UNIVERSITY OF DJILLALI LIABES
FACULTY OF EXACT SCIENCES
SIDI BEL-ABBES

Building a private blockchain for storing student records

Exploring blockchain in the education sector

Thesis will be presented by **Mr.Merbah Mohamed Amine**

In front of the jury composed of

President : **Pr.ABDERRAHMAN YOUSFATE**

Jury : **Pr.ADIL TOUMOUH**

Supervisor : **Pr.KAMEL MOHAMMED FARAOUN**

Table of contents

01

Preliminary study

Problem Analysis

Introduction

Administrations of institutions in the Algerian higher education, finds it hard to manage and protect students records. The digitization of this data can help with this problem but according to the traditional method, that depends on centralized databases, can face the problem of security breaches which can hurt and menace its credibility. That is why they stick with the use of paper based methods. However, it is still time-consuming, hard to manage and difficult for students to access their information.

Problems

→ Data accessibility

Hard for students to access their data at any time, It requires them to rely on their institution (a middleman).

→ Trust and privacy concerns

each institution maintains control over its own student data and credentials. It puts data at danger of being tampered with or deleted, which opens the door to fraud and forgery.

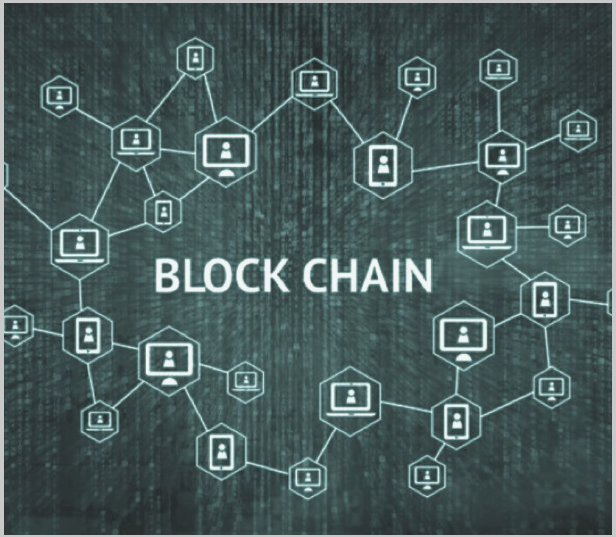
→ Need for a shared common database

Students could have trouble transferring to another higher education institution, whether on the national or the international level. Also, they find it hard to get their credentials to be verified from outer entities like employees, which takes days or even months.

Solution

Blockchain's distributed ledger technology can play an important role in meeting all of those challenges.

"Blockchain is a specific form or subset of distributed ledger technologies, which constructs a chronological chain of blocks, hence the name 'blockchain'. At a technical level, a blockchain can be defined as an immutable ledger for recording transactions, maintained within a distributed network of mutually untrusting peers."



Characteristics

- **Immutability:** It cannot be altered and hence the data cannot be changed with ease
- **Transparency:** Changes in the network are publicly available. Transactions are validated by authorized nodes on the network, so that any change could be detected at any instance of time
- **Traceability:**Timestamp feature in Blockchain helps in recording transactions at each point of time by tracking every movement
- **Decentralization:**refers to the transfer of control and decision-making from a centralized entity to a distributed network.

Uses

Keeping track of student credentials

Protection of intellectual property

ownership of learning credentials

Challenges

require grassroots transformation

The absence of standards

lack of general awareness

02

Basic concepts

Get to know the blockchain

Main Components

the Block and the Chain

Blocks are batches of transactions with a hash of the previous block in the chain

the Node and the network

Blockchain works on the top of a peer-to-peer network that formed of different types of nodes

Cryptography

cryptographic features ensure the reliability and immutability of the data stored on the blockchain

Consensus

Consensus enable network participants to agree on the contents of a blockchain

Types

1

Public Blockchain

They are open source, non-restrictive, fully distributed, decentralized and permissionless

2

Private Blockchain

Referred to as permissioned blockchains where a trusted intermediary is in charge of running it and controlling who can access it

Layers (p1)

Hardware infrastructure layer	Are the servers and peers that forms the network (on premise or on the cloud).
Network layer	responsible for internode communication. Discovery, transactions and block propagation.
Data layer	The structure of the blocks and the transactions stored in it.

Layers (p2)

Consensus layer	Creates a definite set of agreements between nodes across the distributed P2P network.
Application layer	provides a way for a client to communicate with the network.

Goal

building a private/consortium blockchain network that is specific to storing students records. It can enable crucial changes in higher education’s business by drastically reducing data management costs by removing many manual processes, on which its decentralized design provides better privacy and security.

03

Production and design process

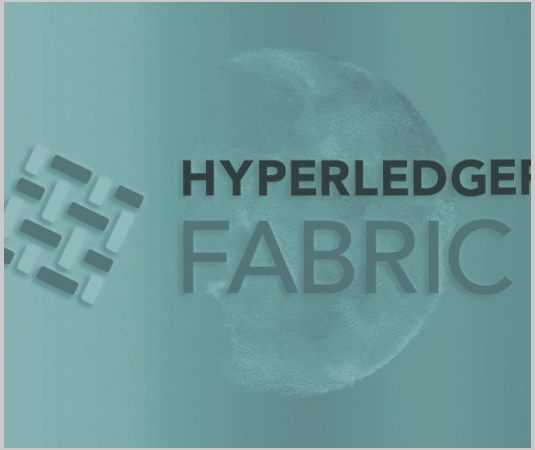
Conception of the system and environment description

Blockchain platforms

lets us take advantage of existing infrastructure, solution, or service related to blockchain in order to facilitate the build of a custom blockchain network ex: Ethereum, Ripple, Hyperledger… .

Hyperledger Fabric

is an enterprise-grade, distributed ledger platform that offers modularity and versatility for a broad set of industry use cases



Fabric Elements

Ledger

contains the current world state of the network and a chain of transaction invocations

Peers

A peer could be a committing peer, endorser peer or an orderer peer

Channels

Each channel has an independent chain of transaction blocks

Chaincode

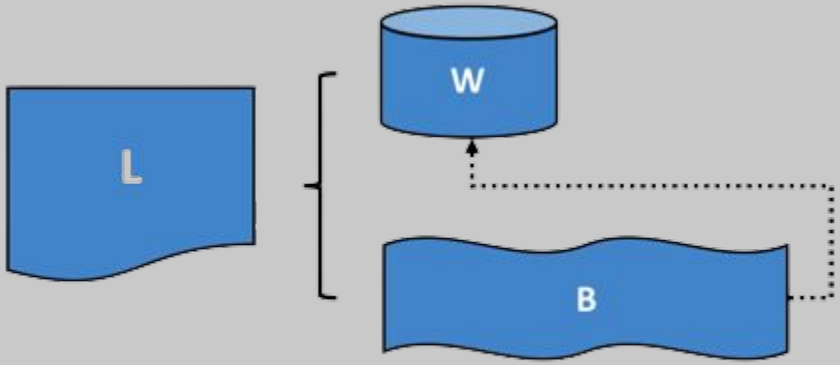
it encapsulates both the asset definitions and the business logic

