BlockChain Architecture and basic working outline.

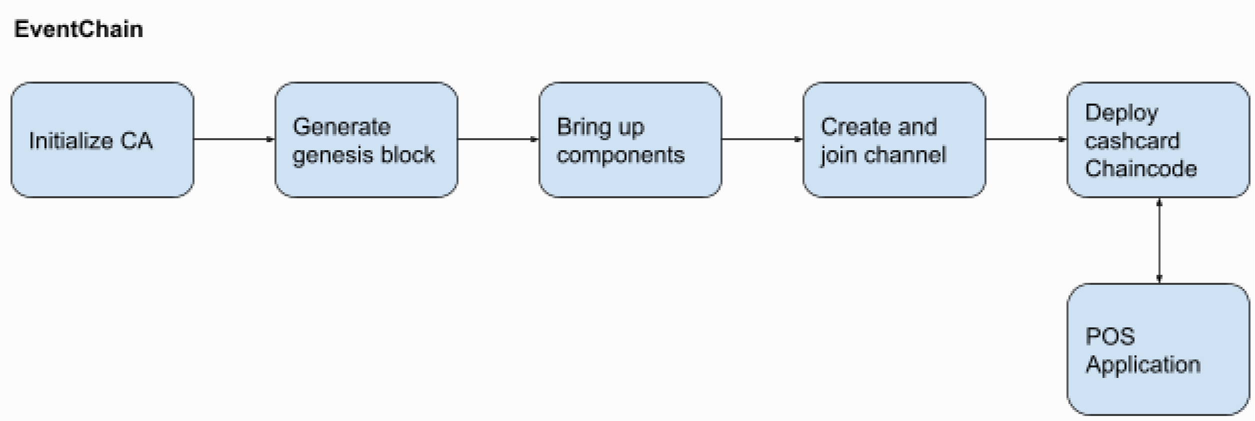
Hyperledger Fabric framework fabric has been used for implementing the solution. I have used the test-network provided in the official documentation in the solution. The architecture and working is as follows.

1. Test Network contains three organizations: one Orderer Organization and two Peer Organizations.
2. Hyperledger Fabric uses public key infrastructure (PKI) to verify the actions of all network participants. Every node, network administrator, and user submitting transactions needs to have a public certificate and private key to verify their identity. These identities need to have a valid root of trust, establishing that the certificates were issued by an organization that is a member of the network
3. Peers are the fundamental components of any Fabric network. Peers store the blockchain ledger and validate transactions before they are committed to the ledger. Peers run the smart contracts that contain the business logic that is used to manage the assets on the blockchain ledger. There are two peer organizations Org1 and Org2 in our network. Each peer organization is deployed with one peer (peer0).
4. An ordering service allows peers to focus on validating transactions and committing them to the ledger. After ordering nodes receive endorsed transactions from clients, they come to consensus on the order of transactions and then add them to blocks. The blocks are then distributed to peer nodes, which add the blocks to the blockchain ledger. In our network, there is one ordering service node (orderer) in the Orderer Organization, which maintains the ordering service of the network.
5. Channels are a private layer of communication between specific network members. Channels can be used only by organizations that are invited to the channel, and are invisible to other members of the network. Each channel has a separate blockchain ledger. Organizations have to be invited to “join” their peers to the channel to store the channel ledger and validate the transactions on the

