Hyperledger Fabric Fundamentals

Morgan Bauer
Open Source Contributor, IBM Open Technologies
mbauer@us.ibm.com
@ibmhb

Hyperledger By The Linux Foundation

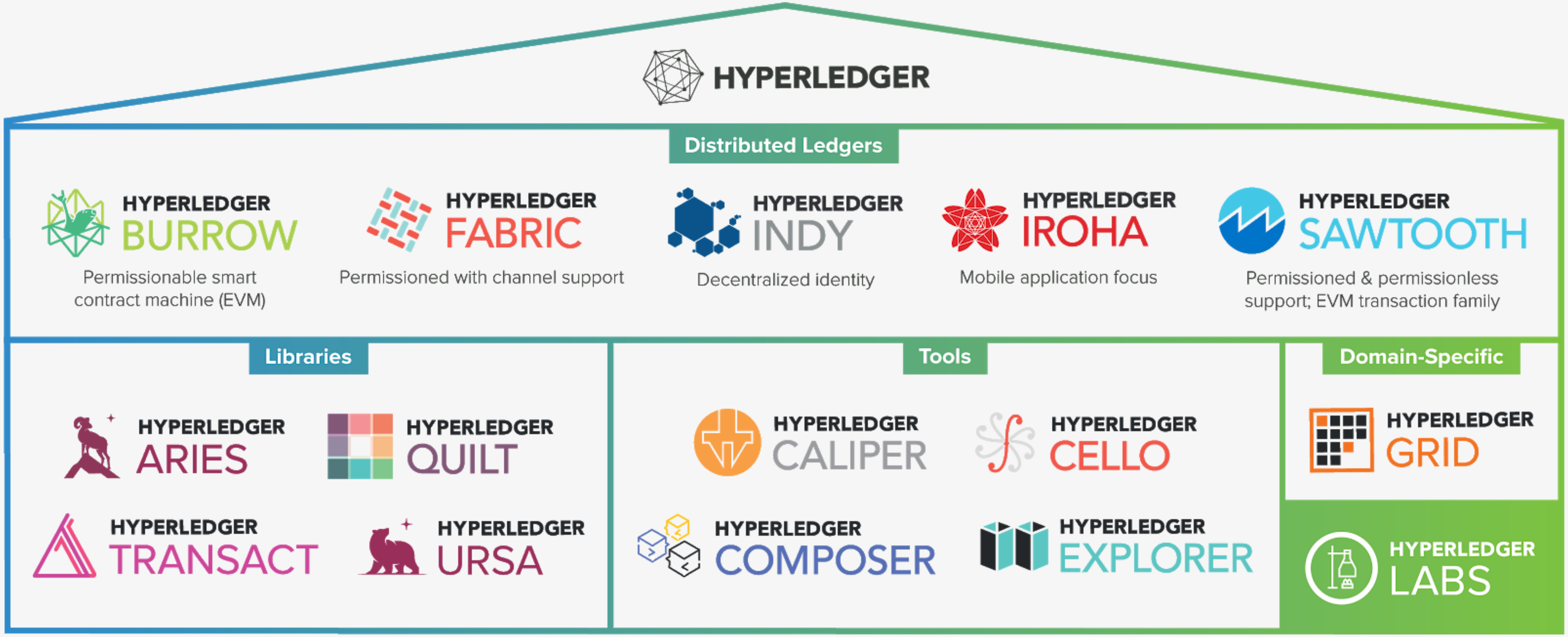
For the last 17 years, The Linux Foundation® has provided unparalleled support for open source communities through financial and intellectual resources, governance structure, IT infrastructure, services, events, and training.

Dedicated to building sustainable ecosystems around open source projects, The Linux Foundation is working with the global technology community to solve the world's hardest problems through open source and **creating the largest shared technology investment in history**.

The Linux Foundation is the umbrella organization for **more than 60 open source projects** accelerating open technology development and commercial adoption. Some of the game-changing initiatives hosted by The Linux Foundation include:

OPENCHAIN
Let's Encrypt
JS Foundation
OPEN CONTAINER INITIATIVE
CLOUD NATIVE COMPUTING FOUNDATION
Xen Project
Core Embedded Linux Project
AUTOMOTIVE GRADE LINUX
LF NETWORKING
CORE INFRASTRUCTURE INITIATIVE
OPNFV
CLOUD FOUNDRY
node JS
yocto PROJECT

Hyperledger Modular Approach



HYPERLEDGER

Distributed Ledgers

HYPERLEDGER BURROW
Permissionable smart contract machine (EVM)

HYPERLEDGER FABRIC
Permissioned with channel support

HYPERLEDGER INDY
Decentralized identity

HYPERLEDGER IROHA
Mobile application focus

HYPERLEDGER SAWTOOTH
Permissioned & permissionless support; EVM transaction family

Libraries

HYPERLEDGER ARIES

HYPERLEDGER QUILT

HYPERLEDGER TRANSACT

HYPERLEDGER URSA

Tools

HYPERLEDGER CALIPER

HYPERLEDGER CELLO

HYPERLEDGER COMPOSER

HYPERLEDGER EXPLORER

Domain-Specific

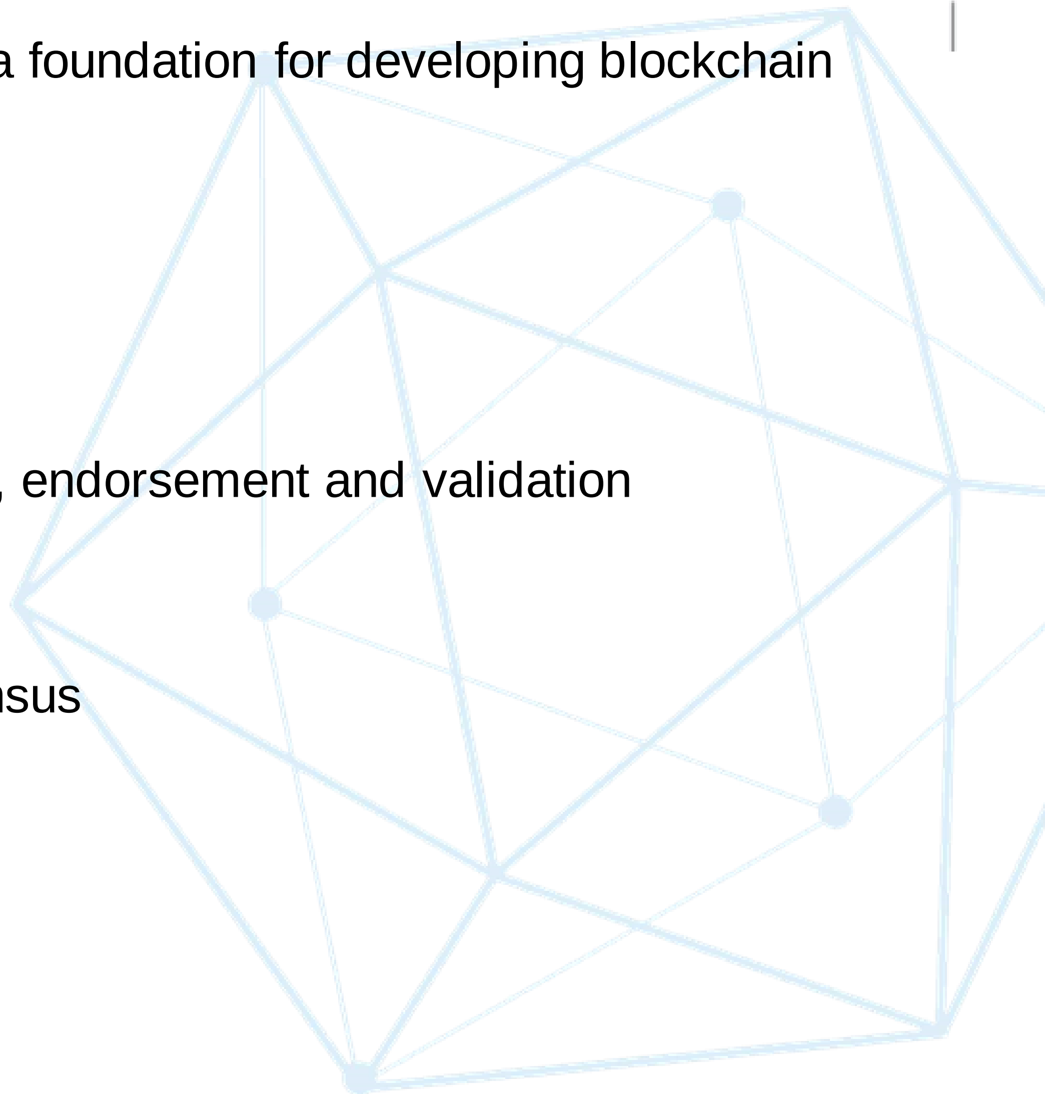
HYPERLEDGER GRID

HYPERLEDGER LABS

What is Hyperledger Fabric

- An implementation of blockchain technology that is intended as a foundation for developing blockchain applications for the enterprise

