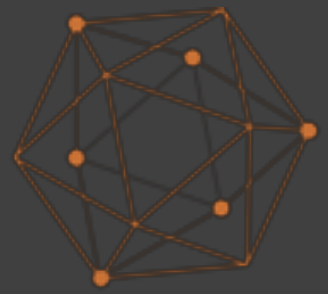


HYPERLEDGER PROJECT

Meetup #2 – Concepts & Foundation

dcentrum Community

Connect | Collaborate | Create

Agenda

The theme will be 20 mins presentations followed by 30 min discussions based on below 3 topics:

- ❑Hyperledger Fabric Concepts in detail – 20 min
  - ❑Channel
  - ❑Certificate Authority (CA)
  - ❑Membership Service Providers (MSP)
  - ❑Peer
  - ❑Orderer
  - ❑Endorser
- ❑Hyperledger Fabric Transaction Flow – 20 min
- ❑Fabcar Example – Open the hood – 20 min
  - ❑Understand what's behind it

- ❑Use case templates and action items for 3rd Meet

DCentrum's 3 Month Blockchain Series: Hyperledger

dcentrum
Connect | Collaborate | Create

Join to experience perfect way to learn and solve real world problems using Hyperledger Fabric with the group of highly motivated Blockchain enthusiasts

Nov 2018 - Feb 2019

Hyderabad

Meeting Frequency
BiWeekly

Total Series Duration
3 Months

First and Second Meeting:

- ✓ Foundation
- ✓ Level Setting
- ✓ Concepts

During initial two meetings, we all will get to know each other and understand the basics of Blockchain to build solid foundation.

Third Meeting:

- ✓ Real World Use Case Selection
- ✓ Architecture Decisions

Will pick a real world use-case by looking at the domain expertise and interest of majority of participants. Group will also work to architect the solution to during this period.

Fourth and Fifth Meeting:

- ✓ Building Solution with collaboration

During this period of 45 days we all will build end-to-end robust enterprise level solution on chosen use-case

Sixth Meeting:

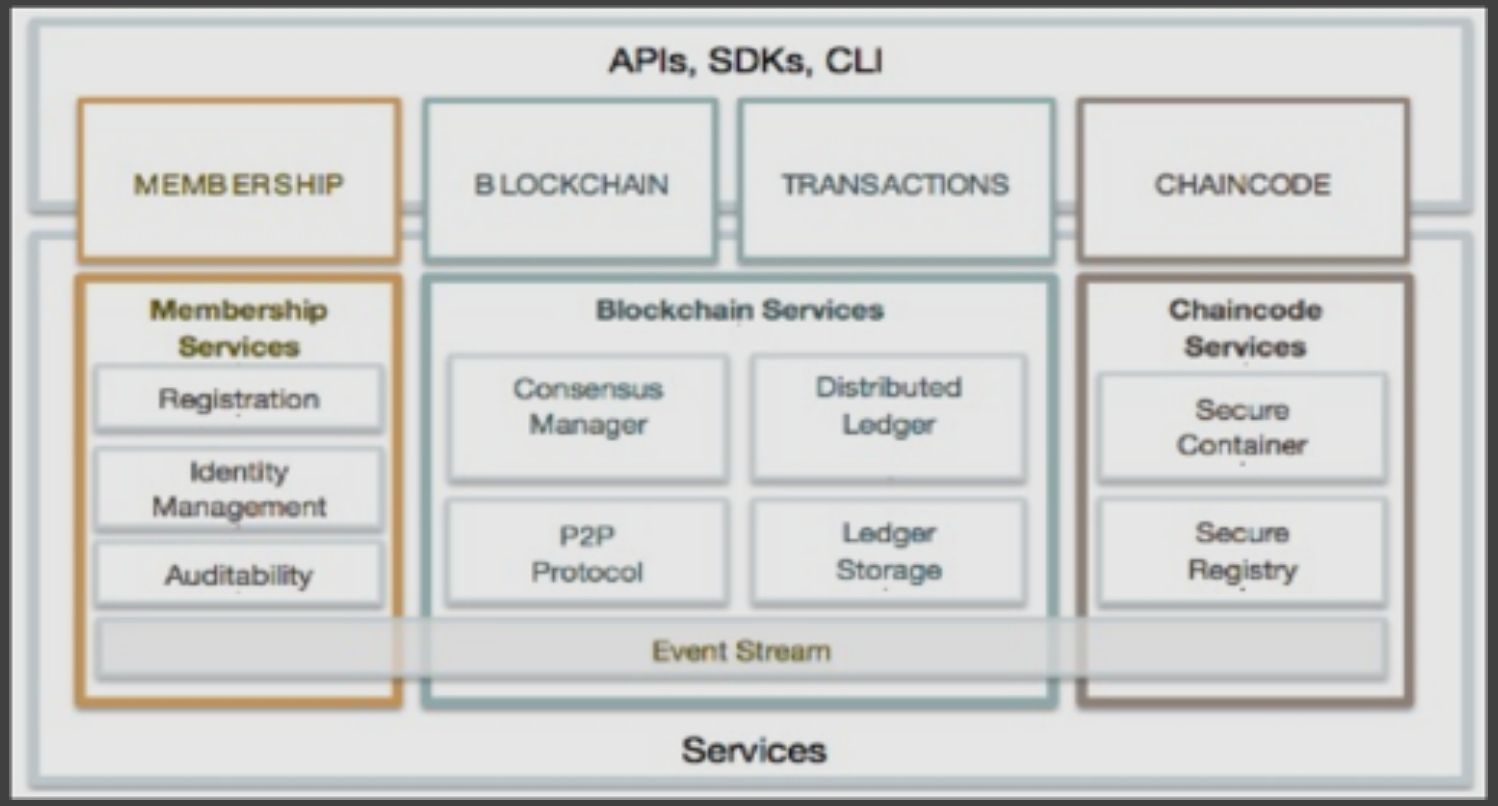
- ✓ Use Case Demo
- ✓ Hackathon

Will host and participate in an open Hackathon, which will be judged by esteemed jury of who's who of Blockchain World

Format for each meeting:

- Welcome and General Discussion: 30 mins
- Presentation from members: 60 mins
- Shared Learning and Discussion: 60 mins
- News Briefs: 10 mins
- Open discussions on general topics: 20 mins

Hyperledger Architecture



IDENTITY
Pluggable, Membership, Privacy and Auditability of transactions.

LEDGER | TRANSACTIONS
Distributed transactional ledger whose state is updated by consensus of stakeholders

SMART-CONTRACT
“Programmable Ledger”, provide ability to run business logic against the blockchain (aka smart contract)

