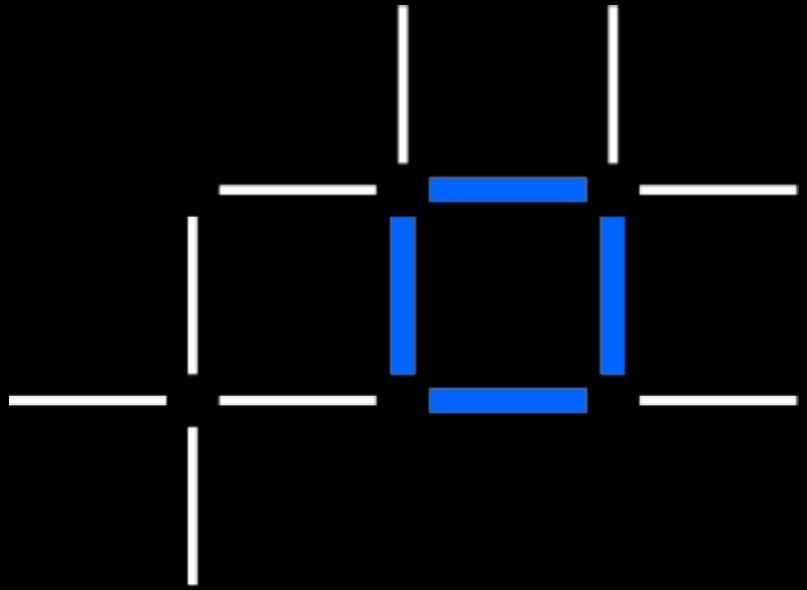
Blockchain Explored, Part 1

Basic components of a Hyperledger Fabric network



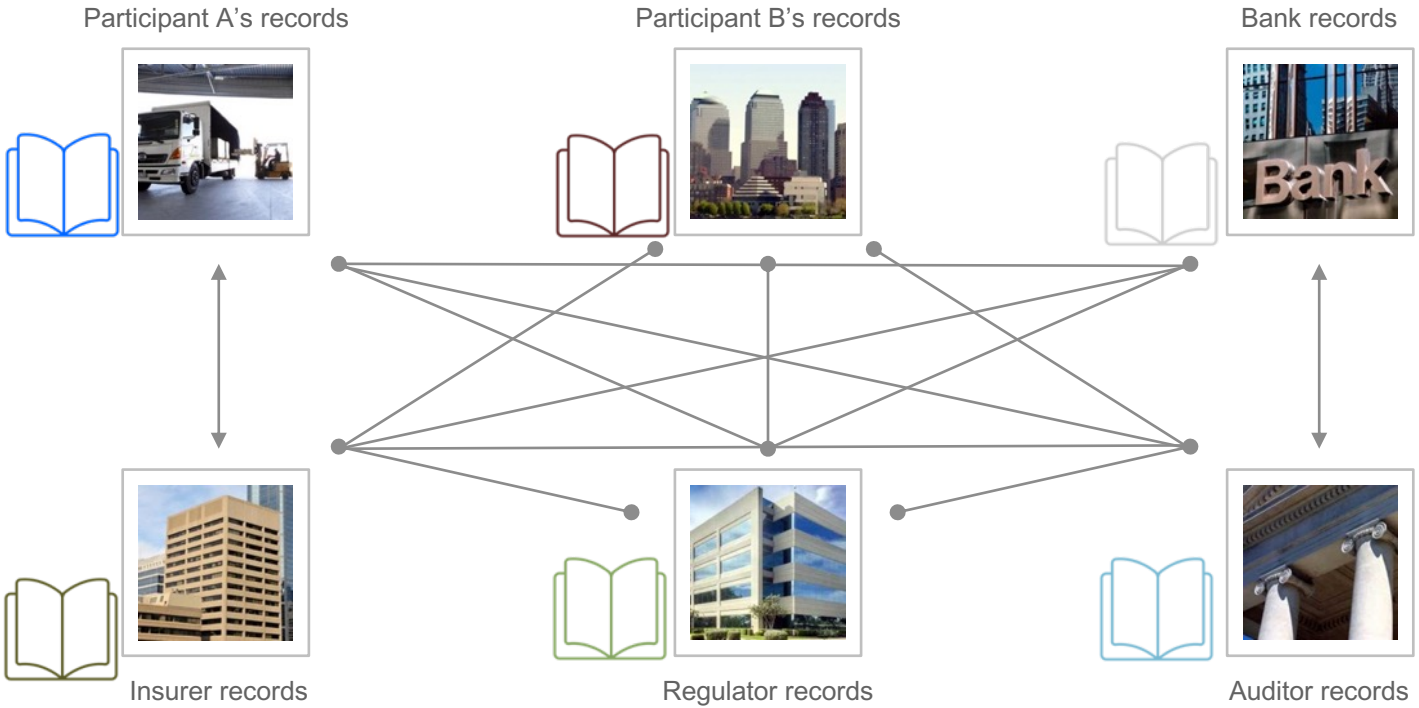
Barry Silliman
IBM Washington Systems Center
silliman@us.ibm.com

