International Journal of Advance Research in Science and Engineering
Volume No. 11, Issue No. 11, November 2022
www.ijarse.com
IJARSE
ISSN 2319 - 8354

A Blockchain based Secured Electronic Health Record System

[1]Rahul Chokhra, [2]Mr. B.L.Pal, [3]Mr. Shiv Kumar

[1]Department of C.S.E. Mewar University, Chittorgarh, Rajasthan India, rahulvic@gmail.com

[2]Professor, Department of C.S.E. Mewar University, Chittorgarh, Rajasthan

[3]Associate Professor, Department of C.S.E. Mewar University, Chittorgarh, Rajasthan

ABSTRACT

Blockchain is the technology, which has gained a lot of attention after it, was presented by Bitcoin. HealthCare is one of the sectors in which this technology has significant possibility and impacts.Electronic health records (EHRs) are medical information of patients in the form of digital data. Current way of storing EHRs is not very secure. It can be stolen and hacked by third party malicious software and hackers.

Blockchain can provide secured way of storing EHRs. Blockchain works on a ledger system, which is distributed among all the participants. Ledger records are immutable and can not be altered. In this project, we are proposing a hyperledger framework based blockchain system, which stores EHRs in a secured way. It also solve issues of data privacy.

Keywords - blockchain, EHR,smart contract, hyperledger

1.INTRODUCTION

1.1 Introduction to Health care system and use of EHRs

The healthcare system consists of businesses that provide medical services, manufacture medical equipment or drugs. In the past few decades, many advancements have happened in healthcare system. A lot of information is associated with every patient. This information is in paper format.With recent developments, paper based information is now being replaced by digital data completely. This digital format data is called Electronic Health Record. EHRs are containing health information of patients and stored digitally in a network. Hospital or a clinician maintains EHR data over time. The electronic medical records comprise all of the important clinical data which is crucial to that patient's care stored with a specific care provider, including MRI reports, past medical examination, immunizations, laboratory reports, and any form of allergies of the patient.EHR records are real-time records, patient specific records that readily available for a patient or a doctor. EHRs can be distributed with other HealthCare provider so that investigations can be done. EHRs are very helpful in the continuous progress in HealthCare. They have improved accuracy and clarity of health information by reducing the occurrences of errors in records.

It advances the conventional way of storing patient's medical records on paper, which were vulnerable to many threats like natural disasters, theft, war, unauthorized manipulation, etc.

EHRs can be accessed anytime and anywhere. By this it, reduce the chances of repetition of tests, delays in treatment, and making patients well aware to make better decisions. EHRs have made the instant interaction

between doctors and patients possible, whenever needed so it has strengthened the relationship between them.

1.2 Introduction to Blockchain

The definition of a blockchain can be given as "a distributed ledger with smart contracts".

- A ledger contains the transactions associated to a business entity. As part of consensus, organizations communicate with each other and maintain similar ledger states.
- A smart contract is program which can fetch data from ledger and create new records at ledger for transaction. Smart contract can manage one or more entity records. It describes how they are created, updated, and queried.

For example, a supply chain of manufacturers, distributors, suppliers, and retailers might use a set of smart contracts and a distributed ledger to maintain a record of transactions. The record details the movement of goods and services through the supply chain.

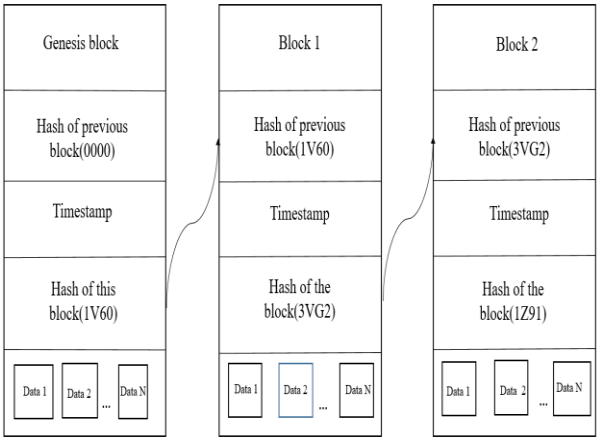


Figure 1.1 Structure of Blockchain

Blockchain structure

• Block: Every block in the Blockchain contains some data and a hash (AKA digital fingerprint) generated from the data contained within the block using cryptography.[2]

• Chain or Linked lists: A hash connects one block to another, chaining them together mathematically. It is the concept that ties block chains together and helps them gain cryptographic trust.

