Logo

Description automatically generated

**Capital University of Science & Technology**

**Islamabad, Pakistan**

**Semester Project**

**BLOCKCHAIN**

**Afshan Liaquat**

**Rimsha Liaquat**

**BCS201019**

**12st june, 2023**

***MEDICAL CHAIN***

**Spring 2023 | Department of Computer Science - FOC**

Contents

[Medical-chain 2](#_Toc137453593)

[Block Chain in Medical-Chain 3](#_Toc137453594)

[Etherium Block Chain : 3](#_Toc137453595)

[Structure of block chain in Medical-Chain: 4](#_Toc137453596)

[Consensus Model use by Etherium in Medical-Chain: 5](#_Toc137453597)

[Data Structure in Medical Chain: 5](#_Toc137453598)

[implementation of medical chain in Block chain: 6](#_Toc137453599)

[Tools and Technology used in Medical-Chain 7](#_Toc137453600)

[Language: 8](#_Toc137453601)

[Access control smart contracts: 9](#_Toc137453602)

[Payment smart contracts: 12](#_Toc137453603)

[Domain model : 15](#_Toc137453604)

# Medical-chain

Medical chain is a block chain-based healthcare company that aims to improve the security, privacy, and accessibility of electronic health records (EHRs) and telemedicine services. Medical chain utilizes block chain technology to provide a decentralized and secure platform for storing and sharing medical data. By leveraging the inherent properties of block chain, such as immutability, transparency, and cryptographic security, Medical chain aims to address some of the key challenges in healthcare data management.

One of the primary offerings of Medical chain is the Medicaid platform, which allows patients to have full control over their medical records. With Medicaid, individuals can grant permission to healthcare providers and other relevant parties to access their medical data securely. This ensures that patients have ownership of their health information and can easily share it with different providers, eliminating the need for redundant tests and improving care coordination. Medical chain provides secure and private video consultations between patients and healthcare providers. These consultations can be recorded and stored on the block chain, creating an auditable and tamper-proof record of the interaction.

# Block Chain in Medical-Chain

Medical chain uses the specific structure of the block chain being used is utilizes a permission or private block chain structure rather than a public blockchain.A private block chain allows access to a predefined single participant who has been granted allow to join and validate transactions on the network.

This type of block chain is suitable for healthcare applications where privacy and data security are paramount, as it provides greater control over who can participate and access the data. Medical chain uses private based block chain, which is limited in organization or a consortium of organizations. This type of block chain is commonly used in enterprise settings, where multiple parties collaborate within a trusted network.

Private block chains offer enhanced privacy, security, and scalability compared to public block chains. These features align with the requirements of healthcare organizations that handle sensitive patient data. Private based block chain are use to ensure the confidentiality and integrity of healthcare data.

# Etherium Block Chain :

Medical chain utilizes the Ethereum block chain for its operations and services in the UK and globally. Ethereum is one of the most well-known and widely used block chain platforms that supports smart contracts and decentralized applications (DApps). Medical chain leverages the features and capabilities of the Ethereum block chain to ensure the secure storage and sharing of medical records, facilitate patient control over their data, and enable interoperability within the healthcare ecosystem. The Ethereum block chain provides the foundation for the decentralized and immutable storage of patient health records and transactional data. Smart contracts are used to manage access permissions, consent, and data sharing between patients and healthcare providers. This allows for transparent and auditable interactions while ensuring the privacy and security of sensitive medical information.

# Structure of block chain in Medical-Chain:

Medical chain is a block chain-based platform designed to facilitate the secure and efficient management of medical records and enhance patient control over their health data. While the specific implementation and structure may vary, a generalized overview of the structure of Medical chain is:

* Block chain Infrastructure: Medical chain utilizes a block chain infrastructure as the underlying technology. The block chain provides a decentralized and immutable ledger where medical data and transactions are recorded securely. Commonly used block chain platforms, such as Ethereum, are often employed for their robustness and smart contract capabilities.
* Nodes and Network Participants: The block chain network consists of various nodes, which are computers or servers that maintain a copy of the block chain and participate in transaction validation and consensus. Network participants can include healthcare providers, hospitals, patients, researchers, and other stakeholders involved in the healthcare ecosystem.
* Smart Contracts: Smart contracts are self-executing contracts with predefined rules and conditions coded into the block chain. In the case of Medical chain, smart contracts are used to manage permissions, access control, and consent mechanisms for sharing medical data between patients and healthcare providers. They enable secure and auditable interactions while preserving patient privacy.
* Patient Wallets: Patients have personal wallets within the Medical chain platform. These wallets store the patient's health data, such as medical records, test results, and personal health information. Patients have control over their data and can grant access permissions to healthcare providers or other authorized parties.
* Healthcare Provider Interface: Healthcare providers have their own interfaces or applications to interact with Medical chain. These interfaces enable healthcare providers to access patient data, update medical records, and communicate securely with patients. Integration with existing electronic health record (EHR) systems is often implemented to streamline data sharing and interoperability.
* Data Encryption and Security: Medical chain emphasizes data security and privacy. Patient data is encrypted using cryptographic algorithms to protect it from unauthorized access. Access control mechanisms, implemented through smart contracts, ensure that only authorized individuals or organizations can view and interact with the data.
* Telemedicine and Virtual Consultations: Medical chain incorporates telemedicine capabilities, allowing patients to engage in virtual consultations with healthcare providers. Secure video calls, messaging, and file sharing functionalities enable remote medical interactions and advice, reducing the need for in-person visits.
* Token Economy: Medical chain may employ its native crypto currency token, such as Med Tokens (MTN), to facilitate transactions and incentivize participation within the platform. These tokens can be used for accessing services, paying for consultations, or rewarding users who contribute to medical research or data sharing.

# Consensus Model use by Etherium in Medical-Chain:

Ethereum, the block chain platform commonly used by Medical chain, currently utilizes a consensus model known as Proof of Stake (PoS). Under the PoS consensus model, validators are chosen to create new blocks and validate transactions based on the number of crypto currency tokens they hold and are willing to "stake" as collateral. Validators are incentivized to act honestly and maintain the security and integrity of the block chain network, as they can earn additional crypto currency rewards for their participation.

Ethereum 2.0 aims to transition from the existing Proof of Work (PoW) consensus model, which is resource-intensive, to a more scalable and energy-efficient PoS model( Wy POW not use). This upgrade will improve the performance and throughput of the Ethereum network, making it more suitable for handling large-scale applications like Medical chain.

Distributed Ledger: Medical chain utilizes a distributed ledger, which is a fundamental feature of block chain technology. The distributed ledger ensures that medical data is stored across multiple nodes or computers in a network, eliminating the need for a central authority and enhancing data integrity and resilience.

# Data Structure in Medical Chain:

In Medical chain, a block chain-based platform for managing medical records, the underlying data structure primarily revolves around the use of a block chain. The block chain itself can be considered as the core data structure used in Medical chain.

A block chain is a distributed and decentralized ledger that organizes data in a sequential chain of blocks. Each block contains a set of transactions or records, and these blocks are linked together using cryptographic hashes, forming an immutable and transparent history of data.

Within the blocks of the block chain, various data structures can be used to organize and store the medical records and related information. Typically, these data structures include:

* **Merle Trees:** Merkle trees are commonly employed in block chain systems to efficiently verify the integrity and consistency of data within a block. They enable quick verification of the presence and integrity of individual transactions or data elements within the block.
* **Data Payload:** The actual medical records and associated information, such as patient data, test results, diagnoses, and treatments, are stored within the data payload of each transaction or block. The specific organization and format of this data depend on the design and requirements of the Medical chain platform.
* **Pointers and References:** To optimize storage and improve efficiency, block chain systems may utilize pointers or references to link related data across different blocks or transactions. These pointers enable efficient retrieval and access to specific medical records or data elements.
* **Encryption and Data Protection:** Medical chain emphasizes the security and privacy of medical data. As such, cryptographic techniques, such as encryption algorithms, are utilized to protect the confidentiality and integrity of the stored data. This ensures that only authorized individuals or entities can access and decrypt the sensitive medical information.

# implementation of medical chain in Block chain:

The implementation of Medical chain on a block chain involves leveraging block chain technology to create a secure and decentralized platform for managing medical records and facilitating interactions within the healthcare ecosystem. Here is an overview of how Medical chain can be implemented on a block chain:

* **Selecting a Suitable Block chain Platform:** Popular block chain platforms like Ethereum, Hyper ledger Fabric, or private/permission block chains can be considered based on factors such as scalability, privacy, consensus mechanism, and smart contract functionality.
* **Designing the Data Structure:** Determine the structure of the medical data to be stored on the block chain. This includes identifying the essential data elements, such as patient information, medical records, test results, diagnoses, treatments, and any other relevant data. The data structure should be designed to efficiently capture and represent the required information.
* **Smart Contract Development:** Develop smart contracts to define the rules and logic governing the interactions on the Medical-chain platform. Smart contracts can be used to manage patient consent, access control, secure data sharing, and other important functionalities. These contracts ensure that transactions and data sharing occur in a secure and automated manner.
* **Identity and Access Management:** Implement robust identity and access management mechanisms to authenticate and authorize participants on the Medical chain platform. This includes verifying the identities of healthcare providers, patients, and other relevant entities. Access control measures ensure that only authorized parties can view and interact with specific medical data.
* **Integration with Healthcare Providers:** Establish integration with healthcare providers' systems and electronic health record (EHR) systems to securely access and transfer existing medical records to the Medical chain platform. Integration protocols and APIs can be developed to facilitate seamless and secure data exchange.
* **Encryption and Privacy Measures**: Implement encryption techniques to protect the privacy and confidentiality of medical data stored on the block chain. Sensitive information should be encrypted using strong cryptographic algorithms to prevent unauthorized access.
* **User Interfaces and Applications:** Develop user-friendly interfaces, such as web or mobile applications, for patients, healthcare providers, and other stakeholders to interact with the Medical chain platform. These interfaces should provide a seamless experience for accessing, sharing, and managing medical records.
* **Testing and Deployment:** Thoroughly test the implemented Medical chain platform to ensure its functionality, security, and performance. Once testing is completed, deploy the platform to a production environment, ensuring appropriate security measures are in place to protect the integrity of the data.
* **Ongoing Maintenance and Upgrades:** Continuously monitor and maintain the Medical chain platform, addressing any bugs, security vulnerabilities, or performance issues. Stay updated with the latest advancements in block chain technology and healthcare industry standards to incorporate relevant upgrades and enhancements.

# Tools and Technology used in Medical-Chain

Medical chain utilizes a combination of tools and technologies to support its block chain-based platform for managing medical records.

* **Block chain Platforms:** Medical chain may utilize established block chain platforms such as Ethereum, Hyper ledger Fabric, or other private/permission block chain frameworks. These platforms provide the foundation for creating secure, transparent, and decentralized networks.
* **Smart Contracts:** Smart contracts are self-executing agreements that run on the block chain. Medical chain likely utilizes smart contracts to automate and enforce various processes, including patient consent management, access control, and secure data sharing.
* **Cryptography:** Medical chain leverages cryptographic techniques to ensure the privacy, integrity, and security of medical data. This includes encryption algorithms, digital signatures, and cryptographic hashing for secure storage and transmission of sensitive information.
* **Identity and Access Management:** Tools and technologies for identity and access management are crucial for authenticating and authorizing users on the Medical chain platform. This may involve secure login mechanisms, multi-factor authentication, and access control protocols.
* **Integration Protocols and APIs:** Medical chain may provide integration capabilities through protocols and APIs (Application Programming Interfaces) to connect with existing healthcare systems, electronic health records (EHRs), or other medical data sources. These integration tools enable seamless and secure data exchange.
* **Web and Mobile Applications:** Medical chain likely offers user-friendly web and mobile applications for patients, healthcare providers, and other stakeholders to access and interact with the platform. These applications provide intuitive interfaces for managing medical records, granting consent, and communicating securely.
* **Data Encryption and Privacy Tools:** To ensure data privacy, Medical chain may employ encryption tools and privacy-enhancing technologies to protect sensitive information. This includes encryption at rest and in transit, anonymization techniques, and data obfuscation methods.
* **Testing and Security Tools:** Quality assurance and security testing tools are essential to ensure the robustness and reliability of the Medical chain platform. These tools may include code analysis, vulnerability scanning, penetration testing, and monitoring systems.
* **Cloud Infrastructure and Storage:** Medical chain may leverage cloud-based infrastructure, such as Amazon Web Services (AWS) or Microsoft Azure, for scalable storage, computing resources, and network infrastructure.

# Language:

Framework(Ethereum Virtual Machine (EVM).

Solidity is a programming language used to write smart contracts on the Ethereum block chain. Smart contracts are self-executing contracts with the terms of the agreement between buyer and seller being directly written into code. They allow for the automation of complex financial and contractual arrangements, without the need for intermediaries such as banks or lawyers.

Solidity is a statically typed, contract-oriented programming language that is influenced by C++, Python, and JavaScript. It is designed to be used for writing smart contracts that can be executed on the Ethereum Virtual Machine (EVM).

Solidity supports inheritance, user-defined types, libraries, and complex user-defined types such as arrays and mappings. It also has built-in support for cryptographic functions, making it easy to implement secure and private transactions.

