

Utilization of Blockchain in Medical Healthcare Record using Hyperledger Fabric

V.Vijayakumar¹, K.M.Sabarivelan², J.Tamizhselvan³, B.Ranjith⁴, B.Varunkumar⁵

Department Of Computer Science and Engineering,

Sri Manakula Vinayagar Engineering College, Puducherry

Email: vijayakumarv@smvec.ac.in¹, sabarivelankm@gmail.com², jayatamizhselvan@gmail.com³, ranjithbakhthavachalam@gmail.com⁴, varunji7470@gmail.com⁵

Abstract- Blockchain is a decentralized network technology. Blockchain consists of a number of blocks. The blocks are connected with other through a chain. Hence the name Blockchain. A block consists of a number of transactions. The links between the blocks are made up of hash values. The hash values are calculated using the transactions in a block and the hash value of the previous block. Healthcare is one of the biggest industry. It also remains as a industry which lacks transparency. At crucial situations the patients medical reports are not readily available. Interoperability between medical organizations is not available due to trust issues. Blockchain is a technology which can provide trust and transparency to its participants. Combining blockchain with healthcare can bring a huge change in the healthcare domain. By including frameworks like Hyperledger, we can provide an industrial standard to healthcare industry along with transparency and trust. All the medical data can be stored in a distributed ledger, which can be used at critical periods for examining report details. Blockchain also provides high security to the data. The data in the blockchain will remain tamper proof.

Keywords- Blockchain, hashing, ledger, consensus, health record, Hyperledger Fabric, composer, rest server

1. INTRODUCTION

Blockchain is a distributed ledger. A ledger is a collection of records.

1.1 Traditional Approach

The general approach follows a client-server architecture. The end user will be the client. All the requests from the client are forwarded to the server. All the processes are carried out in the server and the result alone is forwarded to the client. This approach is also termed as a singleton approach. It is so called because there is a single authority, which controls the entire process. Any attack on the server will result in the overall collapse of the system.

1.1.1 Advantages of Client Server Architecture

- Improved data sharing
- Integration of services
- Shared resource
- Increased data processing capability
- Easy Maintenance

1.1.2 Disadvantages of Client Server Architecture

- Overloaded Servers
- Single point of failure.
- Resource unavailability

1.2 The Blockchain Technology

On contrary to client server architecture, the Blockchain consists of data split across various systems. The individual system is termed as a node. The data is accumulated and stored in a block. All the blocks are connected through links. Hence the name Blockchain.

1.2.1 Components of a Blockchain

- Ledger
- Peer Network
- Membership services
- Smart contracts
- Wallet
- Events
- System Management

1.2.2 Actors in a Blockchain

- Architect: The person who designs the backend working of a Blockchain
- Operators: The people who maintain the copy of the distributed ledger.
- Developers: The person who creates a smart contract using a specific language
- Regulator: The person one who ensures the proper data storage and processing
- End Users: They are the actual users of the application, which is built on a Blockchain

1.2.3 Cryptography in Blockchain

In a trustless environment, Blockchain provides the user with data integrity. This is achieved by using a number of cryptographic algorithms. There are four key terms, which are widely used in cryptography. They are secret, key, function, cipher. Secret refers to the data that has to be protected. The key is used for the process of encryption and decryption.

Function refers to methodology by which the data is protected. The encrypted data is known as cipher. The step by step process which is carried out to convert a normal text into a user un understandable format is known as encryption. The reverse format of the above is known as decryption. The Blockchain uses the public and private key for the cryptographic

function. The original text is encrypted using the senders public key. In order to decrypt it the receivers private key is used.

1.2.4 Cryptographic Hashing in Blockchain

All the values in a block are used to find the hash of the particular block. Even a small change in the value can alter the hash largely.