• Network: The network consists of "full nodes." Nodes are analogous to the machines running an algorithm that secures the network.

Blockchain Terminologies

- Cryptography

In Cryptography, information is encrypted using complex mathematical functions. Only authorized person can receive the information.

- Proof of Work (POW)

It is a protocol. The main responsibility of this protocol is to protect system from malicious attacks such as denial-of-service attack (DDoS).

- Smart Contracts

Smart contracts are programs, which are executed once required conditions are met.These are responsible for making integrity of data amongst all participants.

How blockchain is useful

We know information is key for business. Blockchain provides fastest, shared and completely transparent information. Blockchain shares information by immutable ledger that can be accessed only by permissioned network members. A blockchain network can track orders, payments, accounts, production etc. All members are having single view of truth.

2.LITERATURE REVIEW

Akash Takyar [1] has proposed a Blockchain enabled identity management system to control over the identity. In that, whenever a third party need specific details of a person, a notification will be sent to the individuals owning to the identity.

Vardhini B,Shreyas and Sahana R [2] proposed a system which hosts medical record transactions on the Blockchain to create a smart ecosystem. Hyperledger Fabric, a blockchain system is used to host a smart contract called chain code in a containerized technology that integrate application logic.

G. Shobha and M. Suguna [3] proposed a system to protect individual's identity by allowing users to store the documents to a file system with the help of smart contract. They studied scope and provided implementation details for few fields such as - Block chain in Identity Management, Block Chain in insurance Management, Block chain in Transaction or Payment process.

Yogesh Sharma and B. Balamurugan [4] used a Blockchain based framework Hyperledger Fabric and Composer tool to implement the new system. They deployed a blockchain based EHR network and implement basic functionalities in the network.

Dagher et al. [5] introduced a system called Ancile that uses smart contracts for improved data access control and deception in an Ethereum-based block-chain, and uses advanced cryptographic security techniques.

Huanyu Wu, Lunjie Li, Hye-young Paik and Salil S. Kanhere [7] proposed a multi layer blockchain based EHR system which aims to solve the other research gaps and better support the operational hierarchy in health organizations. They implemented a PDP-like data verification mechanisms to protect data integrity.

Hien Do Hoang, Do Thi Thu Hiena, Thach Canh Nhuta [8] proposed a HoloCare System for providing a single source for the patient's personal healthcare record for other systems to access. HoloCare System leverages blockchain features.

A. B. Haque, A. Muniat, P. R. Ullah and S. Mushsharat [9] proposed a smart healthcare system, which is based on blockchain technology. They provided detail how smart contract can make operations less reliant on manual work.

A. Gupta, R. Rodrigues, A. Tripathi, R. Coutinho and J. Gomes [10] proposed a blockchain based mobile app with an integrated wallet for performing transactions to store and retrieve data on the blockchain network.

F. Tang, S. Ma, Y. Xiang and C. Lin [11] worked on creating a new EHRs paradigm, which can help in dealing with the centralized problem of cloud-based EHRs. They proposed an authentication scheme for blockchain-

based EHRs

J. W. Kim, A. R. Lee, M. G. Kim, I. K. Kim and E. J. Lee [12] developed Patient-centric medication history recording system using blockchain, which is directly capturing QR code printed on the envelop by drug store based on prescription. This system adopted Fast Healthcare Interoperability Resources, which is the international health information exchange standard, as way to improve interoperability.

A. Shahnaz, U. Qamar and A. Khalid [13] presented a framework that could be used for the implementation of blockchain technology in healthcare sector for EHR.

A. Fernandes, V. Rocha, A. F. d. Conceição and F. Horita [14] presented a scalable architecture for sharing electronic health records using a multi-channel Hyperledger blockchain.

3.PROBLEM STATEMENT AND OBJECTIVE

Currently, organizations keep a number of medical records of patients in form of EHRs. A lot of information is associated with every patient.

There are several risks involved with advancement of information technology. Cyber-crime is increasing day by day. EHRs have become more vulnerable to attacks by unauthorized users. Personal information of patients can be theft by malicious users as they can gain unauthorized access. Hackers and malicious users can manipulate the EHRs or use them for their personal benefit. Malicious users are now using very modern software and hacking tools for their benefit. Therefore, it is very necessary to securely store EHRs. There should be some mechanism, which safeguard the information. Data should not be breached by attackers. The cloud-based approach of storing EHRs has issue of data security. Professional hackers on cloud can easily compromise password. Cloud storage has very high data security issue. It is important to safely secure the records and protect the privacy of patients from unauthorized users. A blockchain-based approach for EHRs is very secure and efficient way of storing data over a network.

