



Hyperledger Fabric Architecture

Hyperledger Fabric Implementation in
Birth/Death Certificate

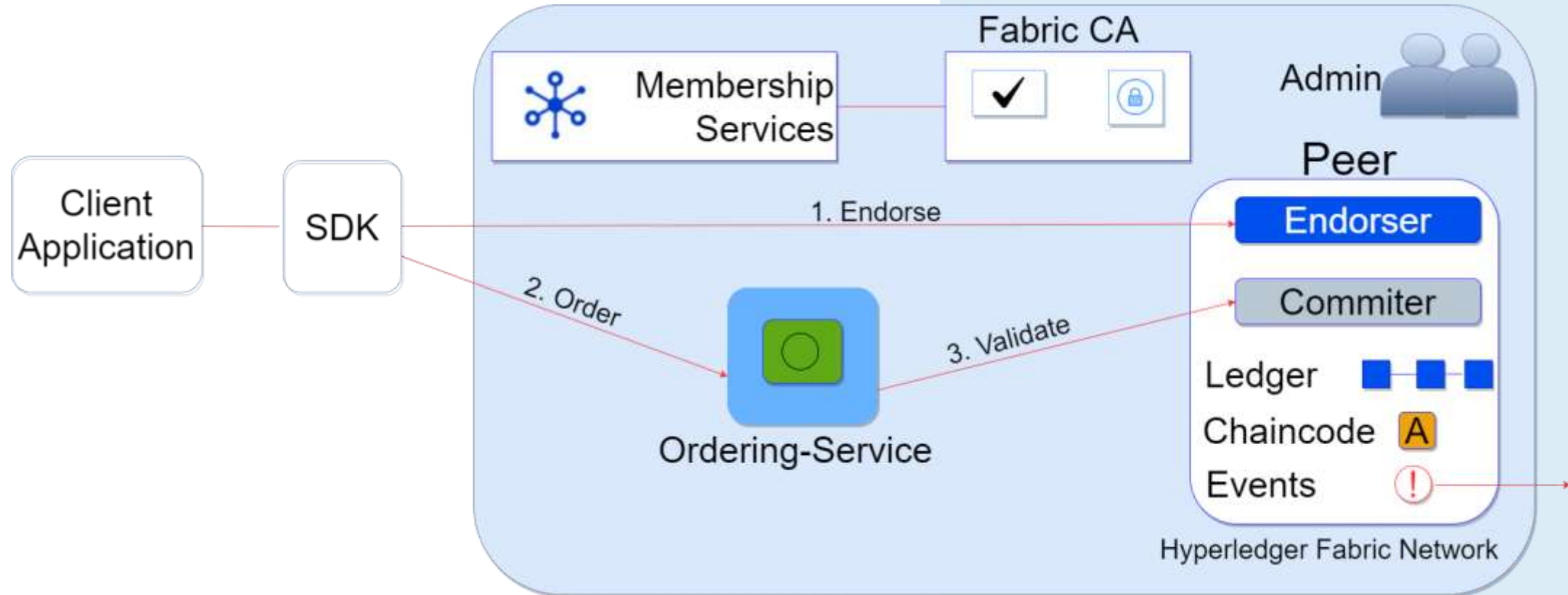


Implemetation

By implementing Hyperledger Fabric in a birth/death certificate system, it is possible to create a secure and tamper-resistant ledger of birth and death records that can be accessed and queried by authorized parties. The use of Blockchain technology can help increase transparency, reduce fraud, and improve the overall efficiency of the system.