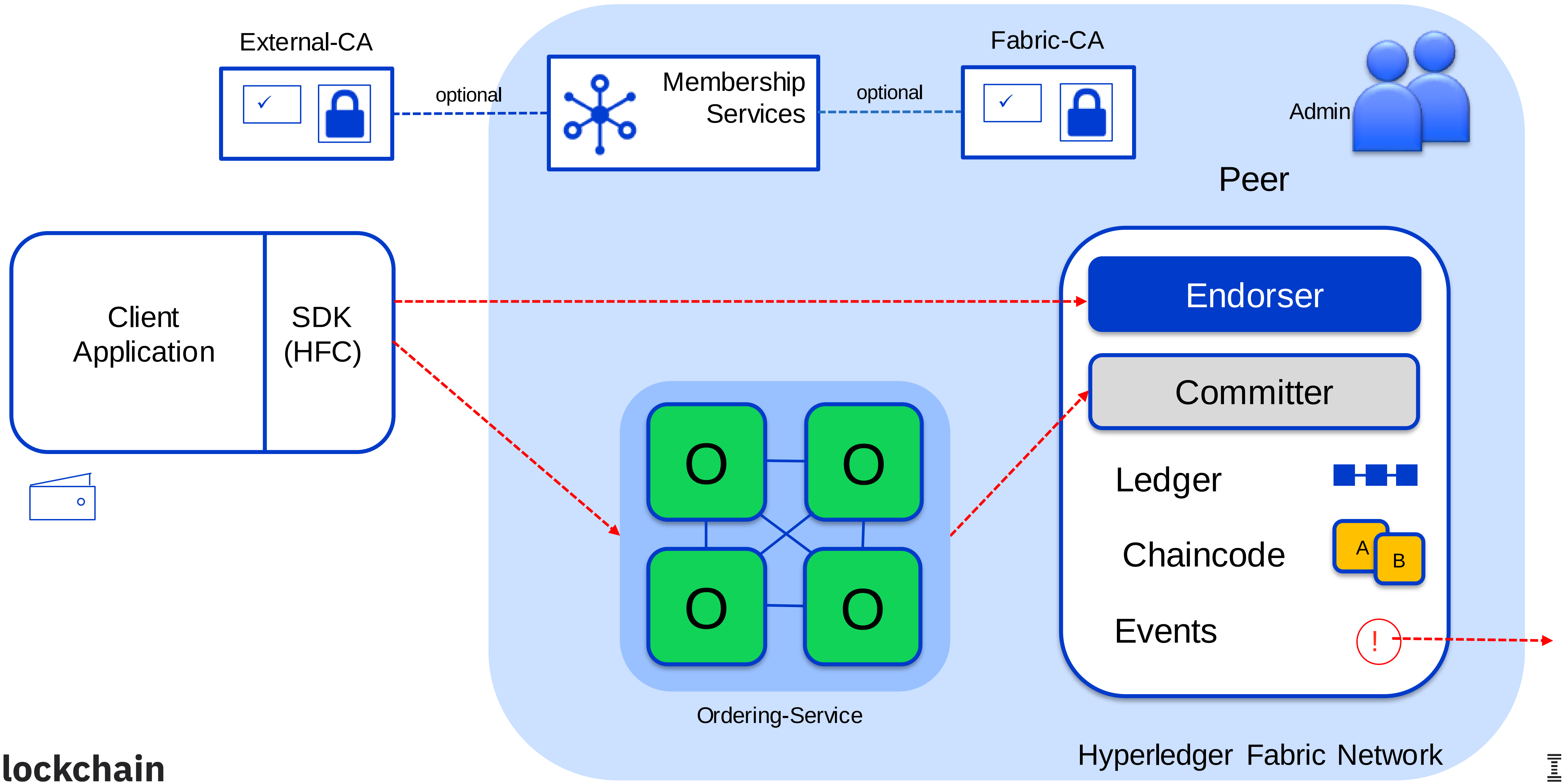
- Key characteristics:
  - Permissioned
  - Highly modular
    - Pluggable consensus, ledger, membership services, endorsement and validation
  - Smart contracts in general purpose languages
  - Privacy
  - No “mining” or native crypto-currency required for consensus
  - Execute-order-validate vs order-execute



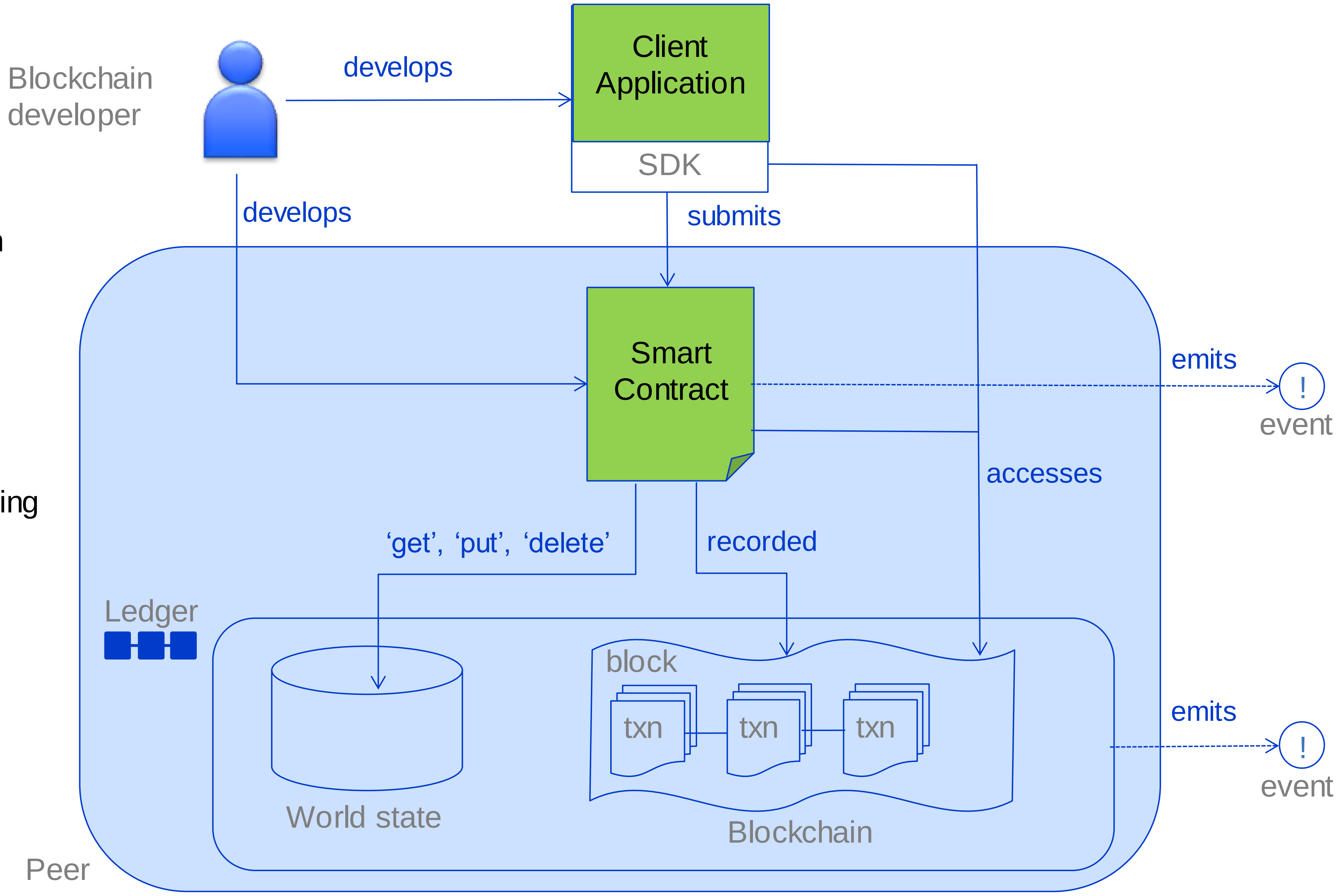
Technical Deep Dive

- [Architectural Overview]
- Network Consensus
- Channels and Ordering Service
- Components
- Network setup
- Endorsement Policies
- Membership Services
- Roadmap

Hyperledger Fabric V1 Architecture



How applications interact with the ledger

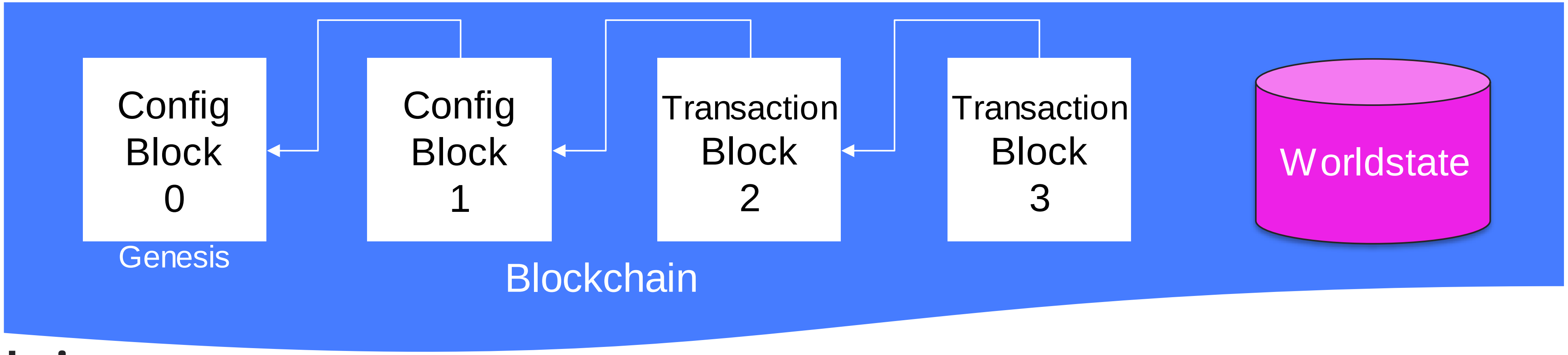


1. Client Application in using Hyperledger Fabric Client (HFC) SDK

2. Smart Contract implemented using chaincode – managing the World state

Fabric Ledger

- The Fabric ledger is maintained by each peer and includes the blockchain and worldstate
- A separate ledger is maintained for each channel the peer joins
- Transaction read/write sets are written to the blockchain
- Channel configurations are also written to the blockchain
- The worldstate can be either LevelDB (default) or CouchDB
  - LevelDB is a simple key/value store
  - CouchDB is a document store that allows complex queries
- The smart contact Contract decides what is written to the worldstate



Technical Deep Dive

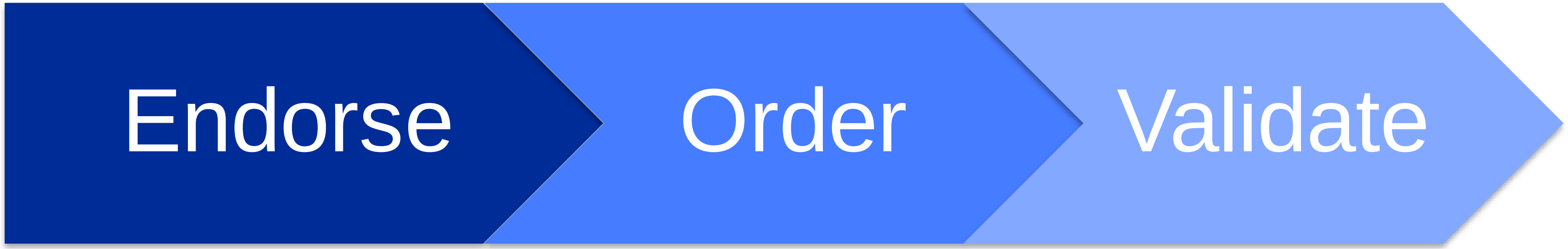
- Architectural Overview
- [Network Consensus]
- Channels and Ordering Service
- Components
- Network setup
- Endorsement Policies
- Membership Services
- Roadmap

Nodes and roles

⬜	**Committing Peer:** Maintains ledger and state. Commits transactions. May hold smart contract (chaincode).
🟦	**Endorsing Peer:** Specialized peer also endorses transactions by receiving a transaction proposal and responds by granting or denying endorsement. Must hold smart contract.
🟩	**Ordering Node:** Approves the inclusion of transaction blocks into the ledger and communicates with committing and endorsing peer nodes. Does not hold smart contract. Does not hold ledger.