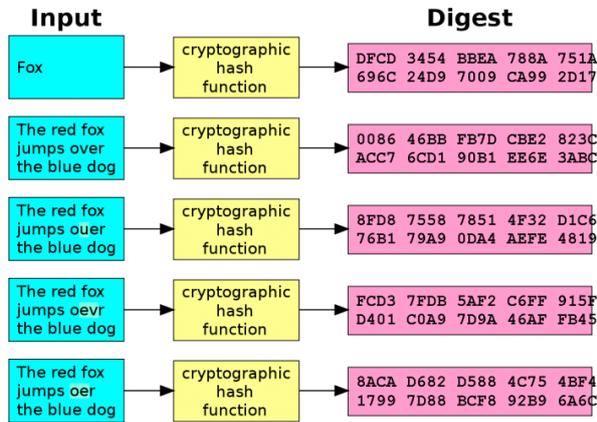


Fig. 1. Hash Value Calculation

1.2.5 Hash links in a Blockchain

In addition to the function the hash is also calculated using the previous blocks hash value.

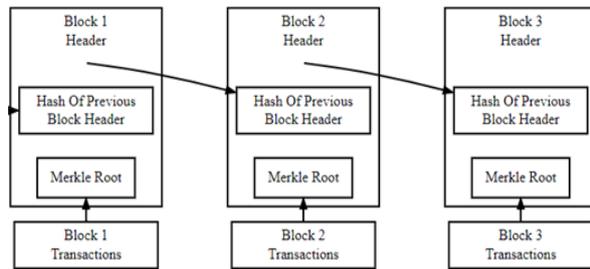


Fig. 2. Blocks of a Blockchain

Thus the change of a small value in the block can affect all the hash values which occur after that.

1.2.6 Merkle tree

It is a data structure which is used to store the hash functions in a Blockchain. It is similar to a tree data structure. It consists of a root node and the child node. The hash values of the leaf nodes are used to find the top hash which is the overall hash of the particular block.

1.2.7 Consensus

Consensus is the methodology of reaching an agreement on adding a set of transactions in a ledger. The widely used consensus in a Blockchain are

- Proof of Work (PoW)
- Proof of Stack (PoS)
- Proof of Elapsed Time (PoET)
- Simplified Byzantine Fault Tolerance (SBFT)
- Proof of Authority (PoA)

2. HYPERLEDGER PROJECTS AND TOOLS

Hyperledger is a collection of open source projects under the Linux Foundation. At present, the hyperledger consists of six frameworks and five other implementation tools.

The frameworks in hyperledger are

- Hyperledger Burrow- It is a permissioned blockchain which is built based on Ethereum smart contract. The project is still under incubation under the Hyperledger community.
- Hyperledger Fabric- It is a permissioned blockchain initially developed by IBM. It was the first project under the Hyperledger community.
- Hyperledger Sawtooth- This project was proposed by Intel. It provides SDK for languages like Java, JavaScript, Python, GO, C++, Rust and Solidity.
- Hyperledger Indy- This project is concerned with digital identity verification.
- Hyperledger Iroha – This project was written in C++. It uses a special type of consensus called as YAC(Yet Another Consensus).
- Hyperledger Grid- This project deals with the supply chain management. It is also under incubation.

The hyperledger tools are

- Hyperledger Caliper- It is used to measure the performance of a particular blockchain.
- Hyperledger Cello- It was proposed by IBM. It provides a toolkit to build projects using Hyperledger.
- Hyperledger Composer- It is built using JavaScript. It provides a set of tools to build a business network.
- Hyperledger Explorer- It is created to provide a user friendly Web-Application. It was a combined project of IBM, Intel.
- Hyperledger Quilt- It provides interoperability between ledger systems. It uses a protocol named Inter Ledger Protocol.
- Hyperledger Ursa- It is used to avoid duplication of Cryptographic works. It consist of two other sub projects.

3. EXISTING SYSTEM

The existing set of rules makes it more complicated for a system to store and handle all records. Even though the storage may be done, retrieving the data and making them secure is a huge task. This is due to a single authority maintenance that is the centralization. In the current system all the data are scattered across various locations. Reports from one health center may not be accepted by others due to trust issues. A track of all the records cannot be

maintained. In order to maintain a track of all medical records, a trust has to be created among all users and organizations. Creating a trust in a centralized network is very difficult. Security could play a major role in containing all data in a centralized server.