Consensus

Transaction endorsement, Ordering, Validation and commitment

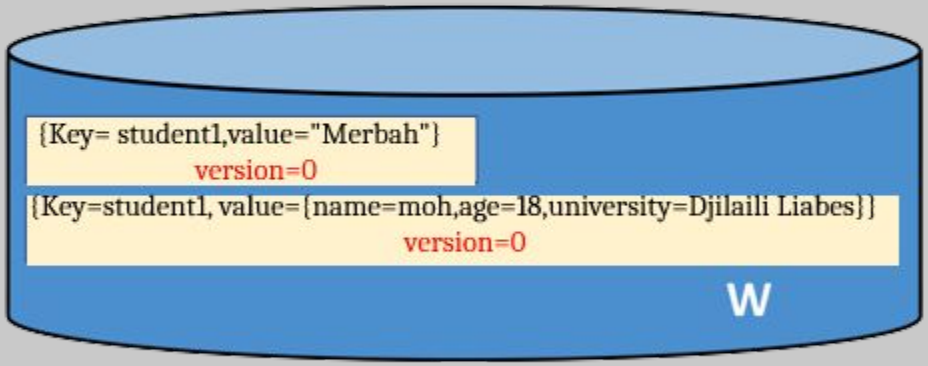
Identities, MSPs and Policies

The ledger



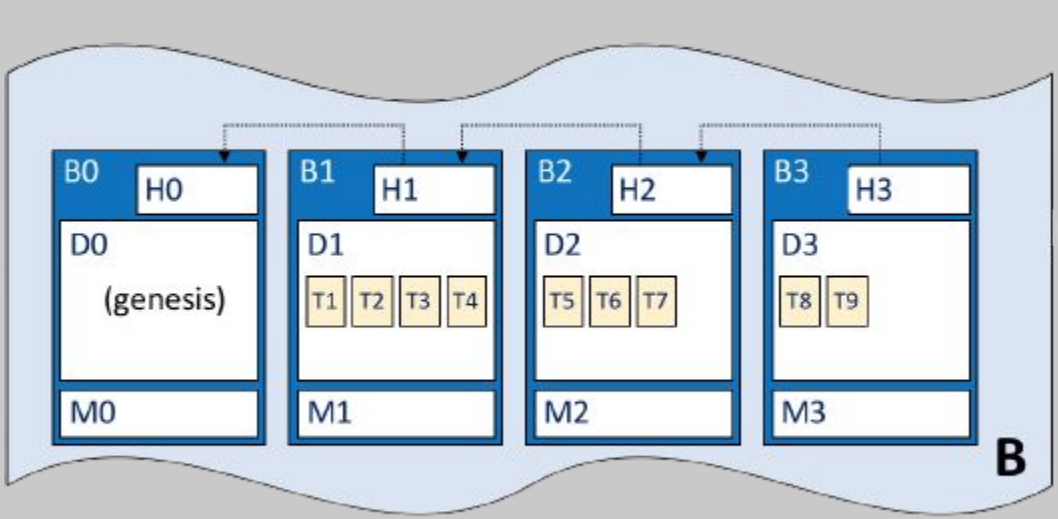
L	Ledger
W	World State
B	Blockchain
L → B, W	L comprises B and W
W ← B	B determines W

World state database



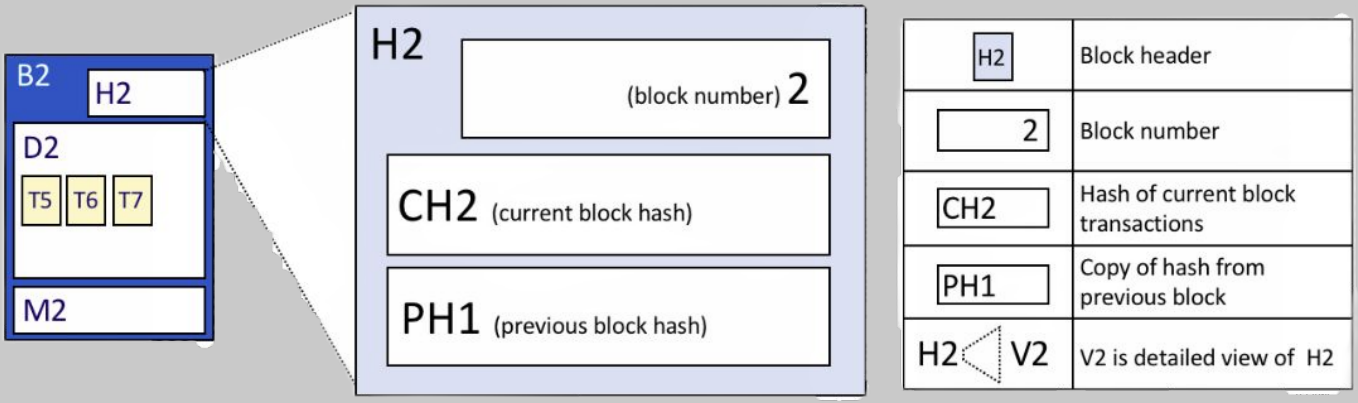
W	Ledger world state
{key=**K**, value = **V** } **version=0**	A ledger state with **key=K**. It contains a set of facts expressed as a simple value, **V**. The state is at version 0.
{key=**K**, value = {**KV**} } **version=0**	A ledger state with **key=K**. It contains a set of facts expressed as a set of key-value pairs {**KV**}. The state is at version 0.

The chain and the Block structure

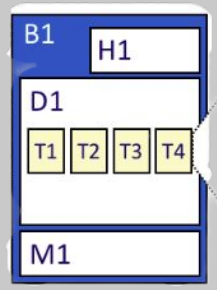


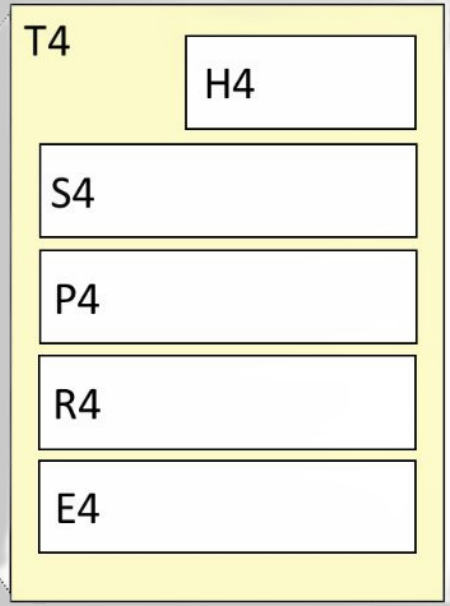
B	Blockchain
B1	Block
H3	Block header
D1	Block data
T5	Transaction
M3	Block metadata
H1 H2	H2 is chained to H1

Blocks Header structure



Transactions structure





T4	Transaction
H4	Header
S4	Signature
P4	Proposal
R4	Response
E4	Endorsements
T4 V4	V4 is detailed view of T4

Peer roles

Committing peer

Each peer maintains the current snapshot of the current state of the ledger

Endorser peer

have chaincode installed. they simulate the transaction execution

Orderer peer

Orderers receive endorsed transactions and assemble them into blocks

Anchor peer

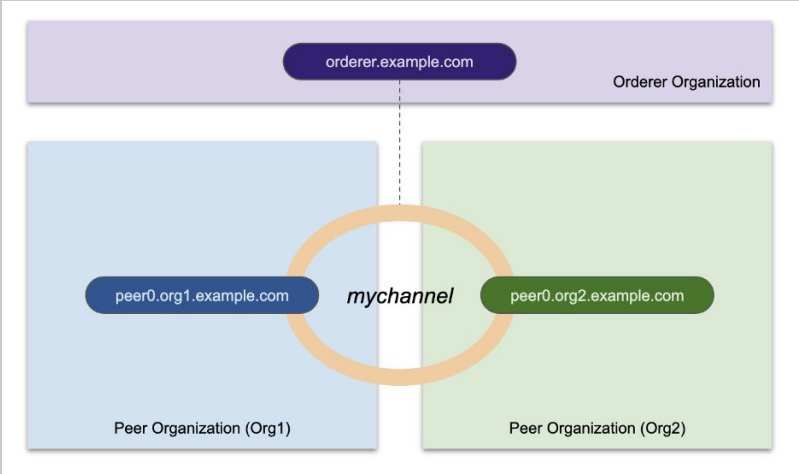
serve as an intermediary between peers from its organization and peers from an external one

Leader peer

take the responsibility of distributing the transactions from the orderer to committing peers of the same organization

Channels

- Channels allow participants to establish a communication path between a subset of participant.
- each channel has a completely separate ledger. This means a completely separate blockchain, and completely separate world states.
- Each peer that joins a channel, has its own identity given by a Membership Service Provider (MSP)

