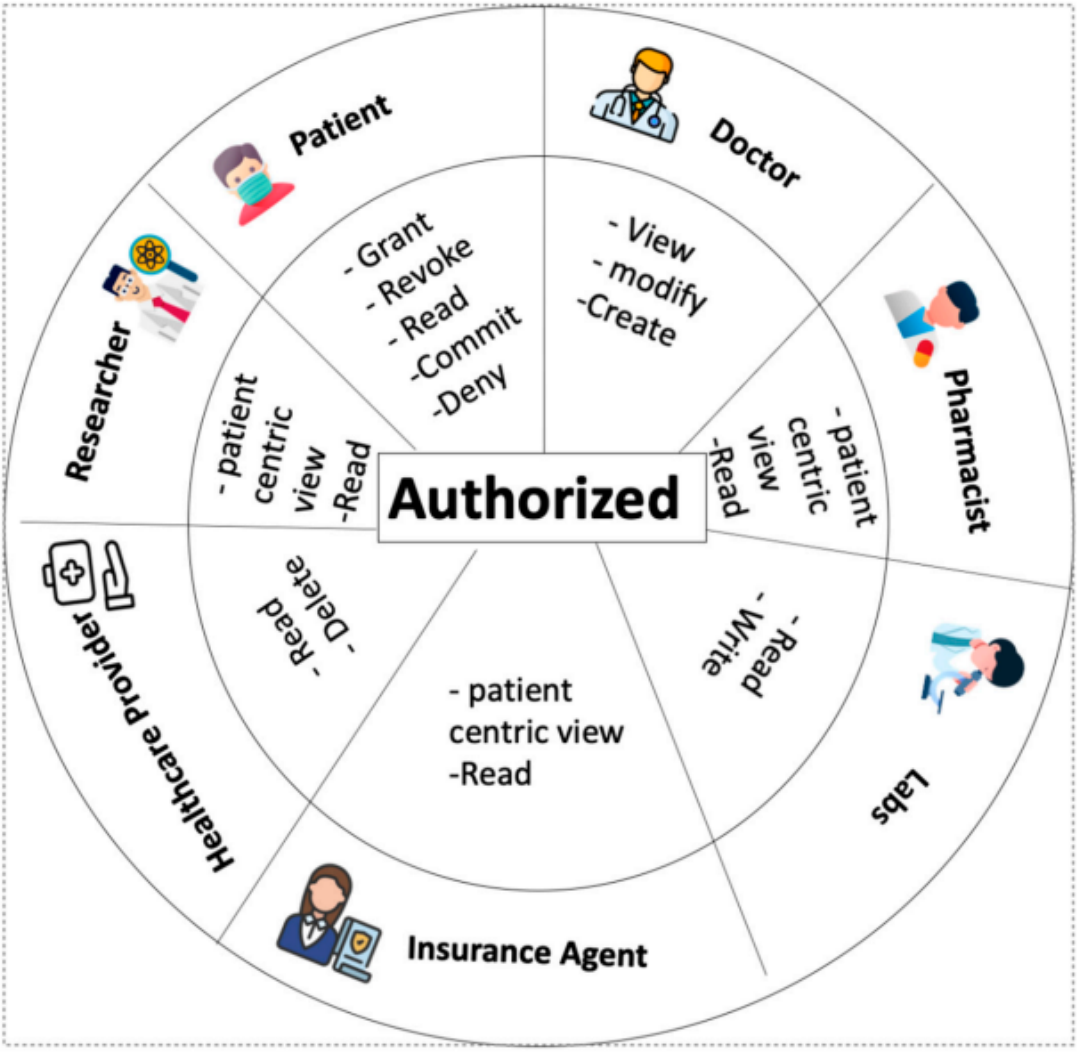


Fig. 3.5 : Stakeholder and rule-based access in PCEHRM.

3.2.6.3 PCEHRM Algorithm

Table 5. Symbols in the algorithm functions.

Symbols	Definition
$\mathcal{P}a_n$	n^{th} Patient
$\mathcal{D}_n$	n^{th} Doctor
$\mathcal{P}h_n$	n^{th} Pharma
$\mathcal{LT}_n$	n^{th} Lab Technician
$\mathcal{HR}_n$	n^{th} Health Record
$\mathcal{P}apubk_n$	n^{th} Patient Public Key
$\mathcal{P}aprk_n$	n^{th} Patient Private Key
$\mathcal{D}pubk_n$	n^{th} Doctor Public key
$\mathcal{D}prk_n$	Doctor Private Key
$\mathcal{S}_{\mathcal{K}}$	n^{th} Session Key
$\mathcal{P}acenv_n$	n^{th} Patient-Centric View
$\mathcal{UP}\_\mathcal{P}acenv_n$	n^{th} Update Patient-Centric View
$\mathcal{HR}_n\_hsh$	n^{th} Health Record Hash Value

Algorithm 1. System_Function()

Input: Doctor $\mathcal{D}_n$, with their Public key $\mathcal{D}pubk_n$, with their Private key $\mathcal{D}prk_n$, with session key $\mathcal{S}_{\mathcal{K}}$ of $\mathcal{HR}_n$ Health_Record. Patient $\mathcal{P}a_n$ with their Public $\mathcal{P}apubk_n$, and Private key $\mathcal{P}aprk_n$.
Output: Boolean (True or False)

1. Function for storing and updating health records.
2. **For** user $\mathcal{U}$ have Access permission to HR
3. Check PCEHRM-SC
4. **If** (permission=="Grant" && role=="Doctor") then
5. Create $\mathcal{P}acenv_n$ for $\mathcal{HR}_n$ in IPFS
6. $\mathcal{P}acenv_n \rightarrow$ Decryption (Encryption ($\mathcal{HR}_n$))
7. Create $\mathcal{S}_{\mathcal{K}}$
8. send Encrypted ($\mathcal{P}apubk_n$ ($\mathcal{S}_{\mathcal{K}}$), $\mathcal{D}pubk_n$($\mathcal{S}_{\mathcal{K}}$), $\mathcal{P}acenv_n$($\mathcal{S}_{\mathcal{K}}$)) to $\mathcal{P}a_n$,
9. $\mathcal{D}_n$ and $\mathcal{P}acenv_n$.
10. create_Update_HR()
11. $\mathcal{HR}_n \rightarrow$ [(Decryption $\mathcal{P}apubrk_n$ (Encrypted $\mathcal{P}apubk_n$ ($\mathcal{HR}_n$)) + Encryption ($\mathcal{UP}\_\mathcal{P}acenv_n$)]
12. $\mathcal{P}a_n \rightarrow$ Commit (IPFS ($\mathcal{HR}_n$))
13. IPFS $\rightarrow \mathcal{HR}_n\_hsh$
14. $\mathcal{HR}_n\_hsh \rightarrow$ Ethereum Blocks
15. **Return** True
16. **Else**
17. Permission=Deny
18. **Return** False
19. **End if**
20. **End For**
21. **End** Function

Algorithm 1 : facilitates doctor access to the patient's health record based on PCEHRM-SC (Smart Contract) .

