Blockchain DLOC Sem VII

HD - Blockchain Platforms

Module - 5 : Hyperledger Blockchain

Instructors : Mrs. Lifna C S

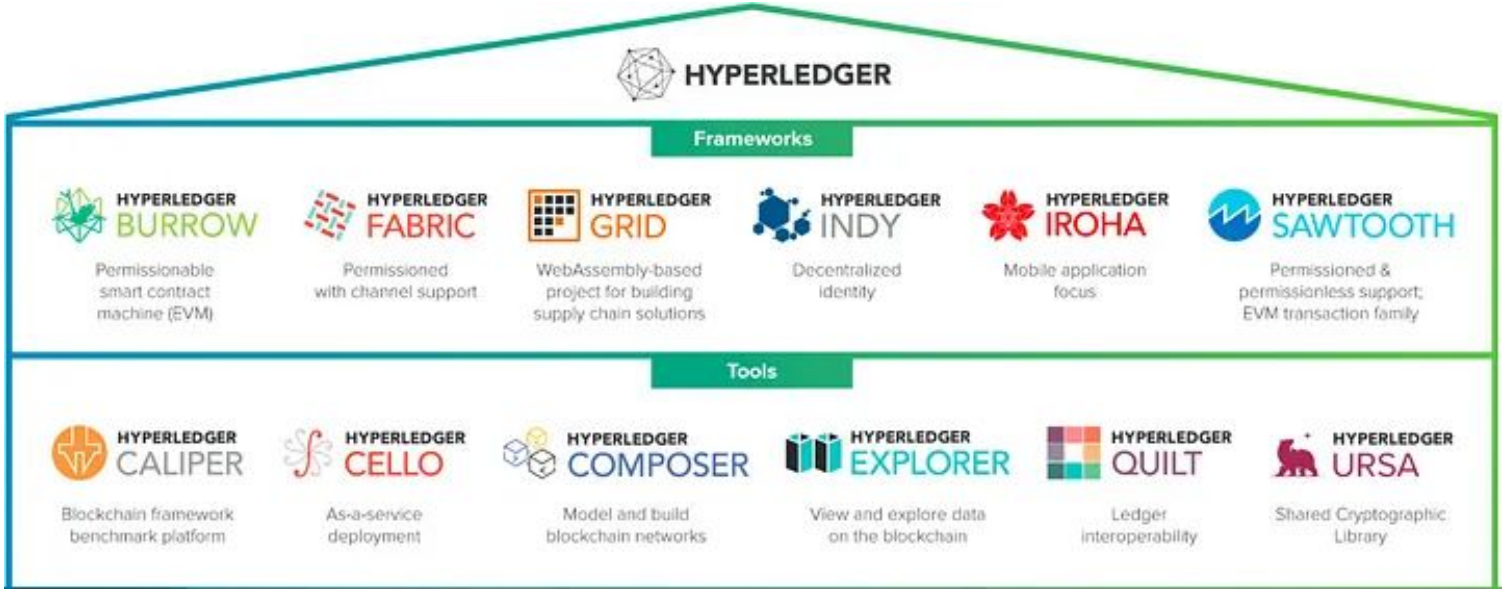
Department of Computer Engineering, VESIT, Mumbai

Topics to be covered

- Introduction to Hyperledger, tools and frameworks,
- Hyperledger Fabric,
- Comparison between Hyperledger Fabric & Other Technologies,
- Distributed Ledgers.
- Hyperledger Fabric Architecture,
- Components of Hyperledger Fabric: MSP, ChainCodes, etc.,
- Transaction Flow,
- Advantages of Hyperledger Fabric Blockchain,
- Working of Hyperledger Fabric
- Creating Hyperledger network,

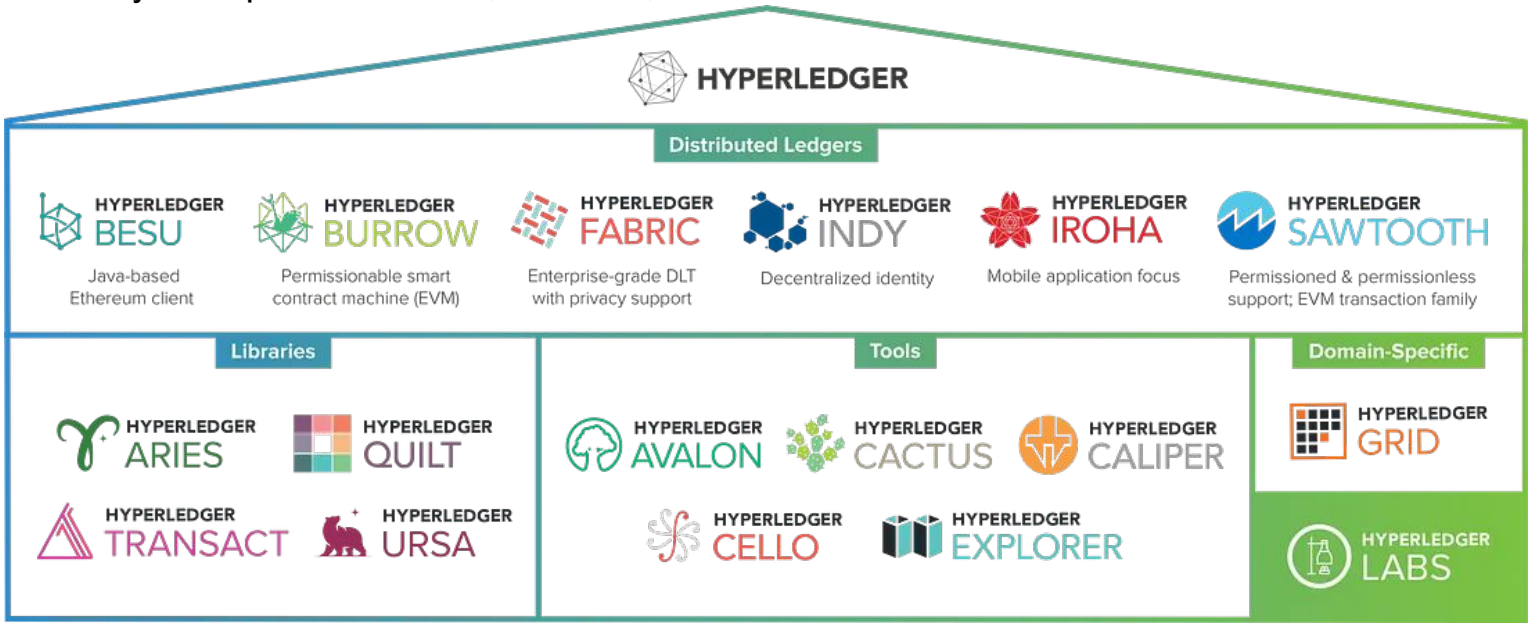
Introduction to Hyperledger - Tools & Frameworks