4. PROPOSED SYSTEM

In the field of healthcare, blockchain can bring a great difference. The data will be distributed among the nodes. Therefore each node will be responsible for handling its own data. This gives the user a control over their data. Organizations can store the medical information's in a distributed ledger. As the data is decentralized, it will be easy to create trust among various participants.

4.1 Setting up the development environment

All the medical data can be stored on a permissioned blockchain. The blockchain can be built using the Hyperledger Fabric framework. We use a tool called Hyperledger composer to implement the fabric system. The workspace environment should be set up to work with the Hyperledger fabric and Hyperledger composer. There are three steps in the environment setup.

- Installing prerequisites
- Installing tools
- Installing Fabric runtime

4.1.1 Installing prerequisites

The following prerequisites are required to run the Fabric runtime.

- Docker Engine- It is a container technology used to build applications. It is a light weight tool.
- Docker Compose- IT is used in order to define and run multi container Docker applications.
- Nodejs- It is a cross platform and open source JavaScript application.
- GIT- It is a version control system. It is used to keep a track about the changes made to an application.
- Python- It is an interpreted high level language

A bash script is available for the Ubuntu users to install all the prerequisites. The following command can be used to download and run the prerequisites.

curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh
chmod u+x prereqs-ubuntu.sh
./prereqs-ubuntu.sh

4.1.2 Installing tools

The following tools are required to build an application using Hyperledger Fabric.

- Composer-cli- The composer command cline interface provides various commands to

execute the composer application.

- Composer REST server- It is used to generate a REST server which consists of methods such as get, put, post, delete.
- Composer playground- It is web user interface where the composer applications can be deployed and tested.
- Yo generator- It is used to generate the composer template and to create an Angular application.
- The tools can be installed using the following commands

npm install -g composer-cli
npm install -g composer-rest-server
npm install -g composer-playground
npm install -g yo generator-hyperledger-composer

4.1.3 Installing Fabric runtime

The following commands can be used to install the fabric runtime

mkdir fabric-dev-servers
cd fabric-dev-servers
curl -O https://raw.githubusercontent.com/hyperledger/composer-tools/master/packages/fabric-dev-servers/fabric-dev-servers.tar.gz
tar -xvf fabric-dev-servers.tar.gz
export FABRIC_VERSION=hlfv12
./downloadFabric.sh
./startFabric.sh
./createPeerAdminCard.sh

A folder is created for the development setup. The required packages are downloaded. The packages are then extracted. The version of the fabric is exported. Here hlfr12 specifies the version of the Fabric to be installed. Three bash scripts are executed to setup all the dev environments.

After the creation and usage of Fabric environment, we need to stop all the instances and remove all the instances from the peer. The following commands are used for this purpose

./stopFabric.sh
./tearDownFabric.sh

The stopFabric halts all the current instances of the fabric in the container. The tearDownFabric removes all docker files related to fabric.

4.2 Business Network Creation

The yo generator is used to create a business network. It provides a template to create the business network.

4.2.1 Business Network Generation

The following command must be executed to install the business network.

```
yo Hyperledger-composer
```

The command creates a network and asks for the following details:

- Business network name
- Description
- Author name
- Author email
- License
- Namespace

4.2.2 Business Network Modelling

The modelling consists of four resource types. They are

- Assets
- Participants
- Transactions
- Events

The asset specifies all the details required for a record. The asset is as follows:

```
asset TradingCard identified by recordId
{
  o String recordId
  o String recordName
  o String recordDescription
  o GameType organisationType
  default="Government" // If no value is provided, it
  takes the default value
  o String prescription
  o String commants
  o Boolean forTrade
  --> Trader owner
}

enum GameType
{
  o Government
  o Private
}
```

The TradingCard is the name of the asset. In order to identify a asset we require a unique identifier. Here the unique identifier is the recordId.

The participants specify all the members involved in the network. The participant record is as follows:

```
participant Trader identified by participantId
{
  o String participantId
  o String participantName
}
```

The transaction specifies the asset which as to be sent from one to the other.

```
transaction TradeCard
```

```
{
  --> TradingCard card
}
```

Here the transaction is carried out using the trading card.