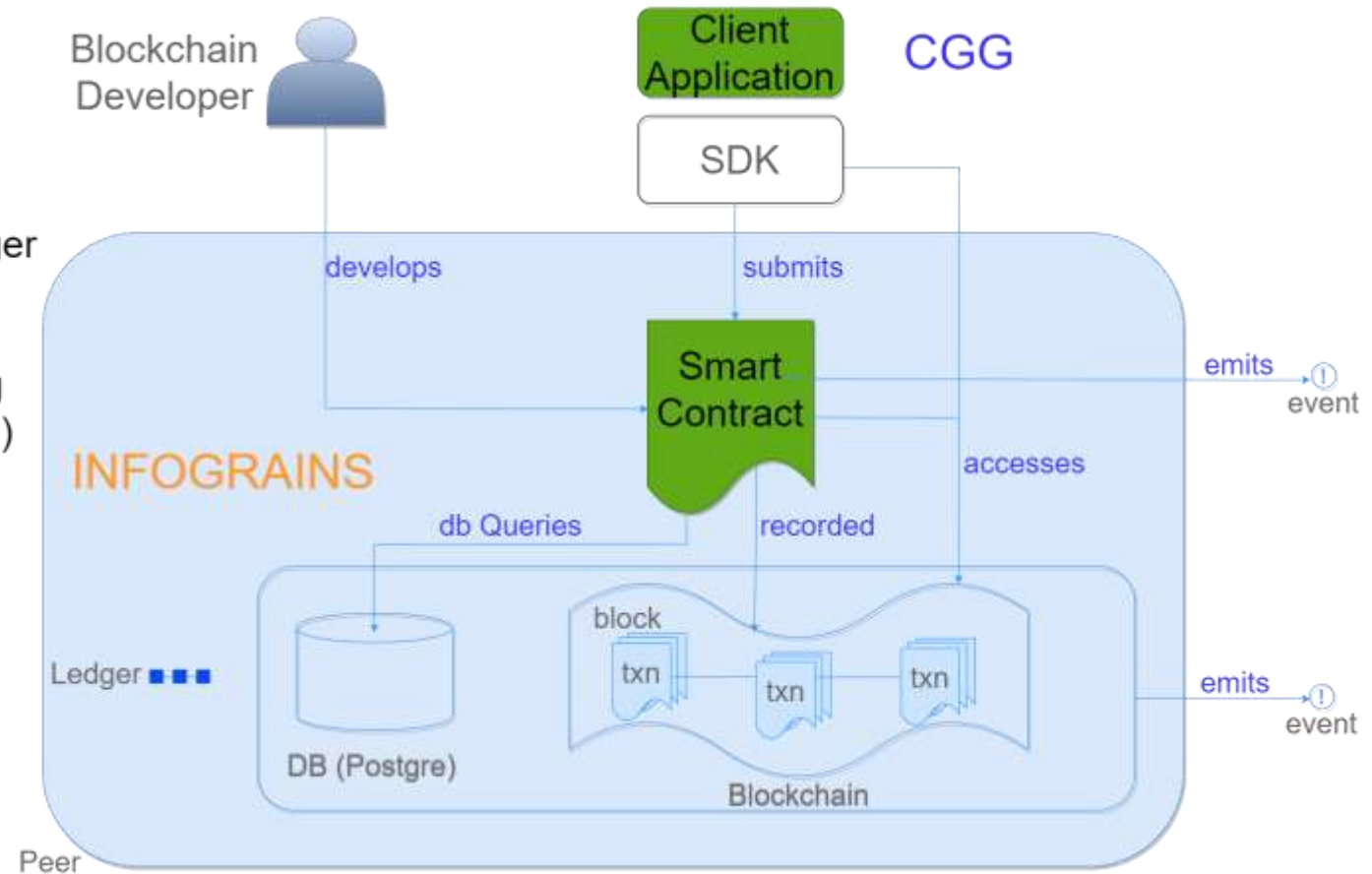


Hyperledger Fabric Architecture

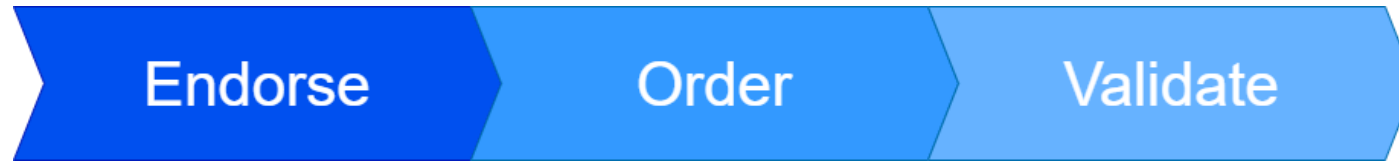


How Application interact with the Ledger

1. Client Application in using Hyperledger Fabric Client (HFC) SDK
2. Smart Contract implemented using chaincode- managing the DB(Postgre)



