Unit - 3

Hyperledger Fabric

3.1 Hyperledger Fabric (A): Decomposing the Consensus Process, Hyperledger Fabric Components, Chaincode Design and Implementation

Hyperledger Fabric

The private enterprise that develops products, solutions and application using plug and play components uses Hyperledger Fabric. It is a modular blockchain framework used by private enterprises.

Hyperledger is developed by Linux foundation in 2015 for open source distributed ledger. It is enterprise-grade framework. Hyperledger Fabric is highly decentralized ledger platform designed for enterprise use.

The hyperledger fabric network has fewer participants in the network as this network is private. The network required permission to access and to segregate business information with transactions can be sped up.

Working Of Hyper Ledger Fabric

- The hyper ledger fabric is open source, proven, enterprise grade and distributed ledger platform.
- It is permissioned network where the data wants to be shared that only get shared by the participant. It has advanced privacy control.
- The smart contract documents are processed in business to automate with self-executing terms between parties written into lines of code.
- In the decentralized blockchain network, the code and agreement are contained across network.
- The hyper ledger creates trust in between the organization. The transaction is trackable and irreversible.

- This enables the business more informed decision quicker, saving time, reducing cost and reducing risks.

Benefits of Hyperledger Fabric

Permissioned network

It establishes decentralized trust in network of permissioned participant rather than open network of anonymous participant.

Confidential transaction

The data is share with only participant wants to share with.

Pluggable architecture

The pluggable architecture is for tailor the blockchain to industry rather that one size fits all approach.

Components of Hyperledger Fabric Network

A network consists of following components.

- Ledger
- Smart contract
- Peer nodes
- Ordering service
- Channel
- Fabric certificate authorities

3.1.1 Decomposing the consensus process

The consensus is the process which provides a guaranteed ordering of transaction and validates the block of transactions.

Consensus provides following functionality.

- Confirms the correctness of transactions in proposed block according to consensus policies and endorsement
- Agrees on order and correctness
- Interfaces and depends on smart contract to verify the correctness of ordered set of transactions in block

The process to achieve the agreement on the single data value among distributed process or system. The Consensus algorithms are used to achieve the reliability in the network involving the multiple unreliable nodes.

The consensus algorithm is used on multi agent systems. It is important on distributed computing for solving issues. The consensus algorithm in hyperledger fabric network provides the guaranteed ordering of transactions.

The algorithm validates the blocks which are having transactions that need to be committed to the ledger. The consensus of hyperledger ensures the following network

It confirms the correctness of all transaction in proposed block according to endorsement and consensus policies. The consensus agrees on the order and correctness.

Hence, the result of execution where the agreement implies on global states. The smart contract layers are used to verify the correctness of an ordered set of transaction in block.

Following are the two properties of consensus which guarantees the agreement among nodes.

Safety:

The same sequence of inputs and result of same output are guaranteed on each node by safety.

The same state change will occur on the each node when it receives an identical series of transaction.

The algorithm behaves identical to a single node system which executes each transaction automatically one at a time.

Liveness:

Each non faulty node eventually receives every submitted transaction. It assumes communication does not fail.

Consensus in Hyper Ledger

There are three phases in hyper ledger fabric consensus.

1. **Endorsement**
    - It is driven policy from which every participant endorses the transactions.
2. **Ordering**
    - This accepts the endorsed transactions.
    - This phase agrees to the order to be committed to the ledger.
3. **Validation**
    - It takes a block of ordered transaction and validates the correctness of the result.
    - It includes checking endorsement policy and double spending.
    - The following figure illustrates the one possible transaction flow in hyper ledger fabric.