Algorithm 2. create_Update_HR ()

Input: $\mathcal{D}_n$, $\mathcal{D}pubk_n$, $\mathcal{D}prk_n$, S_k
Output: Storage of HR

1. Function Doctor $\mathcal{D}pubk_n$
2. **For** Doctor with $\mathcal{D}pubk_n$, S_k
3. $\mathcal{D}_n \rightarrow$ Decrypt($\mathcal{D}pubk_n(S_k)$)
4. $\mathcal{D}_n \rightarrow$ Decrypt($\mathcal{P}acenv_n(S_k)$)
5. $\mathcal{P}acenv_n \rightarrow \mathcal{UP}\_\mathcal{P}acenv_n$
6. IPFS Encrypt($\mathcal{UP}\_\mathcal{P}acenv_n(S_k)$)
7. **End For**
8. **End** Function

Algorithm 2 : involves updating the Health Record (HRn) and managing the patient-centric view.
The CEHRM Algorithm focuses on ensuring privacy and secure management of electronic health records using cryptographic techniques and blockchain technology. It aims to control access and updates to patient data. The PCEHRM Algorithm provides specific details on key generation, session management, and the creation of patient-centric views, emphasizing secure and efficient health record management with controlled access and confidentiality.

3.3 Conclusion

In conclusion, the imperative for patients to retain control over their medical information is addressed through the proposition of a secure, interoperable patient-centric data access management system grounded in blockchain technology. This envisioned system empowers patients with full control over their health record-related data, ensuring secure storage via IPFS. The utilization of tokens further facilitates patients in granting regulated access to their medical data for specific durations, fostering a patient-centric approach in healthcare interactions.

Chapitre 4

REALISATION

4.0.1 Introduction

This chapter focuses on the central point of this work, that is the implementation of the proposed model(PCEHRM) based on blockchain technology and the tools used .

4.1 Tools and Installation Procedures

4.1.1 What is Ganache and Truffle ?

Ganache is a personal blockchain for rapid Ethereum and Filecoin distributed application development. You can use Ganache across the entire development cycle ; enabling you to develop, deploy, and test your dApps in a safe and deterministic environment [16].

Ganache comes in two flavors : a UI and CLI. Ganache UI is a desktop application supporting Ethereum and Filecoin technology. Our more robust command-line tool, ganache, is available for Ethereum development. It offers :

1. console.log in Solidity
2. Zero-config Mainnet and testnet forking
3. Fork any Ethereum network without waiting to sync
4. Ethereum JSON-RPC support
5. Snapshot/revert state
6. Mine blocks instantly, on demand, or at an interval
7. Fast-forward time
8. Impersonate any account (no private keys required !)
9. Listens for JSON-RPC 2.0 requests over HTTP/WebSockets
10. Programmatic use in Node.js

11. Pending Transactions

A world class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier. With Truffle, you get :
Built-in smart contract compilation, linking, deployment and binary management. Advanced debugging with breakpoints, variable analysis, and step functionality. Use console.log in your smart contracts Deployments and transactions through MetaMask with Truffle Dashboard to protect your mnemonic [17].

1. External script runner that executes scripts within a Truffle environment.
2. Interactive console for direct contract communication.
3. Automated contract testing for rapid development.
4. Scriptable, extensible deployment and migrations framework.
5. Network management for deploying to any number of public and private networks.
6. Package management with NPM, using the ERC190 standard.
7. Configurable build pipeline with support for tight integration.

4.1.1.1 Set up and connect the project with Ganache .

Install Truffle . First, we install a development framework Truffle ,by this command line :
npm install -g truffle
In my project, I locate the configuration file for Ethereum, commonly named truffle-config.js (for Truffle) :

```
module.exports = {

  networks: {
    dev: {
      host: "*",
      port: 8545,
      network_id: "*" // Match any network id
    },

  },

  // Set default mocha options here, use special reporters etc.
  mocha: {
    // timeout: 100000
  },

  // Configure your compilers
  compilers: {
    solc: {
      version: "0.8.11",    // Fetch exact version from solc-bin (default: truffle's version)
      // docker: true,        // Use "0.5.1" you've installed locally with docker (default: false)
      settings: {          // See the solidity docs for advice about optimization and evmVersion
       optimizer: {
         enabled: true,
         runs: 200
       },
       //  evmVersion: "byzantium"
      }
    }
  }
```

FIG. 4.1 : Configuration of my truffle.

4.1.1.2 Connect Ganache to my project .

We have created a workspace has the same name of the project ,then I have added a file named **truffle-config.js** to the workspace for connect it with the project .

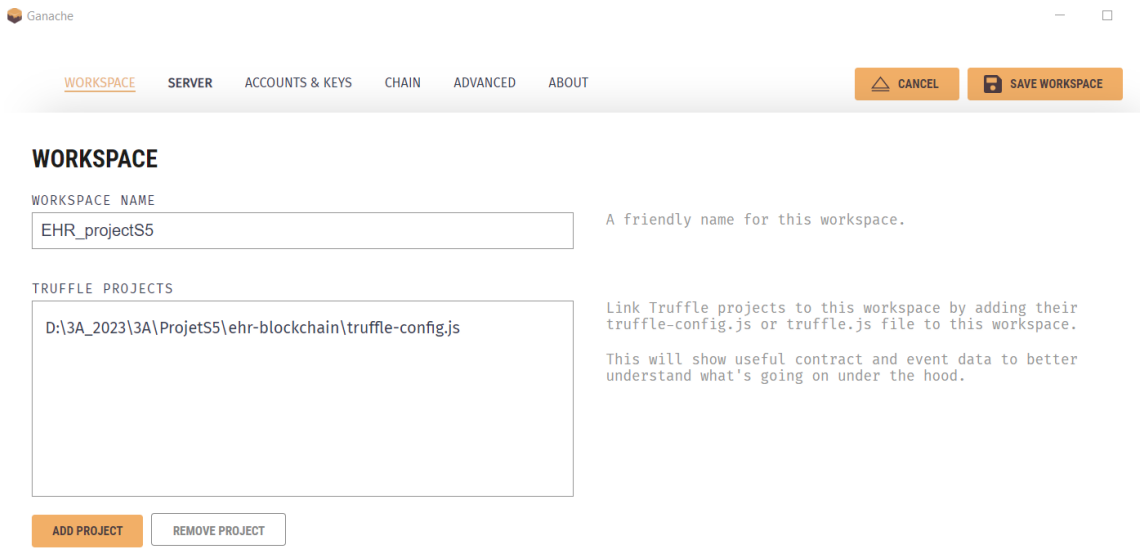


FIG. 4.2 : Configuration of Ganache.