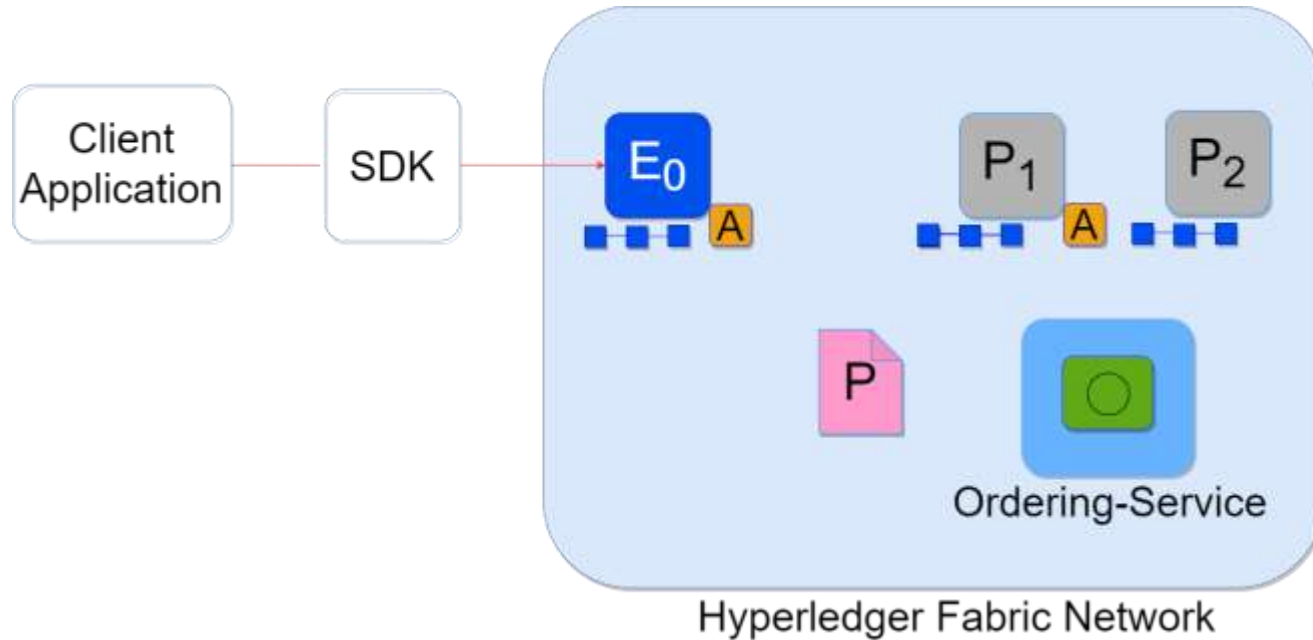
Transaction Flow



When a client application submits a transaction proposal to the network, it is first sent to the endorsing peers for validation. The endorsers execute the chaincode and check whether the proposed transaction is valid according to the rules defined in the chaincode. If the transaction is valid, the endorsers digitally sign the proposal and return it to the client application.

- **Endorser:** Endorsers are responsible for executing the transaction logic and verifying its validity, ensuring that the transaction conforms to the rules defined in the chaincode and that it does not violate any of the endorsement policies.
- **Orderer:** The Orderer is a separate component from the endorsing peers and is responsible for ordering transactions into blocks and distributing them to the peers for validation and commitment to the ledger.
- **Validation:** By validating transactions and blocks before they are added to the ledger, the network ensures that only valid transactions are added and that the ledger is consistent across all nodes in the network.

