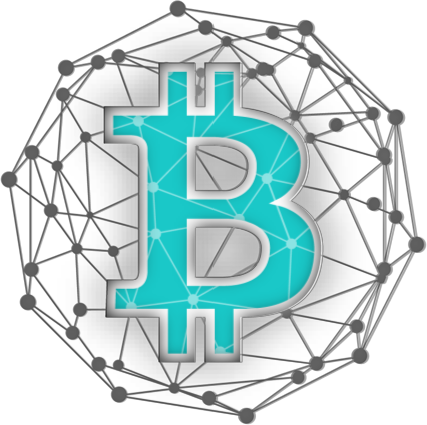
Blockchain Certification Training <https://www.edureka.co/blockchain-training>



|  |
| --- |
| Module 7: Hyperledger |
| Additional Resources |



© Brain4ce Education Solutions Pvt. Ltd.

Module 7: Hyperledger <https://www.edureka.co/blockchain-training>



# ANCHOR PEER

A peer node on a channel that all other peers can discover and communicate with. Each member on a channel has an anchor peer (or multiple anchor peers to prevent single point of failure), allowing for peers belonging to different members to discover all existing peers on a channel.

# BLOCKCHAIN NETWORK

A blockchain network consists of, at minimum, one peer (responsible for endorsing and committing transactions) leveraging an ordering service, and a membership services component (certificate authority) that distributes and revokes cryptographic certificates representative of user identities and permissions.

# BOOTSTRAP

The initial setup of a network. There is the bootstrap of a peer network, during which policies, system chaincodes, and cryptographic materials (certs) are disseminated amongst participants, and the bootstrap of an ordering network. The bootstrap of the ordering network must precede the bootstrap of the peer network, as a peer network is contingent upon the presence of an ordering service. A network need only be “bootstrapped” once.



**CHAINCODE**

Chaincode is software, running on a ledger, to encode assets and the transaction instructions (business logic) for modifying the assets.

**CHAIN MEMBER**

Entities that do not participate in the validation process of a blockchain network, but help to maintain the integrity of a network. Unlike chain transactors, chain members maintain a local copy of the ledger.

# CHAIN TRANSACTOR

Entities that have permission to create transactions and query network data.

# CHAIN VALIDATOR

Entities that own a stake of a chain network. Each chain validator has a voice in deciding whether a transaction is valid, therefore chain validators can interrogate all transactions sent to their chain.

# CHAIN AUDITOR

Entities with the permission to interrogate transactions.

# CHANNEL

A channel is a private blockchain overlay which allows for data isolation and confidentiality. A channel- specific ledger is shared across the peers in the channel, and transacting parties must be properly authenticated to a channel to interact with it. Channels are defined by a configuration-block

© B r a i n 4 c e E d u c a t i o n S o l u t i o n s P v t . L t d Page 1

Module 7: Hyperledger <https://www.edureka.co/blockchain-training>



# COMMITMENT

Each peer on a channel validates ordered blocks of transactions and then commits (writes/appends) the blocks to its replica of the channel ledger. Peers also mark each transaction in each block as valid or invalid.

# CONCURRENCY CONTROL VERSION CHECK

Concurrency control version check is a method of keeping state in sync across peers on a channel. Peers execute transactions in parallel, and before commitment to the ledger, peers check that the data read at execution time has not changed. If the data read for the transaction has changed between execution time and commitment time, then a concurrency control version check violation has occurred, and the transaction is marked as invalid on the ledger and values are not updated in the state database.

# CONFIGURATION BLOCK

Contains the configuration data defining members and policies for a system chain (ordering service) or channel. Any configuration modifications to a channel or overall network (e.g. A member leaving or joining) will result in a new configuration block being appended to the appropriate chain. This block will contain the contents of the genesis block, plus the delta.

# CURRENT STATE



The current state of the ledger represents the latest values for all keys ever included in its chain transaction log. Peers commit the latest values to ledger current state for each valid transaction included in a processed block. Since current state represents all latest key values known to the channel, it is sometimes referred to as world state. Chaincode executes transaction proposals against current state data.