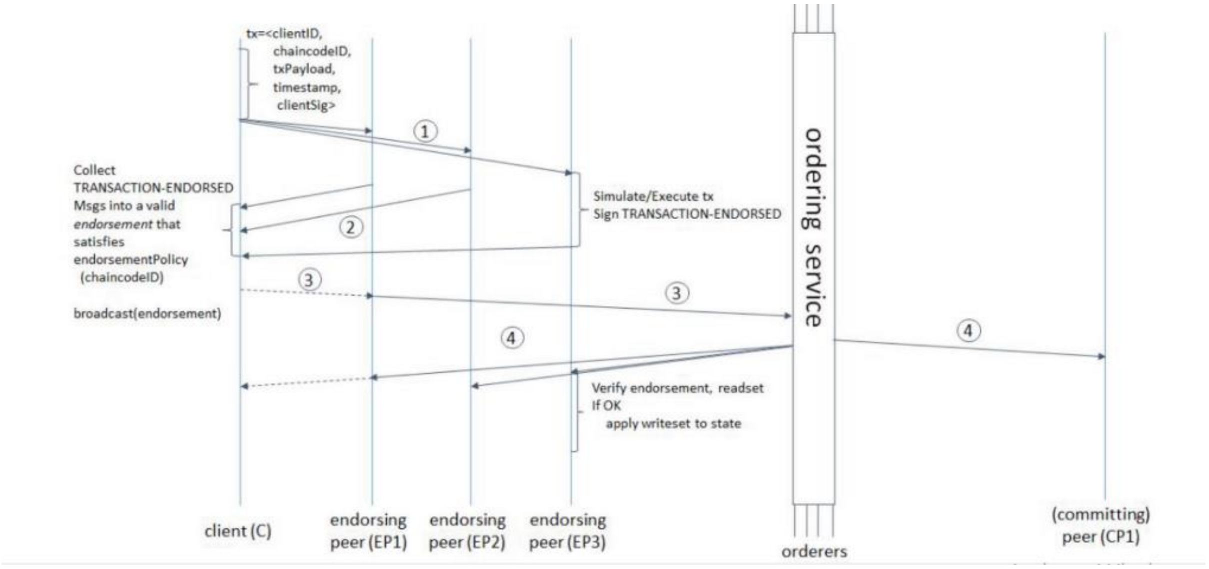
tx=<clientID, chaincodeID, txPayload, timestamp, clientSig>
1
Collect TRANSACTION-ENDORSED Msgs into a valid *endorsement* that satisfies endorsementPolicy (chaincodeID)
2
Simulate/Execute tx Sign TRANSACTION-ENDORSED
3
broadcast(endorsement)
3
ordering service
4
4
Verify endorsement, readset If OK apply writeset to state
client (C)
endorsing peer (EP1)
endorsing peer (EP2)
endorsing peer (EP3)
orderers
(committing) peer (CP1)

Fig. No. 1

All three phases are supported by hyper ledger fabric in pluggable consensus service. The endorsement, ordering and validation models depend on application requirements where applications can plug in different models. The ordering service API has two main basic operations as:

Broadcast (blob)

This operation is called by client for dissemination over the channel to broadcast an arbitrary message blob. This is also called as request (blob).

Deliver (seqno, prevhash, blob)

This operation is called by the ordering service to deliver the message blob with the specified non- negative integer sequence number (seqno) and most recently delivered blob hash (prevhash). It is can output of the ordering service.

3.1.2 Hyper ledger fabric components

Following are the hyper ledger fabric components which help to build hyper ledger network.

- Assets
- Shared ledger
- Smart contract
- Peer nodes
- Channel
- Organization
- Membership services provider (MSP)
- Ordering service

All the components of hyper ledger are explain in detail as follows:

Assets

- The asset means the valuable thing in network.
- The asset has state and ownership.

- The assets are represented in hyper ledger as collection of key value pair.

Shared Ledger

- The ledger is used to keep record of state and ownership.
- The ledger consists of two components as state and block chain.
- The state describes the state of the leger at give point in time.
- The state is database of the ledger.
- Other component block which is used to keep history of transactions.

Smart Contract

- The smart contract is also known as chain code in hyper ledger fabric.
- Smart contract is the software of assets and related transactions.
- The smart contract defines the business logic of the system.
- The interaction with ledger occurred in block chain then chain code is invoked.

Peer Nodes

- Peers are host ledgers and smart contract in the network hence it is a fundamental element of the network.
- The chain code is executed by peers where peer can accesses ledger data.
- The peer can endorse transaction and interfaces with application.
- The endorsement policy defines necessary and sufficient conditions for a valid transaction endorsement.

Channel

- Channels are the collection pf peers where logical structure is formed.
- The peers can create separate ledger of transaction by the capability of channel.

Organization

- The hyper ledger fabric network is built by peers owned by network.

- The peers are contributed by the different organization which are the members of network.
- The collective network get the contribution of individual resources in the network.
- The membership service provider assign the identity to the peers from its owning organization.
- The peers of different organizations can be on the same channel.

Membership Services Provider (MSP)

- The MSP is a certification of authorization used to manage certificates.
- The MSP is used to authenticate the member identity and roles on network.
- The hyper ledger does not allow the unknown identities to transact on network.
- The user ID’s of the participant are validated which enables hyper ledger fabric as private and permissioned network.