Our objective is to use hyperledger based blockchain network for deploying patient's EHR data on secured network. We are also investigating how smart contract can be developed and used in the context of patient's EHR data.

4.PROPOSED SYSTEM

The proposed system is to create a smart ecosystem that solve the healthcare system's current problem related to security and reliability of medical data.

In our new system, all medical record transactions are hosted on the Blockchain network. EHR framework is used for storing medical transactions.

The stakeholders will have to request permission to access a patient's history and commit the transaction to the distributed ledger.

This new blockchain based system has many advantages. It can provide wider availability and data confidentiality. It is also cost-effective.

Following stakeholders are identified.

- Patients

- Clinicians
- Hospitals
- Insurers

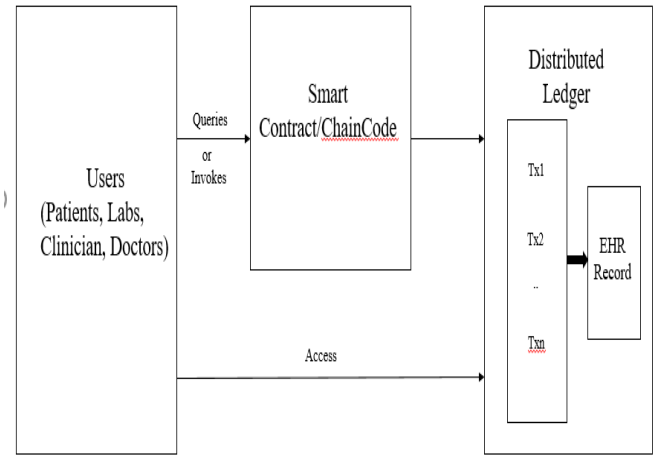


Figure 4.1 System Workflow Diagram

Patient can have different EHR data and as EHR is entirely belong to patient, they have full authority how and whom they want to expose their data.

Clinician capture patients health data and update to ledger. For this patient need to provide access to them.

Admin manages blockchain network. They have responsibility of deploying network and add different contracts in the network. They also monitor encryption and decryption of the transactions.

The transactions are actions performed mostly on the asset in the network like adding a participant in the network, creating a medical record, retrieving specific information from the network, updates in the participant's information.

Following transactions are executed in the system –

1. Create EHR record - By this transaction,we create EHR records in the network. It has fields like ID, owner information and fields which are related to patient's medical history for eg. last consultation with which doctor, consultation date etc. The ID is unique and can be used to identify that specific record in the collection.

2. Grant Access - This transaction provide the access to doctors/clinicians to read or write EHR data.

3. Revoke Access –By this transaction, participant will be removed from the authorized list who can read and write data.

4. Add Participant – By this transaction, new participant can be added to the network.

5. Update Asset –we can update the existing EHR asset using this transaction.

Our proposed system is Hyperledger fabric based blockchain

network.We have two peer organizations and an ordering organization in our network.[6] We will be investigating how smart contract can be created in hyperledger fabric based network.

5.IMPLEMENTATION

To implement this system, we used a blockchain-based framework Hyperledger Fabric [6].

It is a blockchain framework implementation founded by Linux foundation and one of the Hyperledger projects. We are using this framework as it allows components, such as consensus and membership services, to be plug-and-play. It helps container technology to host smart contracts, called "chaincode", that comprise the logic of the system [6]. We have added a new smart contract. For development activities, we have used Linux based Operating system. Some of the prerequisites are mentioned in the table provided below.

Table 5.1 Prerequisites tools and softwares

Sr. No.	Tool/Software	Version
1	Git	2.23.0
2	Curl	7.64.1
3	Docker	19.03.12
4	Docker-Compose	1.27.2
5	Go	1.19.2

Before cloning fabric repository we will be running below commands.

git config --global core.autocrlf false

git config --global core.longpaths true

We will be creating a directory where fabric-samples projects are cloned. We need to run below command.

curl -sSL http://bit.ly/2ysbOFE | bash -s

By this, we will get fabric repository in our local system. The above script will also install fabric and required Hyperledger Fabric Docker Images.

We are deploying a test network by using scripts provided in the fabric-sample repository. We will be creating a new smart contract and deploy to the network.