**DYNAMIC MEMBERSHIP**

Hyperledger fabric supports the addition/removal of members, peers, and ordering service nodes, without compromising the operationality of the overall network. Dynamic membership is critical when business relationships adjust, and entities need to be added/removed for various reasons.

# ENDORSEMENT

Refers to the process where specific peer nodes execute a chaincode transaction and return a proposal response to the client application. The proposal response includes the chaincode execution response

message, results (read set and write set), and events, as well as a signature to serve as proof of the peer’s chaincode execution. Chaincode applications have corresponding endorsement policies, in which the endorsing peers are specified.

# ENDORSEMENT POLICY

Defines the peer nodes on a channel that must execute transactions attached to a specific chaincode application, and the required combination of responses (endorsements). A policy could require that a transaction be endorsed by a minimum number of endorsing peers, a minimum percentage of endorsing peers, or by all endorsing peers that are assigned to a specific chaincode application. Policies can be curated based on the application and the desired level of resilience against misbehavior (deliberate or not) by the endorsing peers. A transaction that is submitted must satisfy the endorsement policy before

© B r a i n 4 c e E d u c a t i o n S o l u t i o n s P v t . L t d Page 2

Module 7: Hyperledger <https://www.edureka.co/blockchain-training>



being marked as valid by committing peers. A distinct endorsement policy for install and instantiate transactions is also required.

# HYPERLEDGER FABRIC CA

Hyperledger fabric ca is the default certificate authority component, which issues pki-based certificates to network member organizations and their users. The ca issues one root certificate (rootcert) to each member and one enrolment certificate (ecert) to each authorized user.

# GENESIS BLOCK

The configuration block that initializes a blockchain network or channel, and serves as the first block on a chain.

# GOSSIP PROTOCOL

The gossip data dissemination protocol performs three functions: 1) manages peer discovery and channel membership; 2) disseminates ledger data across all peers on the channel; 3) syncs ledger state across all peers on the channel.

# INITIALIZE

A method to initialize a chaincode application.



**INSTALL**

The process of placing a chaincode on a peer’s file system.

**INSTANTIATE**

The process of starting and initializing a chaincode application on a specific channel. After instantiation,

peers that have the chaincode installed can accept chaincode invocations.

# INVOKE

Used to call chaincode functions. A client application invokes chaincode by sending a transaction proposal to a peer. The peer will execute the chaincode and return an endorsed proposal response to the client application. The client application will gather enough proposal responses to satisfy an endorsement policy, and will then submit the transaction results for ordering, validation, and commit. The client application may choose not to submit the transaction results. For example, if the invoke only queried the ledger, the client application typically would not submit the read-only transaction, unless there is desire to log the read on the ledger for audit purpose. The invoke includes a channel identifier, the chaincode function to invoke, and an array of arguments.

# LEADING PEER

Each member can own multiple peers on each channel that it subscribes to. One of these peers is serves as the leading peer for the channel, to communicate with the network ordering service on behalf of the member. The ordering service “delivers” blocks to the leading peer(s) on a channel, who then distribute them to other peers within the same member cluster.

© B r a i n 4 c e E d u c a t i o n S o l u t i o n s P v t . L t d Page 3

Module 7: Hyperledger <https://www.edureka.co/blockchain-training>



# LEDGER

A ledger is a channel’s chain and current state data which is maintained by each peer on the channel.

# MEMBER

A legally separate entity that owns a unique root certificate for the network. Network components such as peer nodes and application clients will be linked to a member.

## MEMBERSHIP SERVICE PROVIDER

The membership service provider (msp) refers to an abstract component of the system that provides credentials to clients, and peers for them to participate in a hyperledger fabric network. Clients use these credentials to authenticate their transactions, and peers use these credentials to authenticate transaction processing results (endorsements). While strongly connected to the transaction processing components of the systems, this interface aims to have membership services components defined, in such a way that alternate implementations of this can be smoothly plugged in without modifying the core of transaction processing components of the system.