Propose Transaction










Application proposes transaction

Endorsement Policy:

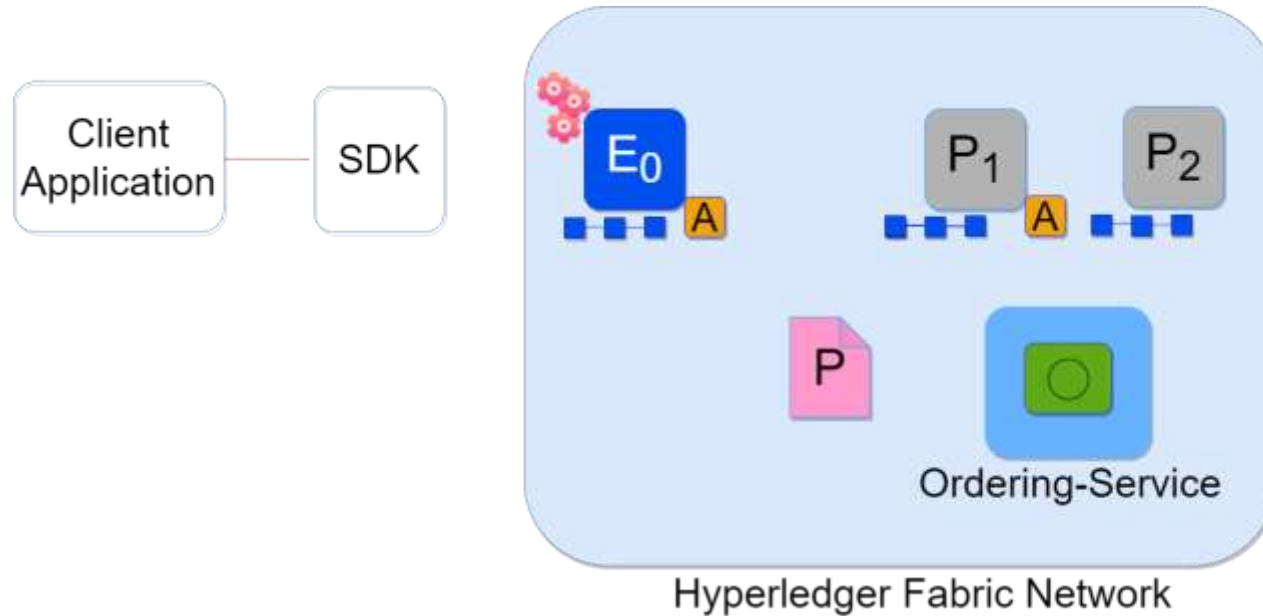
- E₀ must sign
- P₁, P₂ are not part of the policy

Client Application submits a transaction proposal for Smart Contract A. It must target the required peer {E₀}

Key:

Endorser		 Ledger
Committing Peer		 Application
Ordering Node		
Smart Contract		 Endorsement Policy

Execute Proposal










Endorsers Execute Proposals

E₀ will execute the proposed transaction. This execution will not update the ledger.