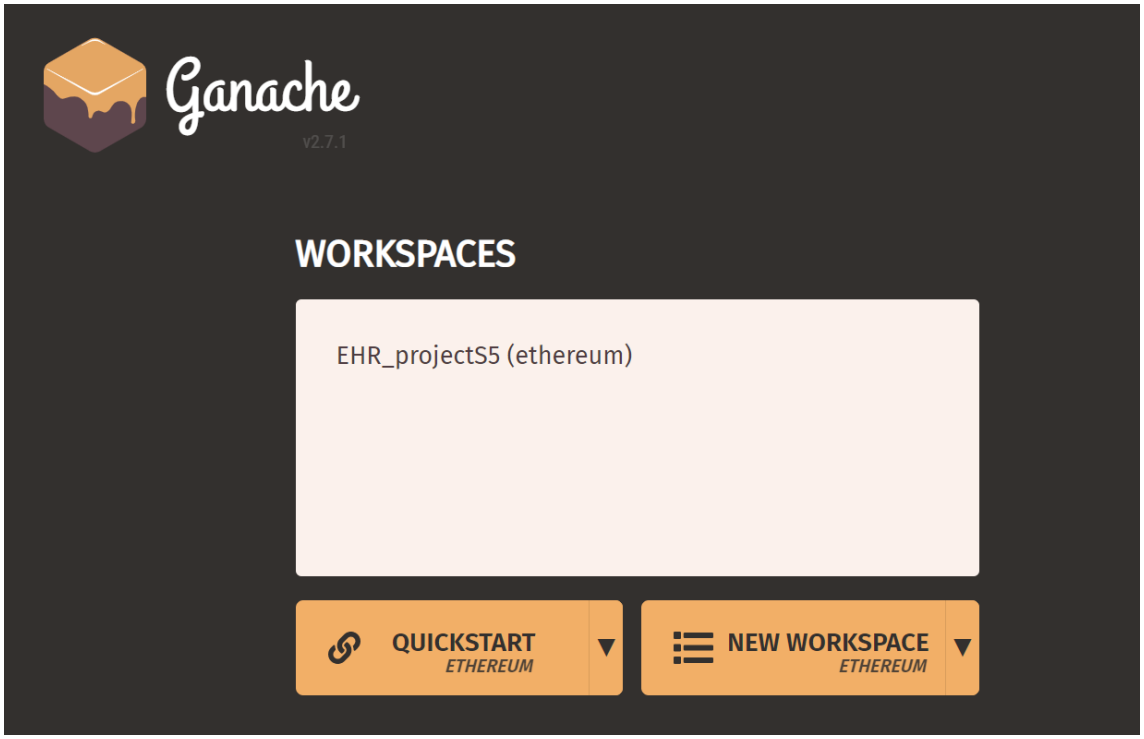


FIG. 4.3 : Configuration of Ganache.

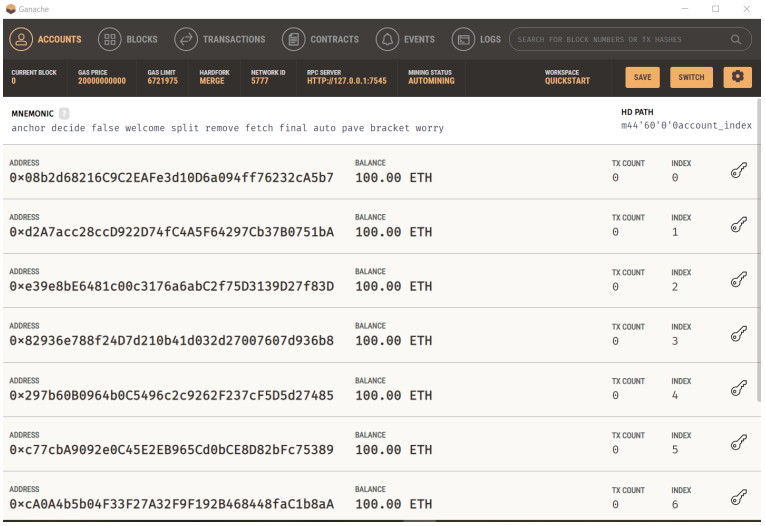


FIG. 4.4 : Configuration of Ganache.

4.1.2 Set up a MetaMask .

4.1.2.1 What is MetaMask .

MetaMask is a cryptocurrency wallet and browser extension that allows users to manage their Ethereum-based assets and interact with decentralized applications (DApps) on the Ethereum blockchain. It functions as a bridge between your web browser and the Ethereum blockchain, providing a user-friendly interface for interacting with blockchain-based applications.
Key features of MetaMask include : Wallet Management,Browser Extension,Secure Key Management,Interaction with DApps,Transaction Signing et Network Switching [18].

FIG. 4.5 : MetaMask.

4.1.3 Connect Metamask to ganache

After creating an account on MetaMask, I connected it with Ganache using the RPC URL server that exists on Ganache.