- An open-source collaborative project hosted by The Linux Foundation
- Aimed at advancing cross-industry blockchain technologies
- Enterprise-grade and permissioned
- Focused on modularity and scalability



Introduction to Hyperledger Fabric

- A modular and extensible architecture
- Supports plug-and-play components
- Permissioned network with identity management
- Smart contracts written in Go, JavaScript, or Java (Chaincode)
- Used by enterprises like IBM, Walmart, Maersk



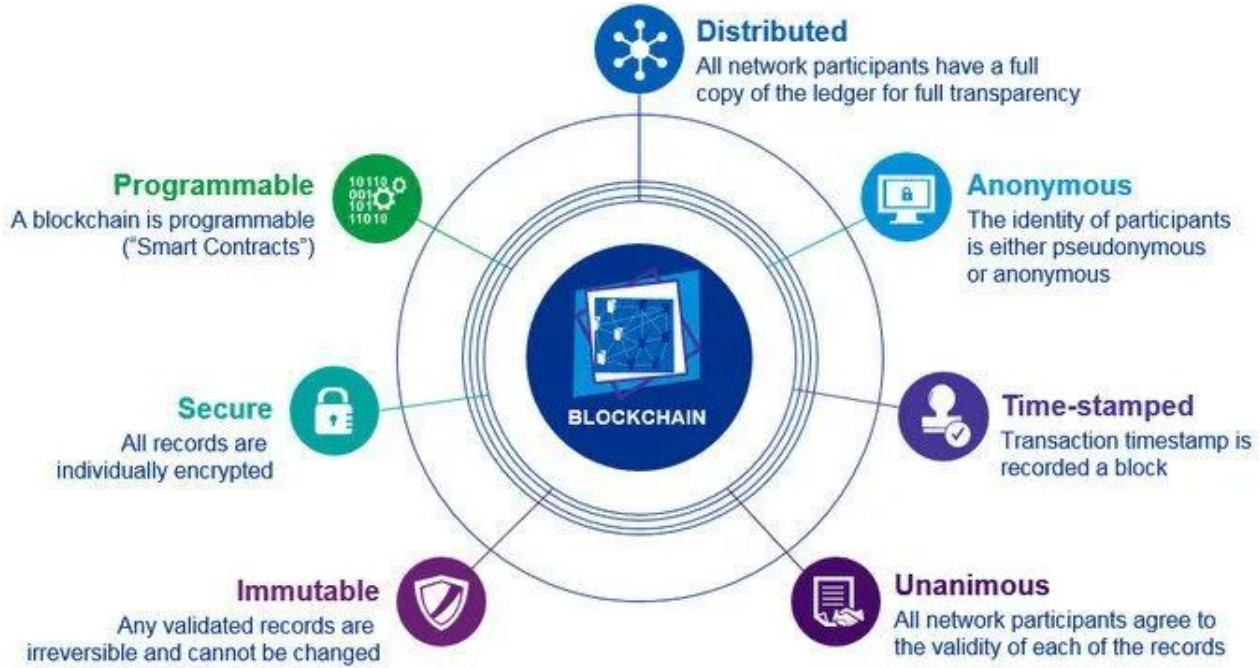
Comparison between Hyperledger Fabric & Other Technologies

Feature	Hyperledger Fabric	Ethereum	Bitcoin
Access	Permissioned	Permissionless	Permissionless
Smart Contracts	Chaincode (Go, JS)	Solidity	Not supported
Consensus	Pluggable (Raft, Kafka)	PoW/PoS	PoW
TPS	Thousands	30–100	~7
Privacy	High (channels, private data)	Public	Public

Distributed Ledgers

- A database that is consensually shared and synchronized
- Maintained across multiple nodes/locations
- No central administrator
- Immutable and transparent