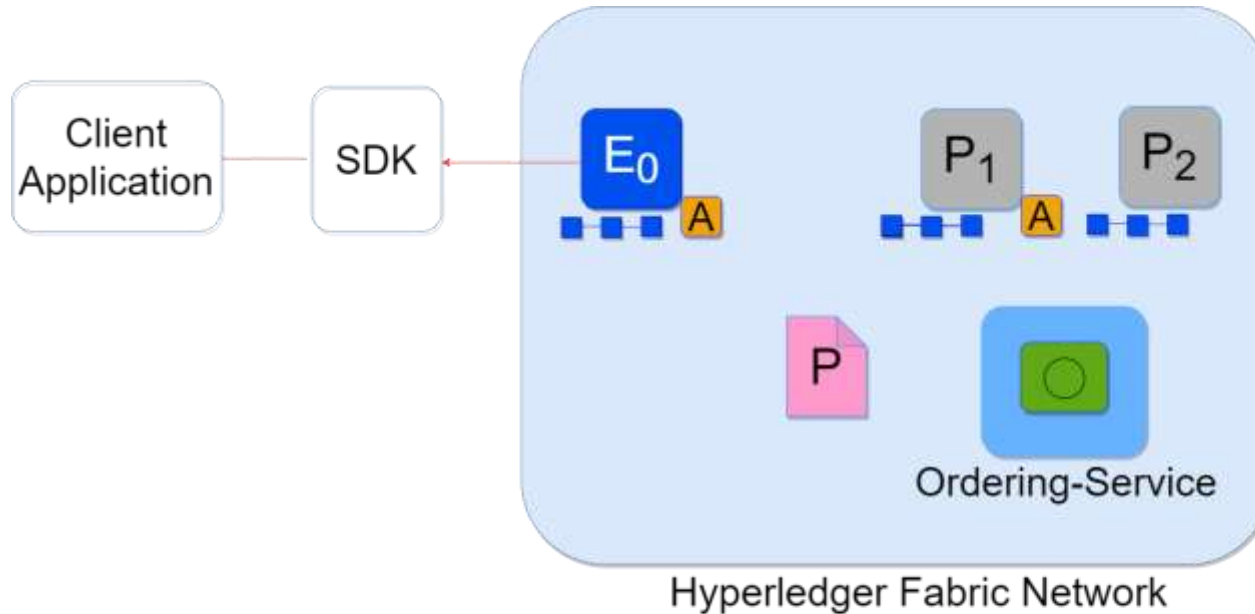
Execution will capture the set of Read and Written data called RW sets which will now flow in the fabric.

Transactions can be signed and encrypted.

Key:

Endorser		 Ledger
Committing Peer		 Application
Ordering Node		
Smart Contract		 Endorsement Policy

Proposal Response



Application Receives Responses

RW sets are asynchronously returned to application

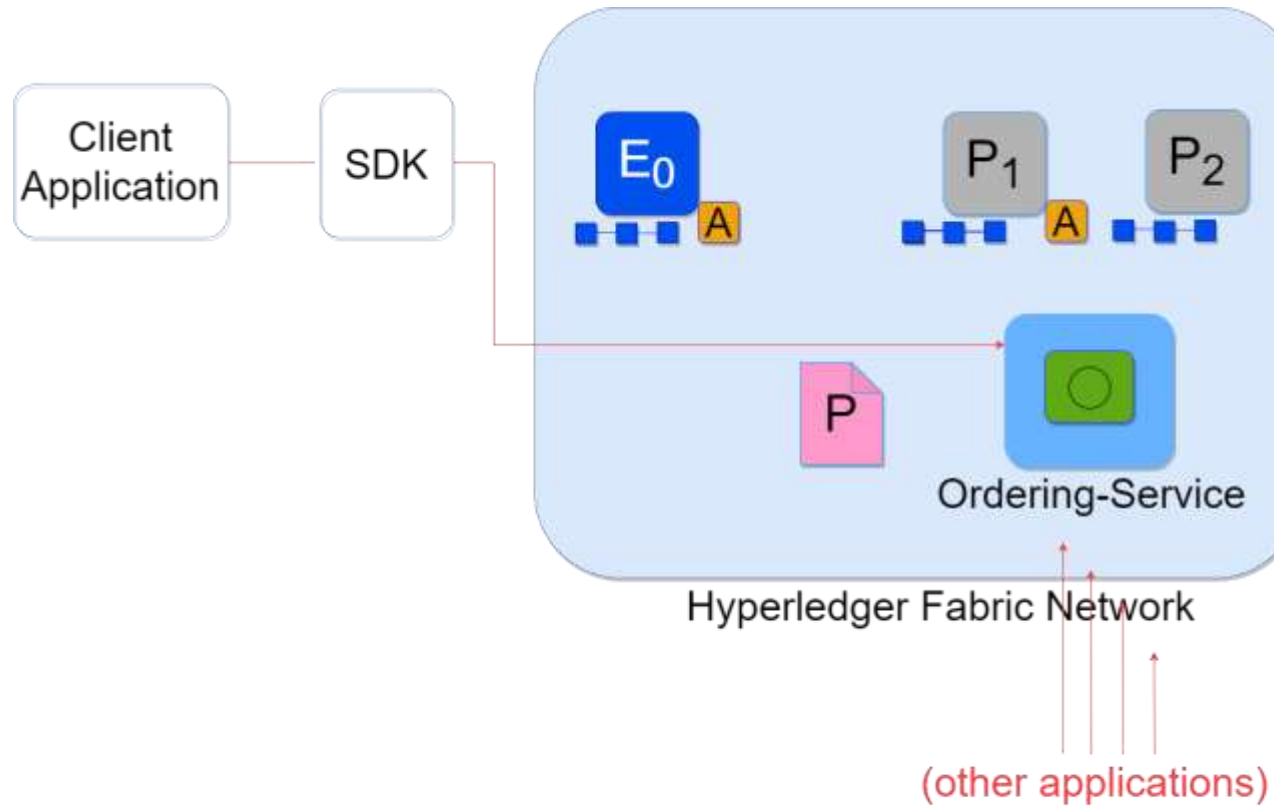
The RW sets are signed by endorser and also includes record version number

