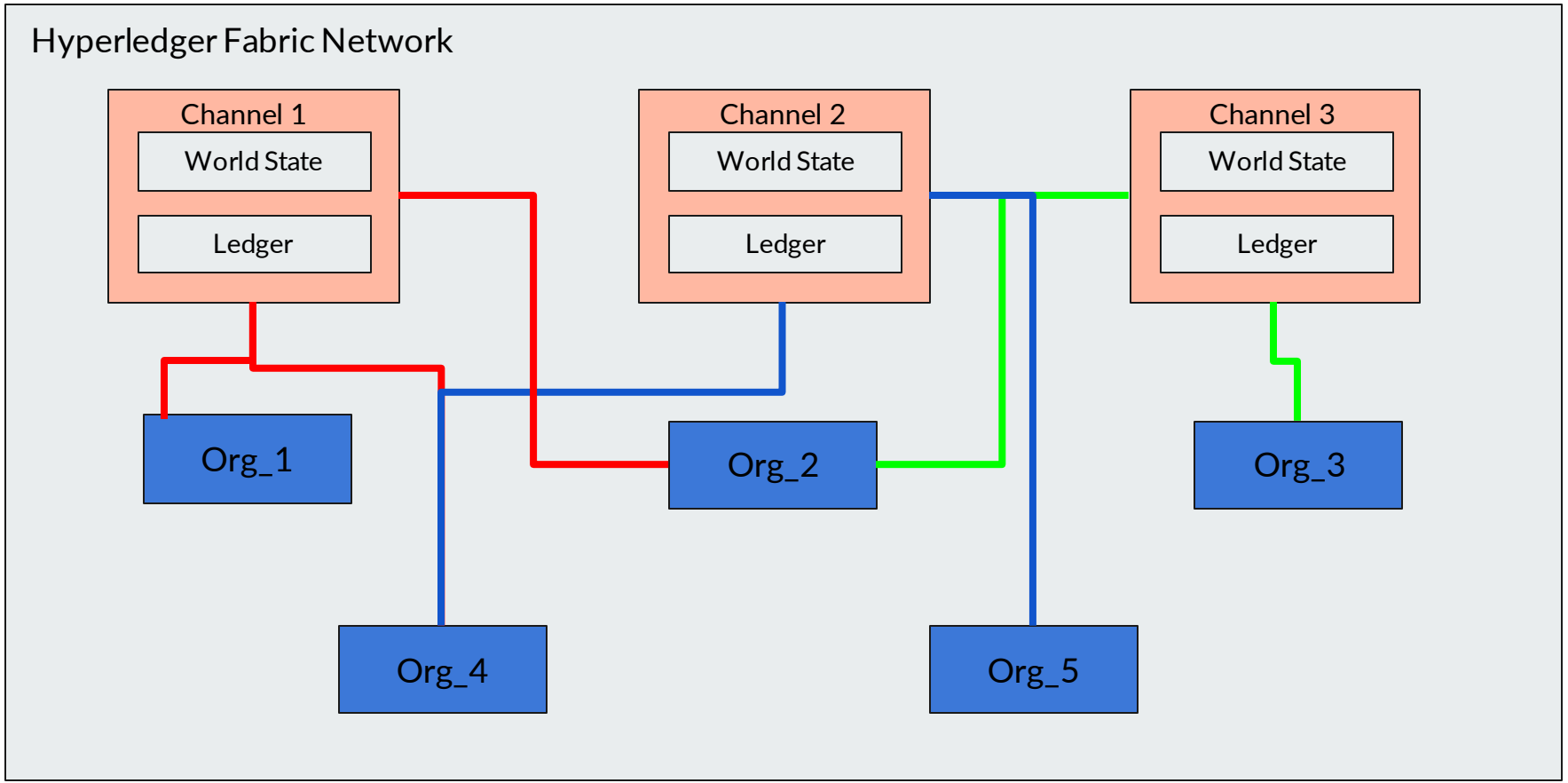
Theory and Applications of Blockchain (CS61065) - Tutorial 2

Hyperledger Fabric


Hyperledger Fabric Network
Channel 1
World State
Ledger
Channel 2
World State
Ledger
Channel 3
World State
Ledger
Org_1
Org_2
Org_3
Org_4
Org_5