- It includes two peer organizations and an ordering organization.
- For simplicity, a single node Raft ordering service is configured.
- To reduce complexity, all certificates are issued by the root CAs.
- The sample network deploys a Fabric network with Docker Compose.

Smart contract Implementation code-

The fabric-contract-api provides the contract interface, a high level API for application developers to implement Smart Contracts. Within Hyperledger Fabric, Smart Contracts are also known as Chaincode. Working with this API provides a high-level entry point to writing business logic. We will be doing implementation using Go language.

When using the contract api, each chaincode function that is called is passed a transaction context "ctx", from which you can get the chaincode stub (GetStub()), which has functions to access the ledger (e.g. GetState()) and make requests to update the ledger (e.g. PutState()).[6]

International Journal of Advance Research in Science and Engineering
Volume No. 11, Issue No. 11, November 2022
www.ijarse.com
IJARSE
ISSN 2319 - 8354

EHR Asset Transfer Chaincode -

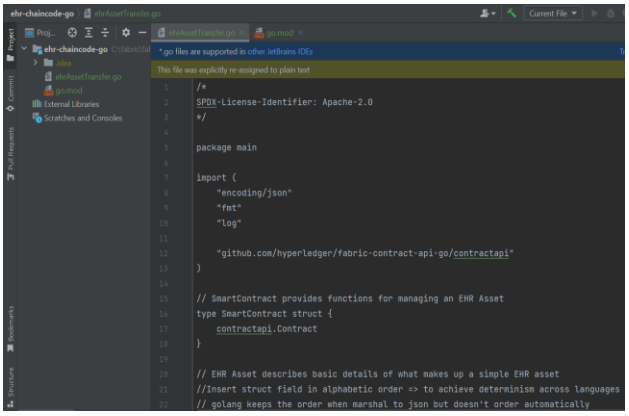


Figure 5.1EHR-Chaincode Project

A chaincode typically handles business logic agreed to by members of the network, so it similar to a "smart contract". A chaincode can be invoked to update or query the ledger in a proposal transaction.

Our application is a basic sample chaincode to initialize a ledger with assets, create, read, update, and delete assets, check to see if an asset exists, and transfer assets from one owner to another. Create directory at below location .

cd fabric/fabric-samples/asset-transfer-basic

mkdir ehr-chaincode-go

cd ehr-chaincode-go

go mod init ehr-chaincode-go

touch ehrAssetTransfer.go

We will import the fabric contract API package and define our SmartContract.

```
import (
    "encoding/json"
    "fmt"
    "log"

    "github.com/hyperledger/fabric-contract-api-go/contractapi"
)
```

Figure 5.2 Required Import

Now we will add a struct EHRAsset to represent simple assets on the ledger

```
type EHRAsset struct {
    BillAmount  int    `json:"BillAmount"`
    ID          string `json:"ID"`
    PatientName string `json:"PatientName"`
    TypeOfEHR   string `json:"TypeOfEHR"`
}
```

Figure 5.3EHR Asset with fields

Initializing the Chaincode -

Next, we will implement the InitLedger function to populate the ledger with some initial data.

```
// InitLedger adds a base set of EHR assets to the ledger
func (s *SmartContract) InitLedger(ctx contractapi.TransactionContextInterface) error {
    assets := []EHRAsset{
        {ID: "ehr1", PatientName: "Mark", TypeOfEHR: "LABREPORT",  BillAmount: 300},
        {ID: "ehr2", PatientName: "Rob", TypeOfEHR: "CONSULTATION", BillAmount: 400},
        {ID: "ehr3", PatientName: "Alex", TypeOfEHR: "LABREPORT",  BillAmount: 500},
        {ID: "ehr4", PatientName: "Amar", TypeOfEHR: "CONSULTATION",  BillAmount: 600},
        {ID: "ehr5", PatientName: "Andy", TypeOfEHR: "CONSULTATION",  BillAmount: 700},
        {ID: "ehr6", PatientName: "Jack", TypeOfEHR: "LABREPORT",  BillAmount: 800},
    }

    for _, asset := range assets {
        assetJSON, err := json.Marshal(asset)
        if err != nil {
            return err
        }

        err = ctx.GetStub().PutState(asset.ID, assetJSON)
        if err != nil {
            return fmt.Errorf("failed to put to world state. %v", err)
        }
    }
```