(This information will be checked much later in the consensus process.)

Key:

Endorser		Ledger
Committing Peer		Application
Ordering Node		
Smart Contract		Endorsement Policy

Order Transaction










Responses Submitted for Ordering

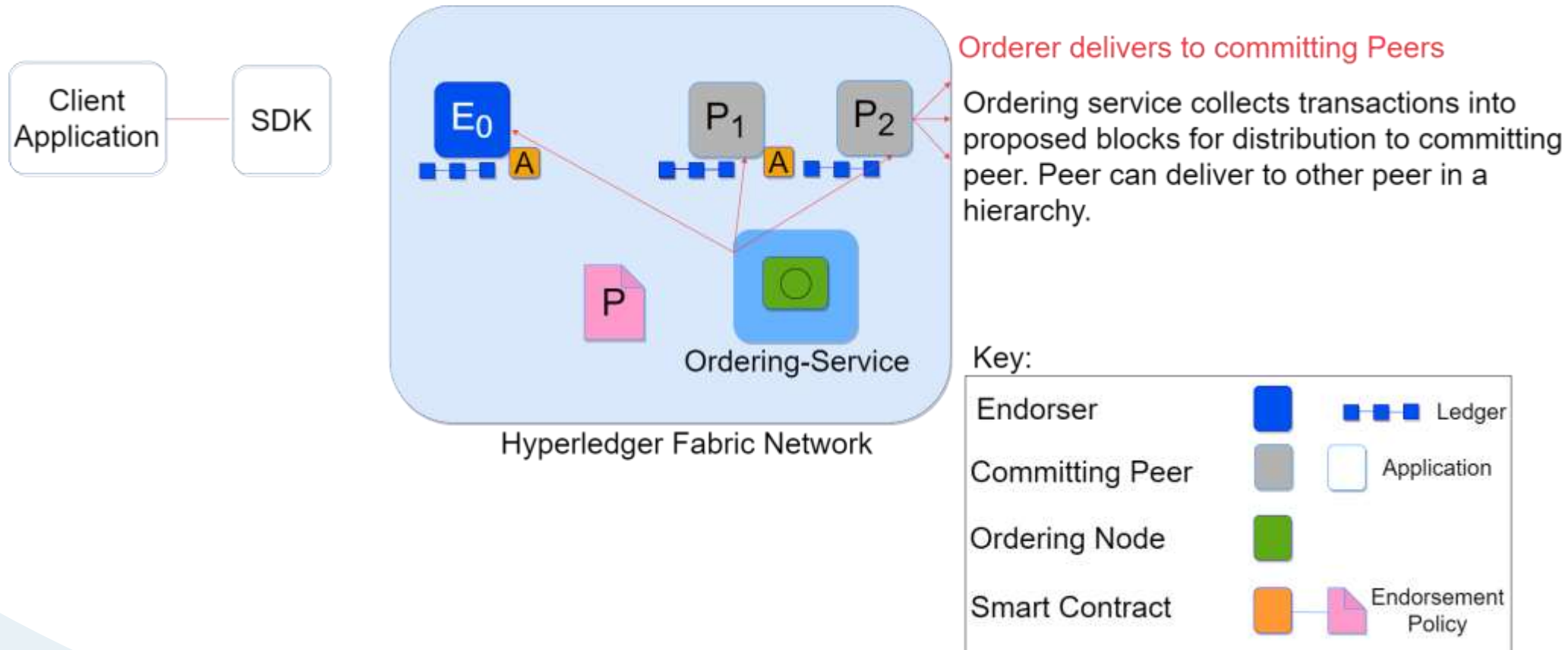
Application submits responses as a transaction to be ordered