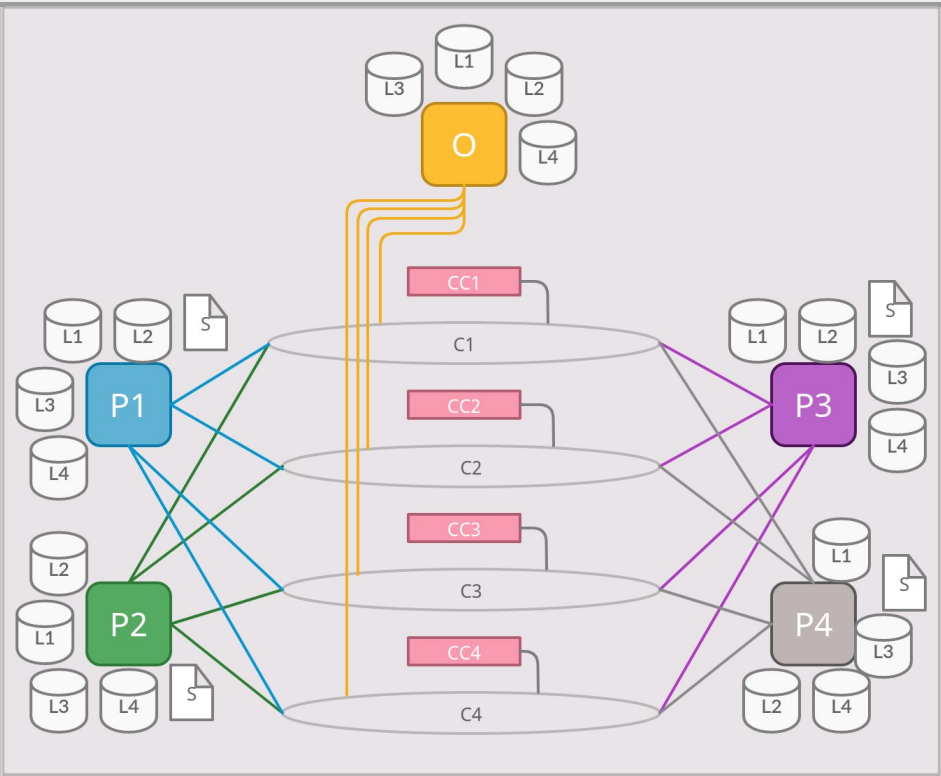


Membership Service Provider (MSP)

MSP is a set of folders that are added to the configuration of the network and is used to define an organization both inwardly (organizations decide who its admins are) and outwardly (by allowing other organizations to validate that entities have the authority to do what they are attempting to do)

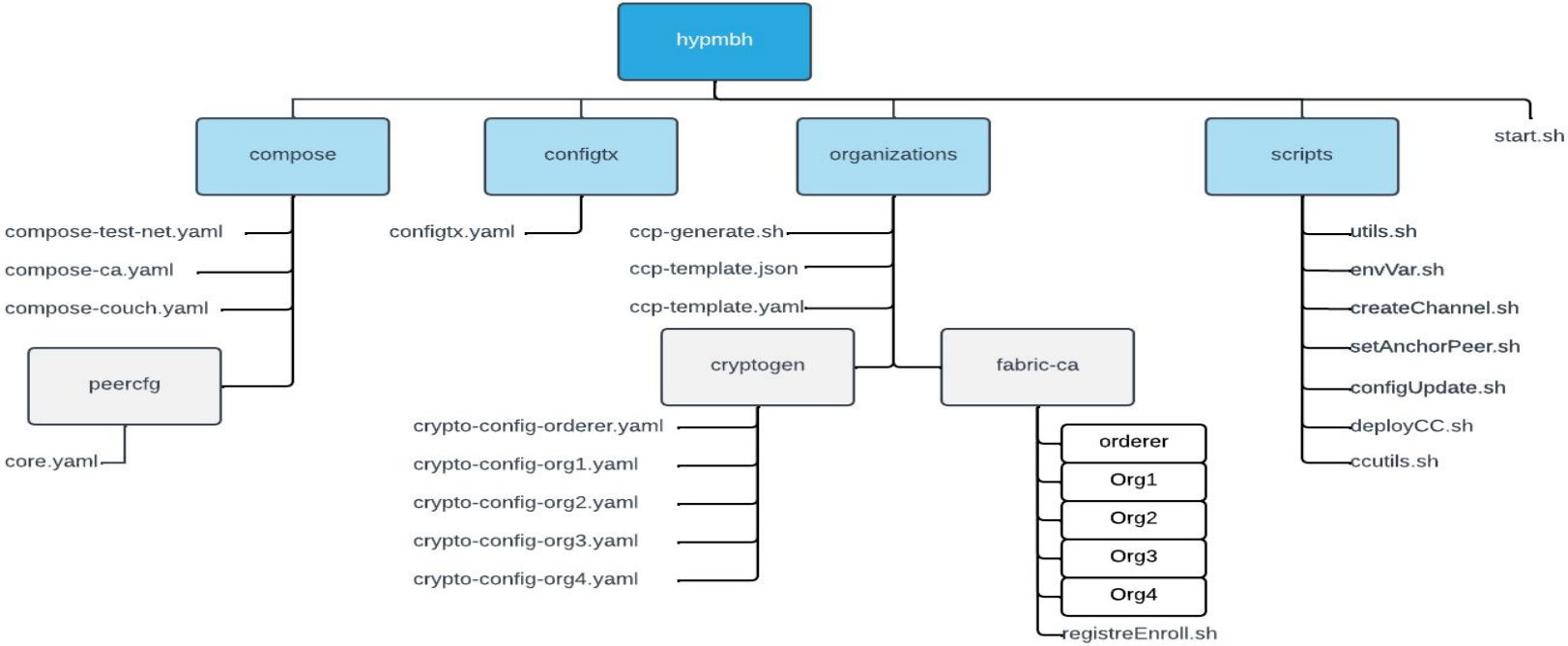
Whereas Certificate Authorities generate the certificates that represent identities, the MSP contains a list of permissioned identities.It is the MSP that turns an identity into a role by identifying specific privileges an actor has on a node or channel.

Network design



- This network composed of four peers of four organizations and one ordered organization.
- It contains four channels, each one of it has its own role.
- Each organization is part of all the channels therefore holds a copy of their respective chaincode

Network Implementation



Network launch

./start.sh up -ca -s couchdb

1- Starting the CAs containers and registering the CAs admins

```
Using docker and docker-compose
Starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'couchdb' with crypto from 'Certificate Authorities'
Generating certificates using Fabric CA
Creating network "fabric_test" with the default driver
Creating ca_org3    ... done
Creating ca_org2    ... done
Creating ca_orderer ... done
Creating ca_org1    ... done
Creating ca_org4    ... done
```

```
Creating Org1 Identities
Enrolling the CA admin
+ fabric-ca-client enroll -u https://admin:adminpw@localhost:7054 --caname ca-org1 --tls.certfiles /home/mbh/Desktop/project/hypmbh/organizations/fabr
ic-ca/org1/ca-cert.pem
2022/06/01 08:57:11 [INFO] Created a default configuration file at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/f
abric-ca-client-config.yaml
2022/06/01 08:57:11 [INFO] TLS Enabled
2022/06/01 08:57:11 [INFO] generating key: &{A:ecdsa S:256}
2022/06/01 08:57:11 [INFO] encoded CSR
2022/06/01 08:57:11 [INFO] Stored client certificate at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/msp/signcert
s/cert.pem
2022/06/01 08:57:11 [INFO] Stored root CA certificate at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/msp/cacerts
/localhost-7054-ca-org1.pem
2022/06/01 08:57:11 [INFO] Stored Issuer public key at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/msp/IssuerPub
licKey
2022/06/01 08:57:11 [INFO] Stored Issuer revocation public key at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/ms
p/IssuerRevocationPublicKey
```

Network launch

2- registering the Org peers, admins and users

```
Registering peer0
+ fabric-ca-client register --caname ca-org1 --id.name peer0 --id.secret peer0pw --id.type peer --tls.certfiles /home/mbh/Desktop/project/hypmbh/organ
izations/fabric-ca/org1/ca-cert.pem
2022/06/01 08:57:11 [INFO] Configuration file location: /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/fabric-ca-cl
ient-config.yaml
2022/06/01 08:57:11 [INFO] TLS Enabled
2022/06/01 08:57:11 [INFO] TLS Enabled
Password: peer0pw
Registering user
+ fabric-ca-client register --caname ca-org1 --id.name user1 --id.secret user1pw --id.type client --tls.certfiles /home/mbh/Desktop/project/hypmbh/org
anizations/fabric-ca/org1/ca-cert.pem
2022/06/01 08:57:12 [INFO] Configuration file location: /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/fabric-ca-cl
ient-config.yaml
2022/06/01 08:57:12 [INFO] TLS Enabled
2022/06/01 08:57:12 [INFO] TLS Enabled
Password: user1pw
Registering the org admin
+ fabric-ca-client register --caname ca-org1 --id.name org1admin --id.secret org1adminpw --id.type admin --tls.certfiles /home/mbh/Desktop/project/hyp
mbh/organizations/fabric-ca/org1/ca-cert.pem
2022/06/01 08:57:12 [INFO] Configuration file location: /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/fabric-ca-cl
ient-config.yaml
2022/06/01 08:57:12 [INFO] TLS Enabled
2022/06/01 08:57:12 [INFO] TLS Enabled
Password: org1adminpw
```