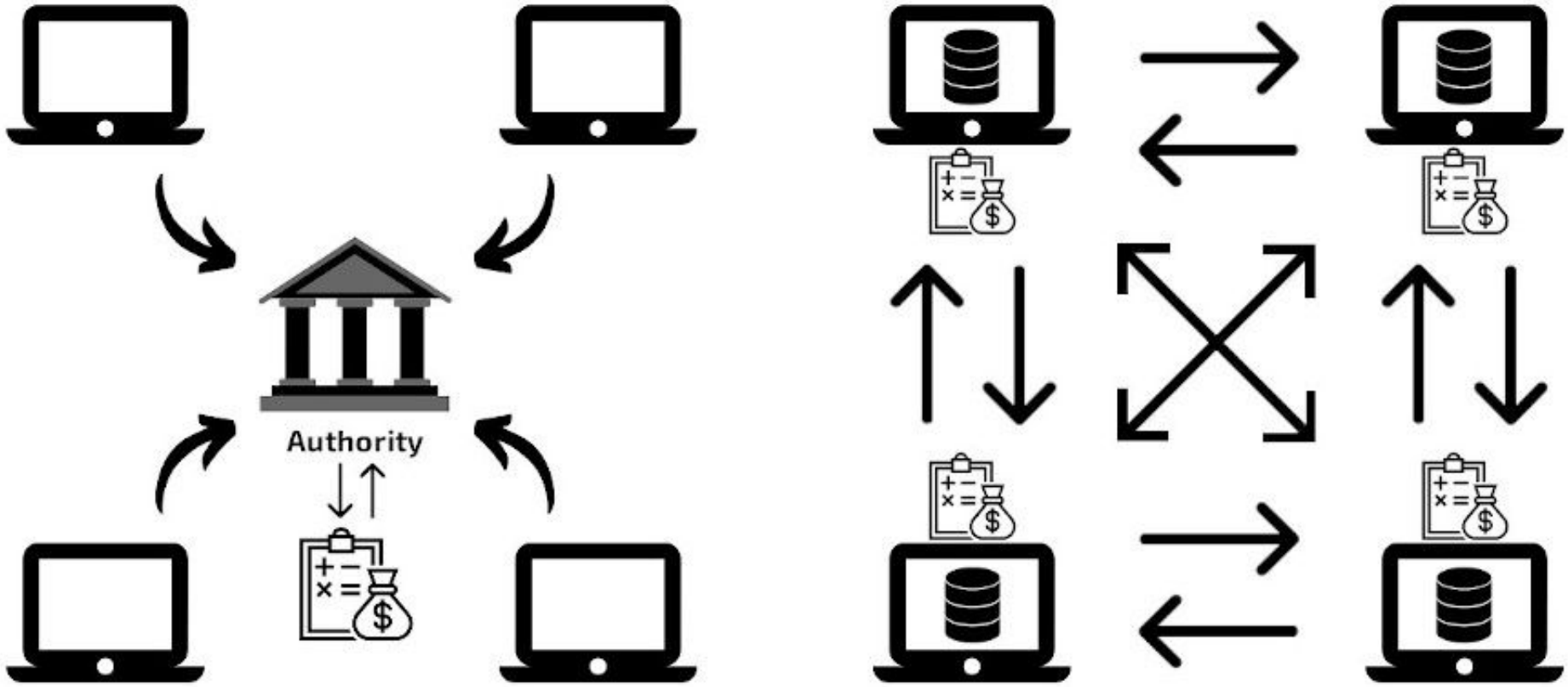
Properties of Distributed Ledger Technology



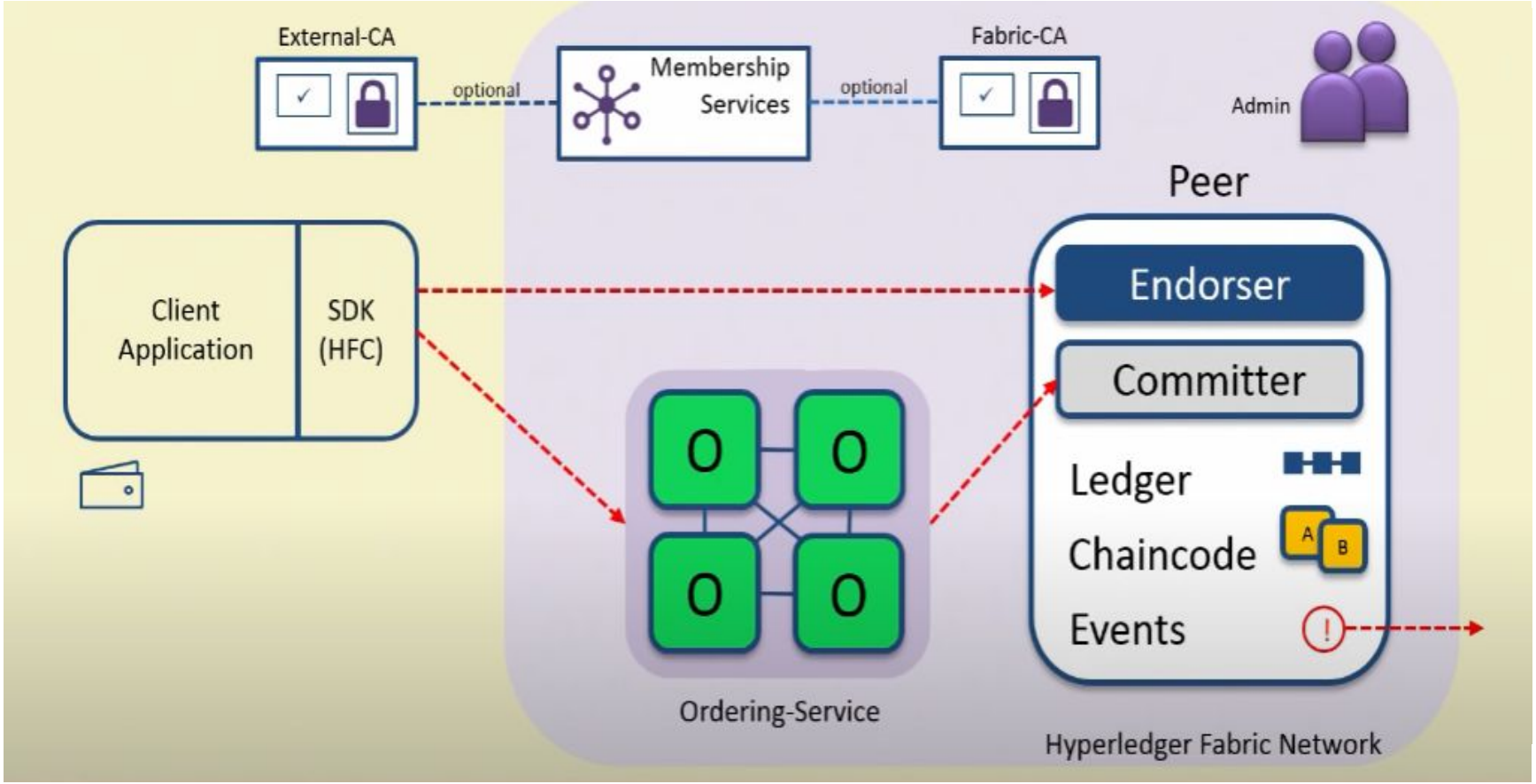
Distributed Ledger Vs Centralized Ledger