APIs, Events, SDKs
Multi-language native SDKs allow developers to write DLT apps

Hyperledger Concepts

- Channel
- Certificate Authority (CA)
- Peer
- Orderer
- Endorser

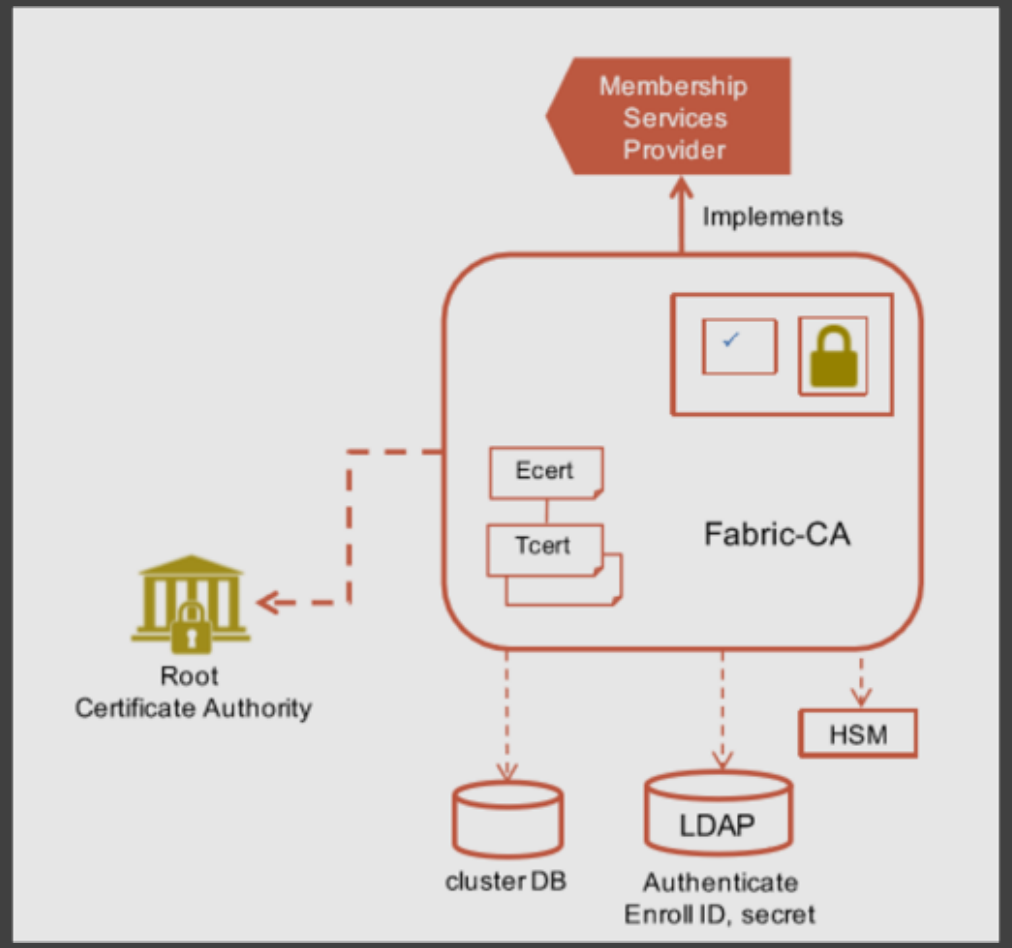
Channel

- A data partitioning mechanism to control transaction visibility only to stakeholders
- Consensus takes place within a channel by members of the channel
  - ✓Other members on the network are not allowed to access the channel and will not see transactions on the channel
- A chaincode may be deployed on multiple channels, each instance is isolated within its channel
  - ✓ A chaincode may query another chaincode in other channel (ACL applied)

Certificate Authority (CA)

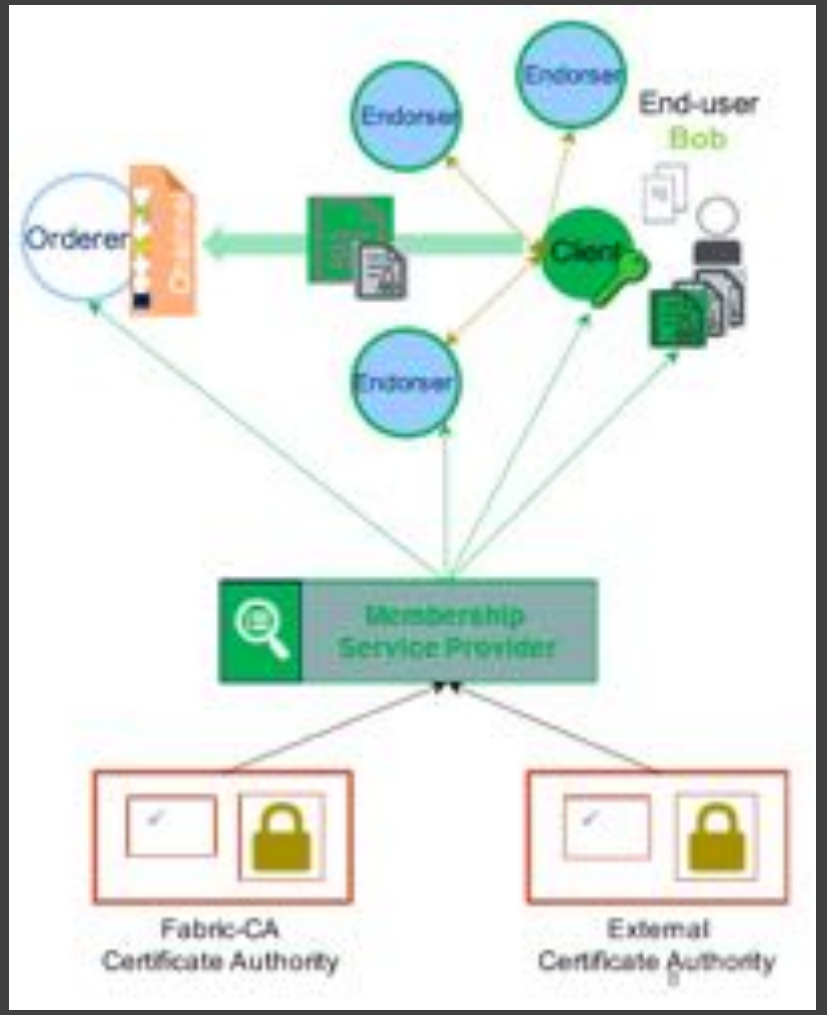
A **certificate authority** (CA) is a trusted entity that issues electronic documents that verify a digital entity's identity on the Internet. The electronic documents, which are called digital **certificates**, are an essential part of secure communication and play an important part in the public key infrastructure

- Default implementation of the Membership Services Provider Interface.
- Issues Ecerts (long-term identity) and Tcerts (disposable certificate)
- Supports clustering for HA characteristics
- Supports LDAP for user authentication
- Supports HSM



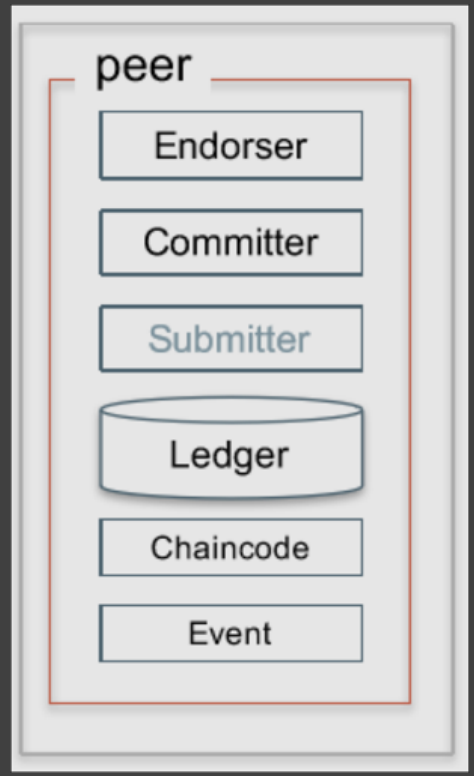
Membership Service Providers (MSP)

- An abstraction of identity provider
  - *<MSP.id, MSP.sign, MSP.verify, MSP.validateid, MSP.admin>*
  - govern application, endorser and orderer identities
- Used as building blocks for access control frameworks
  - at the system level (read/write access on system controls, and channel creation)
  - at the channel level (read/write access),
  - at the chaincode level (invocation access)
- Represent a consortium or a member



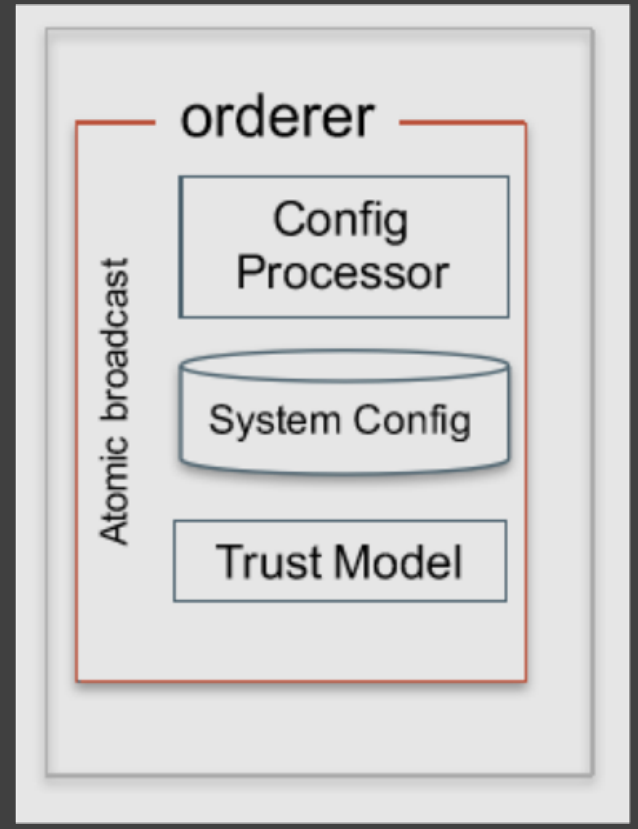
Peer

- A Peer is a node on the network maintaining state of the ledger and managing chaincodes
- Any number of Peers may participate in a network
- A Peer can be an endorser, committer and/or submitter (submitter has not been implemented). An endorser is always a committer
  - An endorser executes and endorses transactions
  - A committer verifies endorsements and validates transaction results
- A Peer manages event hub and deliver events to the subscribers
- Peers form a peer-to-peer gossip network



Orderer

- A group of Orderers runs a communication service, called ordering service, to provide atomic broadcast
- Provides ordering of operations, before the data is committed in the ledger it has to pass through the orderer.
- Orderer will creates the blocks, that will be part of the Blockchain.
- Once the block is full, orderer will send the blocks to the peers, peers will commit to the block to the ledger.
- Ordering service is responsible for:
  - ✓ Verification.
  - ✓ Security.
  - ✓ Policy management.