Network launch

3- generating all the peer's MSPs and certificates

```
Generating the peer0 msp
+ fabric-ca-client enroll -u https://peer0:peer0pw@localhost:7054 --caname ca-org1 -M /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations
/org1.example.com/peers/peer0.org1.example.com/msp --csr.hosts peer0.org1.example.com --tls.certfiles /home/mbh/Desktop/project/hypmbh/organizations/f
abric-ca/org1/ca-cert.pem
2022/06/01 08:57:12 [INFO] TLS Enabled
2022/06/01 08:57:12 [INFO] generating key: &{A:ecdsa S:256}
2022/06/01 08:57:12 [INFO] encoded CSR
2022/06/01 08:57:12 [INFO] Stored client certificate at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/peers/peer0.
org1.example.com/msp/signcerts/cert.pem
2022/06/01 08:57:12 [INFO] Stored root CA certificate at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/peers/peer0
.org1.example.com/msp/cacerts/localhost-7054-ca-org1.pem
2022/06/01 08:57:12 [INFO] Stored Issuer public key at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/peers/peer0.o
rg1.example.com/msp/IssuerPublicKey
2022/06/01 08:57:12 [INFO] Stored Issuer revocation public key at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/pe
ers/peer0.org1.example.com/msp/IssuerRevocationPublicKey
Generating the peer0-tls certificates
+ fabric-ca-client enroll -u https://peer0:peer0pw@localhost:7054 --caname ca-org1 -M /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations
/org1.example.com/peers/peer0.org1.example.com/tls --enrollment.profile tls --csr.hosts peer0.org1.example.com --csr.hosts localhost --tls.certfiles /
home/mbh/Desktop/project/hypmbh/organizations/fabric-ca/org1/ca-cert.pem
2022/06/01 08:57:12 [INFO] TLS Enabled
2022/06/01 08:57:12 [INFO] generating key: &{A:ecdsa S:256}
2022/06/01 08:57:12 [INFO] encoded CSR
2022/06/01 08:57:12 [INFO] Stored client certificate at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/peers/peer0.
org1.example.com/tls/signcerts/cert.pem
2022/06/01 08:57:12 [INFO] Stored TLS root CA certificate at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/peers/p
eer0.org1.example.com/tls/tlscacerts/tls-localhost-7054-ca-org1.pem
2022/06/01 08:57:12 [INFO] Stored Issuer public key at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/peers/peer0.o
rg1.example.com/tls/IssuerPublicKey
2022/06/01 08:57:12 [INFO] Stored Issuer revocation public key at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/pe
ers/peer0.org1.example.com/tls/IssuerRevocationPublicKey
```

Network launch

4- generate all admin's and user's MSPs and certificates

```
Generating the user msp
+ fabric-ca-client enroll -u https://user1:user1pw@localhost:7054 --caname ca-org1 -M /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations
/org1.example.com/users/User1@org1.example.com/msp --tls.certfiles /home/mbh/Desktop/project/hypmbh/organizations/fabric-ca/org1/ca-cert.pem
2022/06/01 08:57:12 [INFO] TLS Enabled
2022/06/01 08:57:12 [INFO] generating key: &{A:ecdsa S:256}
2022/06/01 08:57:12 [INFO] encoded CSR
2022/06/01 08:57:12 [INFO] Stored client certificate at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/users/User1@
org1.example.com/msp/signcerts/cert.pem
2022/06/01 08:57:12 [INFO] Stored root CA certificate at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/users/User1
@org1.example.com/msp/cacerts/localhost-7054-ca-org1.pem
2022/06/01 08:57:12 [INFO] Stored Issuer public key at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/users/User1@o
rg1.example.com/msp/IssuerPublicKey
2022/06/01 08:57:12 [INFO] Stored Issuer revocation public key at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/us
ers/User1@org1.example.com/msp/IssuerRevocationPublicKey
Generating the org admin msp
+ fabric-ca-client enroll -u https://org1admin:org1adminpw@localhost:7054 --caname ca-org1 -M /home/mbh/Desktop/project/hypmbh/organizations/peerOrgan
izations/org1.example.com/users/Admin@org1.example.com/msp --tls.certfiles /home/mbh/Desktop/project/hypmbh/organizations/fabric-ca/org1/ca-cert.pem
2022/06/01 08:57:12 [INFO] TLS Enabled
2022/06/01 08:57:12 [INFO] generating key: &{A:ecdsa S:256}
2022/06/01 08:57:12 [INFO] encoded CSR
2022/06/01 08:57:13 [INFO] Stored client certificate at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/users/Admin@
org1.example.com/msp/signcerts/cert.pem
2022/06/01 08:57:13 [INFO] Stored root CA certificate at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/users/Admin
@org1.example.com/msp/cacerts/localhost-7054-ca-org1.pem
2022/06/01 08:57:13 [INFO] Stored Issuer public key at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/users/Admin@o
rg1.example.com/msp/IssuerPublicKey
2022/06/01 08:57:13 [INFO] Stored Issuer revocation public key at /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/us
ers/Admin@org1.example.com/msp/IssuerRevocationPublicKey
```

Network launch

5- running the peers containers and their databases containers

```
Generating CCP files for Org1 Org2 Org3 and Org4
Creating volume "compose_orderer.example.com" with default driver
Creating volume "compose_peer0.org1.example.com" with default driver
Creating volume "compose_peer0.org2.example.com" with default driver
Creating volume "compose_peer0.org3.example.com" with default driver
Creating volume "compose_peer0.org4.example.com" with default driver
WARNING: Found orphan containers (ca_orderer, ca_org3, ca_org2, ca_org1, ca_org4) for this project. If you removed or renamed this service in your com
pose file, you can run this command with the --remove-orphans flag to clean it up.
Creating couchdb3            ... done
Creating couchdb2            ... done
Creating couchdb0               ... done
Creating couchdb1            ... done
Creating orderer.example.com ... done
Creating peer0.org3.example.com ... done
Creating peer0.org4.example.com ... done
Creating peer0.org2.example.com ... done
Creating peer0.org1.example.com ... done
Creating cli                    ... done
```

Channels creation

./start.sh CreateChannel -c <channel name>

1- generation of the genesis block that contains the configuration choices of that channel

```
Using docker and docker-compose
Creating channel 'channel1'.
If network is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'couchdb
Network Running Already
Generating channel create transaction 'channel1.tx'
+ configtxgen -profile FourOrgsApplicationGenesis -outputBlock ./channel-artifacts/channel1.block -channelID channel1
2022-06-01 10:31:08.439 CET 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2022-06-01 10:31:08.447 CET 0002 INFO [common.tools.configtxgen.localconfig] completeInitialization -> orderer type: etcdraft
2022-06-01 10:31:08.447 CET 0003 INFO [common.tools.configtxgen.localconfig] completeInitialization -> Orderer.EtcdRaft.Options unset, setting to tick
_interval:"500ms" election_tick:10 heartbeat_tick:1 max_inflight_blocks:5 snapshot_interval_size:16777216
2022-06-01 10:31:08.447 CET 0004 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/mbh/Desktop/project/hypmbh/configtx/c
onfigtx.yaml
2022-06-01 10:31:08.471 CET 0005 INFO [common.tools.configtxgen] doOutputBlock -> Generating genesis block
2022-06-01 10:31:08.471 CET 0006 INFO [common.tools.configtxgen] doOutputBlock -> Creating application channel genesis block
2022-06-01 10:31:08.471 CET 0007 INFO [common.tools.configtxgen] doOutputBlock -> Writing genesis block
+ res=0
```

Channels creation

2- The creation of the first channel

```
Creating channel channel1
Using organization 1
+ osnadmin channel join --channelID channel1 --config-block ./channel-artifacts/channel1.block -o localhost:7053 --ca-file /home/mbh/Desktop/project/hypmbh/organizations/ordererOrganizations/example.com/tlsca/tlsca.example.com-cert.pem --client-cert /home/mbh/Desktop/project/hypmbh/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.crt --client-key /home/mbh/Desktop/project/hypmbh/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.key
+ osnadmin channel list -o localhost:7053 --ca-file /home/mbh/Desktop/project/hypmbh/organizations/ordererOrganizations/example.com/tlsca/tlsca.example.com-cert.pem --client-cert /home/mbh/Desktop/project/hypmbh/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.crt --client-key /home/mbh/Desktop/project/hypmbh/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.key
Status: 200
{
        "systemChannel": null,
        "channels": [
                {
                        "name": "channel1",
                        "url": "/participation/v1/channels/channel1"
                }
        ]
}

+ res=0
Status: 201
{
        "name": "channel1",
        "url": "/participation/v1/channels/channel1",
        "consensusRelation": "consenter",
        "status": "active",
        "height": 1
}

Channel 'channel1' created
```