## MEMBERSHIP SERVICES

Membership services authenticates, authorizes, and manages identities on a permissioned blockchain network. The membership services code that runs in peers and orderers both authenticates and authorizes blockchain operations. It is a pki-based implementation of the membership services provider (msp) abstraction.

## MULTI-CHANNEL

The fabric will allow for multiple channels with a designated ledger per channel. This capability allows for multilateral contracts where only the restricted participants on the channel will submit, endorse, order, or commit transactions on that channel. As such, a single peer can maintain multiple ledgers without compromising privacy and confidentiality.

## ORDERING SERVICE

A defined collective of nodes that orders transactions into a block. The ordering service exists independent of the peer processes and orders transactions on a first-come-first-serve basis for all channels on the network. The ordering service is designed to support pluggable implementations beyond the out-of-the-box solo and kafka varieties. The ordering service is a common binding for the overall network; it contains the cryptographic identity material tied to each member.

## ORDERER

One of the network entities that form the ordering service. A collection of ordering service nodes (osns)

will order transactions into blocks according to the network’s chosen ordering implementation. In the

case of “solo”, only one osn is required. Transactions are “broadcast” to orderers, and then “delivered” as

blocks to the appropriate channel.

## PEER

A network entity that maintains a ledger and runs chaincode containers to perform read/write operations to the ledger. Peers are owned and maintained by members.

## PERMISSIONED NETWORK

© B r a i n 4 c e E d u c a t i o n S o l u t i o n s P v t . L t d Page 4

Module 7: Hyperledger <https://www.edureka.co/blockchain-training>



A blockchain network where any entity (node) is required to maintain a member identity on the network. End users must be authorized and authenticated to use the network.

## POLICY

There are policies for endorsement, validation, chaincode management and network/channel management.

## PROPOSAL

A request for endorsement that is aimed at specific peers on a channel. Each proposal is either an instantiate or an invoke (read/write) request.

## QUERY

A query is a chaincode invocation which reads the ledger current state but does not write to the ledger. The chaincode function may query certain keys on the ledger, or may query for a set of keys on the ledger. Since queries do not change ledger state, the client application will typically not submit these read-only transactions for ordering, validation, and commit.

Although not typical, the client application can choose to submit the read-only transaction for ordering, validation, and commit, for example if the client wants auditable proof on the ledger chain that it had knowledge of specific ledger state at a certain point in time.

## SOFTWARE DEVELOPMENT KIT (SDK)

The hyperledger fabric client sdk provides a structured environment of libraries for developers to write and test chaincode applications. The sdk is fully configurable and extensible through a standard interface. Components, including cryptographic algorithms for signatures, logging frameworks and state stores, are easily swapped in and out of the sdk. The sdk provides apis for transaction processing, membership services, node traversal and event handling. The sdk comes in multiple flavors: node.js, java. And python.

## STATE DATABASE

Current state data is stored in a state database for efficient reads and queries from chaincode. Supported databases include leveldb and couchdb.

## SYSTEM CHAIN

Contains a configuration block defining the network at a system level. The system chain lives within the ordering service, and like a channel, has an initial configuration containing information such as: msp information, policies, and configuration details. Any change to the overall network (e.g. A new org joining, or a new ordering node being added) will result in a new configuration block being added to the system chain.

The system chain can be thought of as the common binding for a channel or group of channels. For instance, a collection of financial institutions may form a consortium (represented through the system chain), and then proceed to create channels relative to their aligned and varying business agendas.

## TRANSACTION

Invoke or instantiate results that are submitted for ordering, validation, and commit. Invokes are requests to read/write data from the ledger. Instantiate is a request to start and initialize a chaincode on a

© B r a i n 4 c e E d u c a t i o n S o l u t i o n s P v t . L t d Page 5

Module 7: Hyperledger <https://www.edureka.co/blockchain-training>



channel. Application clients gather invoke or instantiate responses from endorsing peers and package the results and endorsements into a transaction that is submitted for ordering, validation, and commit.

© B r a i n 4 c e E d u c a t i o n S o l u t i o n s P v t . L t d Page 6