Transaction Endorsement

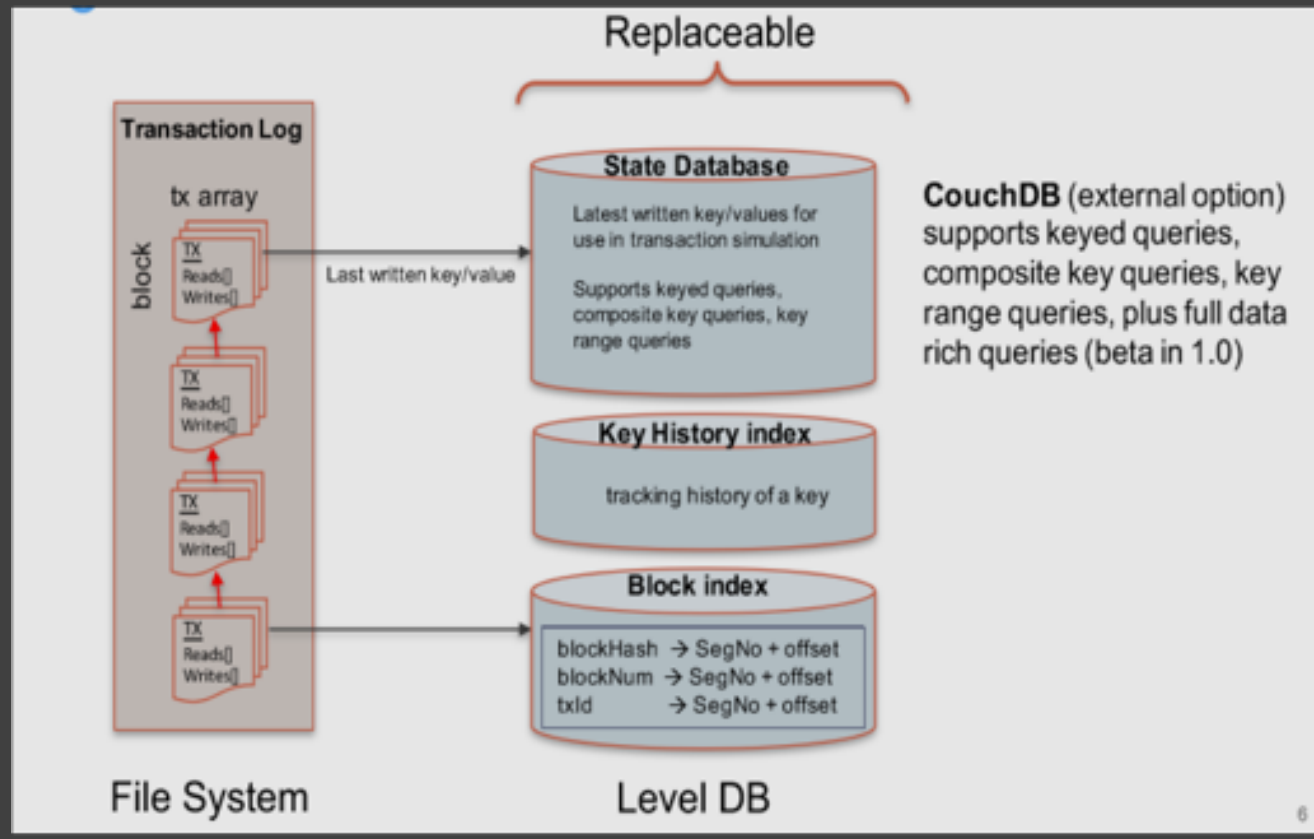
- An endorsement is a signed response of the result of a transaction execution
- An endorsement policy encapsulates the requirement for a transaction to be accepted by the stakeholders, either explicit or implicit
  - A signature from both member1 and member2
  - Either a signature from both member1 and member2 or a signature from member3
  - A signature from John Doe
- The endorsement policy is specified during a chaincode instantiation on a channel
- Each channel-chaincode may have different endorsement policy

Chaincode

- A chaincode is programmatic code deployed on the network, where it is executed and validated by chain validators together during the consensus process.
- Developers can use chaincode's to develop business contracts, asset definitions, and collectively-managed decentralized applications
- There are generally two ways to develop business contracts: the first way is to code individual contracts into standalone instances of chaincode; the second way, and probably the more efficient way, is to use chaincode to create decentralized applications that manage the life cycle of one or multiple types of business contracts, and let end users instantiate instances of contracts within these applications.
- Chaincode can be written in any programming language and executed in containers. The first fully supported chaincode language is Golang.

Ledger

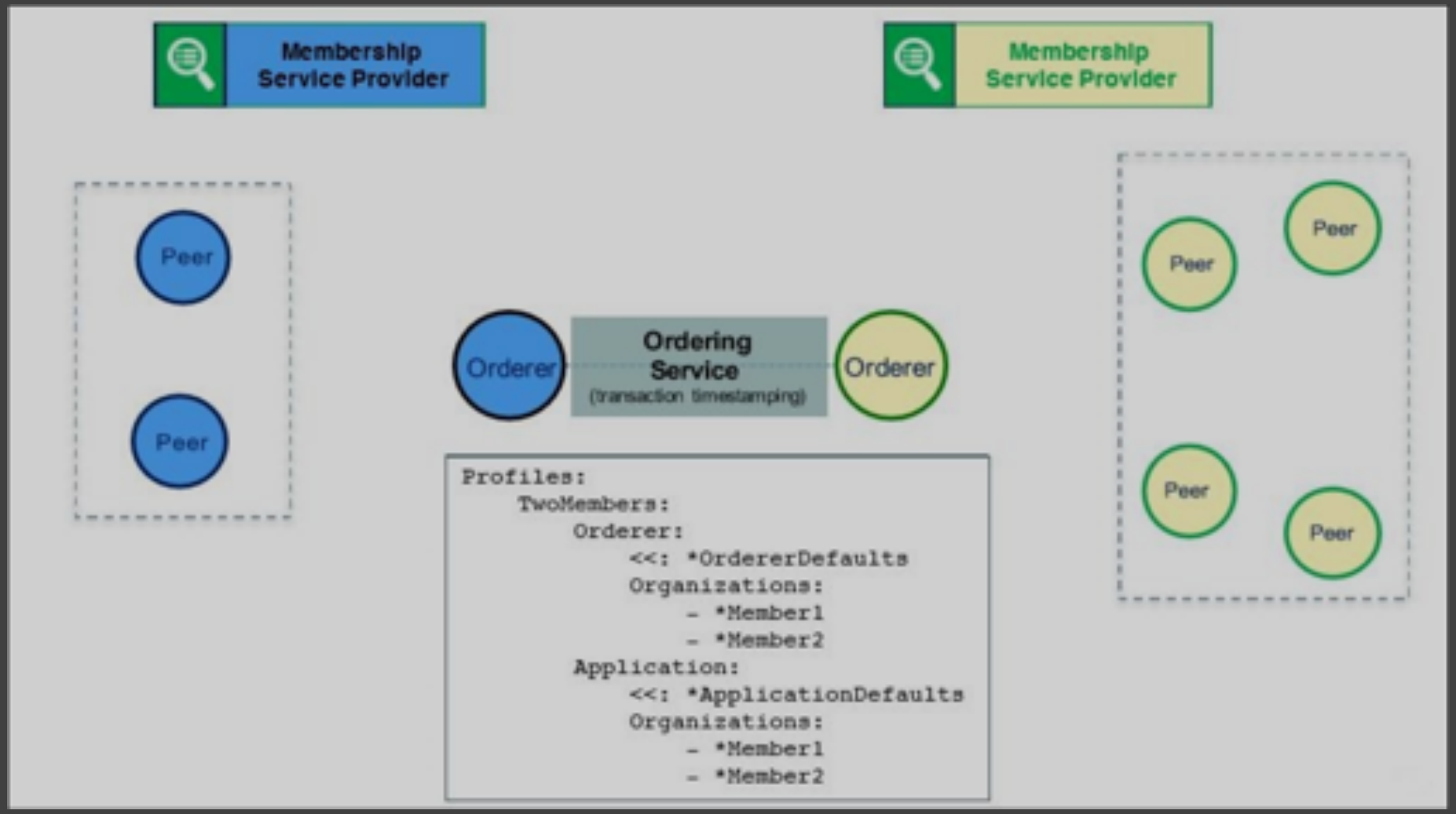
- In Hyperledger fabric State database options include LevelDB and Couch DB.
- LevelDB is the default state database embedded in the peer process and stores chaincode data as key-value pairs.
- CouchDB can store any binary data that is modeled in chaincode .But as a JSON document store, CouchDB additionally enables rich query against the chaincode data, when chaincode values (e.g. assets) are modeled as JSON data.