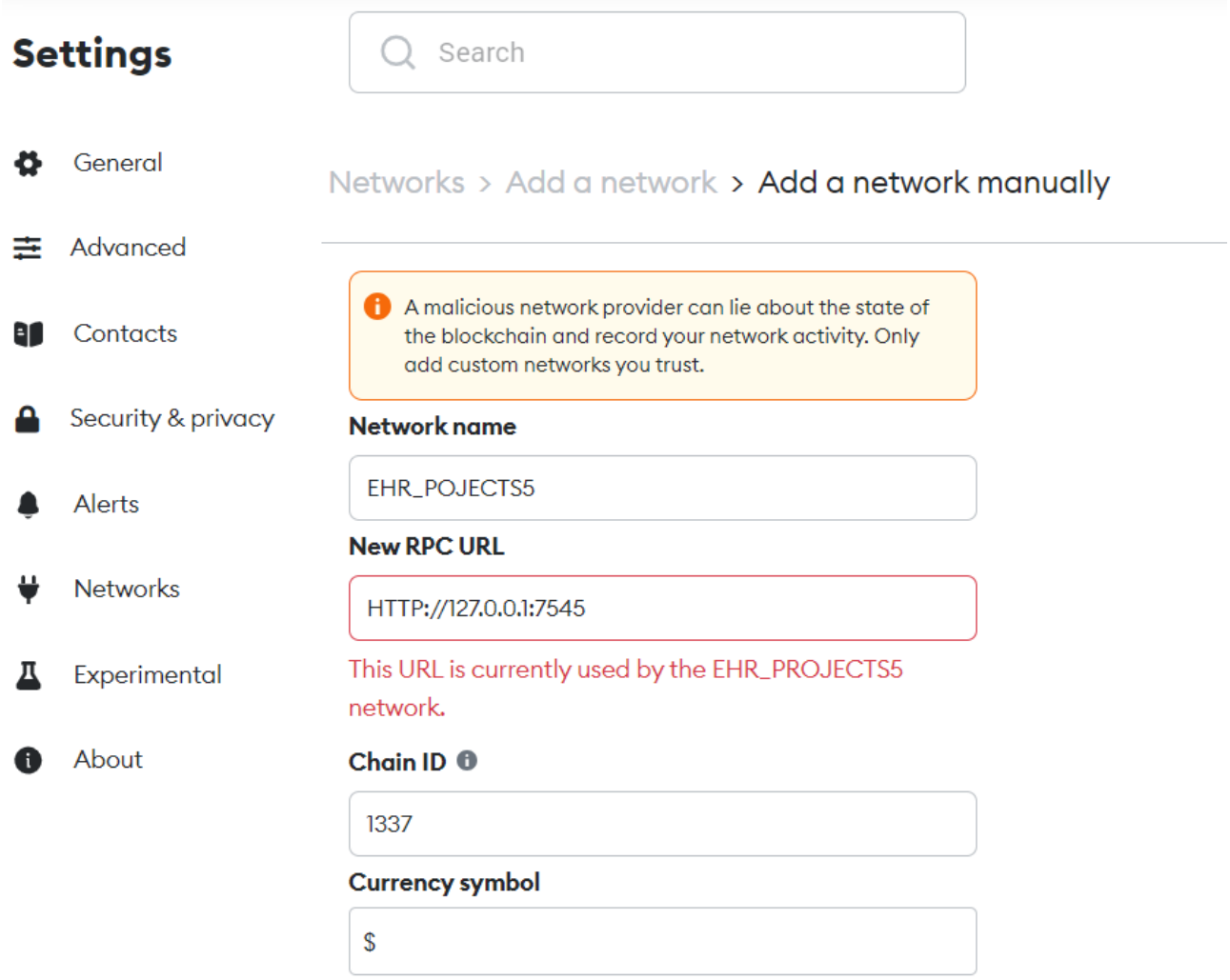


FIG. 4.6 : Configuration of MetaMask to connect it with Ganache.

This is my account on MetaMask that manages the connection between Ganache and My project EHR PROJECTS5 as follow .

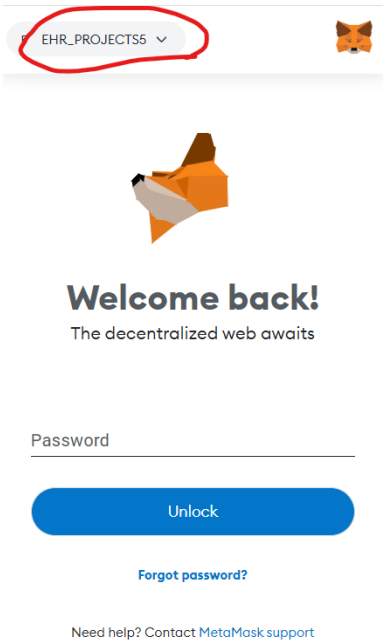


FIG. 4.7 : Configuration of MetaMask to connect it with Ganache.

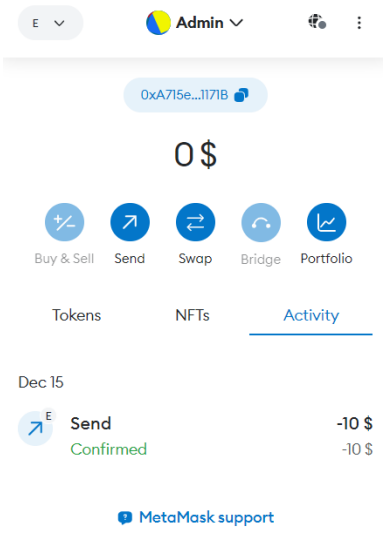


FIG. 4.8 : Configuration of MetaMask to connect it with Ganache.

We need to ensure that the localhost is accessed using the same browser where the MetaMask extension is installed .

4.2 Implementation .

4.2.1 Code Structure

This implementation has been performed using **Node.js**,**Angular.js**,**TypeScript** . and also **WEB3** but the Smart Contract has been implemented by **Solidity**.