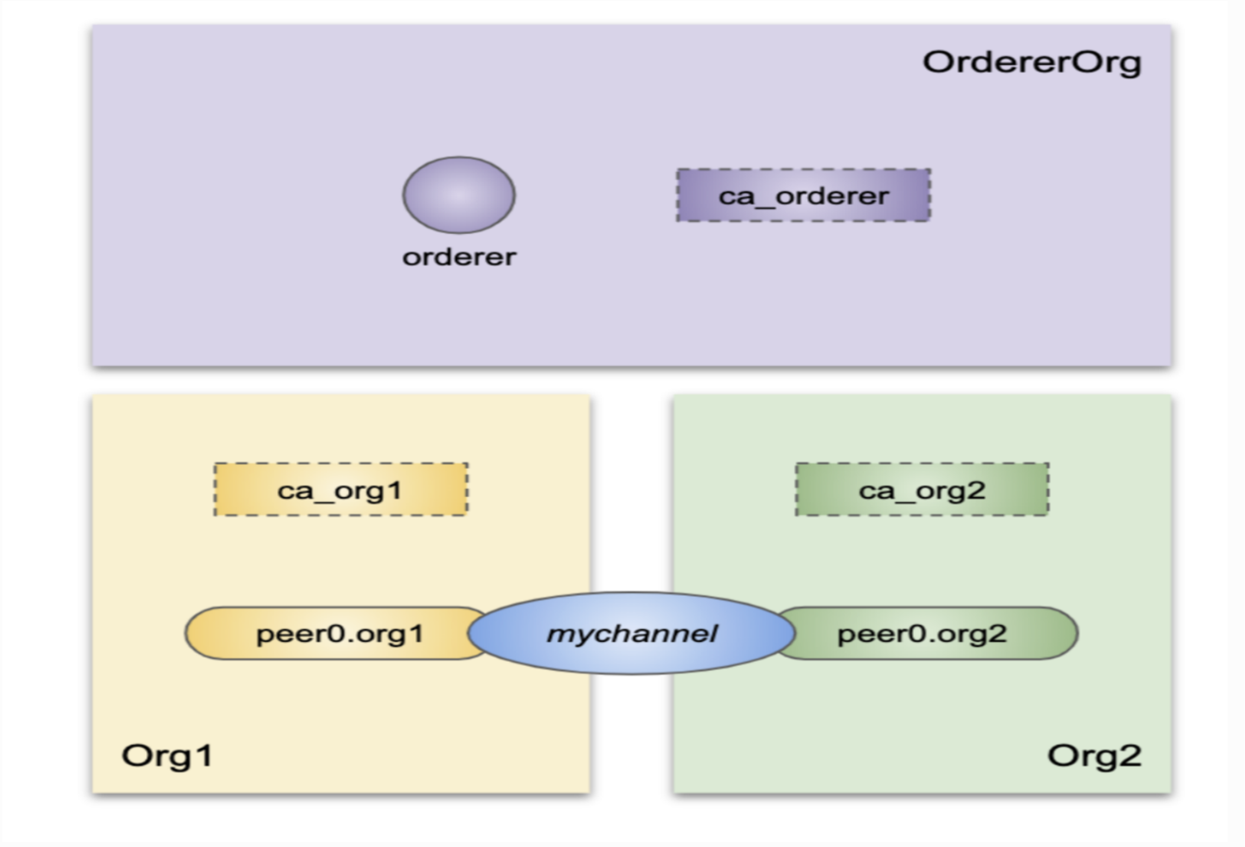
channel. We have created one channel named *‘mychannel’* between the two organizations.

6. After the channel is created, one can start using smart contracts to interact with the channel ledger. Smart contracts contain the business logic that governs assets on the blockchain ledger. In Fabric, smart contracts are deployed on the network in packages referred to as chaincode. 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. Before a chaincode can be deployed to a channel, the members of the channel need to agree on a chaincode definition that establishes chaincode governance. When the required number of organizations agree, the chaincode definition can be committed to the channel, and the chaincode is ready to be used. We have installed the chaincode needed for executing the transactions of eventChain. Javascript was used for the same.
7. 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. Finally after the test network is up and running our chaincode we use our application to interact with it. The application is designed as a Nodejs script.

Workflow of network



EventChain Architecture



Prerequisites needed for running Hyperledger Fabric

Please follow the instructions in

1. https://hyperledger-fabric.readthedocs.io/en/latest/prereqs.html
2. https://hyperledger-fabric.readthedocs.io/en/latest/install.html
3. Install nodejs v14.x by following the instructions here - https://node.dev/node-binary
4. Install npm by - sudo apt-get install npm

Building the EventChain and running the application

1. Untar the attached tar file - eventChain.tar.gz.
2. Go to test-network folder - cd test-network/
3. Run the script network-start.sh to bring up the network - ./network-start.sh
4. For running the application go to the folder - cd ../cashcard/application-javascript/

5. Before executing the application for the first time please install all the necessary dependencies by executing - npm install
6. Delete any previously stored certificates - rm -r wallet*
7. Interact with the application by passing suitable command line arguments. To get help regarding the usage run - node eventChain.js -h
8. After interacting with the application stop the blockchain network
    a. Go to test-network folder - cd test-network/
    b. Run the script network-stop.sh to shut down the network - ./network-stop.sh

Common errors and fixes

1. Error with docker permission denied - https://stackoverflow.com/questions/48957195/how-to-fix-docker-got-permission-denied-issue
2. Failed to register user : Error: fabric-ca request register failed with errors [[{ code: 20, message: 'Authentication failure' }]]
    ******** FAILED to run the application: Error: Identity not found in wallet: appUser
    -> Delete the cashcard/application-javascript/wallet_retail and cashcard/application-javascript/wallet_cash directory and retry this application. The certificate authority must have been restarted and the saved certificates for the admin and application user are not valid. Deleting the wallet store will force these to be reset with the new certificate authority.

References

1. https://hyperledger-fabric.readthedocs.io/en/latest/test_network.html
2. https://kctheservant.medium.com/from-first-network-to-test-network-a-new-test-network-introduced-in-fabric-v2-0-81faa924ce29