Channels creation

3- joining the peers

```
Joining org1 peer to the channel...
Using organization 1
+ peer channel join -b ./channel-artifacts/channel1.block
+ res=0
2022-06-01 10:17:45.751 CET 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2022-06-01 10:17:46.797 CET 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
Joining org2 peer to the channel...
Using organization 2
+ peer channel join -b ./channel-artifacts/channel1.block
+ res=0
2022-06-01 10:17:49.850 CET 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2022-06-01 10:17:50.807 CET 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
Joining org3 peer to the channel...
Using organization 3
+ peer channel join -b ./channel-artifacts/channel1.block
+ res=0
2022-06-01 10:17:53.864 CET 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2022-06-01 10:17:54.920 CET 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
Joining org4 peer to the channel...
Using organization 4
+ peer channel join -b ./channel-artifacts/channel1.block
+ res=0
2022-06-01 10:17:57.997 CET 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2022-06-01 10:17:58.911 CET 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
```

Channels creation

4- setting the anchor peer for each organization

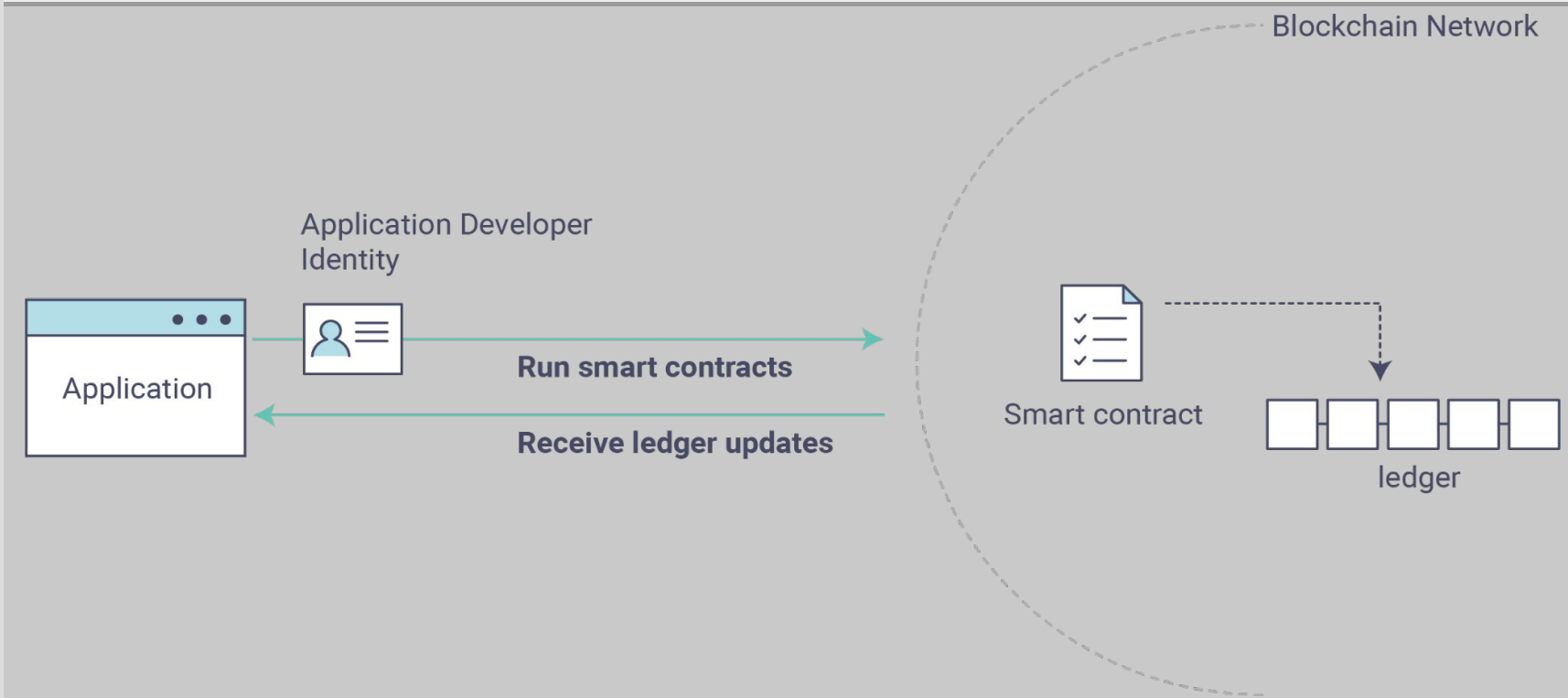
```
Setting anchor peer for org1...
Using organization 1
Fetching channel config for channel channel1
Using organization 1
Fetching the most recent configuration block for the channel
+ peer channel fetch config config_block.pb -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com -c channel1 --tls --cafile /o
pt/gopath/src/github.com/hyperledger/fabric/peer/organizations/ordererOrganizations/example.com/tlsca/tlsca.example.com-cert.pem
2022-06-01 09:18:00.201 UTC 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2022-06-01 09:18:00.218 UTC 0002 INFO [cli.common] readBlock -> Received block: 0
2022-06-01 09:18:00.218 UTC 0003 INFO [channelCmd] fetch -> Retrieving last config block: 0
2022-06-01 09:18:00.220 UTC 0004 INFO [cli.common] readBlock -> Received block: 0
Decoding config block to JSON and isolating config to Org1MSPconfig.json
+ configtxlator proto_decode --input config_block.pb --type common.Block --output config_block.json
+ jq '.data.data[0].payload.data.config' config_block.json
Generating anchor peer update transaction for Org1 on channel channel1
+ jq '.channel_group.groups.Application.groups.Org1MSP.values += {"AnchorPeers":{"mod_policy": "Admins","value":{"anchor_peers": [{"host": "peer0.org1
.example.com","port": 7051}]},"version": "0"}}' Org1MSPconfig.json
+ configtxlator proto_encode --input Org1MSPconfig.json --type common.Config --output original_config.pb
+ configtxlator proto_encode --input Org1MSPmodified_config.json --type common.Config --output modified_config.pb
+ configtxlator compute_update --channel_id channel1 --original original_config.pb --updated modified_config.pb --output config_update.pb
+ configtxlator proto_decode --input config_update.pb --type common.ConfigUpdate --output config_update.json
```

Channels creation

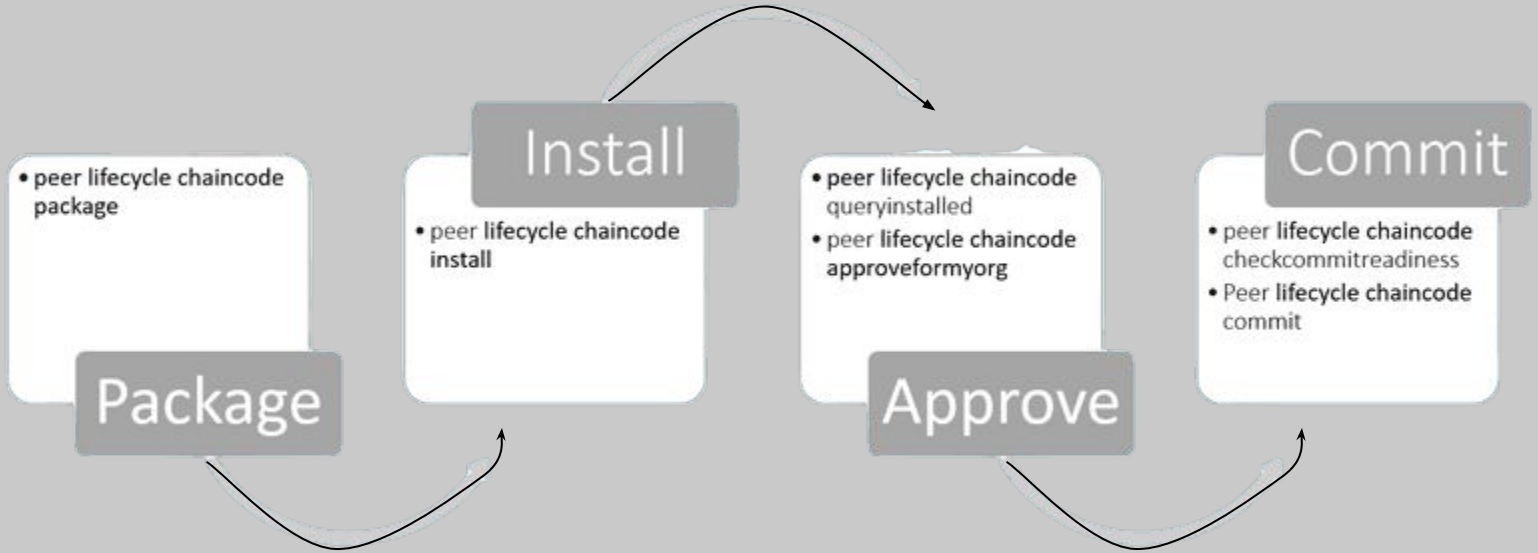
5-repeating the process for creating the other channels

```
{
        "systemChannel": null,
        "channels": [
                {
                        "name": "channel1",
                        "url": "/participation/v1/channels/channel1"
                },
                {
                        "name": "channel2",
                        "url": "/participation/v1/channels/channel2"
                },
                {
                        "name": "channel3",
                        "url": "/participation/v1/channels/channel3"
                },
                {
                        "name": "channel4",
                        "url": "/participation/v1/channels/channel4"
                }
        ]
}
```