Figure 5.4Initialization ofEHR Asset

Create an EHR Asset on the ledger -

```
// CreateEHRAsset issues a new EHR asset to the world state with given details.
func (s *SmartContract) CreateEHRAsset(ctx
contractapi.TransactionContextInterface, id string, patientName string, typeOfEHR string, billAmount int) error {
    exists, err := s.EHRAssetExists(ctx, id)
    if err != nil {
        return err
    }
    if exists {
        return fmt.Errorf("the asset %s already exists", id)
    }

    asset := EHRAsset{
        ID:             id,
        PatientName:    patientName,
        TypeOfEHR:      typeOfEHR,
        BillAmount:     billAmount,
    }
    assetJSON, err := json.Marshal(asset)
    if err != nil {
        return err
    }
```

Figure 5.5Create EHR Asset

Now as we have populated the ledger with initial EHR asset and created an asset, we will be reading EHR asset.

Read EHR asset -

Asset from ledger can be fetched by contract interface.

using GetStub() and GetState().Id of required asset need to be passed. For eg. ctx.GetStub().GetState(id)

Update EHR asset -

```
// UpdateEHRAsset updates an existing EHR asset in the world state with provided parameters.
func (s *SmartContract) UpdateEHRAsset(ctx
contractapi.TransactionContextInterface, id string, patientName string,  typeOfEHR string, billAmount int) error {
    exists, err := s.EHRAssetExists(ctx, id)
    if err != nil {
        return err
    }
    if !exists {
        return fmt.Errorf("the asset %s does not exist", id)
    }
    // overwriting original asset with new asset
    asset := EHRAsset{
        ID:             id,
        PatientName:    patientName,
        TypeOfEHR:      typeOfEHR,
        BillAmount:     billAmount,
    }
    assetJSON, err := json.Marshal(asset)
    if err != nil {
        return err
    }
    return ctx.GetStub().PutState(id, assetJSON)
}
```

Figure 5.6Update EHR Asset

Delete an EHR asset –

Asset from ledger can be deleted by contract interface.

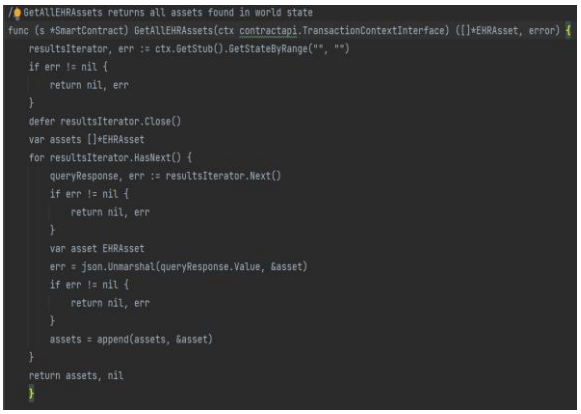
using GetStub() and DelState().Id of asset need to be passed to delete particular asset. For eg. ctx.GetStub().DelState(id)

Check if an asset exist or not at ledger -

Asset from ledger can be verified for existence by contract interface using GetStub() and getState().Id of asset need to be passed. For eg. ctx.GetStub().getState(id)

After getting asset, it can be verified if it is null or not.

Get all the EHR assets from ledger -



```
// GetAllEHRAssets returns all assets found in world state
func (s *SmartContract) GetAllEHRAssets(ctx contractapi.TransactionContextInterface) ([]*EHRAsset, error) {
    resultsIterator, err := ctx.GetStub().GetStateByRange("", "")
    if err != nil {
        return nil, err
    }
    defer resultsIterator.Close()
    var assets []*EHRAsset
    for resultsIterator.HasNext() {
        queryResponse, err := resultsIterator.Next()
        if err != nil {
            return nil, err
        }
        var asset EHRAsset
        err = json.Unmarshal(queryResponse.Value, &asset)
        if err != nil {
            return nil, err
        }
        assets = append(assets, &asset)
    }
    return assets, nil
}
```

Figure 5.7getall EHR Asset from ledger

Function main () is called when smart contract is deployed to the network. It can have call to contract api.

We will be adding all the functions explained earlier to ehrAssetTransfer.go file.

To update the module's requirements, we need to run below commands.

go mod tidy

go mod vendor

6.Result and Analysis

We will be deploying a test network by using scripts that are provided in the fabric-sample repository.