Ordering Service

- The blocks of transaction of ordering service packages are sent on peers of network.
- The network allows the transaction entries confirm on channel.
- The endorsing peers and the peers are communicated on channel.
- The Solo and Kafka mechanisms are used for ordering configuration.

3.1.3 Chaincode Design and Implementation

The chaincode is a program which is written in java, go, node.js. It implements prescribed interface. The chain code executes in secure docker container which is isolated from the endorsing peer process. The applications of participants are used to initialize and manages the ledger state through transaction submitted.

The members of the network handle the business logic on network. The chain code can be considered as smart contract. The chain code has its exclusive states to create the network. The state created by chaincode cannot be accessed by the any other chain code. The appropriate permission of a chain code get the access of another chaincode although it is same network.

Chain code implementation

- The chaincode can be written in Golang or java. The chaincode can be of three types such as public, confidential (private) or access controlled.
- The code files are act as smart contract in chaincode which users can interact with via APIs.
- The chaincode involves the call function which is used to give result in state change and consequently updates the ledger in the network.
- The chain code also involves other functions in network as they can be used for query the ledger and do not result in any state change.
- The shim interface creating is the first step in chain code implementation.
- The shim interfacing is used in accessing the state variables and transactions context of chain code using APIs.
- The chain code can be in Golang or java code.
- Following are various functions that are used to implement the chaincode.

  - Init()
  - Invoke()
  - Query()
  - 4()

Init()

The chain code is going to deploy on the ledger then init() function is invoked.

The init () function is used to initialize the chain code.

As the ledger updates the transaction accordingly the state will be change in chain code.

Invoke()

The function invoked() is performed when contracts are executed.

The invoke() includes the parameters along with an array of argument in the function.

This function allows the stat change and writes to the ledger.

Query()

The function query() is used for current state of deployed chaincode.

The query() function does not make changes in ledger.

4()

The function 4() is executed for peer which deploys its own copy of chaincode.

The chaincode is registered by the 4() function.

Following figure shows the overview of hyperledger fabric.