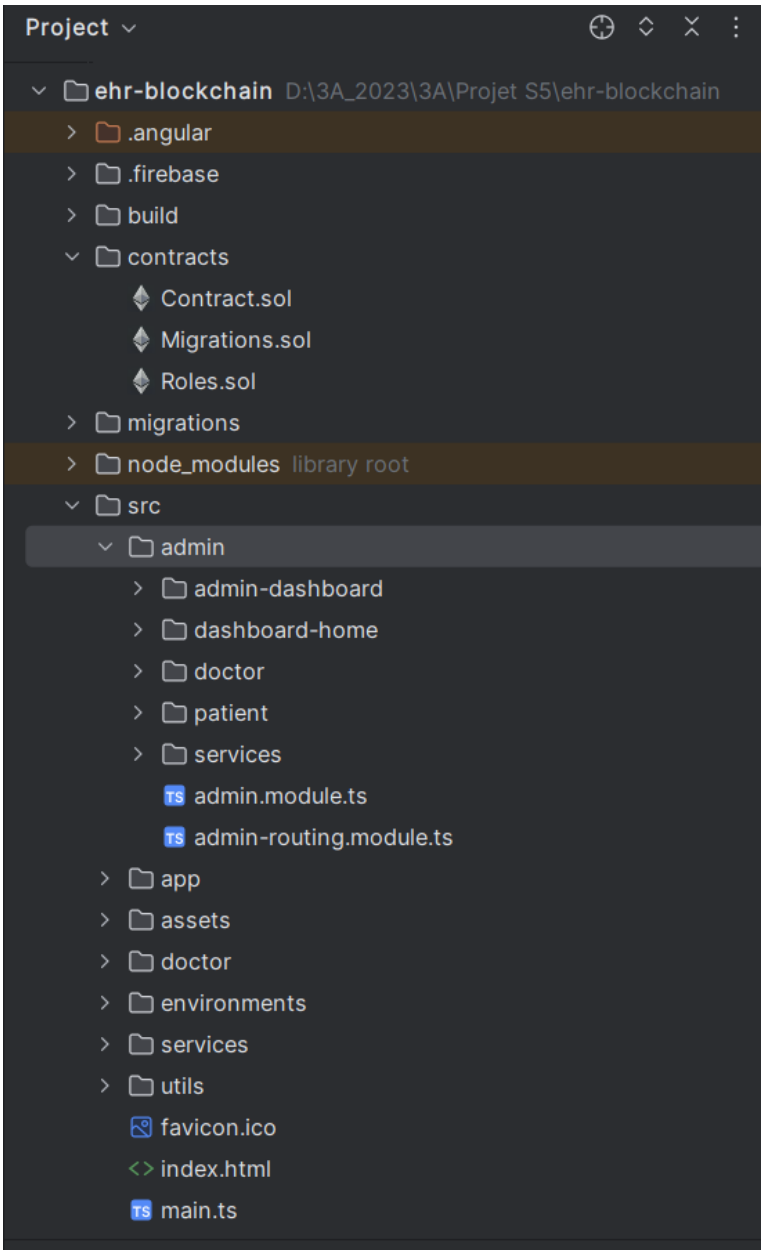


FIG. 4.9 : Tree Structure.

4.2.2 Smart Contract (SC)

Contract.sol

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.21;

import "./Roles.sol";

contract Contract {
  using Roles for Roles.Role;

  Roles.Role private admin;
  Roles.Role private doctor;

  struct Doctor {
    address id;
    string drHash;
  }

  mapping(address => Doctor) Doctors;

  address[] public DrIDs;

  constructor() {
    admin.add(msg.sender);
  }

  //get Admin
  function isAdmin() public view returns (bool) {
    return admin.has(msg.sender);
  }

  //Add Doctor
  function addDrInfo(address dr_id, string memory _drInfo_hash) public {
```

FIG. 4.10 : Contract.sol

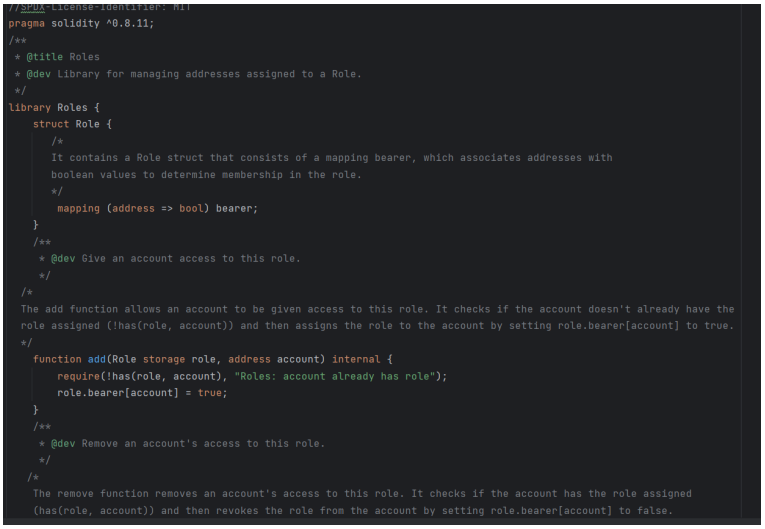
```
  //Add Doctor
  function addDrInfo(address dr_id, string memory _drInfo_hash) public {
    require(admin.has(msg.sender), "Only For Admin");
    Doctor storage drInfo = Doctors[dr_id];
    drInfo.id = dr_id;
    drInfo.drHash = _drInfo_hash;
    DrIDs.push(dr_id);
    doctor.add(dr_id);
  }

  function getAllDrs() public view returns (address[] memory) {
    return DrIDs;
  }

  function getDr(address _id) public view returns (string memory) {
    return (Doctors[_id].drHash);
  }

  // check is Doctor

  function isDr(address id) public view returns (bool) {
    return doctor.has(id);
  }

}
```

FIG. 4.11 : Contract.sol

This smart contract defines a system for managing roles within a EHR (Electronic Health Record). It includes roles for administrators and doctors, each managed by a set of functions. The contract stores information about doctors, identified by their Ethereum

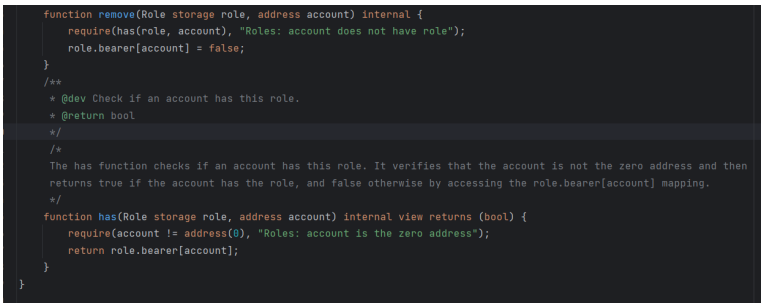
addresses, and their associated data hashes. The administrator can add new doctors and verify their admin status. Additionally, the contract provides functions to retrieve a list of all doctors, retrieve individual doctor information, and check if a given address corresponds to a registered doctor.

Roles.sol



```
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.11;
/**
 * @title Roles
 * @dev Library for managing addresses assigned to a Role.
 */
library Roles {
    struct Role {
      /*
      It contains a Role struct that consists of a mapping bearer, which associates addresses with
      boolean values to determine membership in the role.
      */
       mapping (address => bool) bearer;
    }
    /**
     * @dev Give an account access to this role.
     */
  /*
  The add function allows an account to be given access to this role. It checks if the account doesn't already have the
  role assigned (!has(role, account)) and then assigns the role to the account by setting role.bearer[account] to true.
  */
    function add(Role storage role, address account) internal {
        require(!has(role, account), "Roles: account already has role");
        role.bearer[account] = true;
    }
    /**
     * @dev Remove an account's access to this role.
     */
   /*
    The remove function removes an account's access to this role. It checks if the account has the role assigned
    (has(role, account)) and then revokes the role from the account by setting role.bearer[account] to false.
```

FIG. 4.12 : Roles.sol