Start the network -

The network can be started using below commands.

cd fabric-samples/test-network

./network.sh up

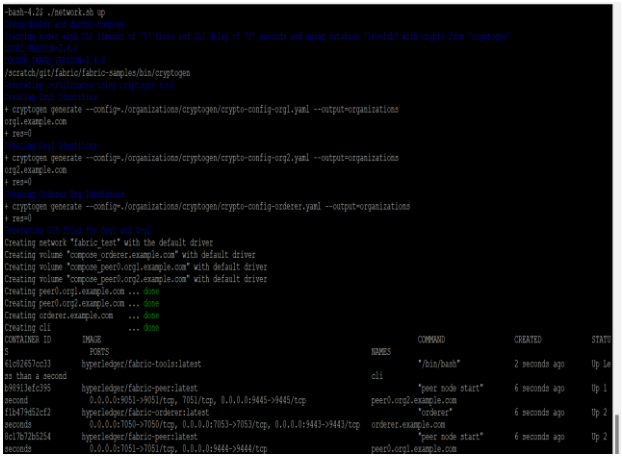


Figure 6.1 network started

As shown in Fig 6.1 we are able to start a network.

Creating a channel -

We willadd new channel to network. Here channel name is myChannel as provided in the below mentioned command.

./network.shcreateChannel -cmychannel

Figure 6.2 channel created

As shown in Fig 6.2 we are able to create a channel on the network.

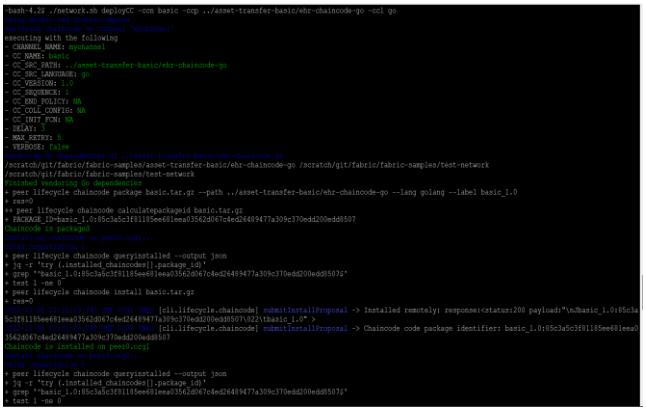
Starting new EHR chaincode on the channel –

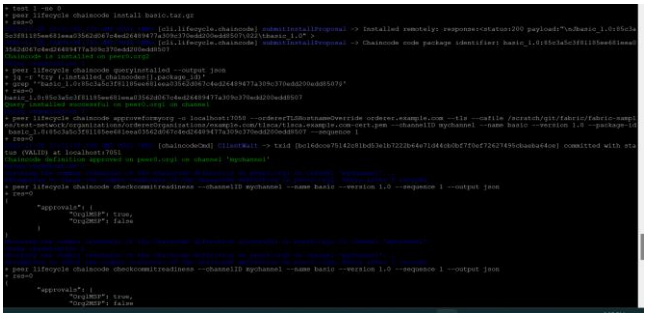
As we have created new channel, now we can use EHR smart contract to communicate with channel ledger. Smart contracts contain the business logic that governs assets on the blockchain ledger. Applications run by members of the network can invoke smart contracts to create assets on the ledger, as well as change and transfer those assets. Applications also query smart contracts to read data on the ledger.

A chaincode is installed on the peers of an organization and then deployed to a channel, where it can then be used to endorse transactions and interact with the blockchain ledger.

We can start a chaincode on the channel using the following command.

./network.shdeployCC-ccnbasic-ccp../asset-transfer-basic/ehr-chaincode-go-cclgo





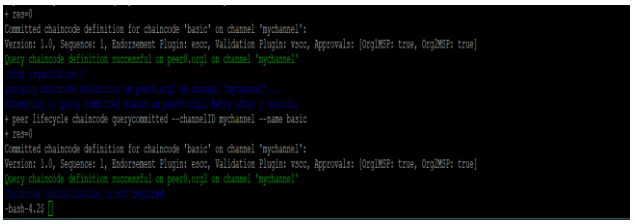


Figure 6.3 chaincode started on channel

As shown in Fig 6.3 we are able to deploy a chaincode on the channel.

deployCC command install the asset-transfer-basic/ehr-chaincode-go chain code on peer0.org1.example.com and peer0.org2.example.com and then deploy the chaincode on the channel mychannel.

Communicating with network -

Before communicating with network we require to set few variables in bashrc files. We can use export commands as mentioned here.