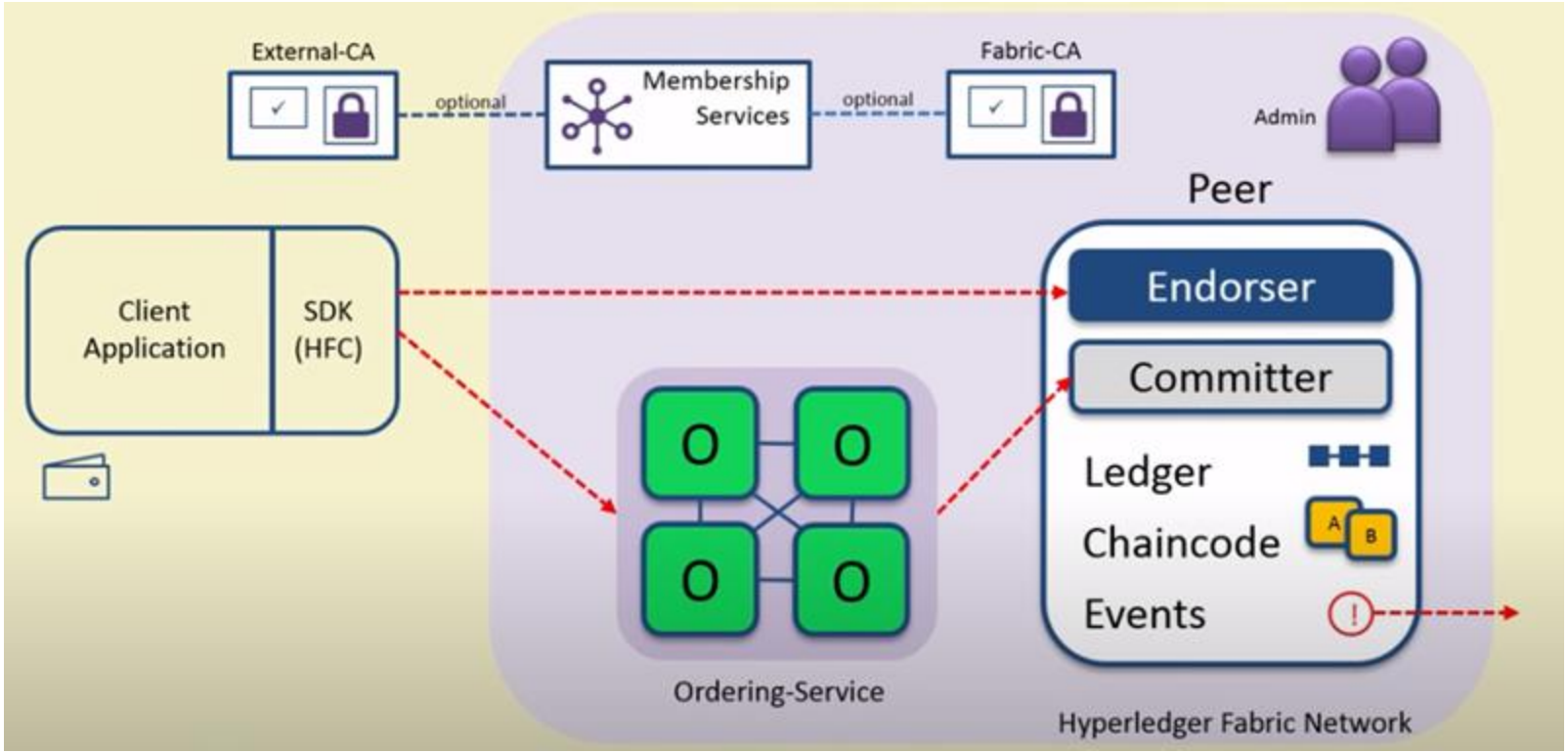
Organization

- Membership Service Providers (MSP) : Identity management of members
- Administrators : To manage the members (enroll, remove etc.)
- Users : Typically applications connecting to the blockchain network
- Peers : Hosting the chaincode and storing the ledger
- Orderers (optional) : Ordering of transactions