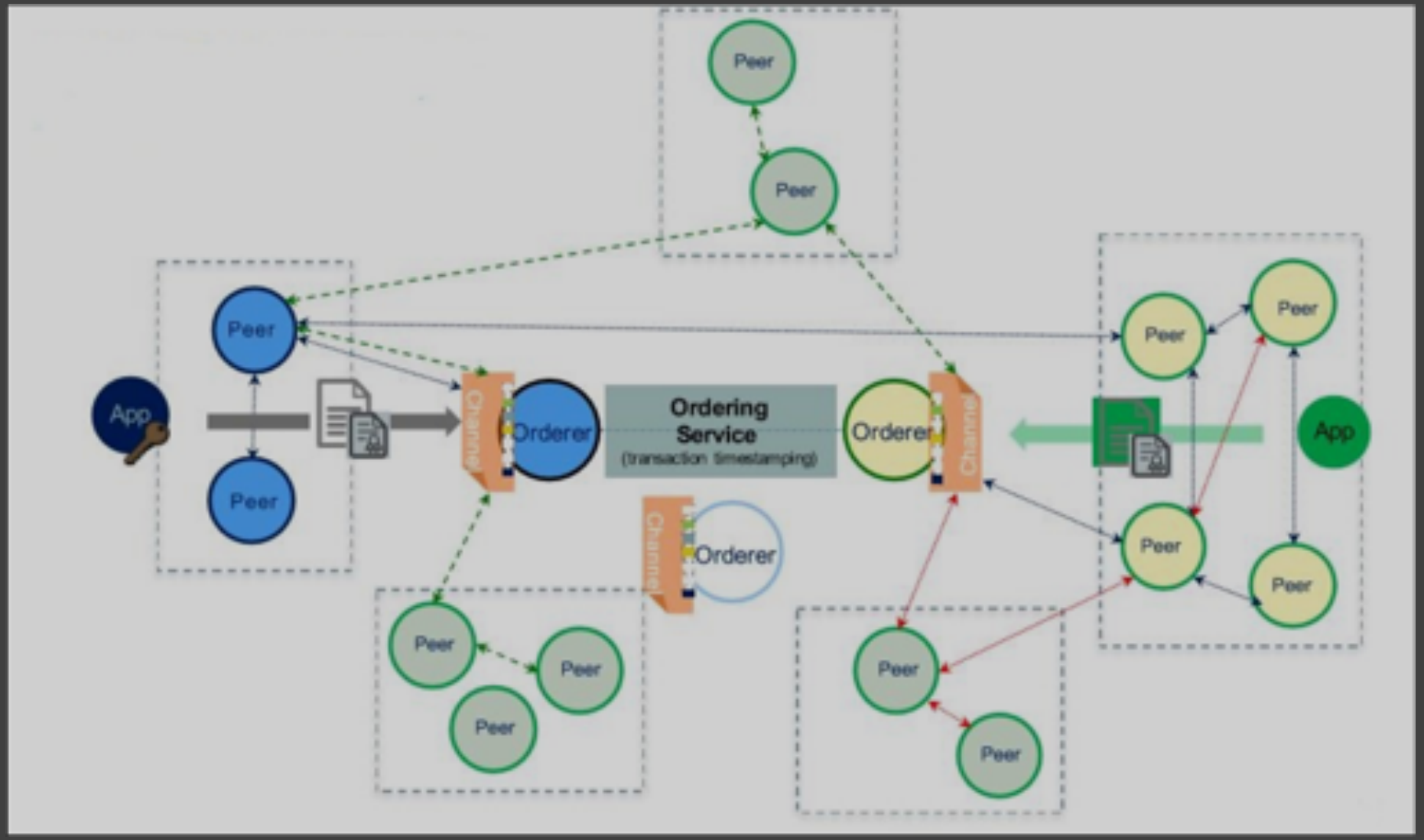
Consensus Redefined

- Consensus = Transaction Endorsement + Ordering + Validation
- Endorsement: Each stakeholder decides whether to accept or reject a transaction
- Ordering: Sort all transactions within a period into a block to be committed in that order
- Validation: Verify transaction endorsement satisfied the policy and transaction transformation is valid according to multi-version concurrency control (MVCC)