```
    function remove(Role storage role, address account) internal {
        require(has(role, account), "Roles: account does not have role");
        role.bearer[account] = false;
    }
    /**
     * @dev Check if an account has this role.
     * @return bool
     */
    /*
    The has function checks if an account has this role. It verifies that the account is not the zero address and then
    returns true if the account has the role, and false otherwise by accessing the role.bearer[account] mapping.
    */
    function has(Role storage role, address account) internal view returns (bool) {
        require(account != address(0), "Roles: account is the zero address");
        return role.bearer[account];
    }
}
```

FIG. 4.13 : Roles.sol

This smart contract for managing addresses assigned to a specific role. It includes a struct called ”Role,” which consists of a mapping named ”bearer” associating addresses with boolean values to determine membership in the role. This smart contract provides functions to add, remove, and check if an account has a specific role. The ”add” function grants an account access to the role, the ”remove” function revokes access, and the ”has” function checks if an account has the role.

Migrations.sol

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.22 <0.9.0;

contract Migrations {
  address public owner = msg.sender;
  uint public last_completed_migration;

  modifier restricted() {
    require(
      msg.sender == owner,
      "This function is restricted to the contract's owner"
    );
    _;
  }

  function setCompleted(uint completed) public restricted {
    last_completed_migration = completed;
  }
}
```

FIG. 4.14 : Migrations.sol

This smart contract provides a basic structure for managing migration-related information with an emphasis on access control, ensuring that only the contract owner can update the migration status ,it could be part of a larger system where migration processes involve more complex operations .

4.2.3 Blockchain services .

```
import { resolve } from 'dns';
import Web3 from 'web3';

const Contract = require('../../build/contracts/Contract.json');

declare let window: any;

5+ usages
@Injectable({
  providedIn: 'root',
})
export class BlockchainService {
  account: any = [];
  netId: any;
  web3: any;

  address: any;
  contract: any;
  netWorkData: any;
  abi: any;

  admin: any;

  balance: any;
```

FIG. 4.15 : blockchain.service.ts class

I was declared these variables to store information related to Ethereum accounts **(account)**, network ID **(netId)**, Web3 instance **(web3)**, contract address **(address)**,

contract instance **(contract)**, network data **(netWorkData)**, and **abi**.There are additional properties for admin and balance which might be used to store admin-related information and the **account balance**, respectively .

FIG. 4.16 : blockchain.service.ts class

This constructor help us to perform many tasks as follow :

getWeb3Provider

a method, presumably an asynchronous operation returning a Promise.

this.web3.eth.getAccounts((err : any, accs : any) => ...) ;

Uses the web3 instance (presumably obtained from getWeb3Provider) to asynchronously get the *Ethereum accounts.* The callback function receives any errors (err) and the accounts (accs). Sets the account property of the class to the first Ethereum account (accs[0]).

this.web3.eth.getBalance(this.account).then((r : any) => ...) ;

Uses the web3 instance to asynchronously get the balance of the Ethereum account (this.account).

this.web3.eth.net.getId().then((r : number) => ...) ;

Uses the web3 instance to asynchronously get the network ID.

this.abi = Contract.abi ;

Sets the abi property of the class to api of the smart contract. The api is obtained from an imported Contract.json file.

this.netWorkData = Contract.networks[this.netId] ;

Sets the netWorkData property of the class to the network data corresponding to the current network ID. The network data is likely information about the deployed contract

on the Ethereum network.

window.ethereum.on('accountsChanged', (acc : any) => ...) ;
Sets up an event listener for changes in Ethereum accounts. When the accounts change, the callback function logs the new accounts and reloads the page using window.location.reload()
.

```
1 usage
checkIsAdmin(): Promise<boolean> {
  return new Promise( executor: (resolve, reject) : void => {
    this.getContract().then(c => {
      this.getCurrentAcount().then(a : string => {
        console.log(a);
        c.methods.isAdmin().call({ from: a }).then((r: any) : void => {
          console.log(r);
          if (r) {
            resolve( value: true)
          }
          reject( reason: false)
        })
      }).catch((er: any) : void => {
        console.log(er);

      })
    })
  })
}
```

FIG. 4.17 : blockchain.service.ts class

checkIsAdmin Method :
Checks if the current Ethereum account is an admin. Returns a promise that resolves to true if the current account is an admin and rejects with false otherwise.
Calls the **getContract** method to obtain the contract instance.
Calls the **getCurrentAcount** method to get the current Ethereum account.
Calls the **isAdmin** method on the contract, passing the current account, and resolves or rejects based on the result.

```
2 usages
async getWeb3Provider(): Promise<Web3> {
  if (window.ethereum) {
    window.web3 = new Web3(window.ethereum);
    window.ethereum.enable();
    // console.log(window.web3);

    this.web3 = window.web3;
    this.account = await this.web3.eth.getAccounts().then((acc: string[]) => {
      // console.log(acc);
      return acc[0];
    });
    return window.web3;
  } else if (window.web3) {
    window.web3 = new Web3(window.web3.currentProvider);
    return window.web3;
  } else {
    return window.web3;
  }
}
```

FIG. 4.18 : blockchain.service.ts class

If **window.ethereum** is available, enables it and sets up the Web3 instance.
else if **window.web3** is available, sets up the Web3 instance using the current provider.

```
2 usages
getCurrentAcount(): Promise<string> {
  return new Promise( executor: (resolve, reject) : void  => {
    if (this.web3) {
      this.web3.eth.getAccounts().then((acc: string[]) : void  => {
        // console.log(acc[0]);
        resolve(acc[0]);
      });
    } else {
      reject( reason: null);
    }
  });
}

2 usages
getWeb3(): Web3 {
  return this.web3;
}

2 usages
getBalance(): any {
  return this.balance;
}
```