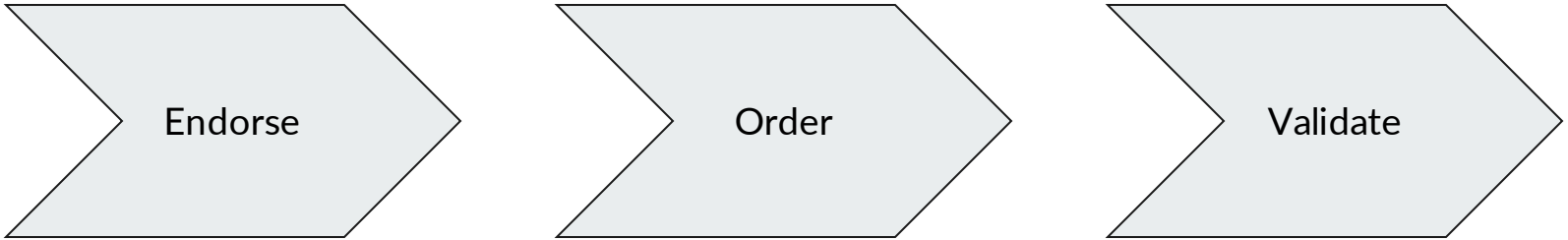
Each organization has an ID and a network can have multiple participating organizations

Image courtesy: NPTEL

Entities involved



Flow of Transactions



Client sends the transaction proposal to endorser. The endorsers return results with their signatures. The client checks if the endorsement policy is being satisfied. If yes, it sends the transaction to the orderer.

The orderer simply orders the transactions, puts them into blocks and sends it to all the peers to be validated. It does not have to look at the transactions.

The peers validate the transactions. (Check for double spending, enforcement of the endorsement policy etc.)

Setup prerequisites

Recommended to set up on a linux based system

- Git https://git-scm.com/downloads
- CURL https://curl.se/download.html
- Docker Engine https://docs.docker.com/engine/install/
- NodeJS (recommend to use nvm (Node Version Manager))
- Docker Compose (version 1.29.2)

```
>> sudo groupadd docker
>> sudo usermod -aG docker $USER
```

To bypass using sudo every time while using docker
REQUIRED TO DO

Installing Hyperledger Fabric

Determine a location on your machine where you want to place the fabric-samples repository and enter that directory in a terminal window.

>> mkdir fabric

>> cd fabric

>> curl -sSL https://bit.ly/2ysbOFE | bash -s -- <fabric_version> <fabric-ca_version>

Example:

>> curl -sSL https://bit.ly/2ysbOFE | bash -s -- 2.2.4 1.5.2

(Recommended to install 2.2.4)

>> export PATH=<path to download location>/fabric-samples/bin:$PATH

Starting the test network with CAs

```
>> cd fabric-samples/test-network

>> ./network.sh  up -ca
```

The script will start containers for each member of the test-network namely:

- orderer
- peer0.org1.example.com
- peer0.org2.example.com
- ca_org1
- ca_org2
- ca_orderer
- cli

Creating a channel and deploying chaincode

>> ./network.sh createChannel -ca

- Creates a channel with name "mychannel"

>> ./network.sh createChannel -c <channel name> -ca

- To create a channel with a custom name

Deploying chain-code

>> ./network.sh -ccn <chain-code-name> –ccp <path-to-chaincode> -ccl <chaincode-language>

- To deploy chaincode

Example:
>> ./network.sh -ccn basic –ccp ../asset-transfer-basic/chaincode-javascript -ccl javascript

Other optional arguments:

- -ccep : To specify the endorsement policy

Running an application to invoke the chaincode

Example: Go to the asset-transfer-basic/application-javascript directory

```
seemant@ubuntu:~/fabric-samples/asset-transfer-basic/application-javascript
$ ls
app.js  node_modules  package.json  package-lock.json  wallet
```

The wallet directory stores the crypto material for a user issued by a CA. The CA is respawned every time we do ./network.sh up.

Before running app.js

- Remove the wallet directory before running app.js.
- Run **>> npm install**

To run the app.js file : >> node app.js

Other examples of interest

- **asset-transfer-events**

  While deploying the chaincode for this example set the -ccep flag to “OR(‘Org1MSP.peer’, ‘Org2MSP.peer’)”

  Example on how to use event listeners

References

- Endorsement policies — hyperledger-fabricdocs main documentation
- Private data — hyperledger-fabricdocs main documentation
- Private Data — hyperledger-fabricdocs main documentation - Implicit private data
- Hyperledger Fabric SDK for Node.js Interface: Network - BlockEventListeners
- Hyperledger Fabric SDK for Node.js Interface: Contract - ContractEventListeners

NPTEL links:

- https://youtu.be/4ujj5knD3pg?si=joLsrcNJDv1I7EMk : Fabric Membership and Identity Management
- https://youtu.be/xjliVltyLRk?si=IRReh8BBsUDSnWm- : Fabric Components details
- https://youtu.be/nBXr7dLXAbE?si=x4vaXCHnM_nLyOGs : Fabric Transaction flow