The event specifies a notification if any transaction has occurred.

```
event TradeNotification
{
  --> TradingCard card
}
```

4.2.3 Adding Business Logic

The business logic specifies when to transfer the asset. The logic is written in JavaScript.

```
async function buyCard(trade) {
  const currentParticipant = getCurrentParticipant();
  // Get information about the transaction instantiator
  if (trade.card.forTrade)
  {
    trade.card.owner = currentParticipant; // Change
    the property
    return
  }
  getAssetRegistry("org.example.biznet.TradingCard")
  .then(assetRegistry => {
    return assetRegistry.update(trade.card); //
    Update the network registry
  })
  .then(() => {
    let event = getFactory().newEvent(
      "org.example.biznet",
      "TradeNotification"
    ); // Get a reference to the event specified in the
    modeling language
    event.card = trade.card;
    emit(event); // Fire off the event
  });
}
```

Here the business logic is used to transfer a medical record from the doctor to the patient.

4.2.4 Adding permissions

The permissions are used to restrict the participants control over the network.

```
rule NetworkAdminUser {
  description: "Grant business network administrators
  full access to user resources"
  participant:
  "org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "*"
  action: ALLOW
}
```

```
rule NetworkAdminSystem {
  description: "Grant business network administrators
full access to system resources"
  participant:
"org.hyperledger.composer.system.NetworkAdmin"
  operation: ALL
  resource: "org.hyperledger.composer.system.*"
  action: ALLOW }
```

```
rule AllParticipantsHaveAccessToAllResources {
  description: "Allow all participants to have access to
all resources and make transactions"
  participant: "ANY"
  operation: ALL
  resource: "org.example.biznet.*"
  action: ALLOW
}
```

4.3 Business network deployment

The above created files are archived so that they can be deployed on the composer. The following command is used to archive the files.

```
composer archive create --sourceType dir --
sourceName .
```

The following command is used to install the business network

```
composer network install --archiveFile cards-trading-
network@0.0.1.bna --card PeerAdmin@hlfv1
```

The following command is used to deploy the business network

```
composer network start --networkName cards-
trading-network --networkVersion 0.0.1 --
networkAdmin admin --networkAdminEnrollSecret
adminpw --card PeerAdmin@hlfv1 --file cards-
trading-admin.card
```

The business card must be imported to the network. It is done by using the following command.

```
composer card import --file cards-trading-
admin.card
```

We can check the business network by pinging to the business network. The ping is done by using the following command.

```
composer network ping --card admin@cards-trading-
network
```

We can also check the business network by using the composer Web user interface.

In order to enable the Web Interface the following command must be executed.

composer-playground

The assets, participants can be created using the web interface.

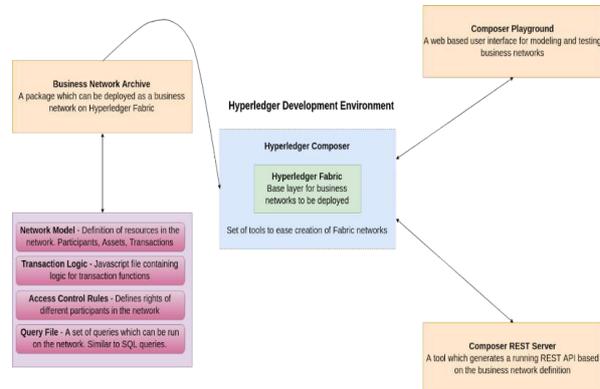


Fig. 3. System Architecture

4.4 Creating a REST API Server

The REST server provide methods such as get, put, post, delete. These methods can be used to interact with the business network layer. The front-end application cannot directly communicate with the network layer. In order to communicate with the business network layer, the front-end application communicates with the REST server which in turn communicates with the business network layer. The following command is used to generate the REST API server.

composer-rest-server

It will ask for the following details

- Business network card name
- Namespace usage
- API key
- Authentication passport
- WebSocket
- TLS Security