Hyperledger Fabric Consensus

Consensus is achieved using the following transaction flow:

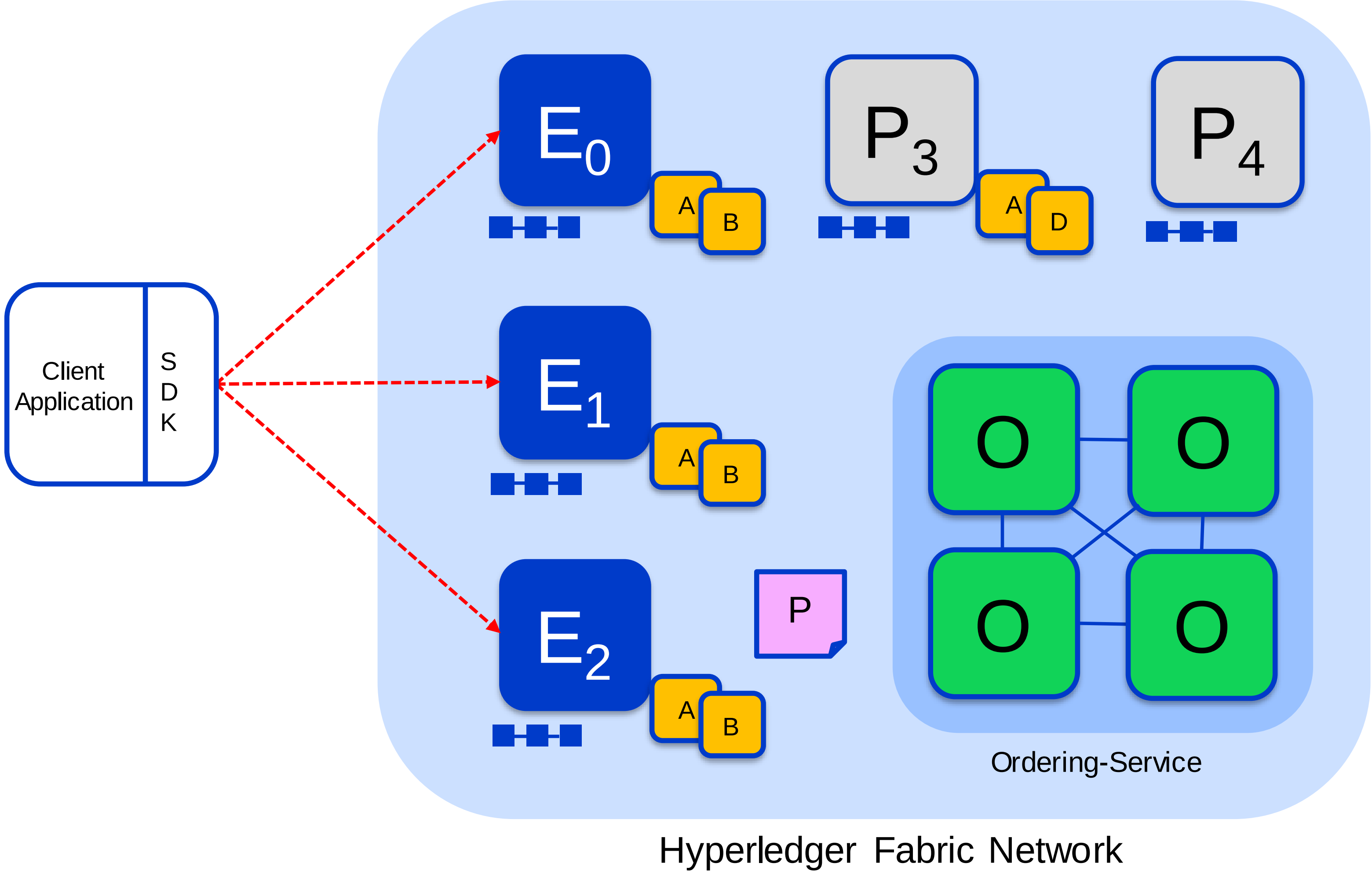


Endorse

Order

Validate

Sample transaction: Step 1/7 – Propose transaction



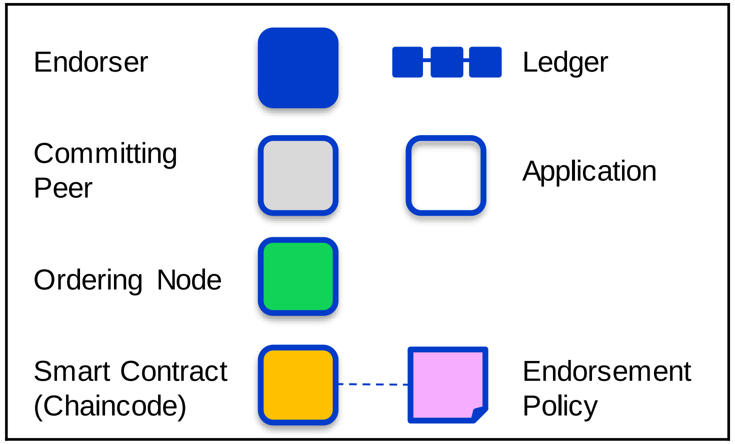
Application proposes transaction

Endorsement policy:

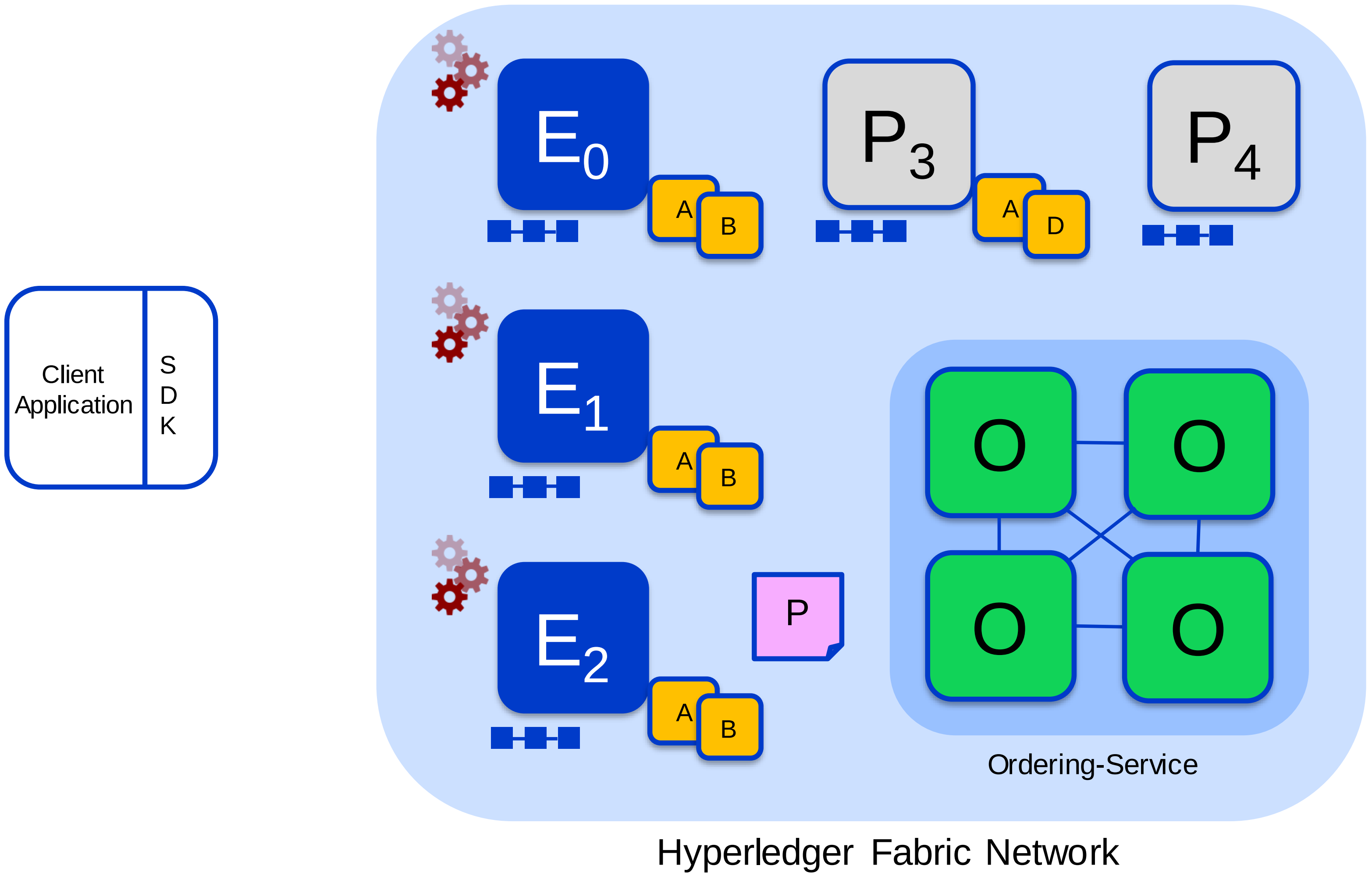
- “E_0, E_1 and E_2 must sign”
- (P_3, P_4 are not part of the policy)