Chaincode


Blockchain Network
Application Developer Identity
Application
Run smart contracts
Receive ledger updates
Smart contract
ledger

Chaincode lifecycle



Chaincode deployment

./start.sh deployCC -C <channel name> -ccn <chancode label> -ccp <chaincode path> -ccl <chaincede language>

1- packaging the chaincode

```
Using docker and docker-compose
deploying chaincode on channel 'channel1'
executing with the following
- CHANNEL_NAME: channel1
- CC_NAME: student
- CC_SRC_PATH: ../chaincode/
- CC_SRC_LANGUAGE: go
- CC_VERSION: 1.0
- CC_SEQUENCE: 1
- CC_END_POLICY: NA
- CC_COLL_CONFIG: NA
- CC_INIT_FCN: NA
- DELAY: 3
- MAX_RETRY: 5
- VERBOSE: false
Vendoring Go dependencies at ../chaincode/
/home/mbh/Desktop/project/chaincode /home/mbh/Desktop/project/hypmbh
/home/mbh/Desktop/project/hypmbh
Finished vendoring Go dependencies
+ peer lifecycle chaincode package student.tar.gz --path ../chaincode/ --lang golang --label student_1.0
+ res=0
Chaincode is packaged
```

Chaincode deployment

2- installing the package on each peer

```
Installing chaincode on peer0.org1...
Using organization 1
+ peer lifecycle chaincode install student.tar.gz
+ res=0
2022-06-06 13:00:23.208 CET 0001 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Installed remotely: response:<status:200 payload:"\nLstudent_
1.0:d25777dc88ee73ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac\022\013student_1.0" >
2022-06-06 13:00:23.208 CET 0002 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Chaincode code package identifier: student_1.0:d25777dc88ee73
ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac
Chaincode is installed on peer0.org1
Install chaincode on peer0.org2...
Using organization 2
+ peer lifecycle chaincode install student.tar.gz
+ res=0
2022-06-06 13:00:43.525 CET 0001 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Installed remotely: response:<status:200 payload:"\nLstudent_
1.0:d25777dc88ee73ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac\022\013student_1.0" >
2022-06-06 13:00:43.525 CET 0002 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Chaincode code package identifier: student_1.0:d25777dc88ee73
ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac
Chaincode is installed on peer0.org2
Install chaincode on peer0.org3...
Using organization 3
+ peer lifecycle chaincode install student.tar.gz
+ res=0
2022-06-06 13:01:04.159 CET 0001 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Installed remotely: response:<status:200 payload:"\nLstudent_
1.0:d25777dc88ee73ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac\022\013student_1.0" >
2022-06-06 13:01:04.159 CET 0002 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Chaincode code package identifier: student_1.0:d25777dc88ee73
ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac
Chaincode is installed on peer0.org3
Install chaincode on peer0.org4...
Using organization 4
+ peer lifecycle chaincode install student.tar.gz
+ res=0
2022-06-06 13:01:23.709 CET 0001 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Installed remotely: response:<status:200 payload:"\nLstudent_
1.0:d25777dc88ee73ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac\022\013student_1.0" >
2022-06-06 13:01:23.709 CET 0002 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Chaincode code package identifier: student_1.0:d25777dc88ee73
ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac
Chaincode is installed on peer0.org4
```

Chaincode deployment

3- Check if the package has installed

```
Using organization 1
+ peer lifecycle chaincode queryinstalled
+ res=0
Installed chaincodes on peer:
Package ID: student_1.0:d25777dc88ee73ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac, Label: student_1.0
Query installed successful on peer0.org1 on channel
Using organization 2
+ peer lifecycle chaincode queryinstalled
+ res=0
Installed chaincodes on peer:
Package ID: student_1.0:d25777dc88ee73ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac, Label: student_1.0
Query installed successful on peer0.org2 on channel
Using organization 3
+ peer lifecycle chaincode queryinstalled
+ res=0
Installed chaincodes on peer:
Package ID: student_1.0:d25777dc88ee73ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac, Label: student_1.0
Query installed successful on peer0.org3 on channel
Using organization 4
+ peer lifecycle chaincode queryinstalled
+ res=0
Installed chaincodes on peer:
Package ID: student_1.0:d25777dc88ee73ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac, Label: student_1.0
Query installed successful on peer0.org4 on channel
```

Chaincode deployment

4- approving the chaincode

```
Using organization 1
+ peer lifecycle chaincode approveformyorg -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/mbh/Desktop/project
/hypmbh/organizations/ordererOrganizations/example.com/tlsca/tlsca.example.com-cert.pem --channelID channel1 --name student --version 1.0 --package-id
 student_1.0:d25777dc88ee73ee0d6a29a1c741990757fbdea5deea5d86e3c184082bb847ac --sequence 1
+ res=0
2022-06-06 13:01:27.590 CET 0001 INFO [chaincodeCmd] ClientWait -> txid [5d32eacd8f4265b684ed0c8df299a4ccfe50b435b3154e10d864459ac90605f0] committed w
ith status (VALID) at localhost:7051
Chaincode definition approved on peer0.org1 on channel 'channel1'
Using organization 1
Checking the commit readiness of the chaincode definition on peer0.org1 on channel 'channel1'...
Attempting to check the commit readiness of the chaincode definition on peer0.org1, Retry after 3 seconds.
+ peer lifecycle chaincode checkcommitreadiness --channelID channel1 --name student --version 1.0 --sequence 1 --output json
+ res=0
{
        "approvals": {
                "Org1MSP": true,
                "Org2MSP": false,
                "Org3MSP": false,
                "Org4MSP": false
        }
}
Checking the commit readiness of the chaincode definition successful on peer0.org1 on channel 'channel1'
```

```
{
        "approvals": {
                "Org1MSP": true,
                "Org2MSP": true,
                "Org3MSP": true,
                "Org4MSP": true
        }
}
```

Chaincode deployment

5- committing the chaincode to the channel

```
Using organization 1
Using organization 2
Using organization 3
Using organization 4
+ peer lifecycle chaincode commit -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/mbh/Desktop/project/hypmbh/o
rganizations/ordererOrganizations/example.com/tlsca/tlsca.example.com-cert.pem --channelID channel1 --name student --peerAddresses localhost:7051 --tl
sRootCertFiles /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org1.example.com/tlsca/tlsca.org1.example.com-cert.pem --peerAddresses
 localhost:9051 --tlsRootCertFiles /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org2.example.com/tlsca/tlsca.org2.example.com-cert
.pem --peerAddresses localhost:11051 --tlsRootCertFiles /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org3.example.com/tlsca/tlsca.
org3.example.com-cert.pem --peerAddresses localhost:12051 --tlsRootCertFiles /home/mbh/Desktop/project/hypmbh/organizations/peerOrganizations/org4.exa
mple.com/tlsca/tlsca.org4.example.com-cert.pem --version 1.0 --sequence 1
+ res=0
2022-06-06 13:01:55.173 CET 0001 INFO [chaincodeCmd] ClientWait -> txid [e2d10dc29a55373e5fd05cf03a4ee2aa52f4e10e48b015a36ca2d4940788b9af] committed w
ith status (VALID) at localhost:11051
2022-06-06 13:01:55.173 CET 0002 INFO [chaincodeCmd] ClientWait -> txid [e2d10dc29a55373e5fd05cf03a4ee2aa52f4e10e48b015a36ca2d4940788b9af] committed w
ith status (VALID) at localhost:7051
2022-06-06 13:01:55.173 CET 0003 INFO [chaincodeCmd] ClientWait -> txid [e2d10dc29a55373e5fd05cf03a4ee2aa52f4e10e48b015a36ca2d4940788b9af] committed w
ith status (VALID) at localhost:9051
2022-06-06 13:01:55.301 CET 0004 INFO [chaincodeCmd] ClientWait -> txid [e2d10dc29a55373e5fd05cf03a4ee2aa52f4e10e48b015a36ca2d4940788b9af] committed w
ith status (VALID) at localhost:12051
Chaincode definition committed on channel 'channel1'
```