4.5 Creating the Angular Application

The yo generator is used to create the Angular application. The Angular application is the front-end which can be used by the participants in the network. The following command is used to generate the Angular application.

yo hyperledger-composer

It asks for the following details:

- Business Network name
- Project name
- Description
- Author name
- Author email
- License
- Business Network Card name

- Connection with existing REST API
- Rest Server address
- Rest Server port
- Namespace usage

5. CONCLUSION

By integrating Hyperledger Fabric, we can record all medical information. We can track the medical organization that has recorded the details. All the transactions will be stored in the Blockchain. By using query tools, we can access the entire medical record of an individual.

Transaction ID	Date, Time	Entry Type	Participant	Status
Transaction 1	2019-02-28 12:30:39	Address	idm@theblockchain.org	Block record
Transaction 2	2019-02-28 11:58:01	Address	idm@theblockchain.org	Block record
Transaction 3	2019-02-28 11:56:11	Address	idm@theblockchain.org	Block record
Transaction 4	2019-02-28 11:55:27	Address	idm@theblockchain.org	Block record
Transaction 5	2019-02-28 11:54:19	Address	idm@theblockchain.org	Block record
Transaction 6	2019-02-28 11:48:15	Address	idm@theblockchain.org	Block record
Transaction 7	2019-02-28 11:48:05	Address	idm@theblockchain.org	Block record
Transaction 8	2019-02-28 11:48:06	Address	idm@theblockchain.org	Block record

Fig. 4. Recorded Transactions

By using Blockchain in healthcare, we can ensure the trust among various organizations. The overall cost for the treatment can be reduced. Initially it will take time to adapt, but for sure it will bring a change in the field of healthcare. By including Hyperledger Fabric, we can provide a industrial standard in the field of healthcare. The entire patient record can be tracked from the start to the end. The overall standard of the healthcare industry can be increased.

REFERENCES

[1] Liam Bell, William J Buchanan, Jonathan Cameron, Owen “Applications of Blockchain Within Healthcare”,Blockchain in Healthcare TodayTM ISSN 2573-8240 online.

[2] David Randall, Pradeep Goel and Ramzi Abujamra,”Blockchain Applications and Use Cases in Health Information Technology”,J Health Med Inform, an open access journal ISSN: 2157-7420.

[3] Marko Hölbl,Marko Kompara,Aida Kamišali and Lili Nemeč Zlatolas ,“A Systematic Review of the Use of Blockchain in Healthcare “,Symmetry 2018, 10, 470 doi:10.3390/sym10100470.

[4] Vishal Rawal,Prashant Mascarenhas,Manan Shah,Sai Sravan Kondaka,“Blockchain for Healthcare”,CitiusTech 2018.

[5] John D. Halamka, MD, Editor-in-Chief, BHTY , ”Real Blockchain Use Cases for Healthcare”,Blockchain in Healthcare TodayTM ISSN 2573-8240..

[6] Thomas F Heston,“A case study in blockchain health care innovation “,International Journal of

Current Research Vol. 9, Issue, 11, pp.60587-60588, November, 2017.

[7] William J. Gordon , Christian Catalini, “Blockchain Technology for Healthcare: Facilitating the Transition to Patient-Driven Interoperability” , Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology.

[8] Alevtina Dubovitskaya Michael,Schumacher,,Fusheng Wang,“Secure and Trustable Electronic Medical Records Sharing using Blockchain” .

[9] Peng Zhang, Douglas C. Schmidt, Jules White,Gunther Lenz,,”Blockchain Technology Use Cases in Healthcare”.

[10] Matthias Mettler, “Blockchain Technology in Healthcare The Revolution Starts Here”, IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom).

[11] Chet Stagnaro “Innovative, Blockchain Uses in Health Care”

[12] <https://hyperledger.github.io/composer/v0.19/introduction/introduction.html>

[13] [https://hyperledger.github.io/composer/latest/reference/JSDOC- README](https://hyperledger.github.io/composer/latest/reference/JSDOC-README)