Client application submits a transaction proposal for Smart Contract A. It must target the required peers {E_0, E_1, E_2}

Key:



Sample transaction: Step 2/7 – Execute proposal



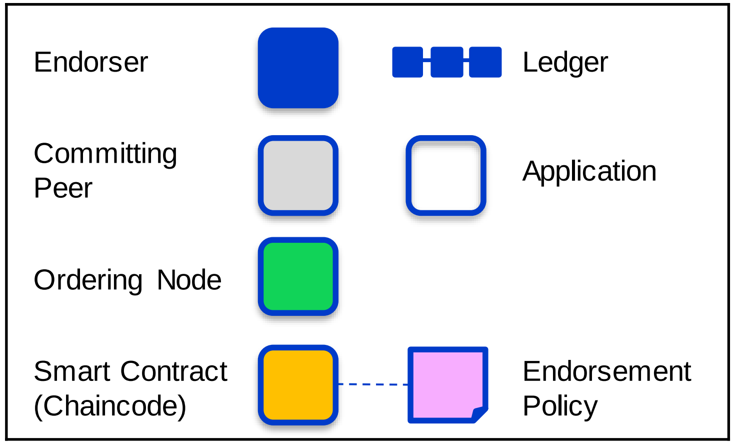
Endorsers Execute Proposals

E_0, E_1 & E_2 will each execute the proposed transaction. None of these executions will update the ledger

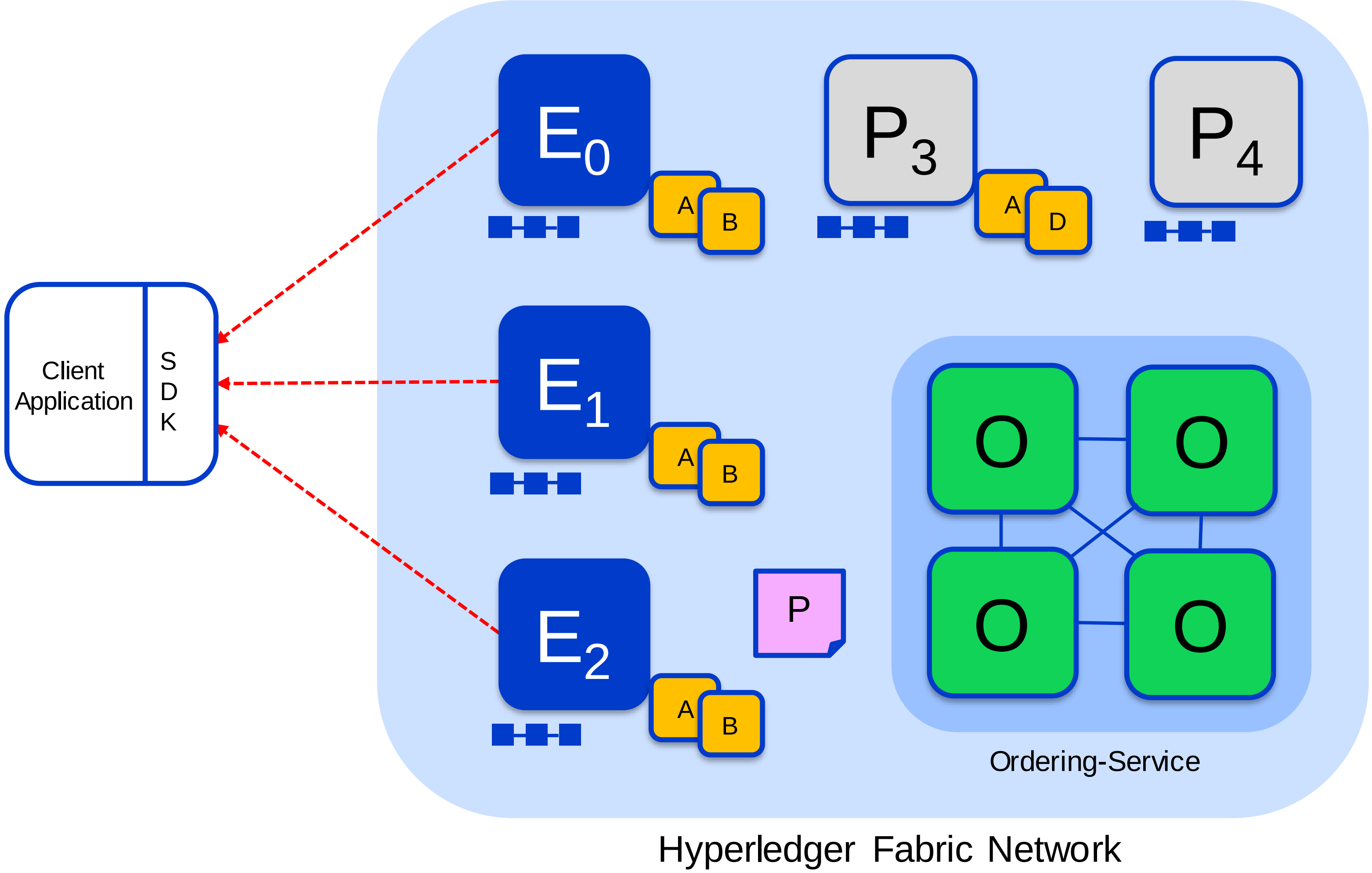
Each execution will capture the set of Read and Written data, called RW sets, which will now flow in the fabric.

Transactions can be signed & encrypted

Key:



Sample transaction: Step 3/7 – Proposal Response



Application receives responses

RW sets are asynchronously returned to application

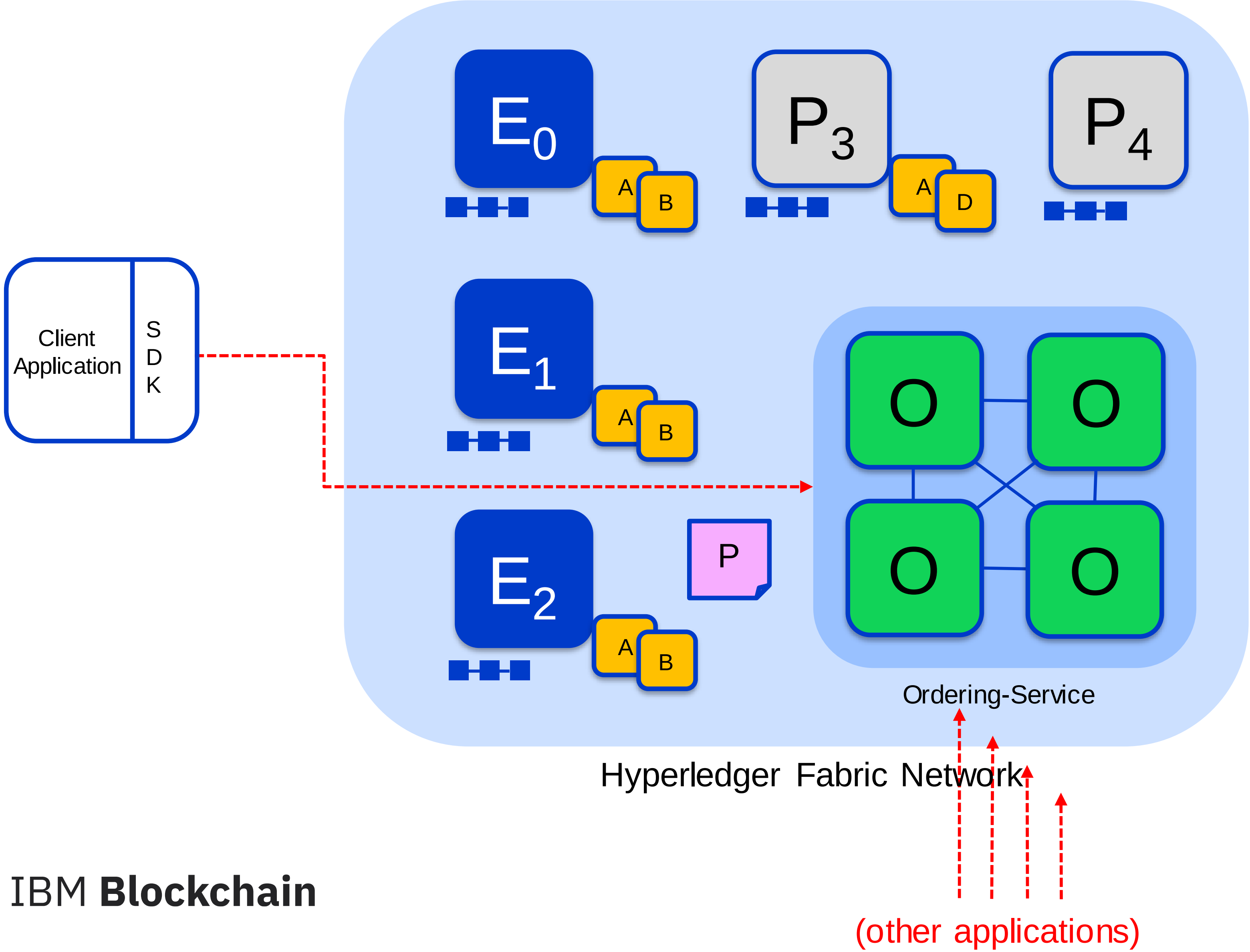
The RW sets are signed by each endorser, and also includes each record version number

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

Key:

Endorser	■	■-■-■	Ledger
Committing Peer	■	□	Application
Ordering Node	🟩		
Smart Contract (Chaincode)	🟨	▨	Endorsement Policy