# Access control smart contracts:

contract Access Control {

// Mapping to store roles and permissions

mapping(address => bool) public healthcareProviders;

mapping(address => bool) public patients;

// Modifier to restrict access to only the contract owner

modifier onlyOwner() {

require(msg.sender == owner, "Only contract owner can call this function");

} // Modifier to restrict access to only healthcare providers

modifier onlyHealthcareProvider() {

require(healthcareProviders[msg.sender], "Only healthcare providers can call this function"); }

// Modifier to restrict access to only patients

modifier onlyPatient() {

require(patients[msg.sender], "Only patients can call this function");

\_;

} // Contract owner address

address private owner;

// Constructor to set the contract owner

constructor() {

owner = msg.sender;

}

// Function to grant healthcare provider role to an address

function grantHealthcareProvider(address provider) public onlyOwner {

healthcareProviders[provider] = true;

}

// Function to revoke healthcare provider role from an address

function revokeHealthcareProvider(address provider) public onlyOwner {

healthcareProviders[provider] = false;

}

// Function to grant patient role to an address

function grantPatient(address patient) public onlyHealthcareProvider {

patients[patient] = true;

}

// Function to revoke patient role from an address

function revokePatient(address patient) public onlyHealthcareProvider {

patients[patient] = false;

}

// Other functions and logic for accessing and managing medical records

}

Consent smart contract:

contract Consent {

// Structure to represent consent details

struct ConsentData {

address patient;

address healthcareProvider;

bool granted;

uint256 expirationDate;

}

// Mapping to store consent records

mapping(uint256 => ConsentData) public consents;

// Event to notify when consent is granted or revoked

event ConsentUpdated(uint256 indexed consentId, address indexed patient, address indexed healthcareProvider, bool granted);

// Function to grant consent

function grantConsent(uint256 consentId, address healthcareProvider, uint256 expirationDate) public {

require(consentId > 0, "Consent ID must be greater than zero");

// Check if consent already exists

require(!consents[consentId].granted, "Consent has already been granted");

// Store consent details

consents[consentId] = ConsentData({

patient: msg.sender,

healthcareProvider: healthcareProvider,

granted: true,

expirationDate: expirationDate

});

// Emit event

emit ConsentUpdated(consentId, msg.sender, healthcareProvider, true);

}

// Function to revoke consent

function revokeConsent(uint256 consentId) public {

require(consentId > 0, "Consent ID must be greater than zero");

// Check if consent exists

require(consents[consentId].granted, "Consent has not been granted");

// Check if the patient is the owner of the consent

require(consents[consentId].patient == msg.sender, "Only the patient can revoke the consent");

// Revoke consent by setting the granted flag to false

consents[consentId].granted = false;

// Emit event

emit ConsentUpdated(consentId, msg.sender, consents[consentId].healthcareProvider, false);

} }

# Payment smart contracts:

contract Payment {

// Struct to represent a payment record

struct PaymentRecord {

address payer; // Address of the payer

address payee; // Address of the payee

uint256 amount; // Amount of the payment

bool isPaid; // Flag to indicate if the payment has been made

}

// Mapping to store payment records based on payment ID

mapping(uint256 => PaymentRecord) public paymentRecords;

// Event to emit when a payment is made

event PaymentMade(uint256 indexed paymentId, address indexed payer, address indexed payee, uint256 amount);

// Function to make a payment

function makePayment(uint256 paymentId, address payee) public payable {

// Check if payment already made

require(paymentRecords[paymentId].payer == address(0), "Payment already made");

// Get the payment amount from the transaction value

uint256 amount = msg.value;

// Create a new payment record

paymentRecords[paymentId] = PaymentRecord({

payer: msg.sender,

payee: payee,

amount: amount,

isPaid: true

});

// Emit the PaymentMade event

emit PaymentMade(paymentId, msg.sender, payee, amount);

}

// Function to get the payment amount for a given payment ID

function getPaymentAmount(uint256 paymentId) public view returns (uint256) {

return paymentRecords[paymentId].amount;

} // Function to allow the payee to withdraw the payment

function withdrawPayment(uint256 paymentId) public {

// Check if the caller is the designated payee

require(paymentRecords[paymentId].payee == msg.sender, "Only the payee can withdraw payment");

// Check if the payment has been made

require(paymentRecords[paymentId].isPaid, "Payment not made yet");

// Get the payment amount

uint256 amount = paymentRecords[paymentId].amount;

// Reset the payment record

paymentRecords[paymentId].isPaid = false;

paymentRecords[paymentId].amount = 0;

// Transfer the payment amount to the payee

payable(msg.sender).transfer(amount);

}

}

# Domain model :