Feature	Centralized Ledger	Distributed Ledger
Control	Single authority controls and manages the ledger	Multiple parties/nodes share control
Data Storage	Stored in a central server/database	Copies are stored across several distributed nodes
Failure Point	Single point of failure – prone to outages or attacks	No single point of failure – resilient and fault-tolerant
Transparency	Limited to the organization	Transparent to all authorized participants
Security	Vulnerable if central server is compromised	Higher security via consensus and cryptography
Speed	Typically faster due to no consensus overhead	May be slower due to consensus mechanisms
Trust Model	Requires trust in a central authority	Trustless – relies on protocol and consensus
Example	Traditional banks, ERP systems	Bitcoin, Ethereum, Hyperledger Fabric

Distributed Ledger Vs Centralized Ledger



Hyperledger Fabric Architecture



Hyperledger Fabric Architecture - Components

1. **Peer Nodes** : Nodes that host ledgers and smart contracts (chaincode).
   - Types:
     - Endorsing Peer: Simulates and signs transaction proposals.
     - Committing Peer: Validates transactions and appends them to the ledger.
     - Leader Peer (optional): Handles communications with the ordering service.

📝 Key Role: Maintain ledger and execute chaincode.

2. **Ordering Service** : Ensures total order of transactions across the network.
   - Consensus Mechanism: Pluggable (Raft, Kafka, etc.)
   - Functionality:
     - Collects endorsed transactions
     - Orders them into blocks
     - Broadcasts blocks to all peers in the channel

📝 Key Role: Achieves consensus without proof-of-work.

Hyperledger Fabric Architecture - Components

3. Membership Service Provider (MSP) : Component that manages identities and permissions.

- Uses: X.509 certificates issued by a Certificate Authority (CA)
- Responsible For:
  - Authenticating users and peers
  - Authorizing actions on the network

📝 Key Role: Provides identity management and access control.

4. **Ledger**

- Structure:
  - Blockchain (Immutable): Stores blocks of transactions.
  - World State (Mutable): Stores current state as key-value pairs.
- Backends: CouchDB (rich queries) or LevelDB (default, simple key-value)

📝 Key Role: Records and maintains all historical and current data.

Hyperledger Fabric Architecture - Components

5. Chaincode (Smart Contracts) : Business logic running on the network.

- Languages: Go, JavaScript (Node.js), Java
- Lifecycle: Install → Approve → Commit → Invoke

📝 Key Role: Defines rules and logic for transactions.

6. Channel : Private sub-network between a subset of participants.

- Use Case: Enables data partitioning and confidentiality.
- Each Channel:
  - Has its own ledger
  - Allows only authorized peers to participate

📝 Key Role: Enables confidentiality within the network.

Hyperledger Fabric Architecture - Components

7. Certificate Authority (CA) : Issues digital certificates for identities in the network.

- Functionality:
  - Registration of identities
  - Enrollment (generating cryptographic credentials)

📝 Key Role: Supports MSP with identity lifecycle.

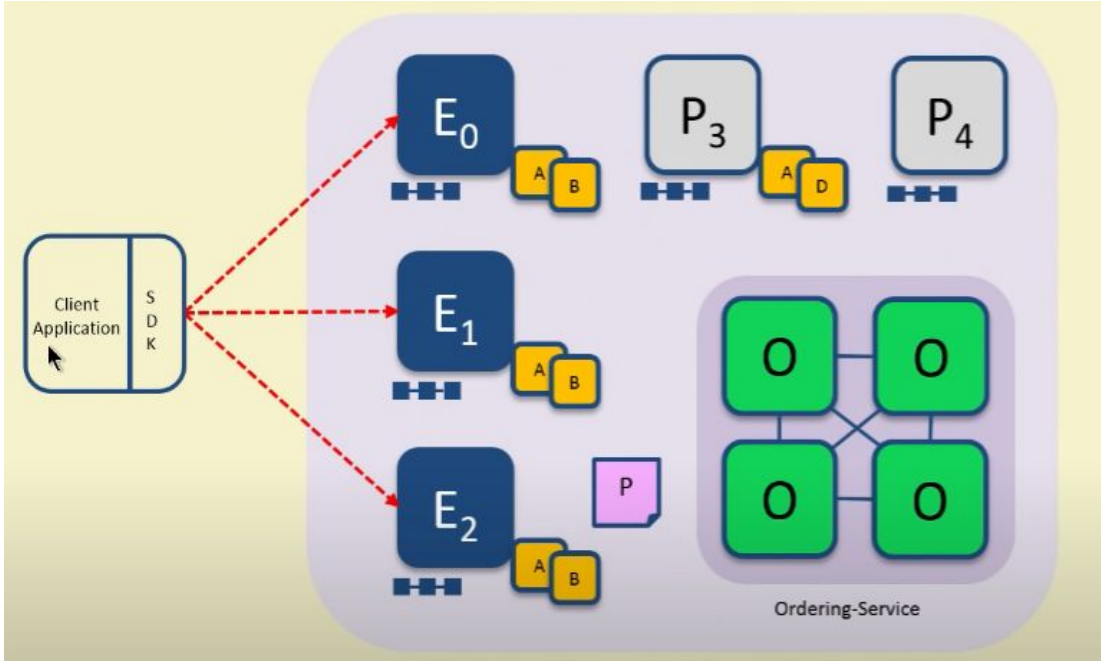
8. Gossip Protocol : Mechanism for peers to disseminate ledger data.