Chaincode deployment

6- checking the commitment of the chaincode

```
Using organization 1
Querying chaincode definition on peer0.org1 on channel 'channel1'...
Attempting to Query committed status on peer0.org1, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID channel1 --name student
+ res=0
Committed chaincode definition for chaincode 'student' on channel 'channel1':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true, Org3MSP: true, Org4MSP: true]
Query chaincode definition successful on peer0.org1 on channel 'channel1'
Using organization 2
Querying chaincode definition on peer0.org2 on channel 'channel1'...
Attempting to Query committed status on peer0.org2, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID channel1 --name student
+ res=0
Committed chaincode definition for chaincode 'student' on channel 'channel1':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true, Org3MSP: true, Org4MSP: true]
Query chaincode definition successful on peer0.org2 on channel 'channel1'
Using organization 3
Querying chaincode definition on peer0.org3 on channel 'channel1'...
Attempting to Query committed status on peer0.org3, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID channel1 --name student
+ res=0
Committed chaincode definition for chaincode 'student' on channel 'channel1':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true, Org3MSP: true, Org4MSP: true]
Query chaincode definition successful on peer0.org3 on channel 'channel1'
Using organization 4
Querying chaincode definition on peer0.org4 on channel 'channel1'...
Attempting to Query committed status on peer0.org4, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID channel1 --name student
+ res=0
Committed chaincode definition for chaincode 'student' on channel 'channel1':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true, Org3MSP: true, Org4MSP: true]
Query chaincode definition successful on peer0.org4 on channel 'channel1'
```

Chaincode Assets(p1)

01 Student

- ID
- FullName
- DateOfBirth
- PlaceOfBirth
- Address
- University
- Faculty
- Speciality
- Major
- Degree
- EnrollementYear
- CurrentYear

- InitLedger()
- CreateStudent()
- ReadStudent()
- UpdateStudent()
- DeleteStudent()
- StudentExists()
- TransferStudent()
- ChangeMajor()
- ChangeDegree()
- ChangeCurrentYear()
- GetAllStudents()

Chaincode Assets(p2)

02 Certificate

- ID
- StudentID
- Title
- IssuePlace
- IssueDate
- DocHash
- Semester1Average
- Semester2Average
- Semester3Average
- Semester4Average
- Semester5Average
- Semester6Average
- YearsAverage
- Description

- InitLedger()
- CreateCerteficat()
- ReadCerteficat()
- UpdateCerteficat()
- DeleteCerteficat()
- CerteficatExists()
- GetAllCerteficats

Chaincode Assets(p3)

02 Project

- ID
- StudentID
- Title
- Advisor
- Course
- Type
- DeposeDate
- RepoHash
- UrlPath
- Partner1
- Partner2
- Partner3
- Partner4
- Partner5
- Partner6
- Grade
- Description

- InitLedger()
- CreateProject()
- ReadProject()
- UpdateProject()
- DeleteProject()
- GetAllProject

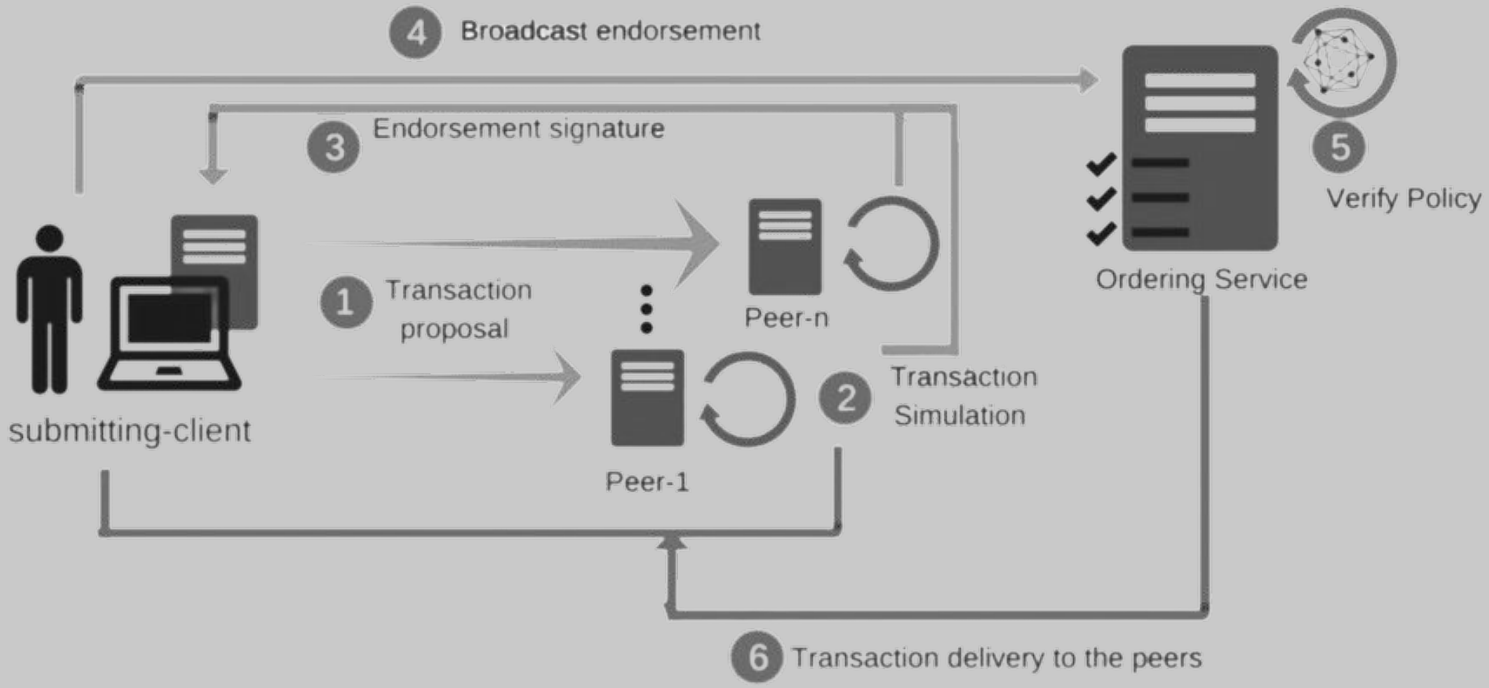
Chaincode Assets(p4)