Sample transaction: Step 4/7 – Order Transaction



Responses submitted for ordering

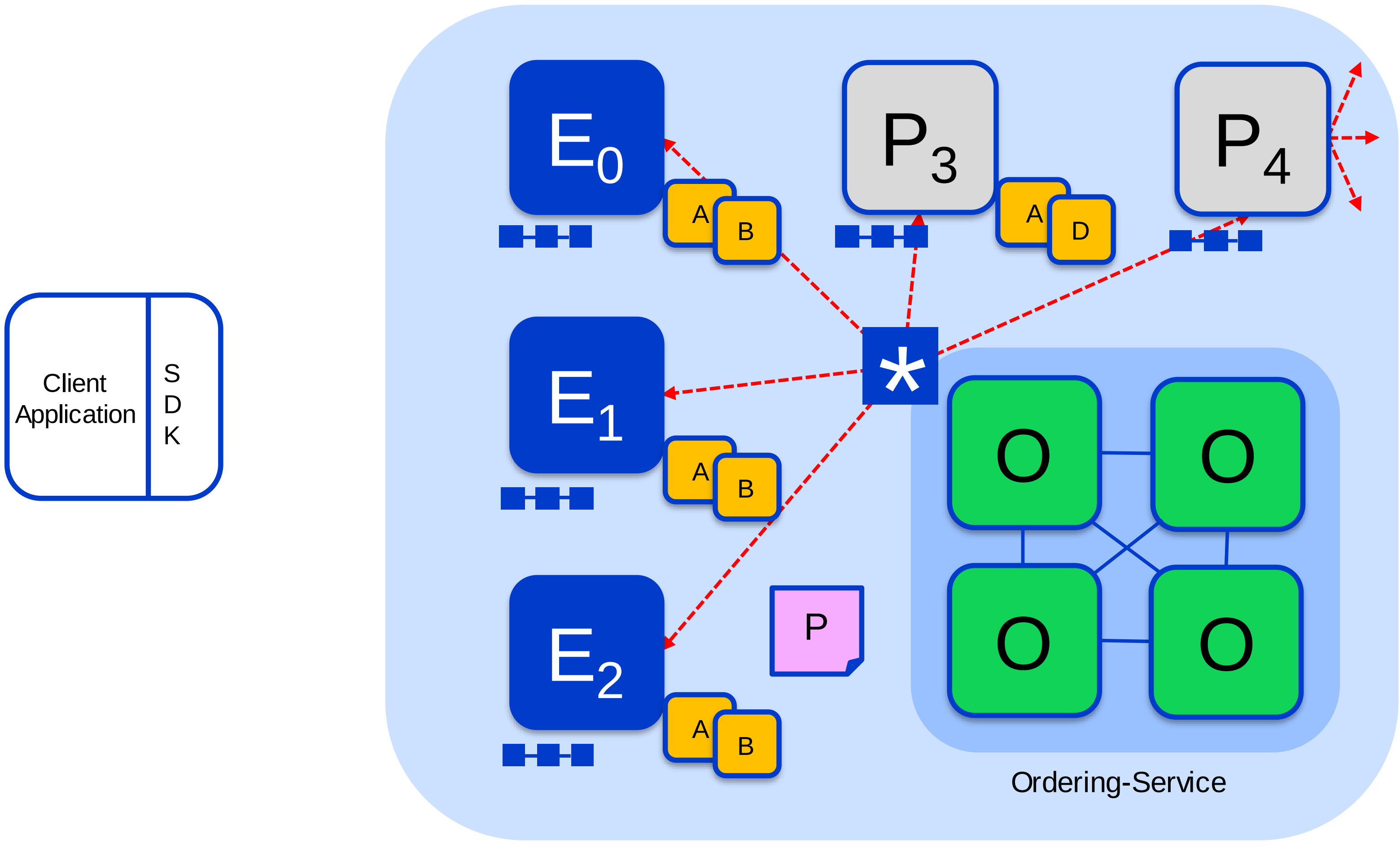
Application submits responses as a transaction to be ordered.

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

Key:

Endorser	🟦	■■■	Ledger
Committing Peer	⬜	□	Application
Ordering Node	🟩		
Smart Contract (Chaincode)	🟨	▨	Endorsement Policy

Sample transaction: Step 5/7 – Deliver Transaction



Hyperledger Fabric Network

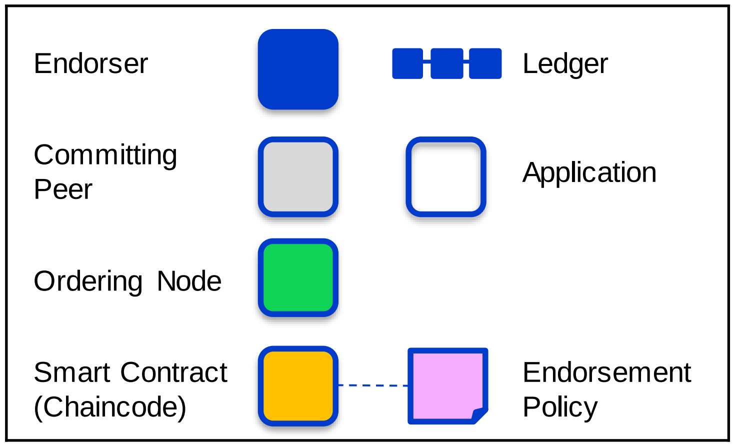
Orderer delivers to committing peers

Ordering service collects transactions into proposed blocks for distribution to committing peers. Peers can deliver to other peers in a hierarchy (not shown)

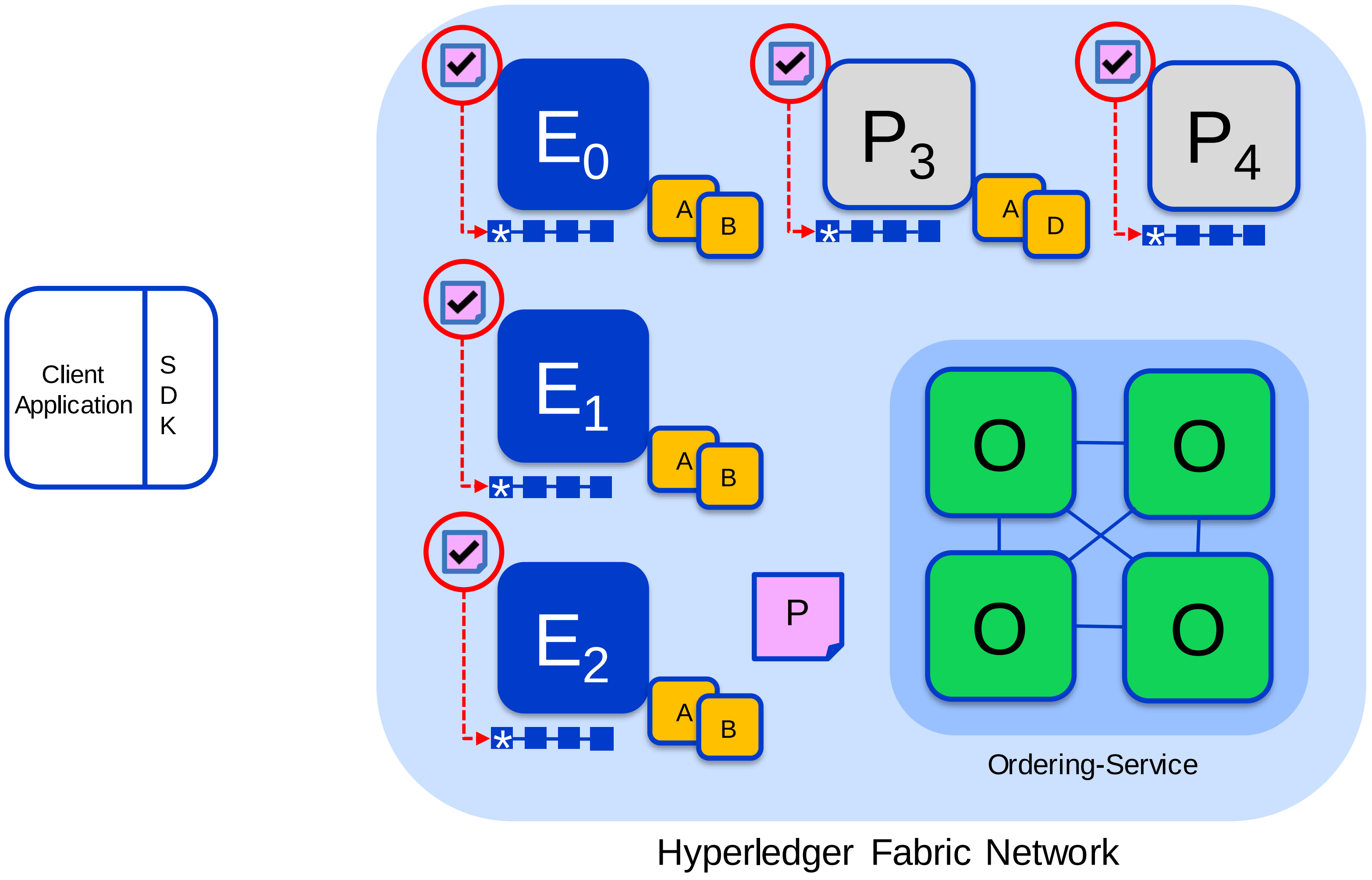
Different ordering algorithms available:

- SOLO (Single node, development)
- Kafka (Crash fault tolerant)
- Raft (Crash fault tolerant)

Key:



Sample transaction: Step 6/7 – Validate Transaction



Committing peers validate transactions

Every committing peer validates against the endorsement policy. Also check RW sets are still valid for current world state