- Functionality:
  - Peer discovery
  - Data synchronization across peers

📝 Key Role: Ensures data consistency and redundancy.

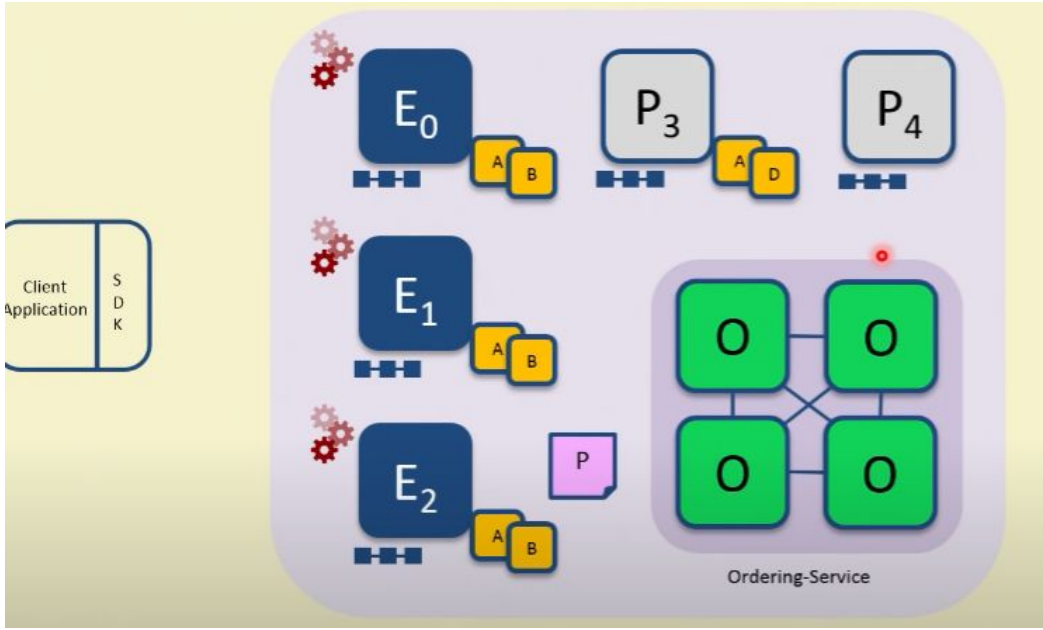
Transaction Flow in Hyperledger Fabric

1. **Client sends transaction proposal**
2. Endorsing peers simulate and sign
3. Client collects endorsements
4. Submits to Ordering Service
5. Ordering Service batches into blocks
6. Blocks delivered to peers and validated
7. Ledger updated



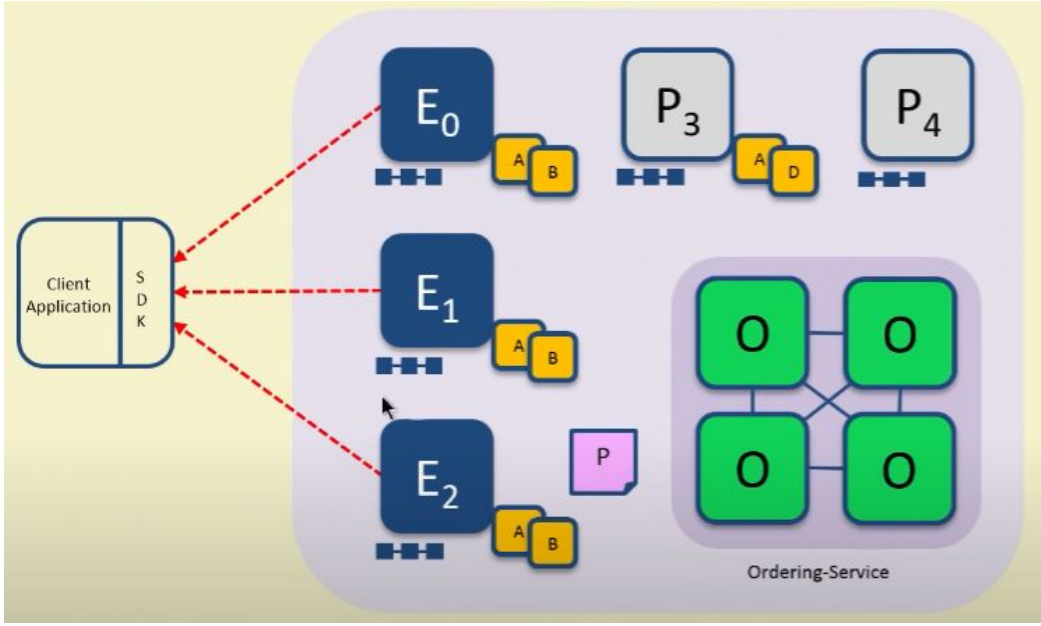
Transaction Flow in Hyperledger Fabric

1. Client sends transaction proposal
2. **Endorsing peers simulate and sign**
3. Client collects endorsements
4. Submits to Ordering Service
5. Ordering Service batches into blocks
6. Blocks delivered to peers and validated
7. Ledger updated