01 Event

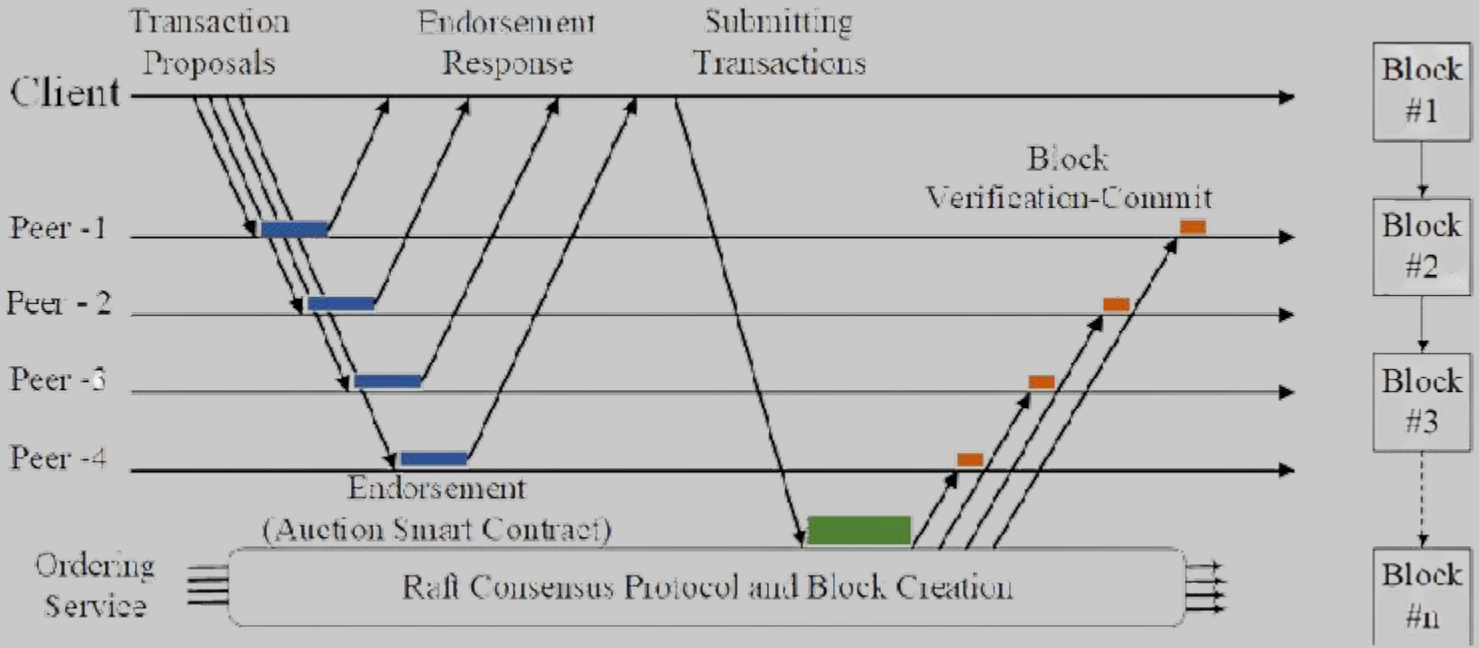
- ID
- StudentID
- Location
- JoinDate
- Orgnaizer
- Description

- InitLedger()
- CreateEvent()
- ReadEvent()
- UpdateEvent()
- DeleteEvent()
- EventExists()
- GetAllEvent

Transaction flow



Transaction flow



Transactions execution

1- Create students:

```
bash-5.1# peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C channel1 -n student --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" --peerAddresses peer0.org3.example.com:11051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org3.example.com/peers/peer0.org3.example.com/tls/ca.crt" --peerAddresses peer0.org4.example.com:12051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org4.example.com/peers/peer0.org4.example.com/tls/ca.crt" -c '{"function":"InitLedger","Args":[]}'
2022-06-09 12:13:48.139 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
```

	id	key	value
☐	student1	student1	{ "rev": "1-3b8e7a505501f92f2478e2f2…
☐	student2	student2	{ "rev": "1-ec87a67b5850aea18d2154…
☐	student3	student3	{ "rev": "1-878d7b047671b6eb896d24…
☐	student4	student4	{ "rev": "1-a3f579ce4c48c66c93d0601…
☐	student5	student5	{ "rev": "1-d8d2ec95c76afc4016059ad…
☐	student6	student6	{ "rev": "1-bcd1cc9ae2f382eec275aadf…

Transactions execution

2- Add a new student:

```
bash-5.1# peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C channel1 -n student --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" --peerAddresses peer0.org3.example.com:11051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org3.example.com/peers/peer0.org3.example.com/tls/ca.crt" --peerAddresses peer0.org4.example.com:12051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org4.example.com/peers/peer0.org4.example.com/tls/ca.crt" -c '{"Args":["CreateStudent","student7","Touzala","1","Math","2018","Statistique","master","Djillali Liabes","Science Exactes","12/12/1999","Sidi Bel Abbes","Sidi Bel Abbes,sfizef"]}'
2022-06-09 13:00:13.985 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
```

	id	key	value
☐	student1	student1	{ "rev": "1-3b8e7a505501f92f2478e2f2…
☐	student2	student2	{ "rev": "1-ec87a67b5850aea18d2154…
☐	student3	student3	{ "rev": "1-878d7b047671b6eb896d24…
☐	student4	student4	{ "rev": "1-a3f579ce4c48c66c93d0601…
☐	student5	student5	{ "rev": "1-d8d2ec95c76afc4016059ad…
☐	student6	student6	{ "rev": "1-bcd1cc9ae2f382eec275aadf…
☐	student7	student7	{ "rev": "1-5ebd88a6ff8806828433797…

Transactions execution

3- transfert a student:

```
bash-5.1# peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C channel1 -n student --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" --peerAddresses peer0.org3.example.com:11051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org3.example.com/peers/peer0.org3.example.com/tls/ca.crt" --peerAddresses peer0.org4.example.com:12051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org4.example.com/peers/peer0.org4.example.com/tls/ca.crt" -c '{"Args":["TransferStudent","student7","Abu Bekr Belkaid"]}'
2022-06-09 14:46:05.814 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200 payload:"Djillali Liabes"
```

```
bash-5.1# peer chaincode query -C channel1 -n student --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" -c '{"Args":["ReadStudent","student7"]}'
{"EnrollementYear":2018,"FullName":"Touzala","ID":"student7","Speciality":"Math","Major":"Statistique","CurrentYear":1,"Degree":"master","University":"Abu Bekr Belkaid","Faculty":"Science Exactes","DateOfBirth":"12/12/1999","PlaceOfBirth":"Sidi Bel Abbes","Address":"Sidi Bel Abbes,sfizef"}
```

Transactions execution

4- delete a student:

```
bash-5.1# peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/orderOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C channel1 -n student --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" --peerAddresses peer0.org3.example.com:11051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org3.example.com/peers/peer0.org3.example.com/tls/ca.crt" --peerAddresses peer0.org4.example.com:12051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org4.example.com/peers/peer0.org4.example.com/tls/ca.crt" -c '{"Args":["DeleteStudent","student3"]}'
2022-06-09 16:37:00.013 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
```

	id	key	value
☐	student1	student1	{ "rev": "1-3b8e7a505501f92f2478e2f2…
☐	student2	student2	{ "rev": "1-ec87a67b5850aea18d2154…
☐	student4	student4	{ "rev": "1-a3f579ce4c48c66c93d0601…
☐	student5	student5	{ "rev": "1-d8d2ec95c76afc4016059ad…
☐	student6	student6	{ "rev": "1-bcd1cc9ae2f382eec275aadf…
☐	student7	student7	{ "rev": "2-87382ea2e32f9322531a77a…

Achieved goal

The network has successfully being created and the data was poorly engineered, it is clearly needs more work on it but we did learn about the blockchain and we did confirmed that its potentials in the educational section could be achieved

Future Work

More implementation details

More data can be included like student courses or attendance history.More functions needs to be implemented and better input verification.

Make good use of the network

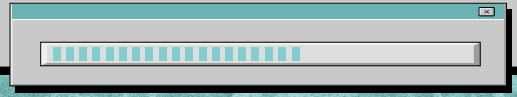
To people use it we need to build applications in the top of this network.These application wil find it easier to be integrated and communicate with the chaincode

Conclusions

At the end, we can say that the project could be much more of assist and help and that we see that Blockchain has the potential to reshape the education industry, the leading members must start learning about and experimenting with the technology now, both individually and as part of a collaboration, so our educational institution can be ready for the future.

Resources

- *A beginner's guide to understanding the layers of blockchain technology*. **https://cointelegraph.com/blockchain-for-beginners/a-beginners-guide-to-understanding-the-layers-of-blockchain-technology**, jan 2021.
- Sudheer Hanumanthakari A. R. Sathya, Sandeep Kumar Panda. *Blockchain Technology: Applications and Challenges*, volume 203 of Intelligent Systems Reference Library. Springer, first edition, 2021.
- John Evans. *Blockchain nodes: An in-depth guide*. **https://nodes.com/**.
- Gwyneth Iredale. *What are the different types of blockchain technology?*,**https://101blockchains.com/types-of-blockchain/**, jan 2021.
- Alok Kumar Jain. *Blockchain goes to school*. Cognizant 20-20 Insights, mar 2019.
- **https://hyperledger-fabric.readthedocs.io/en/latest/**,2020-2022



Thanks!

Do you have any questions?

merbahmohamedamine.22@gmail.com