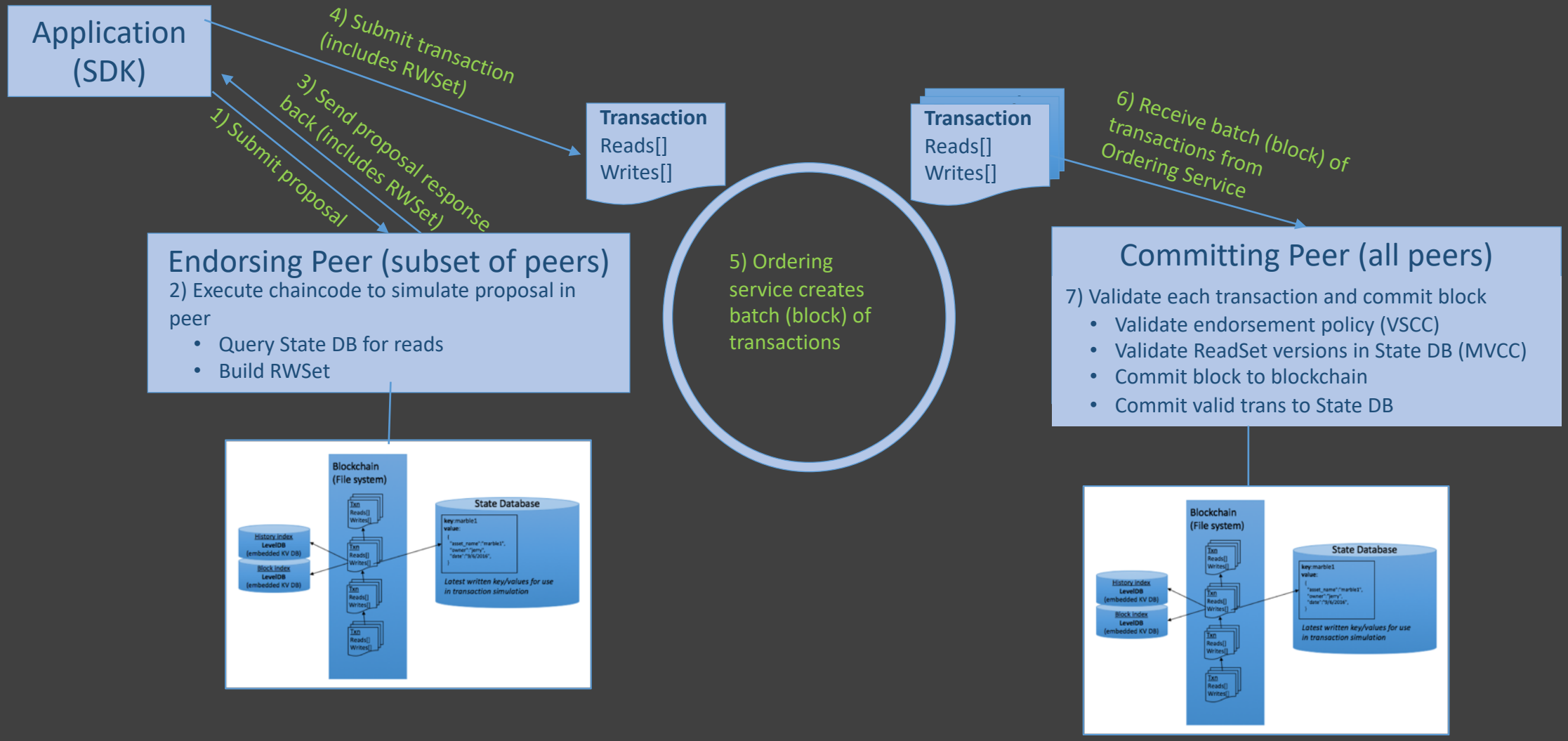
Two-Member Network



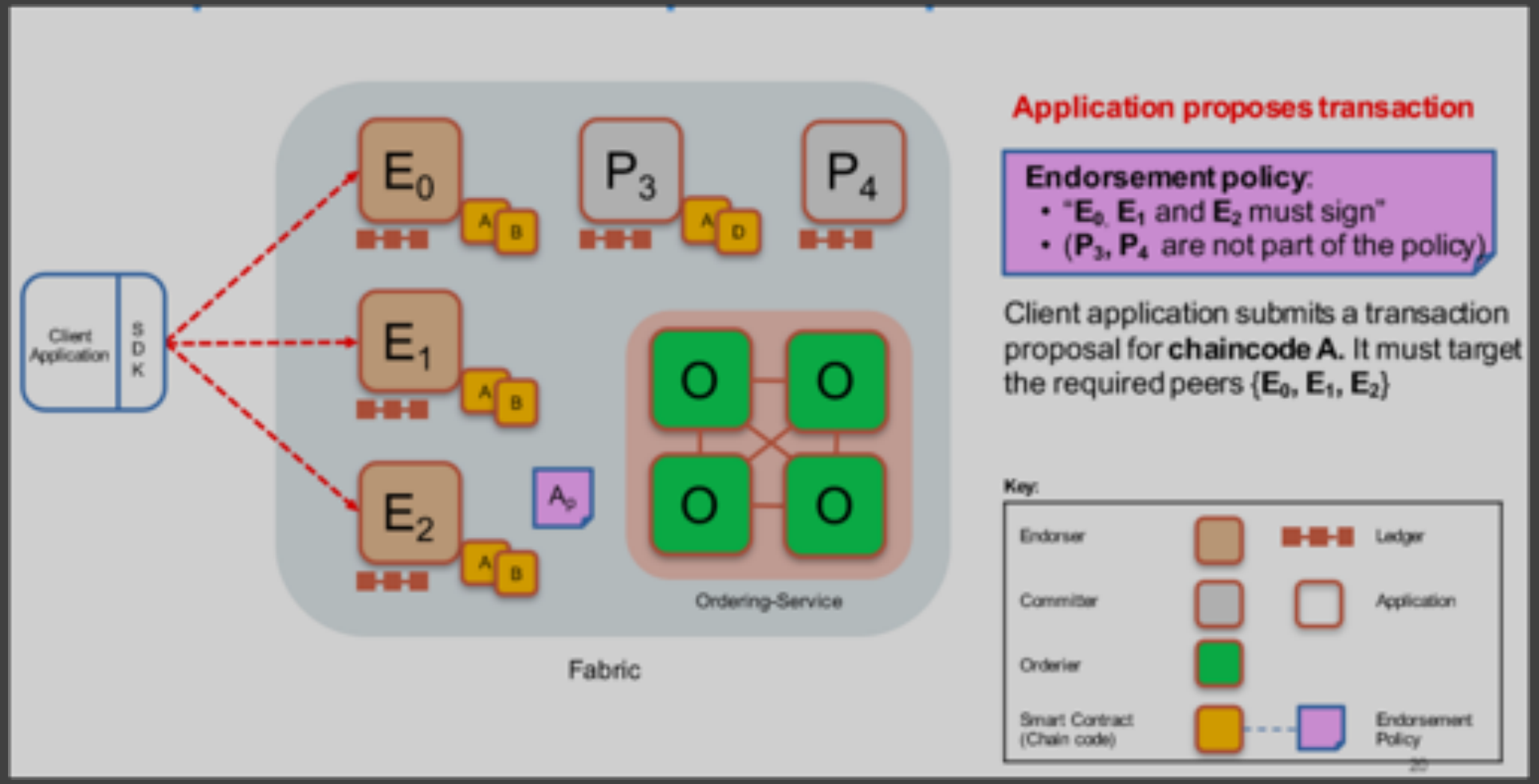
N-Member Network with Multichannel



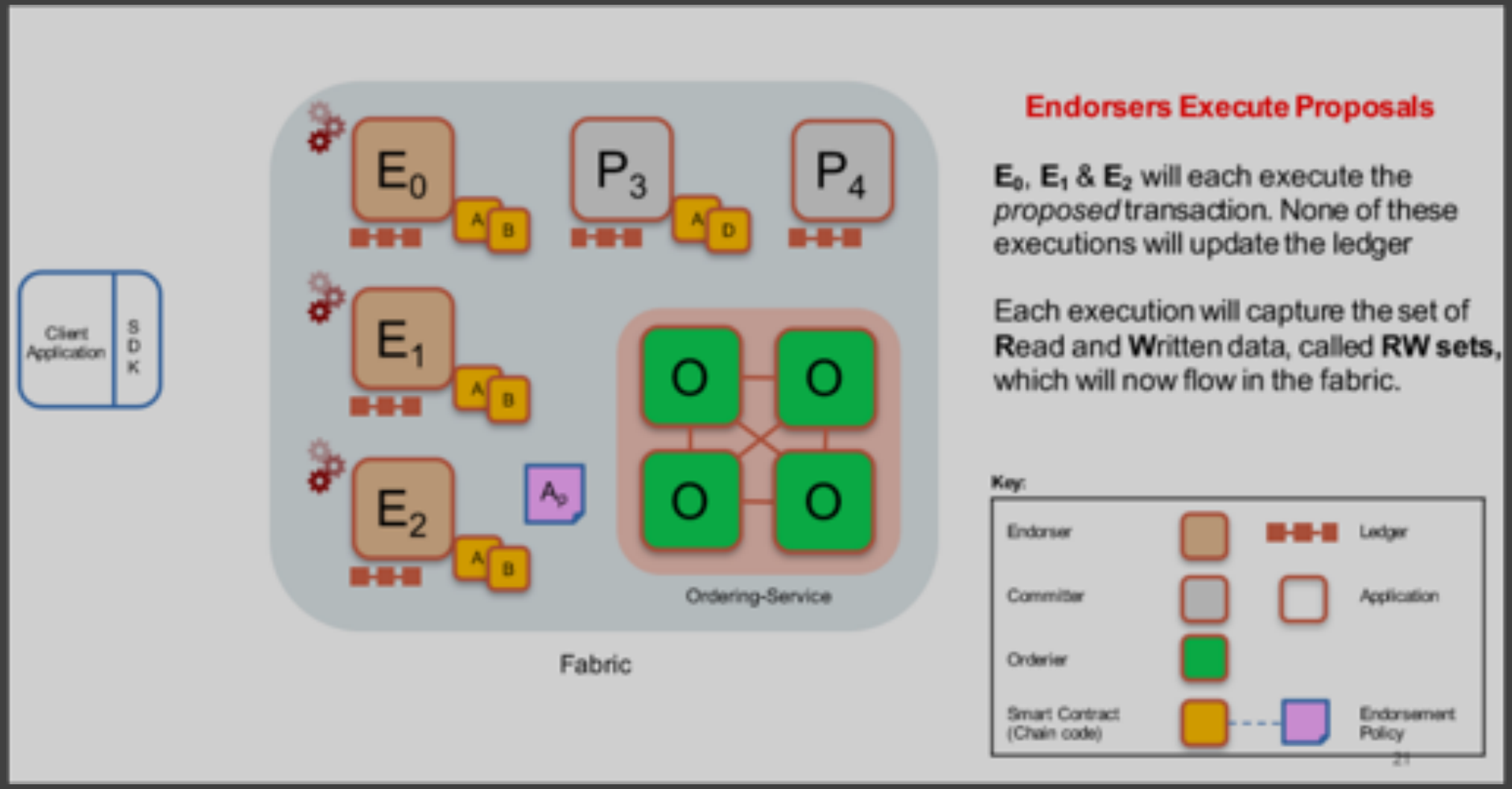
Hyperledger Fabric Transaction Lifecycle