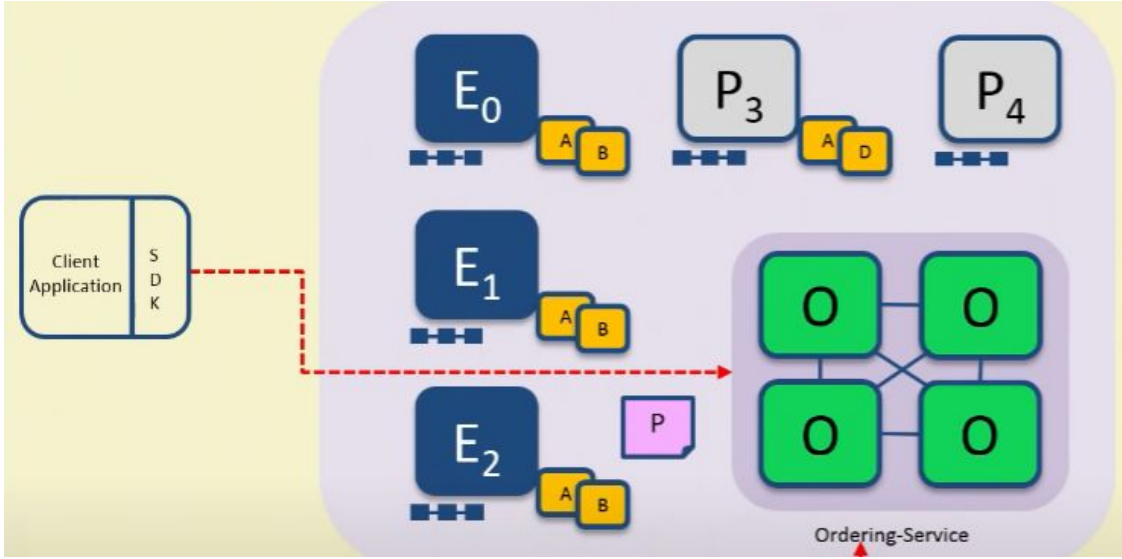
Transaction Flow in Hyperledger Fabric

1. Client sends transaction proposal
2. Endorsing peers simulate and sign
3. **Client collects endorsements**
4. Submits to Ordering Service
5. Ordering Service batches into blocks
6. Blocks delivered to peers and validated
7. Ledger updated



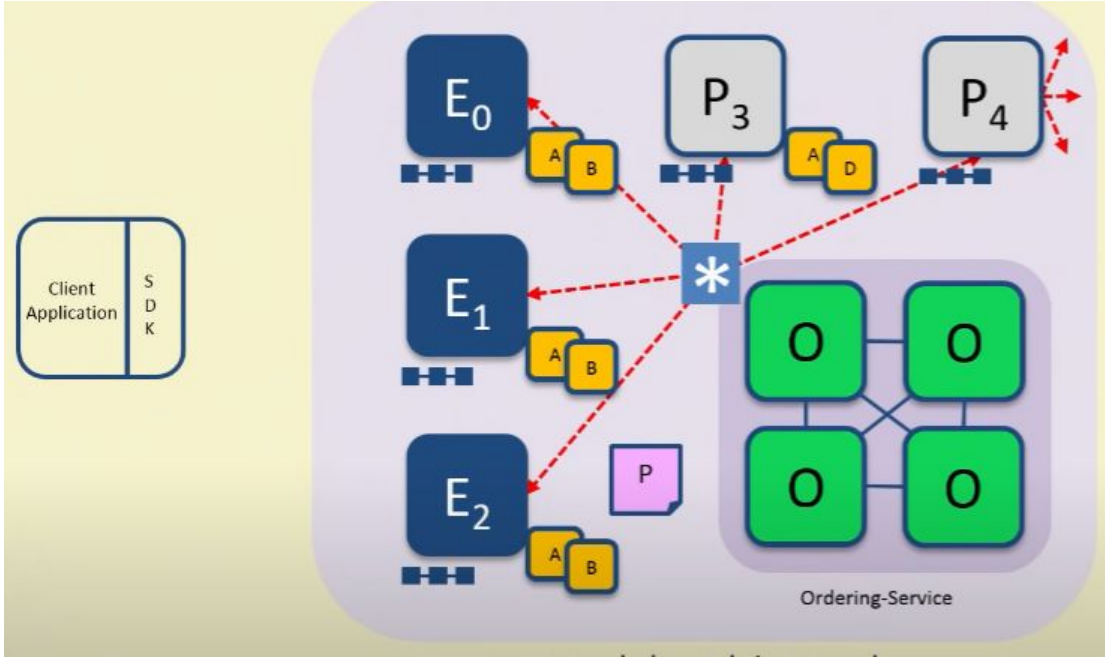
Transaction Flow in Hyperledger Fabric

1. Client sends transaction proposal
2. Endorsing peers simulate and sign
3. Client collects endorsements
4. **Submits to Ordering Service**
5. Ordering Service batches into blocks
6. Blocks delivered to peers and validated
7. Ledger updated



