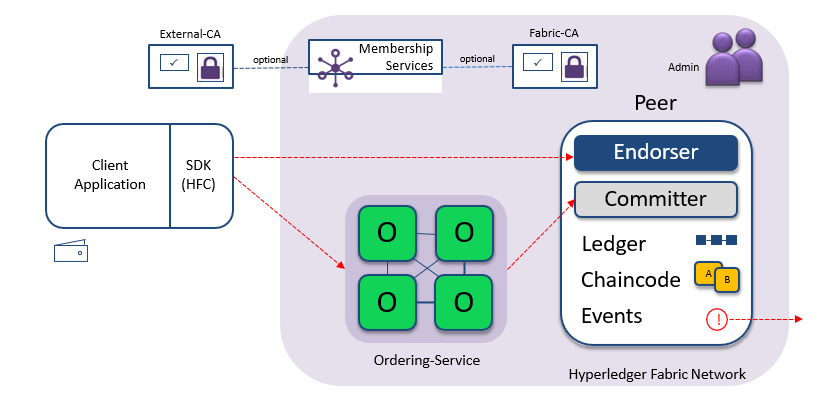
**Lecture 4-8\_Notes**

**Hyperledger Fabric - Transaction Flow**

**8.1 Hyperledger Fabric V1 Architecture**



* In the high-level architecture different key components of the hyper ledger fabric architecture are shown in the above figure.
* There is a membership service providing the notion of identity for the users who are going to be transacting on the block chain.
* This identity is going to be a digital certificate and users are going to be using this digital certificate to sign their transactions and submit them to the block chain and the benefit of signing this transaction is twofold: one they authenticate with the block chain that they are a legitimate user and two it also ensures that they get the right access privileges on the block chain for the transactions they are performing.
* Notions of access control facilitates certain users are allowed to perform certain transactions and if you do not have the right access then your transaction will get rejected.
* The certificate is going to have all that information about you what privileges you can have what attributes you have and so on. It can be used to transact on the block chain. It comes from traditional certificate party; fabric implements a certificate authority.
* This certificate authority can be external there are large number of organizations that provide these services and those can be used, and all of this relies on public key infrastructure.
* Each user will have public key and a private key associated with, the private key needs to be private only to user. The public key is something that user advertise to the entire world, and the certificate authority helps you to advertise that.
* The client application can be written in absolutely any language of choice and we provide a set of SDKs to interact with the block chains, so it is called the hyper ledger fabric client.
* This is the HFC SDK is available in multiple languages, it is available in node js, it is available in Java, it is available in python, etc.0you can use any of those SDKs to perform your transactions on block chain.
* An important component is of course the peer. There can be multiple organizations that are each running one or more peer.
* It is possible that one organization runs multiple peers several reasons for fault tolerance reasons for just separating out different applications it is part of.
* It is possible for one organization to run multiple peers and of course, there are multiple organizations there are run in the network.
* Apart from the peer there is an ordering service, this ordering service can also be run by multiple organizations or one organization you can architect it.
* The goal of the ordering service provides to give you totally ordered set of transactions, different peers might be submitting transactions to the ordering service and the ordering service is just gone overtake this transaction and put them in some order.
* The peer is going to be maintaining the ledger this is the ledger of all transactions all the state information that is getting stored. It has the smart contracts we call them chain code, so this is the code that is gone a be running on the block chain itself. Every node in the network will be executing this chain codes.
* Peer can serve as an endorser, it is going to execute the chain code, it has a copy of the chain code with it.
* It will be responsible for executing the chain code and signing the output of that chain code.
* The committing function can be thought of a separate function. It is possible that certain peers are both endorsers and committers, there are certain other peers are only committers.

**8.2 Modular Architecture**

The modular architecture of Hyperledger Fabric separates the transaction processing workflow into three different stages: smart contracts called chaincode that comprise the distributed logic processing and agreement of the system, transaction ordering, and transaction validation and commitment. This segregation offers multiple benefits:

* A reduced number of trust levels and verification that keeps the network and processing clutter-free
* Improved network scalability
* Better overall performance

Additionally, Hyperledger Fabric’s support for plug-and-play of various components allows for easy reuse of existing features and ready-made integration of various modules. For instance, if a function already exists that verifies the participant’s identity, an enterprise-level network simply needs to plug and reuse this existing module instead of building the same function from scratch.

The participants on the network have three distinct roles:

* Endorser
* Committer
* Committer

In a nutshell, the transaction proposal is submitted to the endorser peer according to the predefined endorsement policy about the number of endorsers required. After sufficient endorsements by the endorser(s), a batch or block of transactions is delivered to the committer(s). Committers validate that the endorsement policy was followed and that there are no conflicting transactions. Once both the checks are made, the transactions are committed to the ledger.