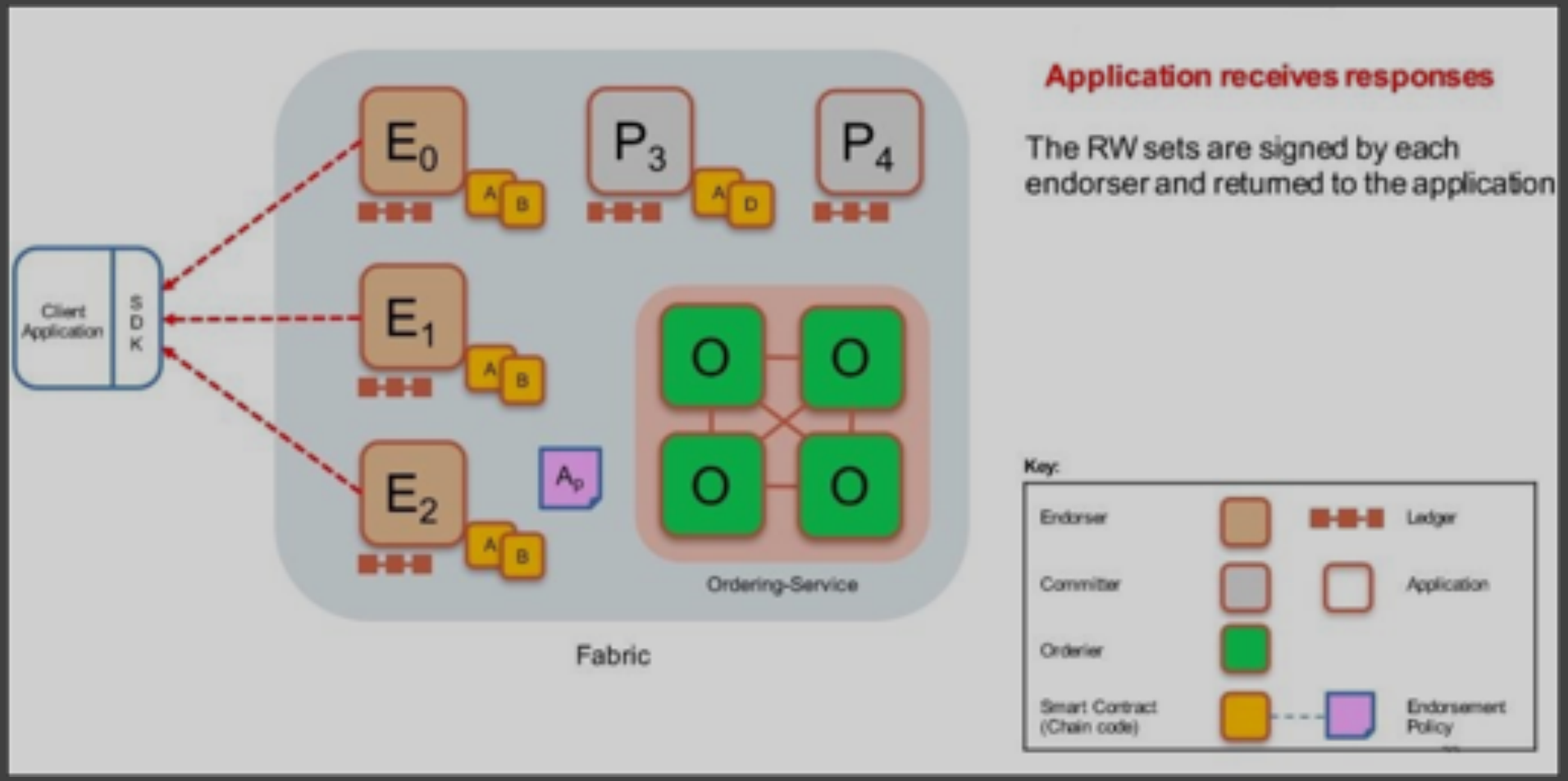
Sample transaction: Step 1/7 –Propose transaction



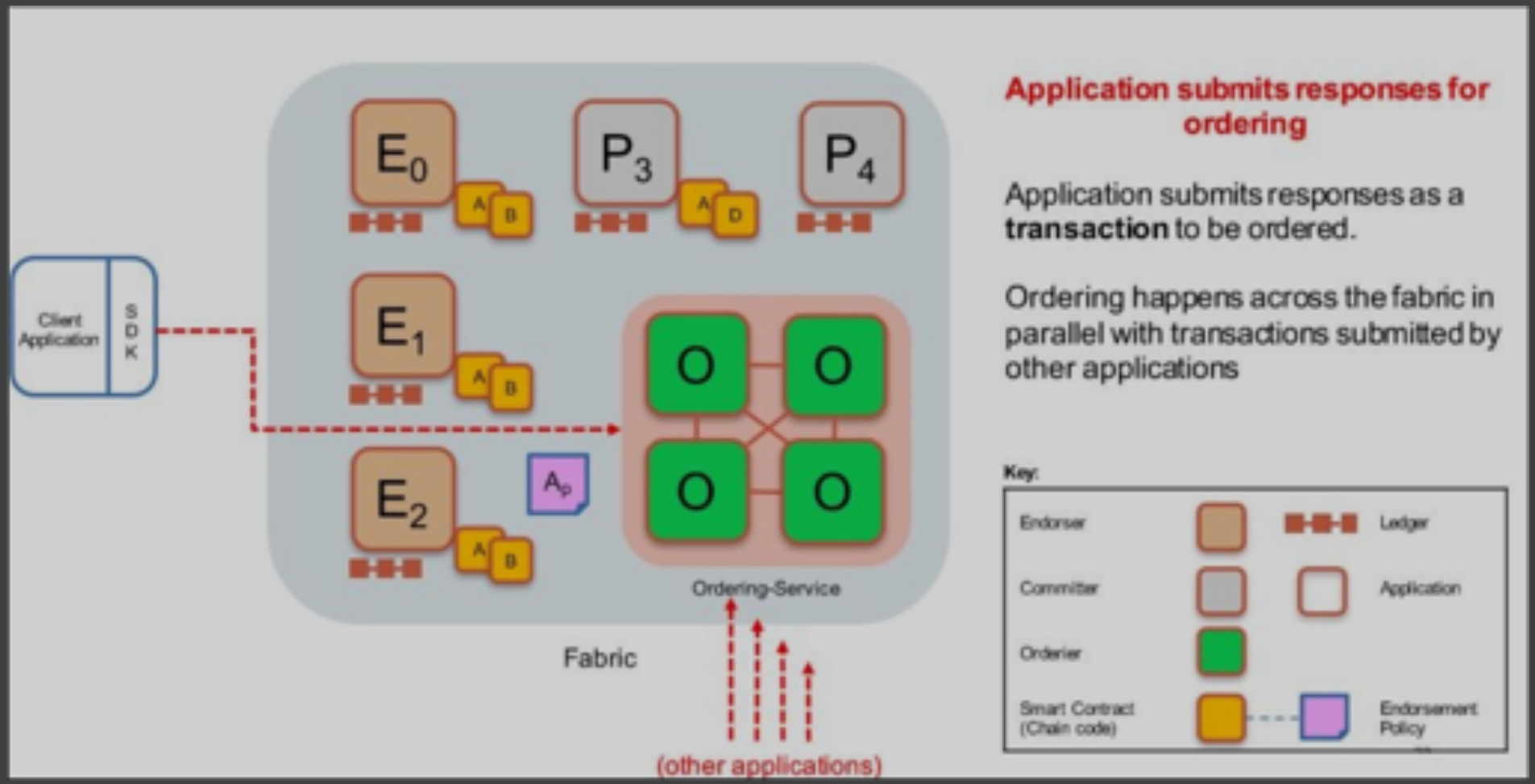
Sample transaction: Step 2/7 – Execute proposal