Validated transactions are applied to the world state and retained on the ledger

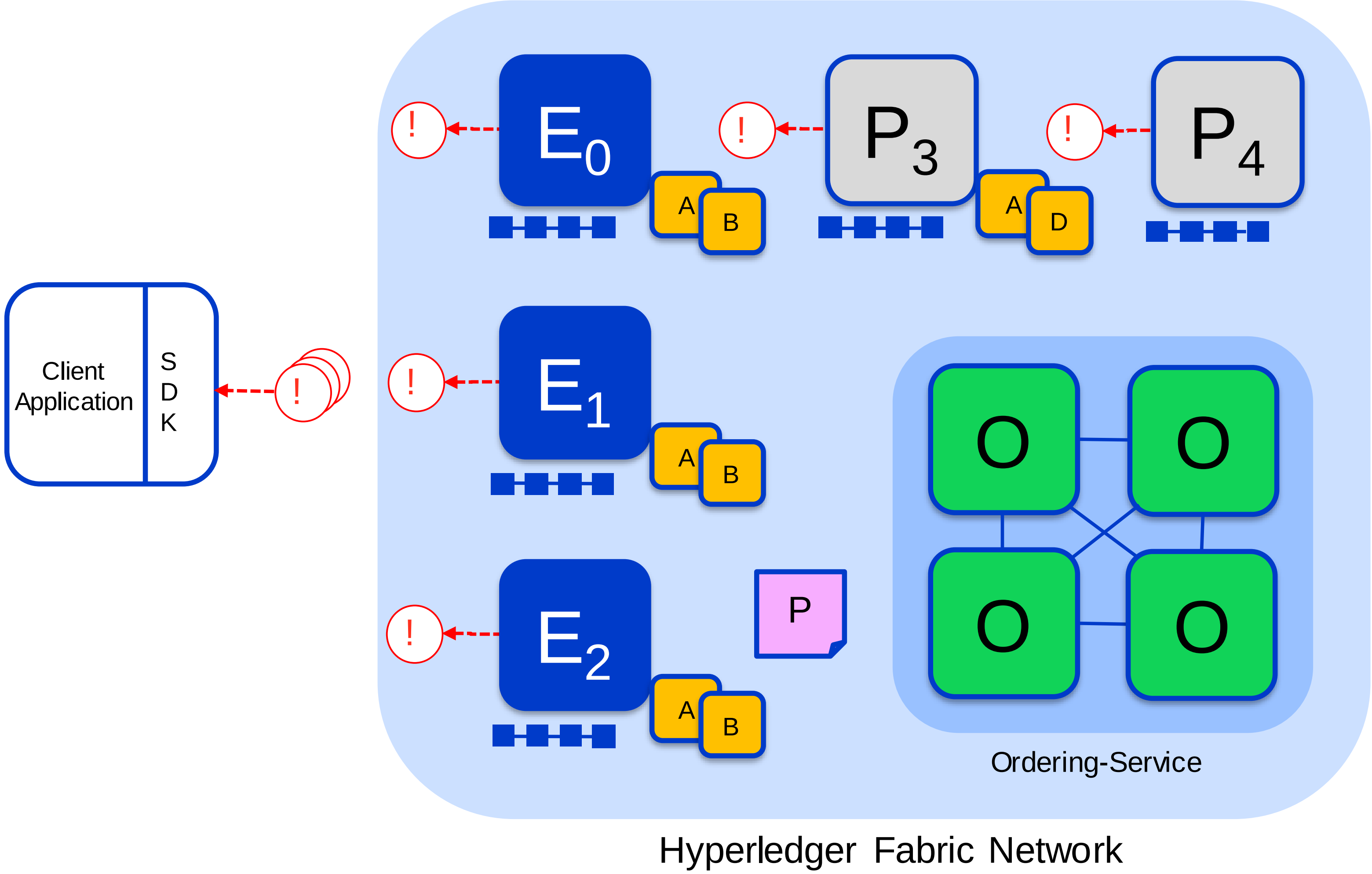
Invalid transactions are also retained on the ledger but do not update world state

Key:

Endorser	■	■-■-■	Ledger
Committing Peer	□	□	Application
Ordering Node	🟩		
Smart Contract (Chaincode)	🟨	▨	Endorsement Policy

IBM Blockchain

Sample transaction: Step 7/7 – Notify Transaction



Committing peers notify applications

Applications can register to be notified when transactions succeed or fail, and when blocks are added to the ledger

Applications will be notified by each peer to which they are connected

Key:

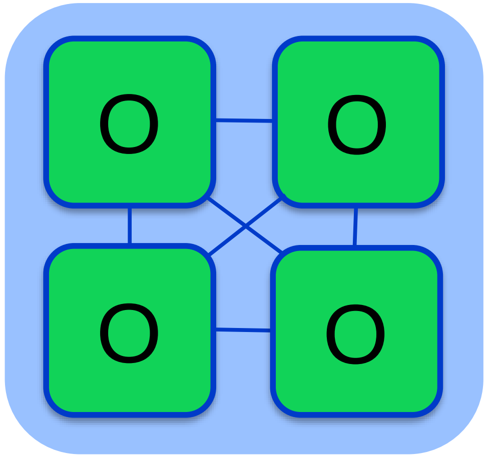
Endorser	🟦	■-■-■	Ledger
Committing Peer	▢	□	Application
Ordering Node	🟩		
Smart Contract (Chaincode)	🟨	🟪	Endorsement Policy

Technical Deep Dive

- Architectural Overview
- Network Consensus
- [Channels and Ordering Service]
- Components
- Network setup
- Endorsement Policies
- Membership Services
- Roadmap

Ordering Service

The ordering service packages transactions into blocks to be delivered to peers. Communication with the service is via channels.



Ordering-Service

Different configuration options for the ordering service include:

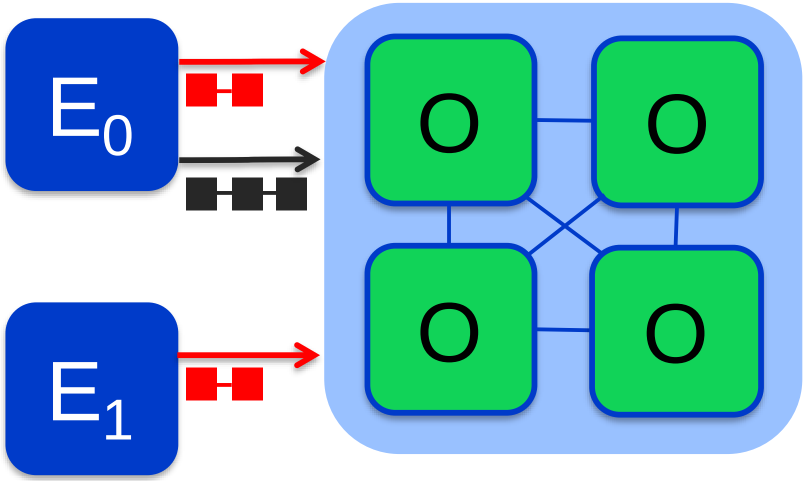
– SOLO
  - Single node for development

– Kafka : Crash fault tolerant consensus
  - 3 nodes minimum
  - Odd number of nodes recommended

– Raft : Crash fault tolerant consensus
  - 3 nodes minimum
  - Odd number of nodes recommended

Channels

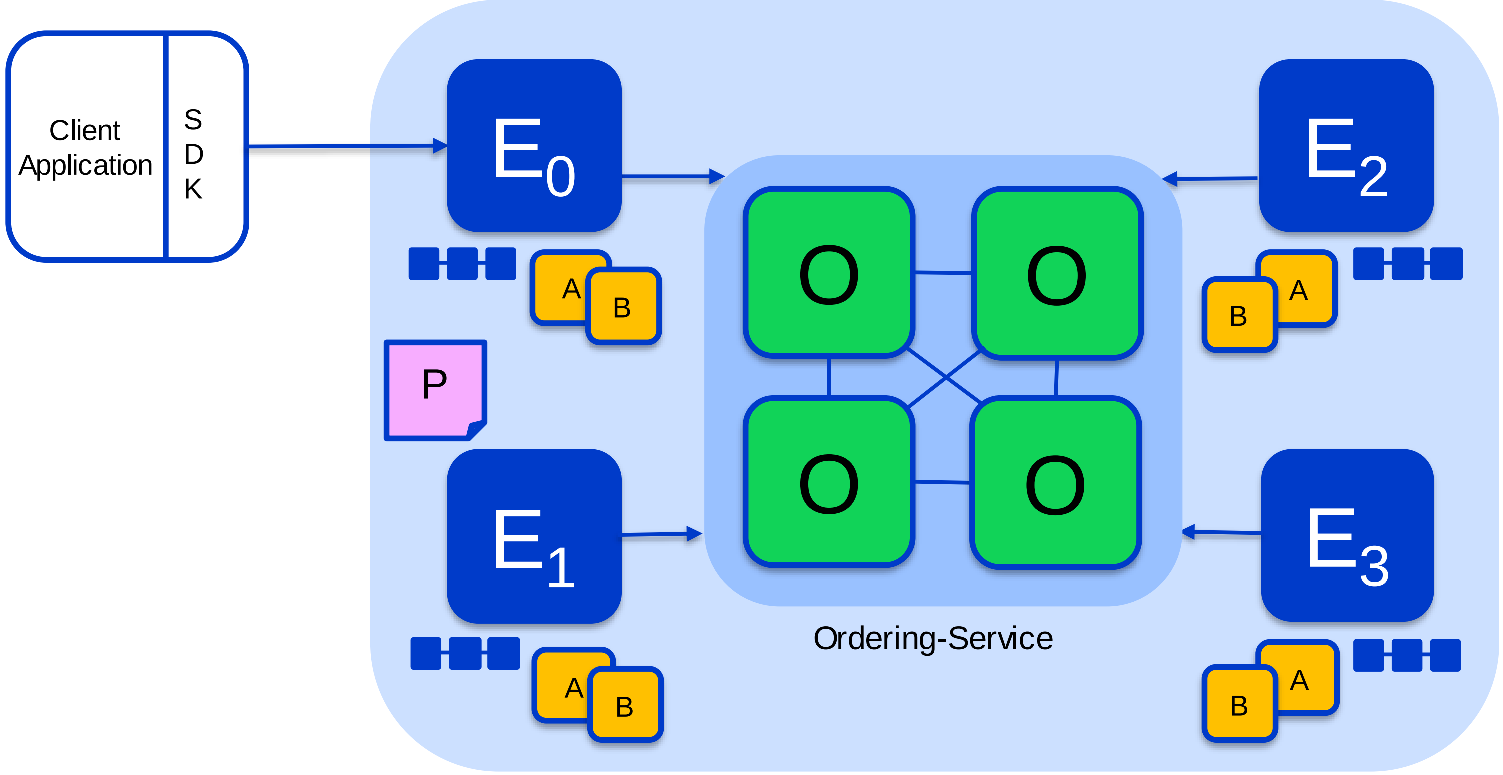
Channels provide privacy between different ledgers