Ordering happens across the fabric in parallel with transactions submitted by other applications

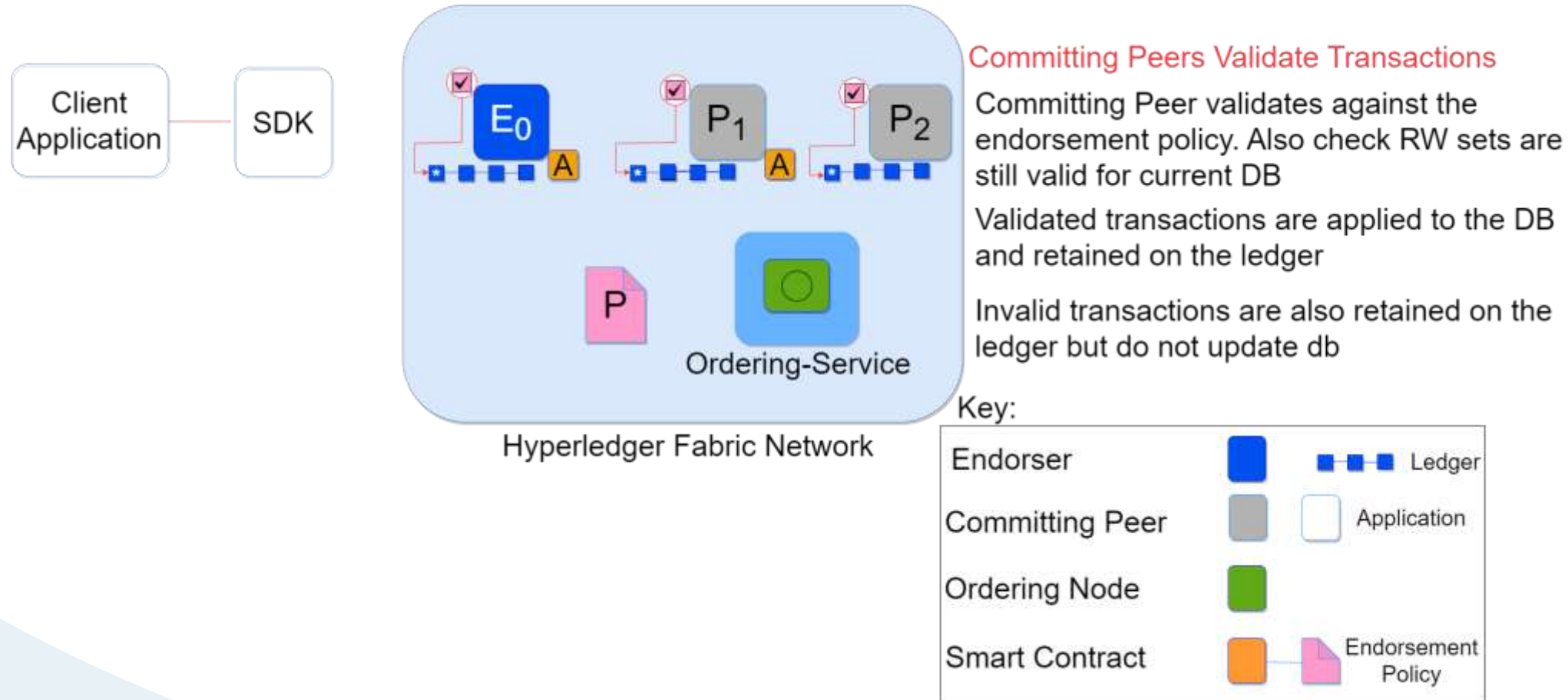
Key:

Endorser		 Ledger
Committing Peer		 Application
Ordering Node		
Smart Contract		 Endorsement Policy

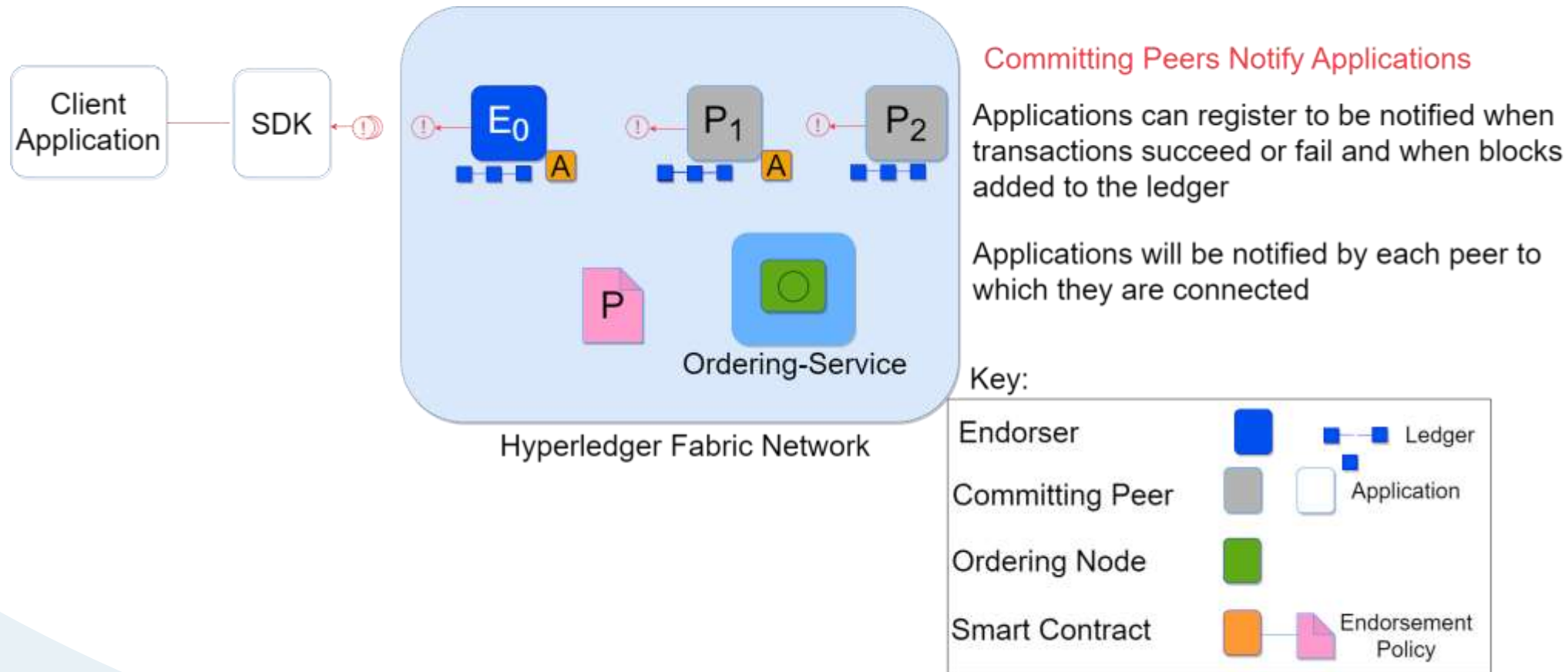
Deliver Transactions



Validate Transactions



Notify Transactions





Thank You

We hope you found this presentation informative and useful.