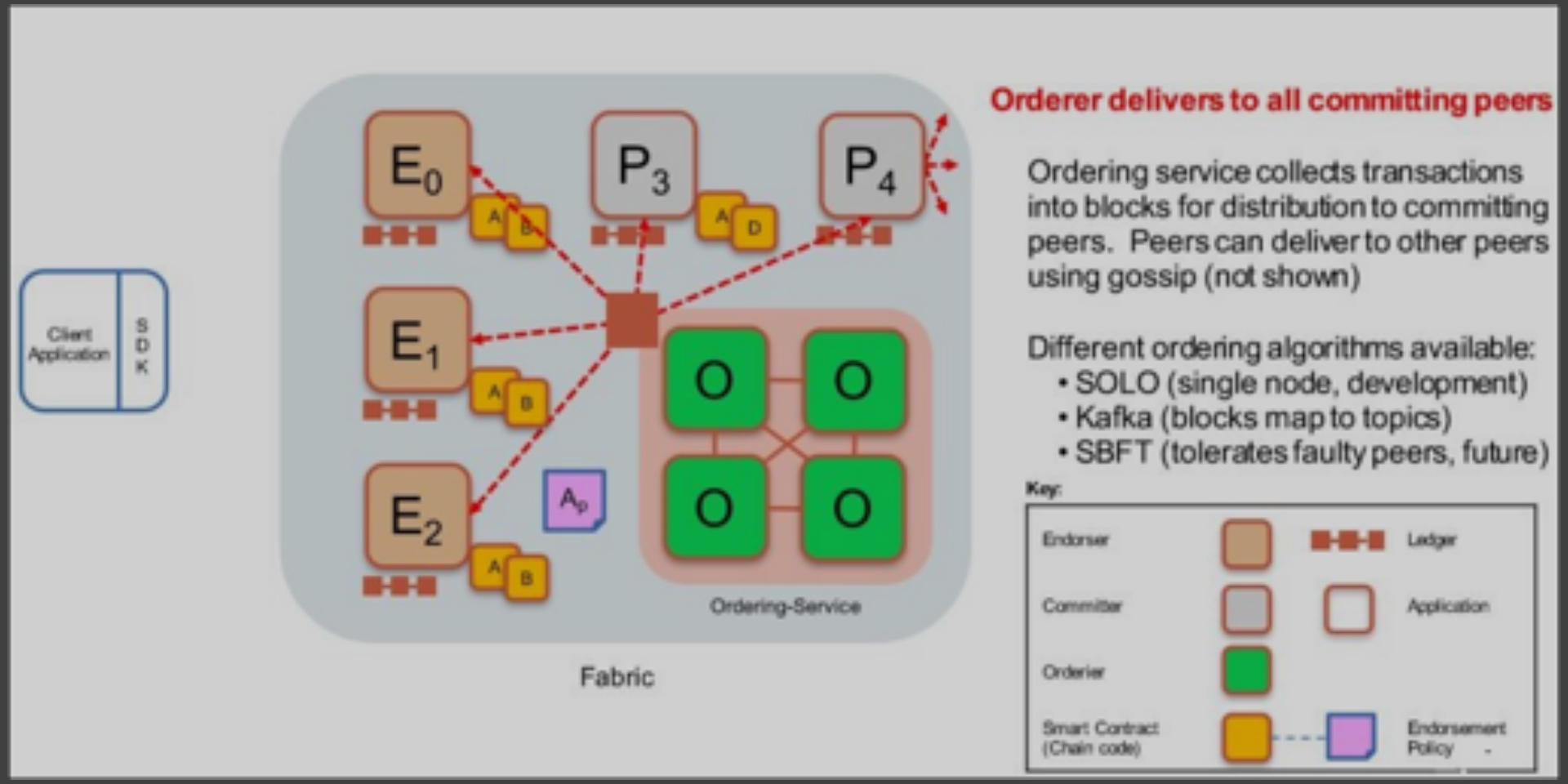
Sample transaction: Step 3/7 – Proposal Response



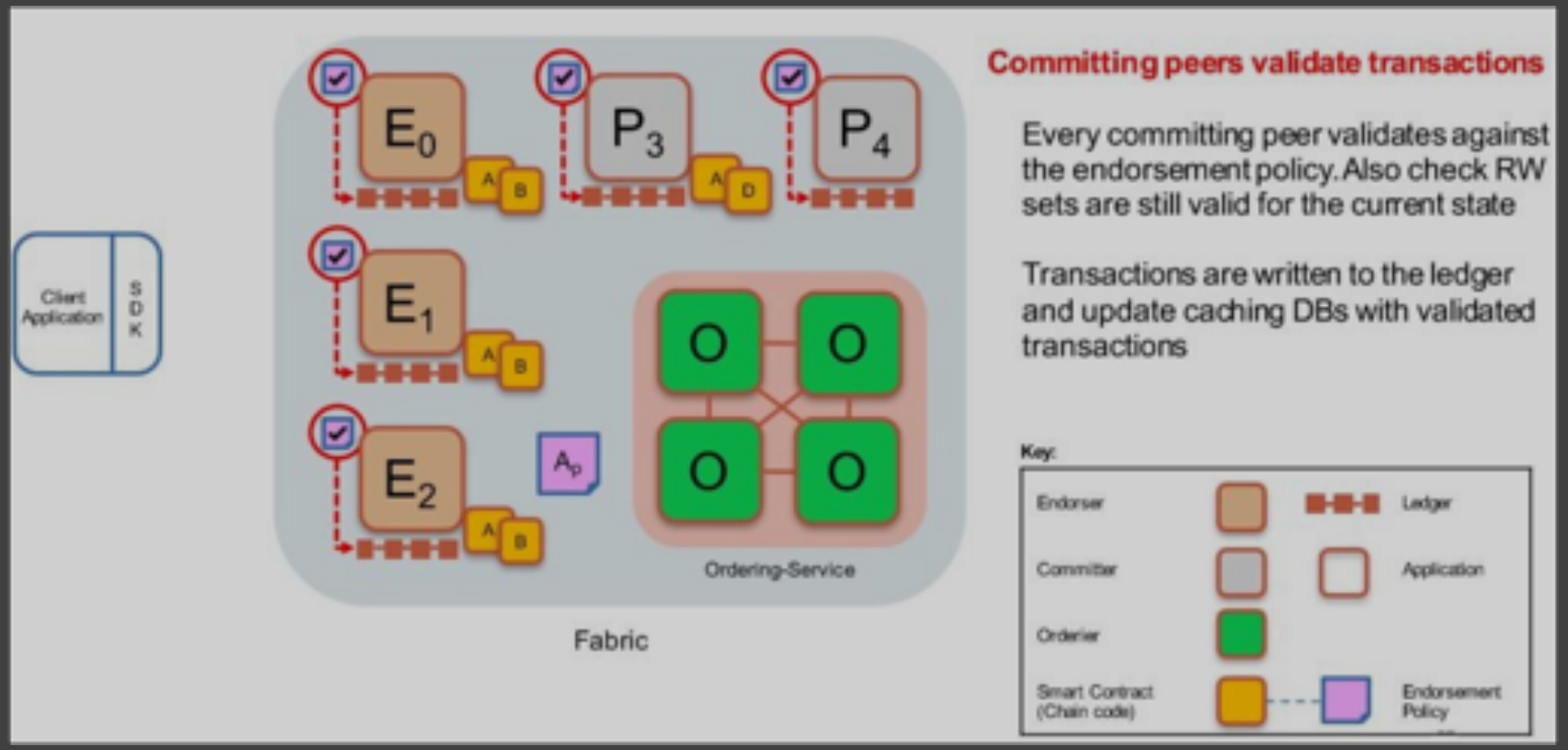
Sample transaction: Step 4/7 –Order Transaction