Ordering-Service

- Ledgers exist in the scope of a channel
  - Channels can be shared across an entire network of peers
  - Channels can be permissioned for a specific set of participants
- Chaincode is installed on peers to access the worldstate
- Chaincode is instantiated on specific channels
- Peers can participate in multiple channels
- Concurrent execution for performance and scalability

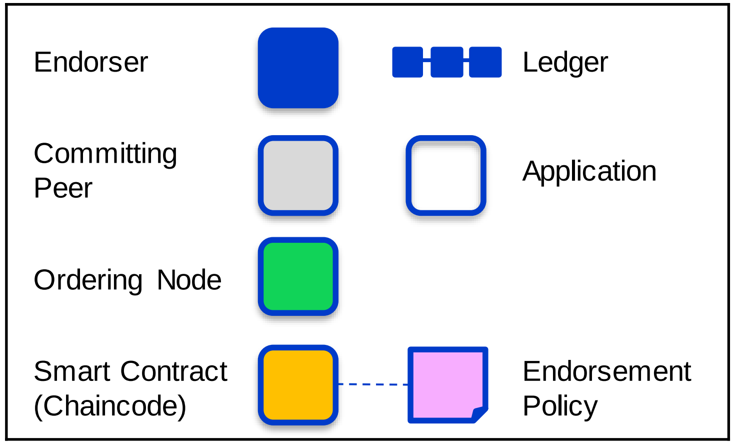
Single Channel Network



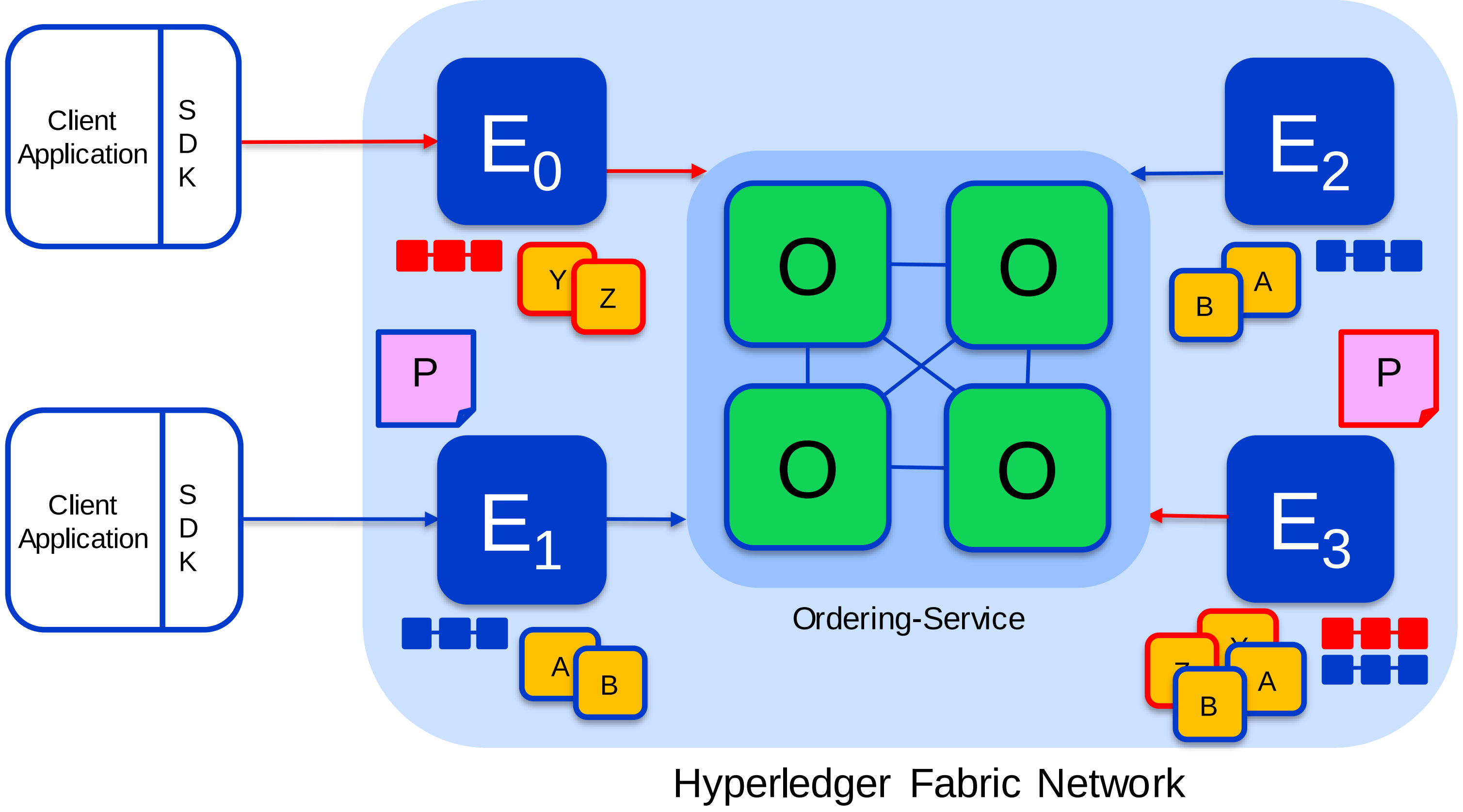
Hyperledger Fabric Network

- All peers connect to the same system channel (blue).
- All peers have the same chaincode and maintain the same ledger
- Endorsement by peers E_0, E_1, E_2 and E_3

Key:



Multi Channel Network



- Peers E_0 and E_3 connect to the red channel for chaincodes Y and Z
- E_1, E_2 and E_3 connect to the blue channel for chaincodes A and B

Key:

Endorser	🟦	▬▬▬	Ledger
Committing Peer	⬜	□	Application
Ordering Node	🟩		
Smart Contract (Chaincode)	🟨	🟪	Endorsement Policy

Get Started Contributing to Hyperledger Projects

- Linux Foundation Account – https://identity.linuxfoundation.org/
  - Email based, or social media login, including Github
- Rocket chat – https://chat.hyperledger.org/
- Hyperledger Wiki – https://wiki.hyperledger.org/
- Jira – https://jira.hyperledger.org/
- Gerrit – https://gerrit.hyperledger.org/

Getting started with Hyperledger Fabric

- Build Your First Network (BYFN) – Network administrator
  - A simple network with 2 organizations running 2 peers, with one channel, a simple chaincode
  - Dockerhub images
  - Uses predefined enrollment certificates and « Solo » Ordering Service
- Extend Your First Network – Network administrator
  - Adds a 3rd organization to BYFN
- Develop Your First Application – Application developer
  - A simple Node.js application
- Start in devmode (minimal set up), then move to network (several peers), and security (membersrvc)
- Several examples to start from (fabcar)

Resources

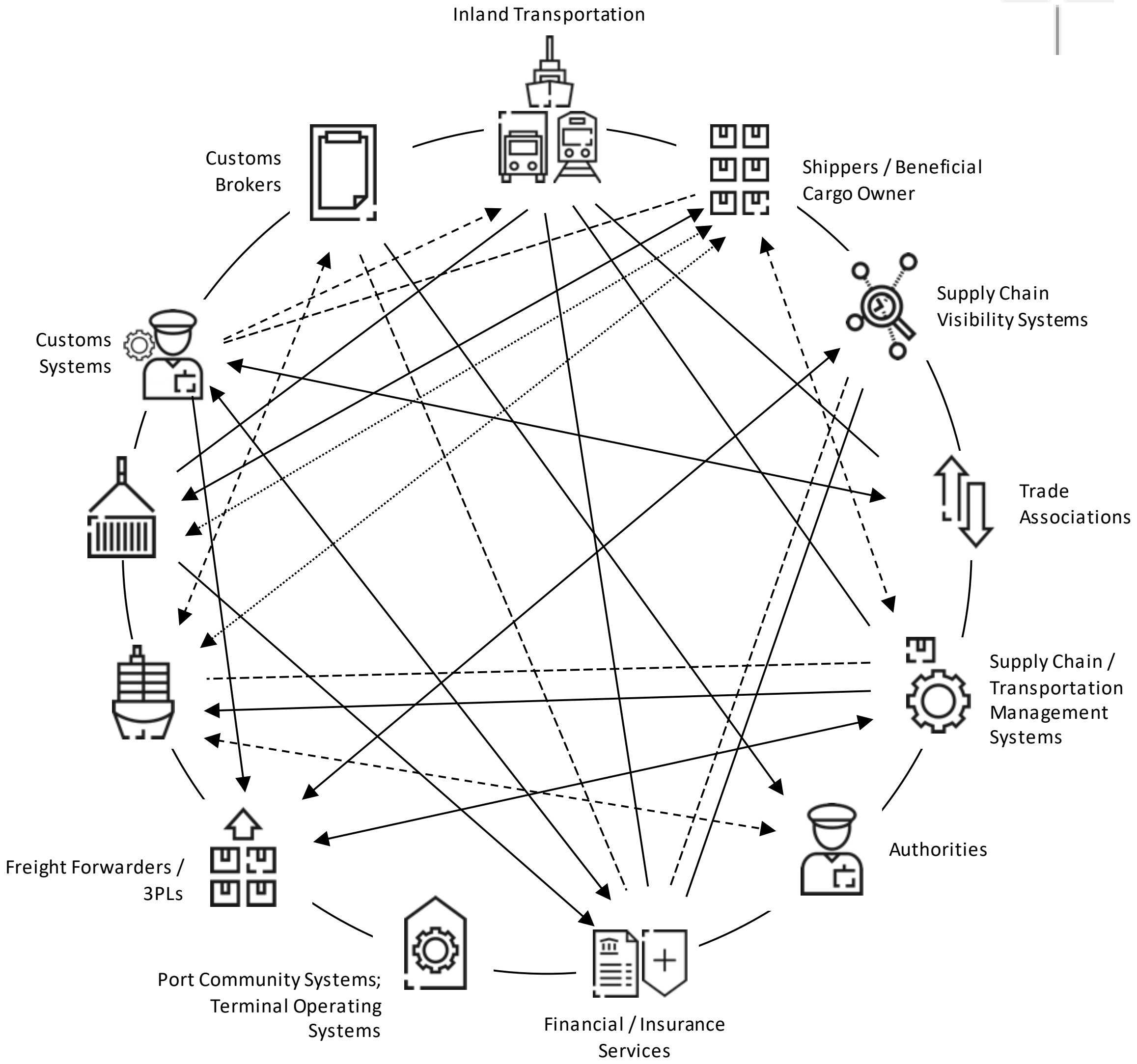
- **Hyperledger** http://hyperledger-fabric.readthedocs.io/en/release/
  - **Docs + tutorials**
- **IBM Code:** https://developer.ibm.com/code/technologies/blockchain/
  - **Code patterns, lectures, howtos, lab, etc**
- **IBM Blockchain Dev Center:** https://developer.ibm.com/blockchain/
  - **Blockchain 101**
- **IBM Blockchain Platform:**
  - **Starter Plan:** https://www.ibm.com/blockchain/getting-started.html