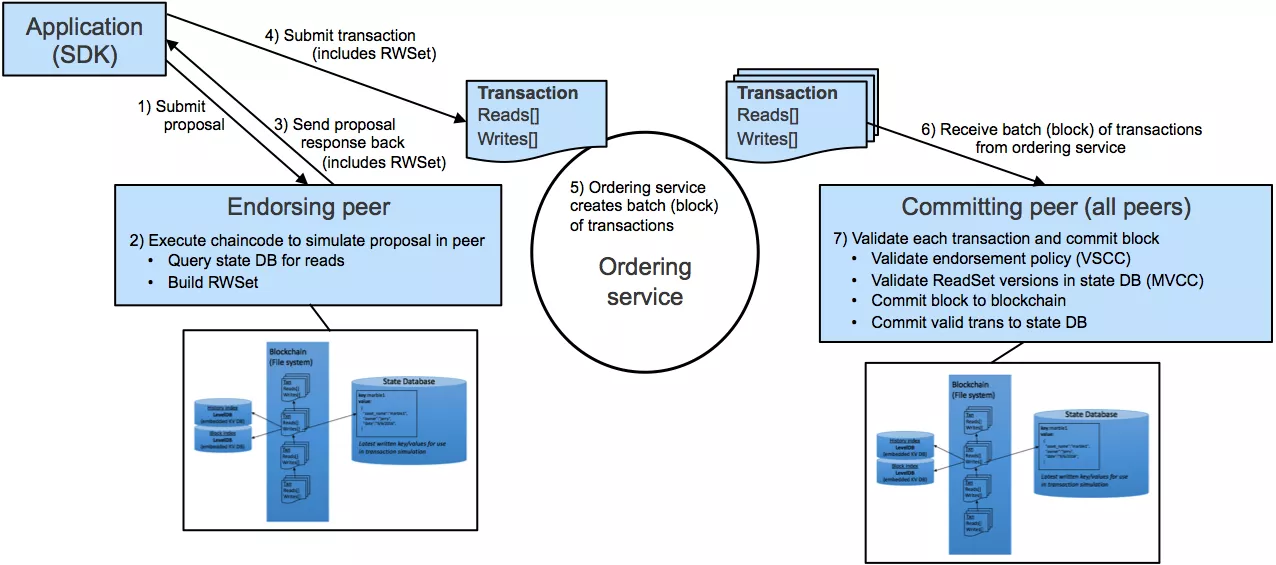


Image source: IBM

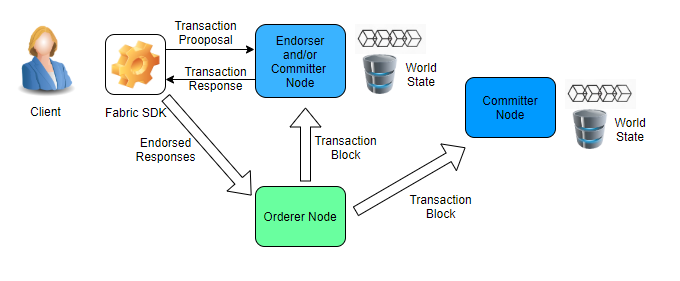
Since only confirming instructions such as signatures and read/write set are sent across the network, the scalability and performance of the network is enhanced. Only endorsers and committers have access to the transaction, and security is improved with a fewer number of participants having access to key data points.

**8.3 Nodes and Roles**

* Committing all the transactions that others are performing so that is a committing peer and it maintains the ledger and the state, but it does not execute any transactions and it does not hold the smart contracts themselves.
* Endorsing peer is specialize notion so beyond just committing all the transactions, it is also responsible for executing the transactions coming up with the output and then agreeing whether this is legitimate transaction or not.
* An ordering node is going to approve the inclusion of transactions into a particular block, it is going to order the transaction that may be seen across the networks from multiple peers and it is going to communicate to all the peers what is the order in which the transaction must be committed.

**8.4 Hyperledger Fabric Basic Transaction Flow**

Before going to each component in detail let us see a high-level transaction flow and basic components involved.



Hyperledger Fabric Basic Transaction Flow

* Here when clients submit the transaction proposal through the Fabric SDK, this proposal is sent to all Endorsing Peers.
* These endorsing peers check the transaction verifies and executes and generate the Read and Write set as output.
* This response is again sent to the client.
* The client collects all responses from all endorsing peers and send them to Orderer.
* Orderer sees all transactions and orders them in ascending order and form a block.
* This block is sent to all committers which checks the transaction and add a new block in their own copy of the ledger.