PART 1

Blockchain Network

1. All blockchain technologies have this concept of nodes.
2. Nodes connect to each other to form the blockchain network.
3. In public, blockchain networks such as Ethereum and Bitcoin, all nodes are equal, but in the case of Hyperledger Fabric, all nodes may not be equal in the Hyperledger fabric distributed ledger technology or blockchain technology.

Concept of Members

1. There is a concept of members.
2. Members are legally separate entities that join the blockchain network to transact. (Members = legally separate entities)
3. These members share one or more distributed ledgers and each of these members host nodes.
4. These nodes are used for submitting the transactions and for managing the state of the ledger within the organization.
5. Each node is assigned an identity by way of certificates.
6. Even the users in the blockchain ecosystem built on Hyperledger fabric are assigned a certificate

Ledger

1. Ledgers are used for managing the state of assets.
2. These assets in the Hyperledger Fabric Network may represent anything in the real world, such as bonds,loans, title to a house, or it can be anything that can be represented digitally.
3. Nodes initiate the transactions which lead to the execution of the chain code, and the transactions are recorded in the ledger in the order they are received.
4. These chain code executions also lead to the change of the state of the assets.
5. There is one ledger per channel, so channels are a way by which transaction privacy is achieved in the Hyperledger Fabric Network.
6. All of the nodes and users in the Hyperledger Fabric Network are assigned an

identity, and that is done by way of the membership service Providers and Certification Authority.

Node

There are three types of nodes in the Hyperledger Fabric Network,

1. orderer nodes:
   1. Orderer Nodes Provide the communication channel for the fabric network.
   2. They are also known as the ordering service.
   3. The primary responsibility of the order nodes is to ensure a consistent state of the ledger across the blockchain network.
   4. They provide a consensus mechanism and make sure that the order of the transactions is maintained.
   5. Order creates the blocks and guarantees atomic delivery of these blocks to the peer nodes in the network.
   6. Order is implemented with message oriented middleware.
   7. Solo, which is a messaging component, is built into the order node.
   8. And And if you use solo, you can have only a single order instance, which is okay for development, but obviously will lead to a single point of failure in production , which is not the way to go.
   9. In a live network, you would use the raft consensus.
   10. The raft consensus mechanism is built into the order of binary like solo.
   11. Another option for setting up the ordering service in production is the Kafka setup. But the challenge with Kafka is that you need to set up the Kafka brokers across the multiple member organizations, and this adds complexity in the network and a number of additional moving parts. So at this point, the recommendation is to go with Raft in a live network.
2. peer nodes: (peers may be tagged a role)
   1. Peer can be a leader, peer or an anchor peer.
   2. And for the lack of better term, a peer that has not been tagged with the leader or the anchor role may be referred to as the regular peer.
   3. Each of the pier instance in a network maintains its own copy of the ledger.
   4. There are two parts in this ledger.
   5. Part one is the transaction log and part two is the state database.
   6. Both of these are implemented by default using Leveldb, which is an in-process database.
   7. Transaction log is immutable.
   8. What that means is that once a transaction is added to the transaction log, it can neither be deleted nor it can be changed.
   9. Peer may be configured to use Couchdb for state database.
   10. That way you can run complex queries against the state data maintained for the chain code.
   11. ACHOR PEERS
   12. A peer node maintains the transaction log, the state database, and it also manages the chain code deployed to it.
   13. The peer node exposes services and these services are built on gRPC.
   14. These services are invoked by the clients, by the other peers and by the orderer for sending blocks to the peer peers, exchange block data by way of gossip, data dissemination protocol.
   15. An organization may set up multiple peers for scalability, isolation and performance reasons, but only the anchor peers are known outside the organization.
   16. In other words, they are the only peer instances within an organization that are discoverable.
   17. Each organization must have at least one anchor peer.
   18. Here is an example two organization Organization A that has two anchor peers and organization B, which has one anchor peer.
   19. The anchor peer is an organization A and B are discoverable.
   20. As a result, the peers may engage in gossip, protocol based data dissemination protocol.
   21. But since the regular peers in Organization B are not discoverable by the anchor peer in organization, a such connection between the two peers is not possible.
3. LEADERS PEER (receive blocks from ORDERER)
   1. Not all peers in the network connect directly to the order for receiving the blocks.
   2. Only the leader peers in the organization receive blocks from the orders.
   3. Regular peer is tagged as a leader.
   4. Either statically or the administrator of the organization can set up the nodes in the organization to dynamically elect a leader.
   5. Leadership assignment is at a channel level.
   6. The idea is that a peer node can join multiple channels, so this assignment is done at the channel level.
   7. Here is how the infrastructure for an organization may look like.
   8. There will be an anchor peer, a leader peer, which may be static or dynamically assigned, and then there will be a set of regular peers.
   9. Leader Peer will receive the new blocks from the order and then by way of the gossip data dissemination protocol, send it to the other peers within the organization.
4. client nodes ( Client Node acts on behalf of the end user.)
   1. They are created by way of SDKs, such as the Golang SDK for Hyperledger Fabric or the Node SDK.
   2. The client node is also known as the submitting client.
   3. Client node are the one that submit the transaction requests to the network.
   4. Channels are a way by which transaction privacy is achieved in the Hyperledger Fabric based blockchain network.
5. CHANNELS (Transaction Isolated With in the channel)
   1. The client node is also known as the submitting client.
   2. Client node are the one that submit the transaction requests to the network.
   3. Channels are a way by which transaction privacy is achieved in the Hyperledger Fabric based blockchain network.
   4. All transactions in a channel are isolated.
   5. In other words, transactions created on that channel are available only to the members of that channel.
   6. Chaincode is deployed to a channel, not to the network.
   7. There is a special system channel known as the Ordering System Channel.
   8. It is also referred to as the bootstrap channel.
   9. This channel gets automatically created at the time of the network initialization peers and hence the organizations may join multiple channels.
   10. Fabric uses PKI for identity management.
   11. It follows the typical process used for identity management by way of certification authority.
   12. This Hyperledger fabric has an implementation of the certification authority CA server.
   13. The server exposes services for identity registration, identity enrollment and certificate management.