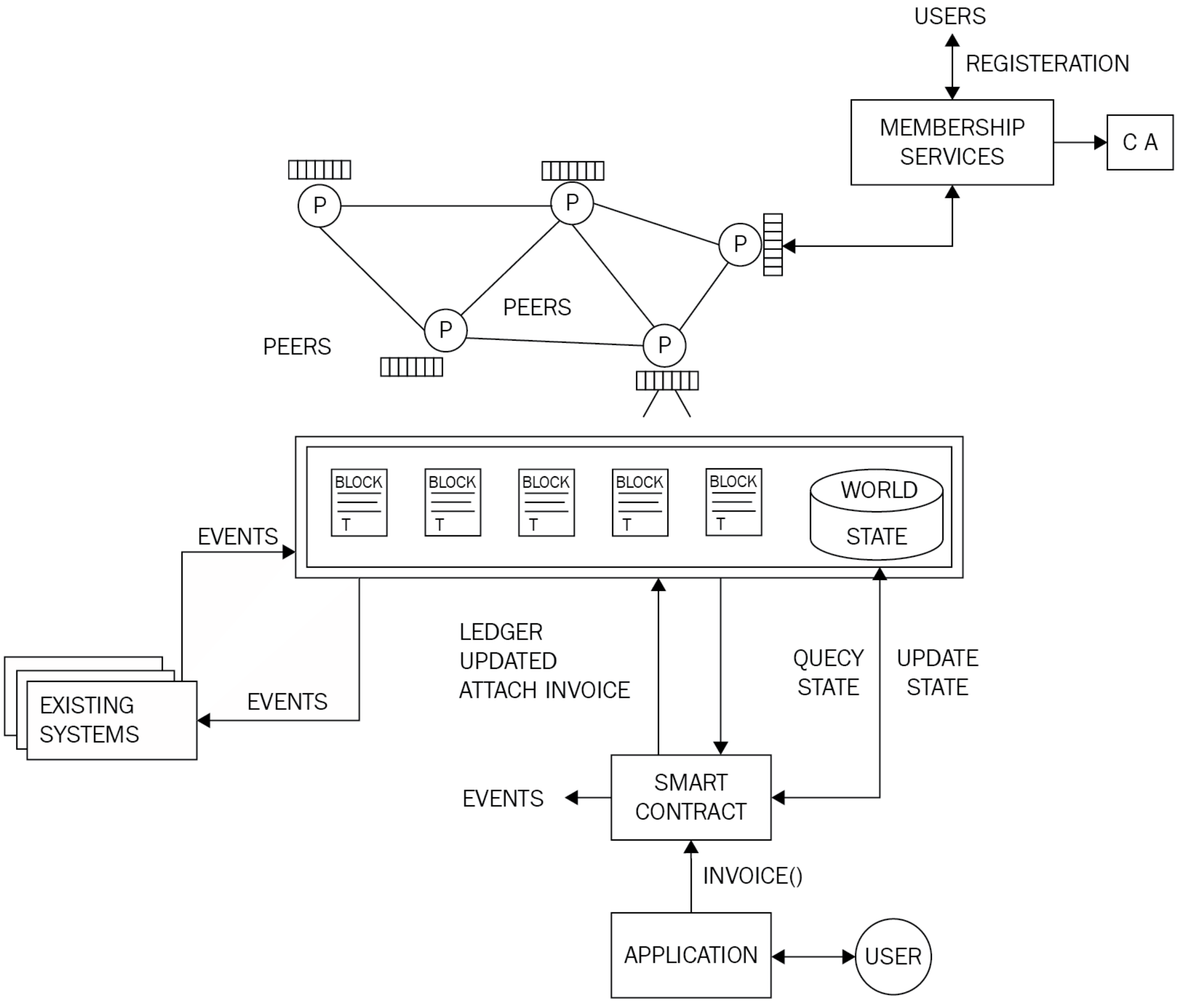

Fig. No. 2

The peer cluster shows in above figure are the all type of nodes such as endorses, committers, orders and so on.

The peers are communicating with each other and each another nodes which has copy of blockchain.

The membership services shown are used to validate and authenticate the peers attach on the network by using certificate of authority (CA)

At the bottom of image the blockchain magnified view is shown.

The blockchain can listen for blockchain events and cam produce events for blockchain.

3.2 Hyperledger Fabric (B): Beyond Chaincode: Fabric Sdk and Front End (B) Hyperledger Composer Tool

3.2.1 Beyond Chaincode the Hyperledger Fabric SDK and Front End

The chaincode implements higher level functionality on top of the blockchain ledger. There are two types of chaincode which runs on peer's control.

Following are two kinds of chaincode as:

- System chaincode
- Normal chaincode

The peer process requires system chaincode. The normal chaincode is the separated part managed by peer. The hyperledger fabric SDK is required for communication between fabric blockchain network and application.

The simple API is used to submit the transactions to the ledger. The SDK of hyperledger can be host from anywhere. The web application of fabric act as front end of SDK and fabric act as back end.

The hyperledger fabric SDK provides a query content of ledger with minimal code. The hyperledger fabric SDK implements the fabric programing model as developing applications.

The source code of each chaincode must be review and sign it to prevent the tampering. The ‘chaincodeDeploymentSpec’ package is used to pass around the chaincode implementation. The chaincode is used when multiple entities agrees on chaincode.

The chaincode Deployment Spec package includes source code, policies for installing the chaincode application and list of entities that have agreed on chaincode.

As the chaincode endorsed properly then it can be installed and instantiated on peers in the network.

The docker is used to run a container with the chaincode inside the peer during instantiation process.

Specifically Kubernetes is designed around pod which are having their own networking model.

3.2.2 Hyperledger Composer Tool

The hyperledger composer is a set of collaboration of tools which are used for building business network which makes the business simple and fast for business owners.

It helps developers to create smart contract and block chain application to solve business problem. The composer is built using JavaScript the leveraging modern tools includes node.js, nmp, CLI and popular editors.

Composer offers business centric abstraction and sample apps with easy to test. It develops the process to crate robust blockchain solution.

The blockchain solution used to derive alignment across business requirement by using technical development.

The hyperledger composer is used for various business person to develop such as:

- Defining assets which are exchanged in blockchain
- Defining the business rules around d transactions
- Defining participant's identity and access controls for their role exist

The developers used hyperledger composer to open toolset for following things:

- Make model reusable
- Generate JavaScript and REST APIs on business network
- Develop and test on web based composer

The following are the main components of hyperledger composer:

- Business Network Archive
- Composer Playground
- REST API Support And Integration Capabilities

Business Network Archive

- It captures the core data in business network.
- It includes business models, transaction logic and access control.

Composer Playground

- It is web based tool allows to learn hyperledger composer.
- It Helps To Model Out Their Business Network.
- It tests the network and deploy network to live instance of blockchain.

REST API Support And Integration Capabilities

- The REST API is used easily to consume client applications.
- It integrates non blockchain application.
- A loopback connector are developed for business network which exposes a running network as REST API.