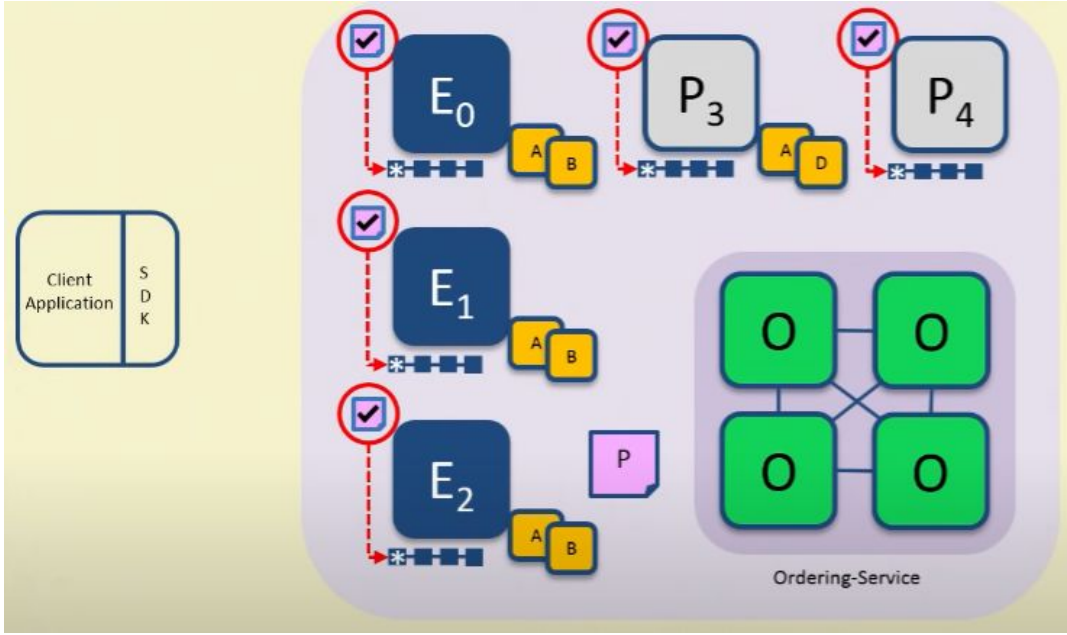
Transaction Flow in Hyperledger Fabric

1. Client sends transaction proposal
2. Endorsing peers simulate and sign
3. Client collects endorsements
4. Submits to Ordering Service
5. **Ordering Service batches into blocks**
6. Blocks delivered to peers and validated
7. Ledger updated



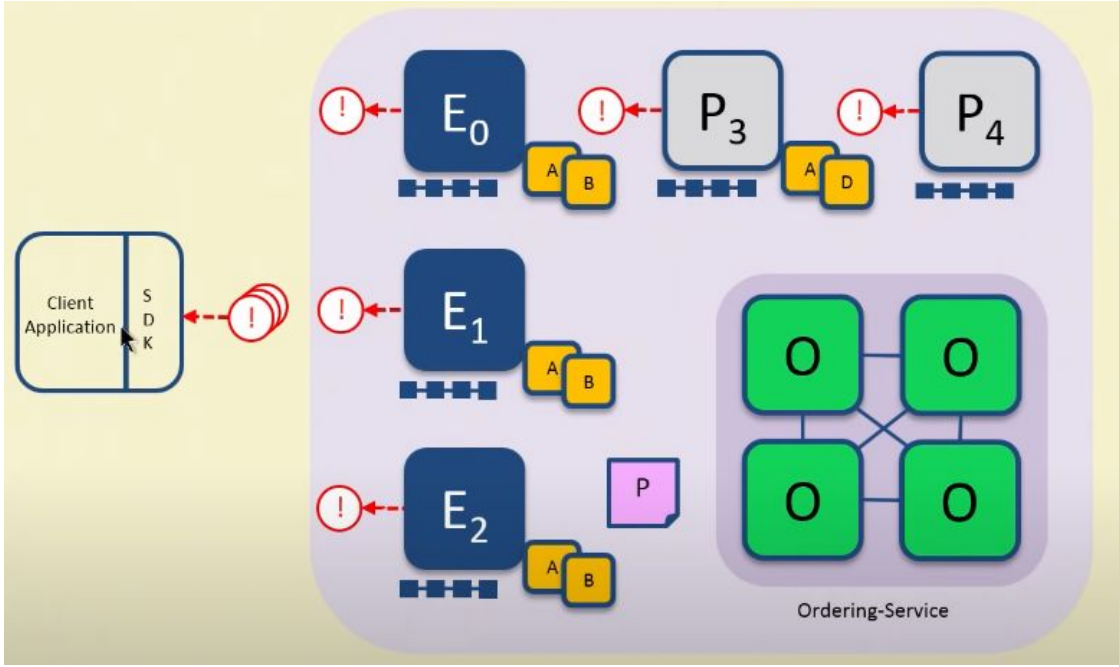
Transaction Flow in Hyperledger Fabric

1. Client sends transaction proposal
2. Endorsing peers simulate and sign
3. Client collects endorsements
4. Submits to Ordering Service
5. Ordering Service batches into blocks
6. **Blocks delivered to peers and validated**
7. Ledger updated