FIG. 4.19 : blockchain.service.ts class

getCurrentAcount Method :
Returns a promise that resolves to the current Ethereum account. Checks if this.web3 is available and retrieves the Ethereum accounts using this.web3.eth.getAccounts.Resolves the promise with the first account if available, otherwise rejects with null.
getWeb3 Method :
Returns the Web3 instance.
getBalance Method :
Returns the balance of the current Ethereum account.

```
  getBalance(): any {
    return this.balance;
  }

  2 usages
  getAccount() {
    return this.account;
  }

  5+ usages
  async getContract(): Promise<any> {
    return new Promise( executor: (resolve, reject) : void  => {
      let check : NodeJS.Timeout  = setInterval( handler: () : void  => {
        if (this.contract != null) {
          resolve(this.contract);
          clearInterval(check);
        }
      }, timeout: 1000);
    });
  }
}
```

FIG. 4.20 : blockchain.service.ts class

getAccount Method :
Returns the current Ethereum account.
getContract Method :
Returns a promise that resolves to the contract instance. Uses a setInterval loop to repeatedly check if the contract instance is not null.Resolves the promise with the contract instance once it is available and clears the interval.

```
import { Injectable } from '@angular/core';
import { create, IPFSHTTPClient } from 'ipfs-http-client';
import { IPFS } from 'src/environments/environment';



5+ usages
@Injectable({
  providedIn: 'root',
})
export class IpfsService {
  ipfs!: IPFSHTTPClient;
  no usages
  constructor() {
    this.ipfs = create( options: { url: IPFS.localIPFS });
  }

  3 usages
  getIPFS() : IPFSHTTPClient  {
    return this.ipfs;
  }
}
```

FIG. 4.21 : blockchain.service.ts class

Constructor :

The constructor initializes the IpfsService class. It uses the create function from the ipfs-http-client library to instantiate an IPFS client. The url property of the IPFS client is set to the local IPFS gateway URL obtained from the Angular environment configuration (IPFS.localIPFS). The instantiated IPFS client is assigned to the ipfs property of the service.

getIPFS Method :

This method returns the IPFS client instance. The purpose of having this method is to provide access to the IPFS client from other components or services in this app.

The **IPFS client** can then be used to interact with the IPFS network, such as adding and retrieving files from the distributed file system.

4.2.4 Interfaces of web app .

In this section I will demonstrate some interfaces of web app . In this figure 4.22,4.23,4.24 I show you how we authenticate across the blockchain network to the EHR (Electronic Health Record web app .)

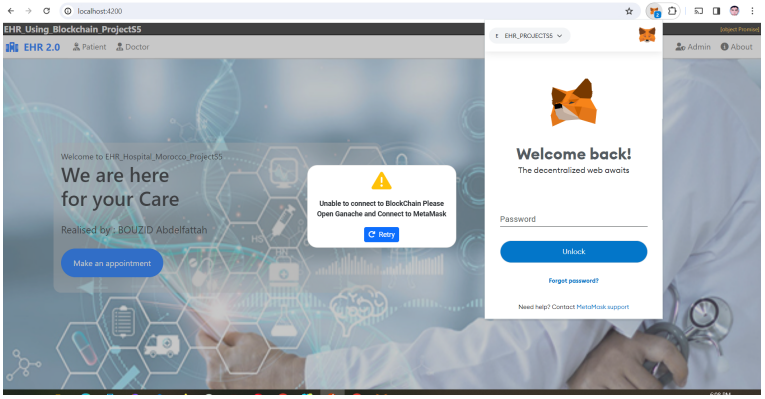


FIG. 4.22 : Authentication on the Blockchain

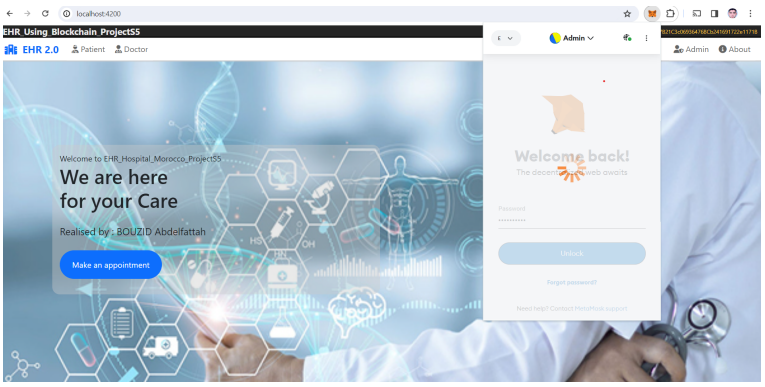


FIG. 4.23 : Authentication on the Blockchain

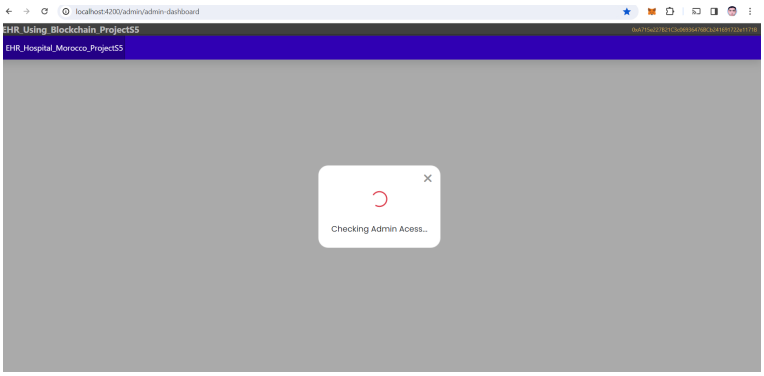


FIG. 4.24 : Authentication on the Blockchain

In this figure 4.25 we demonstrate the admin dashboard that has the authority to add docotor and patient ,also he can consult total patient and doctors and nurses .
The ADDRESS **0xA715e227B21C3c06936476BCb** above represent the the ID of this User Admin from GANACHE BLOCKCHAIN and the **99.98 ETH** represent the balance of him .

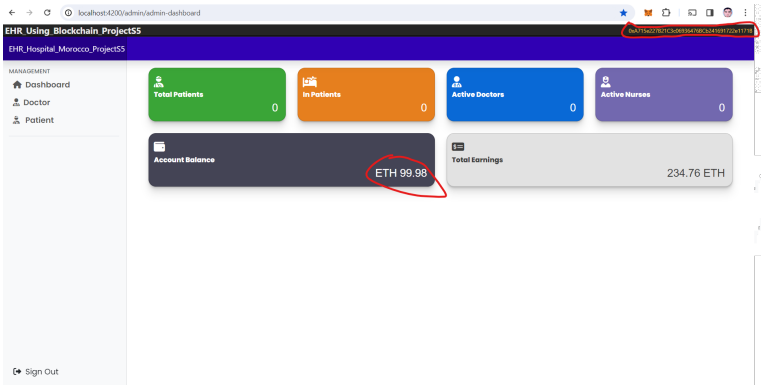


FIG. 4.25 : admin-dashboard

In this figure 4.26 and 4.27 represent the forms for adding a docotor or patient to the EHRBB (Electronic Health Record based on Blockchain).