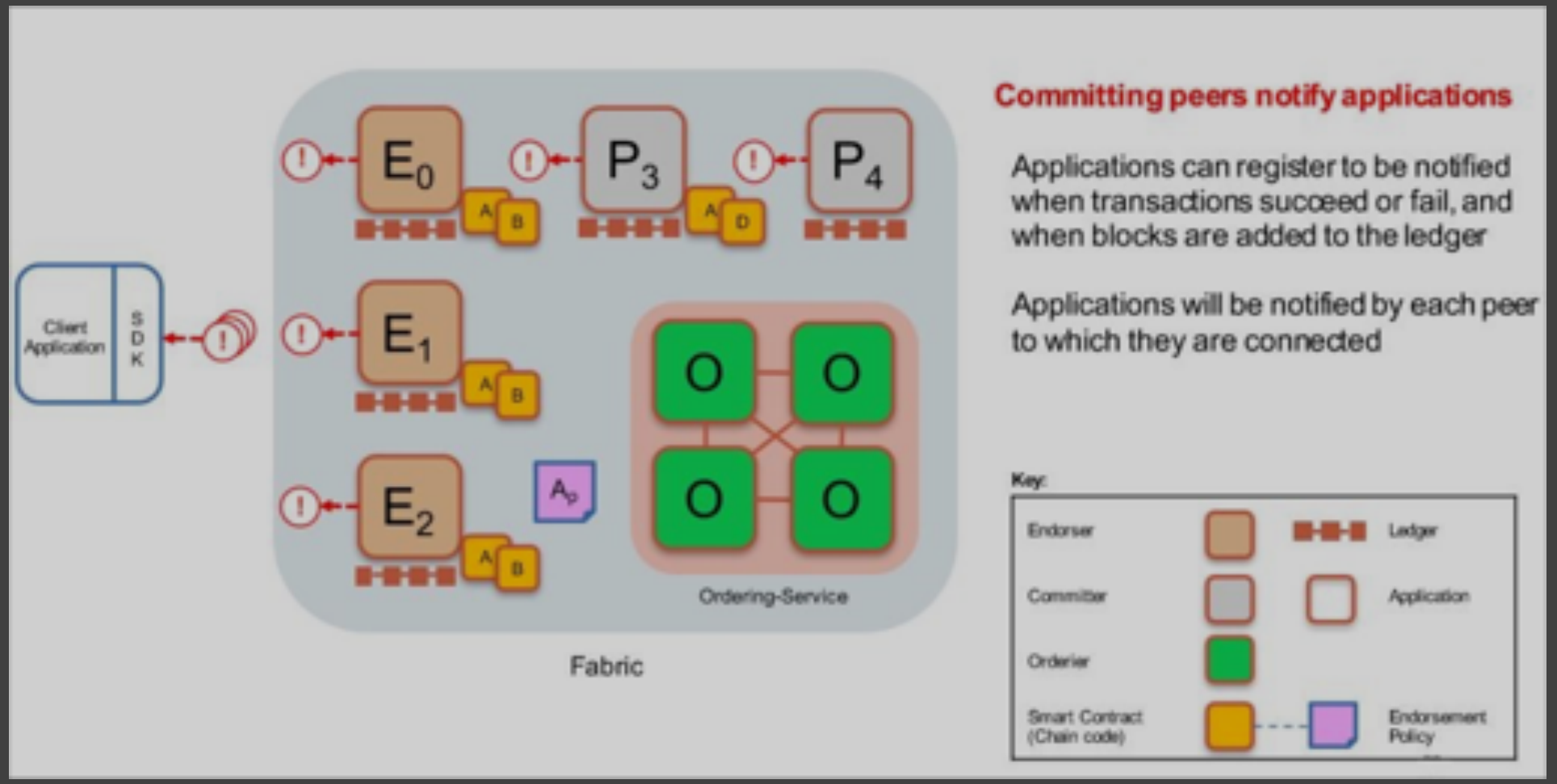
Sample transaction: Step 5/7 – Deliver Transaction



Sample transaction: Step 6/7 – Validate Transaction



Sample transaction: Step 7/7 –Notify Transaction




WHERE?
when?
which?
WHO?
why?
WHEN?
How?
WHERE?
why?
WHAT?

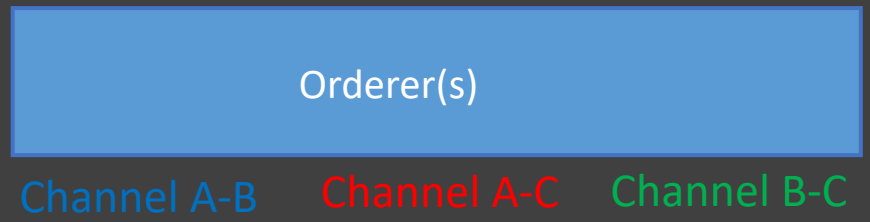
Scenario: Channels for bilateral trades

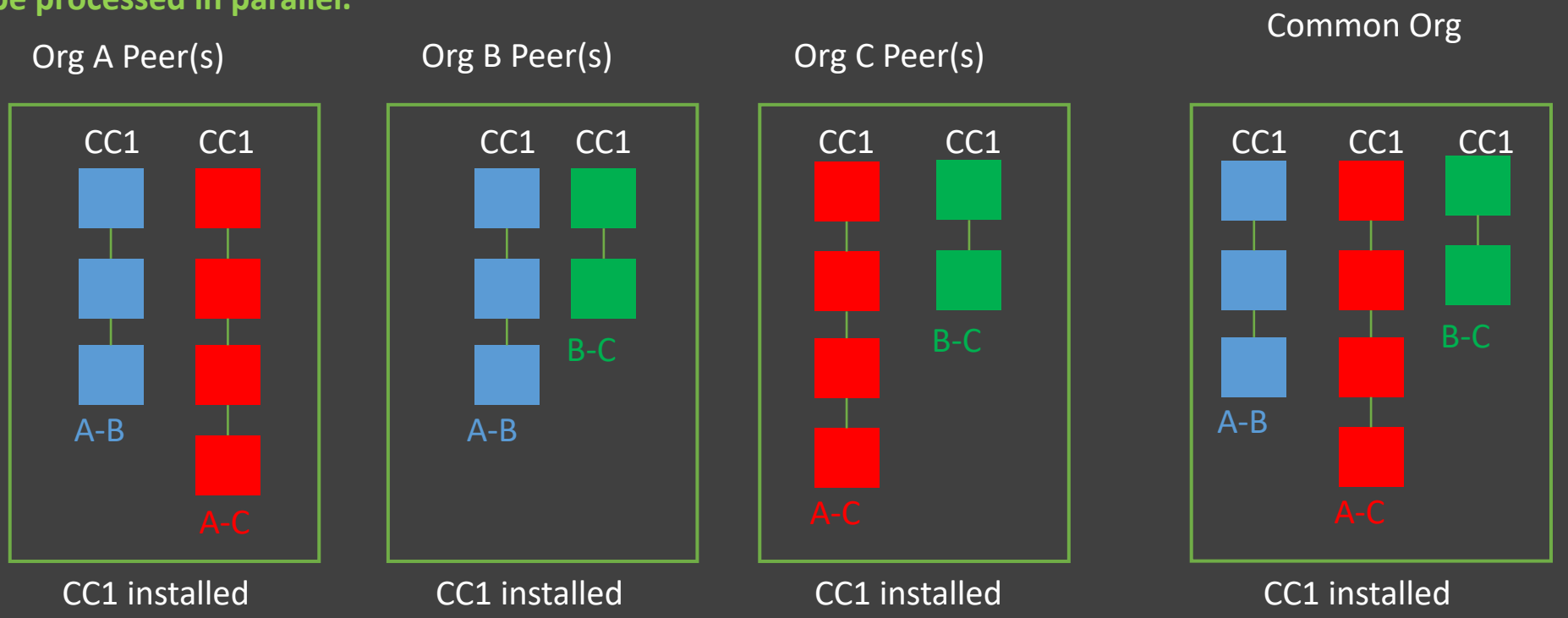
Chaincode1 installed on all 4 peers.
Chaincode1 instantiated on all 3 channels*
**Different chaincodes could be instantiated on different channels.*
**Multiple chaincodes can be instantiated on each channel.*

One distributed ledger per channel.

Org A cannot see transactions between B and C.
Blocks from different channels can be processed in parallel.

➔ ***Privacy + increased throughput***





Bootstrapping a Network

- Decide on members (MSPs) controlling the ordering service
  - Set up MSP configuration for each member (root certs, signing certs, key, admins)
  - Set up policies governing the network (who has privilege to modify config and create channels)
  - Start up orderers with the configuration
- Each member decides on the number of peers to participate – For each peer, issue peer identity (local MSP configuration) and start it up
- At this point, we have a network of peers and orderers – Peers are not yet connected to orderers nor to each other

Setting up Channels, Policies, and Chaincodes

- Depending on the business network, 1 or more channels may be required
- To create a channel, send a configuration transaction to the ordering service specifying members of the channel, ACL policies, anchor peers – The configuration becomes part of the genesis block of the channel – Then notify members to join the channel (a peer may join multiple channels)
- Deploy chaincodes on the channel with appropriate endorsement policy
- Now the network is ready for transacting