TradeLens improves **global trade efficiency**

- TradeLens is an open, extensible platform for sharing shipping events, messages, and documents across all the actors and systems in the supply chain ecosystem.
- It provides shared visibility and shared state for container shipments

Benefits

- Increase speed and transparency for cross border transactions through real time access to container events.
- Reduced cost and increased efficiency through paperless trade



Inland Transportation
Customs Brokers
Shippers / Beneficial Cargo Owner
Customs Systems
Supply Chain Visibility Systems
Trade Associations
Supply Chain / Transportation Management Systems
Authorities
Freight Forwarders / 3PLs
Port Community Systems; Terminal Operating Systems
Financial / Insurance Services

IBM **Blockchain**

Food Trust

What?

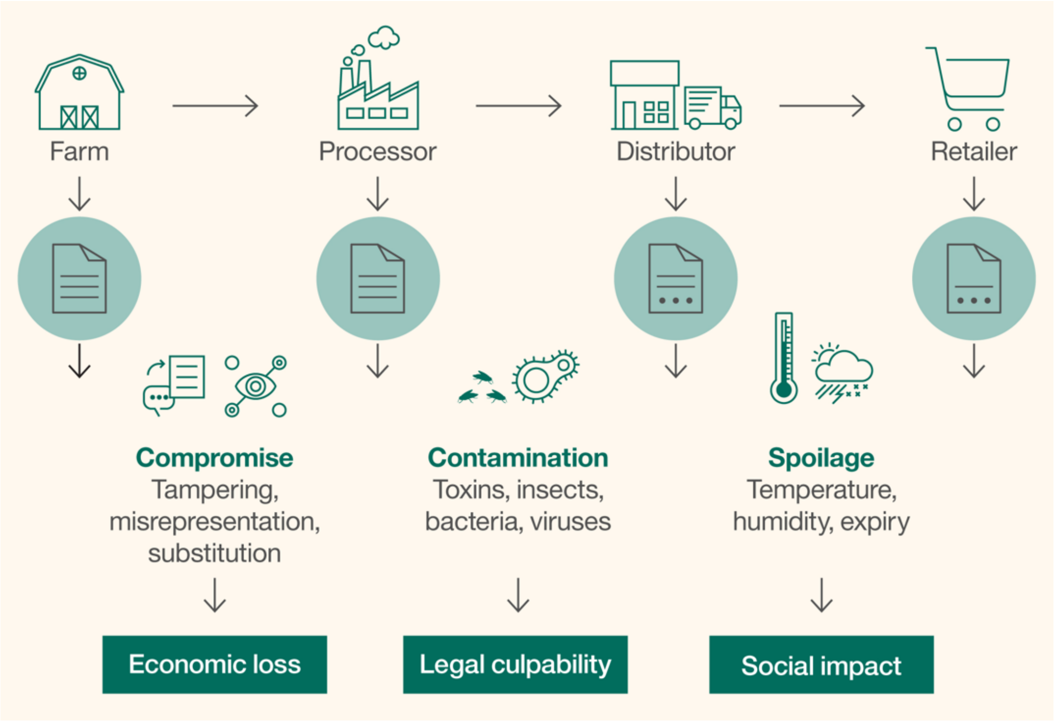
- Provide a trusted source of information and traceability to improve transparency and efficiency across the food network.

How?

- Shared ledger for storing digital compliance documentation, test results and audit certificates network.

Benefits

- Reduce impact of food recalls through instant access to end-to-end traceability data to verify history in the food network and supply chain.
- Help to address the 1 in 10 people sickened and 400,000 fatalities which occur every year from food-born illnesses.



Further examples by (selected) industry

Financial

- Trade Finance
- Cross currency payments
- Mortgages
- Letters of Credit

Public Sector

- Asset Registration
- Citizen Identity
- Medical records
- Medicine supply chain

Retail

- Supply chain
- Loyalty programs
- Information sharing (supplier – retailer)

Insurance

- Claims processing
- Risk provenance
- Asset usage history
- Claims file

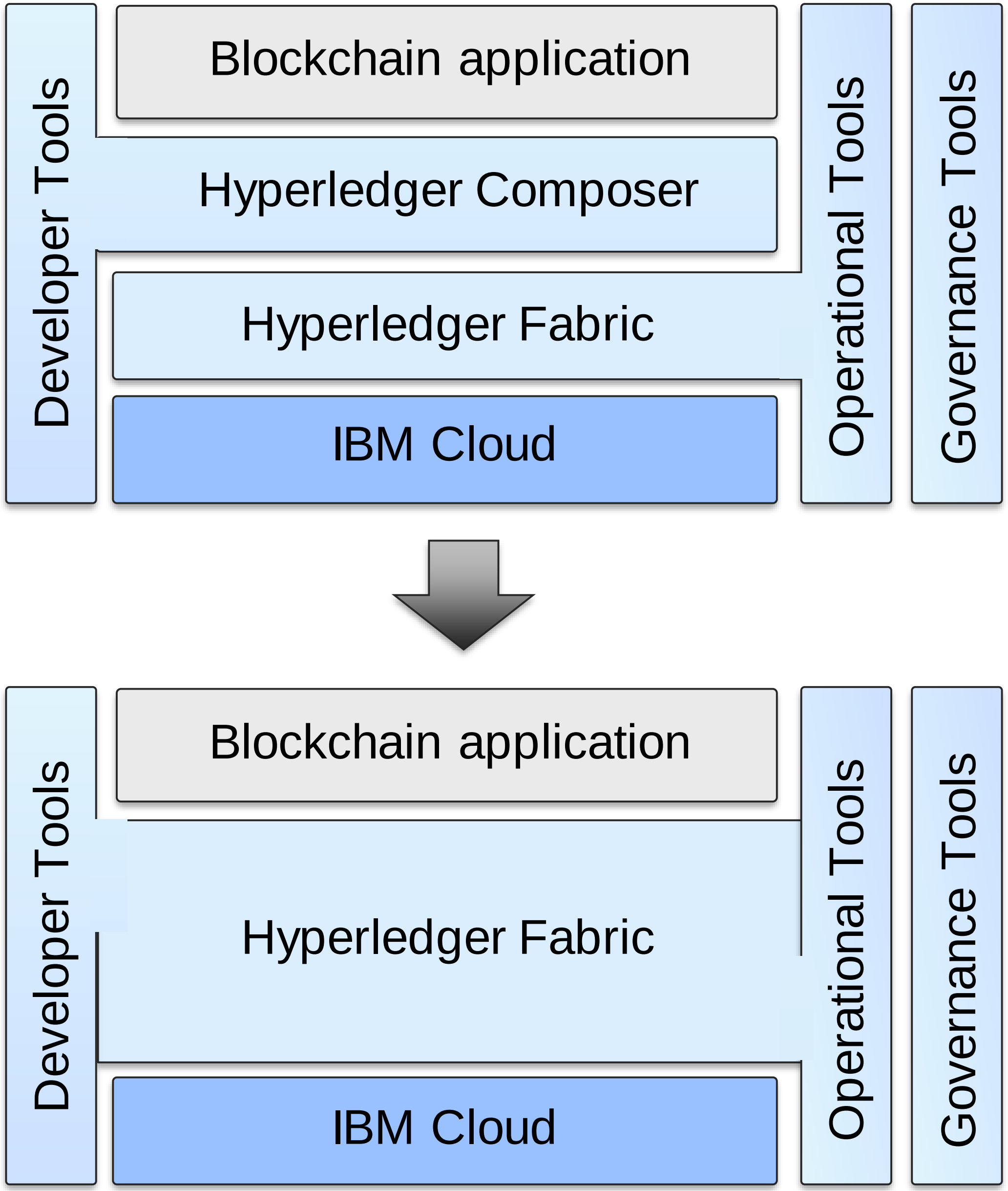
Manufacturing

- Supply chain
- Product parts
- Maintenance tracking

IBM Blockchain Platform

IBM Blockchain Platform is a fully integrated enterprise-ready blockchain platform designed to accelerate the development, governance, and operation of a multi-institution business network

- **Developer tools** that make use of Hyperledger Composer to quickly build your blockchain application
- Hyperledger Fabric provides the ledger, which is managed through a set of intuitive **operational tools**
- **Governance tools** for democratic management of the business network
- Flexible deployment options, including a highly secure and performant **IBM Cloud** environment



Thank you

Morgan Bauer
mbauer@us.ibm.com
@ibmhb

IBM **Blockchain**

www.ibm.com/blockchain

developer.ibm.com/blockchain

www.hyperledger.org

IBM