Peer binaries are available in the bin folder of the fabric-samples repository.

export PATH=${PWD}/../bin:$PATH

We need to set FABRIC_CFG_PATH to point to the core.yaml file in the fabric-samples repository.

export FABRIC_CFG_PATH=$PWD/../config/

We can now set below environment variables that allow you to operate the peer CLI as Org. Here ${PWD} is /fabric-samples/test-network

export CORE_PEER_TLS_ENABLED=true

export CORE_PEER_LOCALMSPID="Org1MSP"

export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

export CORE_PEER_ADDRESS=localhost:7051

Initialize the ledger with assets -

Command- peerchaincodeinvoke-olocalhost:7050--ordererTLSHostnameOverrideorderer.example.com--tls--cafile"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem"-Cmychannel-nbasic--peerAddresseslocalhost:7051--tlsRootCertFiles"${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt"--peerAddresseslocalhost:9051--tlsRootCertFiles"${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt"-c'{"function":"InitLedger","Args":[]}'

Output- if successful, we see output as follows.

INFO [chaincodeCmd] chaincodeInvokeOrQuery ->chaincode invoke successful. result: status:200

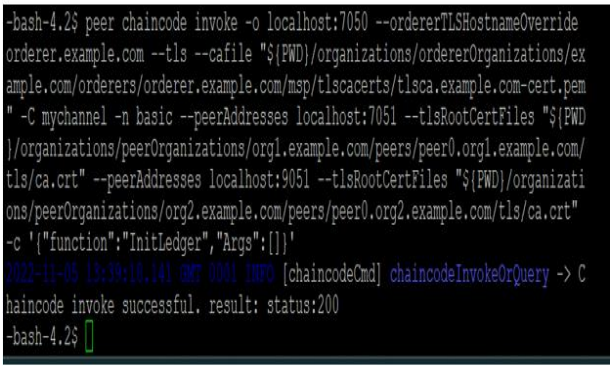


Figure 6.4Initialized ledger

As shown in Fig 6.4, we are able to initialize ledger with few EHR asset.

Get the list of EHR assets from channel ledger -

Command- peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllEHRAssets"]}'

Output-

[{"BillAmount":300,"ID":"ehr1","PatientName":"Mark","TypeOfEHR":"LABREPORT"},{"BillAmount":400,"ID":"ehr2","PatientName":"Rob","TypeOfEHR":"CONSULTATION"},{"BillAmount":500,"ID":"ehr3","PatientName":"Alex","TypeOfEHR":"LABREPORT"},{"BillAmount":600,"ID":"ehr4","PatientName":"Amar","TypeOfEHR":"CONSULTATION"},{"BillAmount":700,"ID":"ehr5","PatientName":"Andy","TypeOfEHR":"CONSULTATION"},{"BillAmount":800,"ID":"ehr6","PatientName":"Jack","TypeOfEHR":"LABREPORT"}]

Read one EHR assets from ledger -

Command- peerchaincodequery-Cmychannel-nbasic-c'{"Args":["ReadEHRAsset","ehr6"]}'

Output-{"BillAmount":800,"ID":"ehr6","PatientName":"Jack","TypeOfEHR":"LABREPORT"}

Update an EHR asset -

Command- peer chaincode invoke -o localhost:7050 −−ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"UpdateEHRAsset","Args":["ehr2","Rob", "CONSULTATION", "800"]}'

Output-

Figure 6.5 Updated asset on ledger

Now when we read the same EHR asset again, value of EHR attribute BillAmount is changed from 400 to 800.

Command- peer chaincode query -C mychannel -n basic -c '{"Args":["ReadEHRAsset","ehr2"]}'

Output-

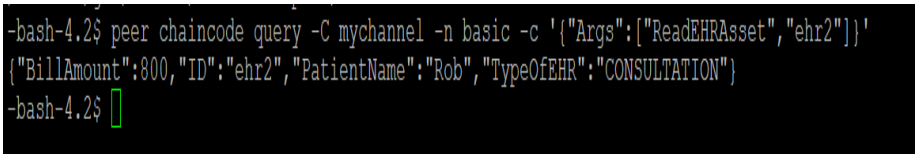


Figure 6.6Read asset on ledger

Create new EHR asset -

Command- peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"CreateEHRAsset","Args":["ehr7","Rahul", "LAB", "800"]}'

Output- 2022-11-06 05:17:51.957 GMT 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200

Now we will read the newly create EHR asset ehr7 using below command.