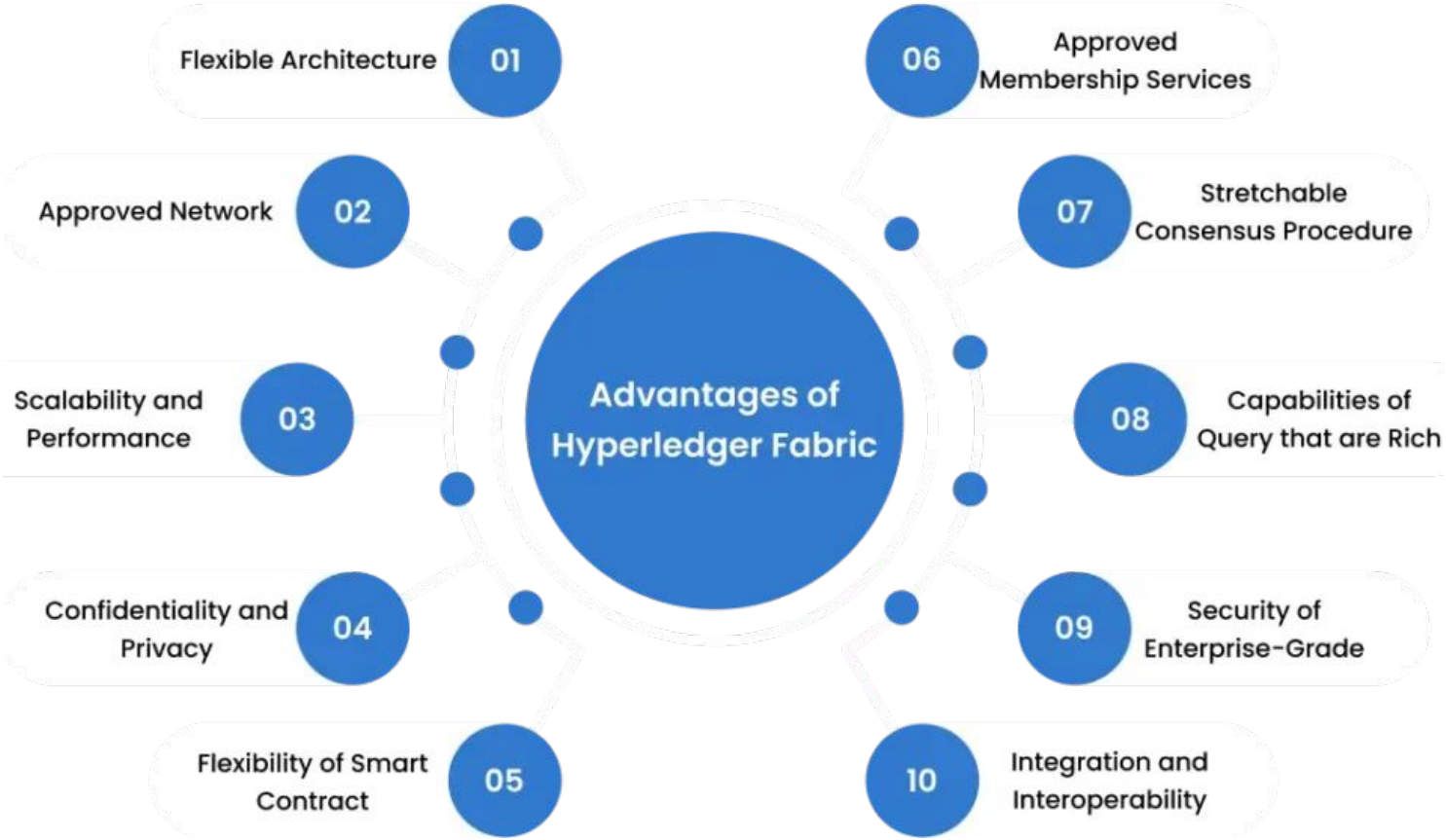


Transaction Flow in Hyperledger Fabric

1. Client sends transaction proposal
2. Endorsing peers simulate and sign
3. Client collects endorsements
4. Submits to Ordering Service
5. Ordering Service batches into blocks
6. Blocks delivered to peers and validated
7. **Ledger updated**



Advantages of Hyperledger Fabric Blockchain



Courtesy : Sunrise Tech

Working of Hyperledger Fabric

1. **Setup**: Define network topology, CA, orderers, peers
2. **Develop**: Write chaincode, define endorsement policies
3. **Deploy:** Install & instantiate chaincode
4. **Interact**: Use SDK or CLI to invoke/query
5. **Monitor**: Use Hyperledger Explorer or logs

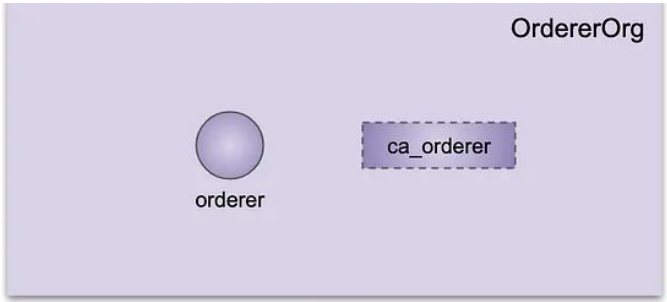
Creating Hyperledger network

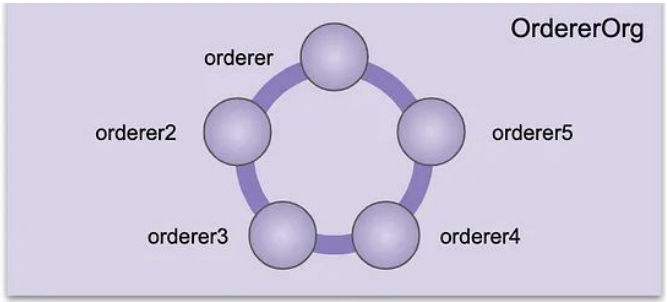
Prerequisites: Docker, Fabric binaries, config files

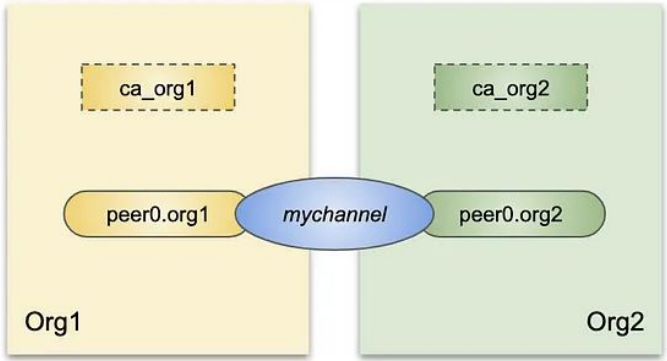
Steps:

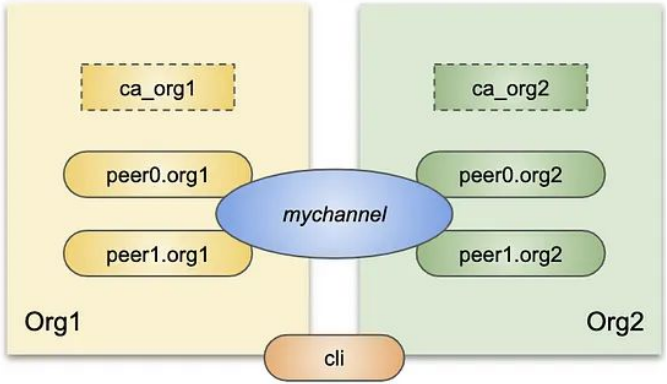
1. Generate crypto material using **cryptogen**
2. Create genesis block and channel config with **configtxgen**
3. Launch network with Docker Compose
4. Join peers to the channel
5. Install and instantiate chaincode

Creating Hyperledger network









Test Network

First Network