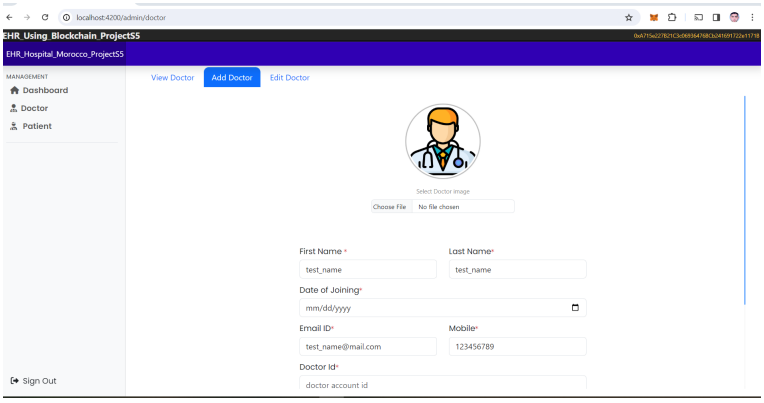


FIG. 4.26 : Add docotor on The blochchain .

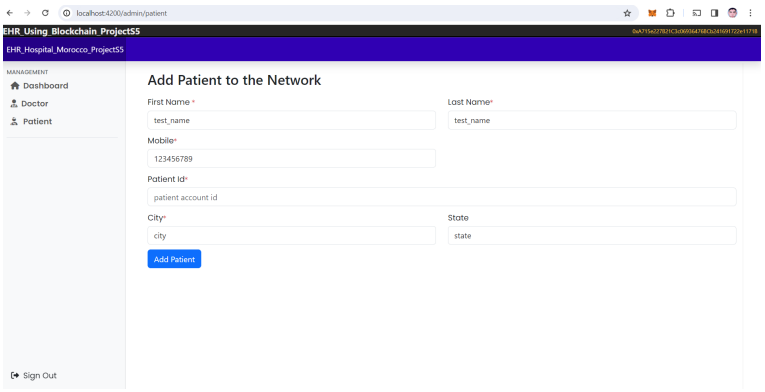


FIG. 4.27 : Add the patient on the blockchain .

4.3 Conclusion

In conclusion, this chapter provides an overview of the implemented components and the tools utilized in this project. However, it's essential to note that the project is ongoing and requires a significant amount of time for its completion.

Conclusion and Perspectives

In conclusion, this project represents a simple widget preliminary implementation of a Generic Patient-Centered Blockchain-Based Electronic Health Record (EHR) Management System. The primary goal is to explore and propose an innovative solution for safeguarding patient and doctor data through a decentralized ledger. The utilization of blockchain technology holds significant promise in addressing security and privacy concerns within the healthcare sector.

While the current simple implementation includes foundational functionalities, it is important to acknowledge that this is just the beginning of a more extensive and comprehensive system. Several aspects can be further developed and enhanced to achieve a fully functional and scalable EHR management solution.

Bibliography

[1] https://www.sciencedirect.com/science/article/pii/S131915782100207X

[2] https://jacobdjaelani.azurewebsites.net/blockchain/

[3] https://4irelabs.com/articles/choosing-a-distributed-ledger-technology/

[4] https://www.researchgate.net/publication/355223429_Quantum_solutions_to_possible_challenges_of_Blockchain_technology

[5] https://www.academia.edu/44112222/Melanie_Swan_Blockchain_BLUEPRINT_FOR_A_NEW_ECONOMY

[6] https://www.ranktracker.com/blog/the-impact-of-blockchain-on-marketing-automation/

[7] https://blog.cfte.education/what-is-p2p-network-blockchain/

[8] https://www.researchgate.net/figure/Flow-of-a-transaction-in-Blockchain-technology fig3_355223429

[9] https://www.researchgate.net/publication/355297166_Systematic_Literature_Review_of_Challenges_in_Blockchain_Scalability

[10] https://101blockchains.com/introduction-to-blockchain-features/

[11] https://builtin.com/blockchain/blockchain-applications

[12] https://smcmaroc.org/wp-content/uploads/2023/01/White-paper-on-e-health-in-Morocco.pdf

[13] https://www.semanticscholar.org/paper/Hyperledger-Healthchain%3A-Patient-Centric-IPFS-Based-Mani-Manickam/497d9dc0eac7c8dd43c2b53857fc25acdef6ae

[14] https://www.deltecbank.com/2023/05/23/ethereums-smart-contracts-explained/#:~:text=Smart%20contracts%20are%20self%2Dexecuting,which%20the%20Ethereum%20blockchain%20stores

[15] https://docs.ipfs.tech/concepts/what-is-ipfs/#defining-ipfs

[16] https://trufflesuite.com/docs/ganache/

[17] https://trufflesuite.com/docs/truffle/

[18] https://www.researchgate.net/publication/361234744_An_InDepth_Review_on_Blockchain_Simulators_for_IoT_Environments

[19] https://www.researchgate.net/publication/364857213_Application_of_blockchain_in_BIM_a_systematic_review

[20] https://docs.metamask.io/wallet/

[21] https://www.researchgate.net/figure/EHR-architecture-of-our-system_fig1_358356059

[22] Thearticle"Digitalchangeappliedtohealth:Whatcontributionsdoesdigitalizationmaketo publishedinthejournalIjafamein2022.https://www.ijafame.org/index.php/ijafame/article/download/613/522/

[23] Thearticle"Digitaltransformation:Moroccoonthepathtohealthtech", publishedinthemagazineLaVieécoin2022.https://www.lavieeco.com/affaires/transformation-digitale-le-maroc-sur-le-chemin-de-la-healthtech/

[24] https://ethereum.org/

[25] https://www.blockchain.com/learning-portal/ether-basics

[26] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1761894/

[27] https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/big%20data%20the%20next%20frontier%20for%20innovation/mgi_big_data_exec_summary.pdf

[28] https://www.researchgate.net/publication/312596137_Big_data_The_next_frontier_for_innovation_competition_and_productivity