Command- peer chaincode query -C mychannel -n basic -c '{"Args":["ReadEHRAsset","ehr7"]}'

Result-{"BillAmount":800,"ID":"ehr7","PatientName":"Rahul","TypeOfEHR":"LAB"}

7. CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

As part of this thesis, we have worked on finding how we can leverage Hyperledger fabric to build a blockchain network, create a channel with two organizations with one peer. We have created a new smart contract with asset as EHR. We are able to showcase the basic functionalities using smart contract such as initialize ledger, create EHR asset, update EHR asset, Read EHR asset etc. We are able to secure EHR asset ensuring privacy of patient's data.

7.2 Future Scope

There are few more scope such as creating web application, which can use smart contract developed by us. We produced a basic smart contract, which deal with EHR asset with minimal fields. It can be enhanced to have more attributes and functions.

REFERENCES -

[1] Takyar, A. (2022, January 10). *Blockchain identity management - Enabling control over identity*. LeewayHertz - Software Development Company. https://www.leewayhertz.com/blockchain-identity-management/

[2] V. B, S. N. Dass, S. R and R. Chinnaiyan, "A Blockchain based Electronic Medical Health Records Framework using Smart Contracts," 2021 International Conference on Computer Communication and Informatics (ICCCI), 2021, pp. 1-4, doi: 10.1109/ICCCI50826.2021.9402689.

[3] G. Shobana and M. Suguna, "Block Chain Technology towards Identity Management in Health Care Application," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019, pp. 531-535, doi: 10.1109/I-SMAC47947.2019.9032472.

[4] Yogesh Sharma, B. Balamurugan,"Preserving the Privacy of Electronic Health Records using Blockchain",Procedia Computer Science,Volume 173,2020,Pages 171-180,ISSN 1877-0509.

[5] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustain. Cities Soc.*, vol. 39, pp. 283–297, May 2018, doi: 10.1016/j.scs.2018.02.014.

[6]Getting started - Run fabric — hyperledger-fabricdocs main documentation. (n.d.). A Blockchain Platform for the Enterprise — hyperledger-fabricdocs main documentation. https://hyperledger-fabric.readthedocs.io/en/latest/getting_started_run_fabric.html

[7] H. Wu, L. Li, H. -y. Paik and S. S. Kanhere, "MB-EHR: A Multilayer Blockchain-based EHR," *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2021, pp. 1-3, doi: 10.1109/ICBC51069.2021.9461075.

[8] H. D. Hoang, D. Thi Thu Hien, T. C. Nhut, P. Dang Truc Quyen, P. T. Duy and V. -H. Pham, "A Blockchain-based Secured and Privacy-Preserved Personal Healthcare Record Exchange System," *2021 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, 2021, pp. 1-5, doi: 10.1109/ICMLANT53170.2021.9690542.

[9] A. B. Haque, A. Muniat, P. R. Ullah and S. Mushsharat, "An Automated Approach towards Smart Healthcare with Blockchain and Smart Contracts," *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2021, pp. 250-255, doi: 10.1109/ICCCIS51004.2021.9397158.

[10] A. Gupta, R. Rodrigues, A. Tripathi, R. Coutinho and J. Gomes, "Blockchain for EHR: an off-chain based approach," *2022 IEEE Region 10 Symposium (TENSYMP)*, 2022, pp. 1-6, doi: 10.1109/TENSYMP54529.2022.9864405.

[11] F. Tang, S. Ma, Y. Xiang and C. Lin, "An Efficient Authentication Scheme for Blockchain-Based Electronic Health Records," in IEEE Access, vol. 7, pp. 41678-41689, 2019, doi: 10.1109/ACCESS.2019.2904300.

[12] J. W. Kim, A. R. Lee, M. G. Kim, I. K. Kim and E. J. Lee, "Patient-centric medication history recording system using blockchain," *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2019, pp. 1513-1517, doi: 10.1109/BIBM47256.2019.8983032.

[13] A. Shahnaz, U. Qamar and A. Khalid, "Using Blockchain for Electronic Health Records," in *IEEE Access*,

vol. 7, pp. 147782-147795, 2019, doi: 10.1109/ACCESS.2019.2946373.

[14] A. Fernandes, V. Rocha, A. F. d. Conceição and F. Horita, "Scalable Architecture for sharing EHR using the Hyperledger Blockchain," *2020 IEEE International Conference on Software Architecture Companion (ICSA-C)*, 2020, pp. 130-138, doi: 10.1109/ICSA-C50368.2020.00032.

[15]Blockchain. (2022, September 16). Wikipedia, the free encyclopedia. Retrieved November 16, 2022, from https://en.wikipedia.org/wiki/Blockchain