Ledgers, Transactions and Contracts

- Ledger: an important **log** of all transactions
  - Describes the inputs and outputs of the business
- Transaction: an **asset transfer** between participants
  - Matt gives a car to Dave (simple)
- Contract: the **conditions** for a transaction to occur
  - If Dave pays Matt money, then car passes from Matt to Dave (simple)
  - If car won't start, funds do not pass to Matt (as decided by third party arbitrator) (more complex)

Problem

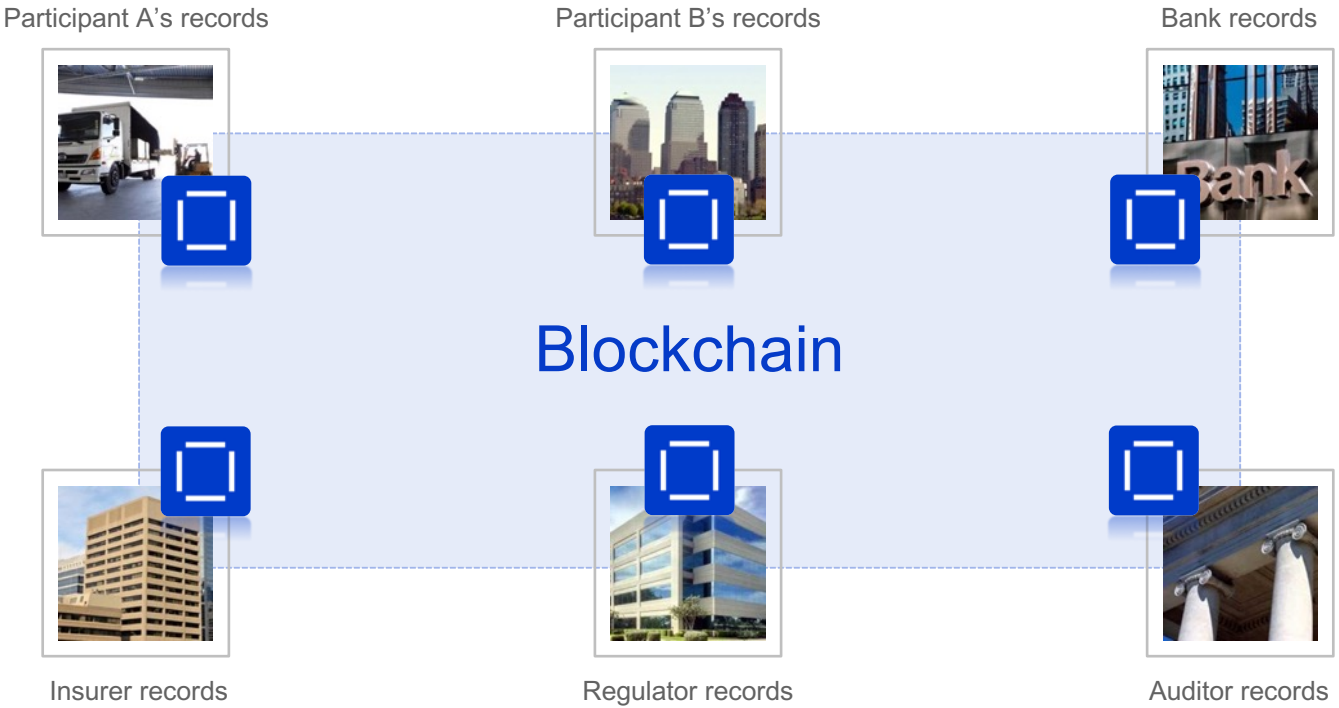
inefficient, expensive, vulnerable



Solution

A shared, replicated, permissioned ledger…

…with consensus, provenance, immutability and finality



Different types of blockchain

- All blockchains aim to provide irrefutable proof that a set of transactions occurred between participants

- Different types of blockchain exist:

  bitcoin is an example of an unpermissioned, public blockchain
  - The first blockchain application
  - Defines a shadow-currency and its ledger
  - Resource intensive

- Blockchains for business generally prioritize
  - **Assets** over cryptocurrency; **Identity** over anonymity; **Selective endorsement** over proof of work

Hyperledger and Linux Foundation

- **Hyperledger is a blockchain for business project under the Linux Foundation organization**

  https://www.hyperledger.org

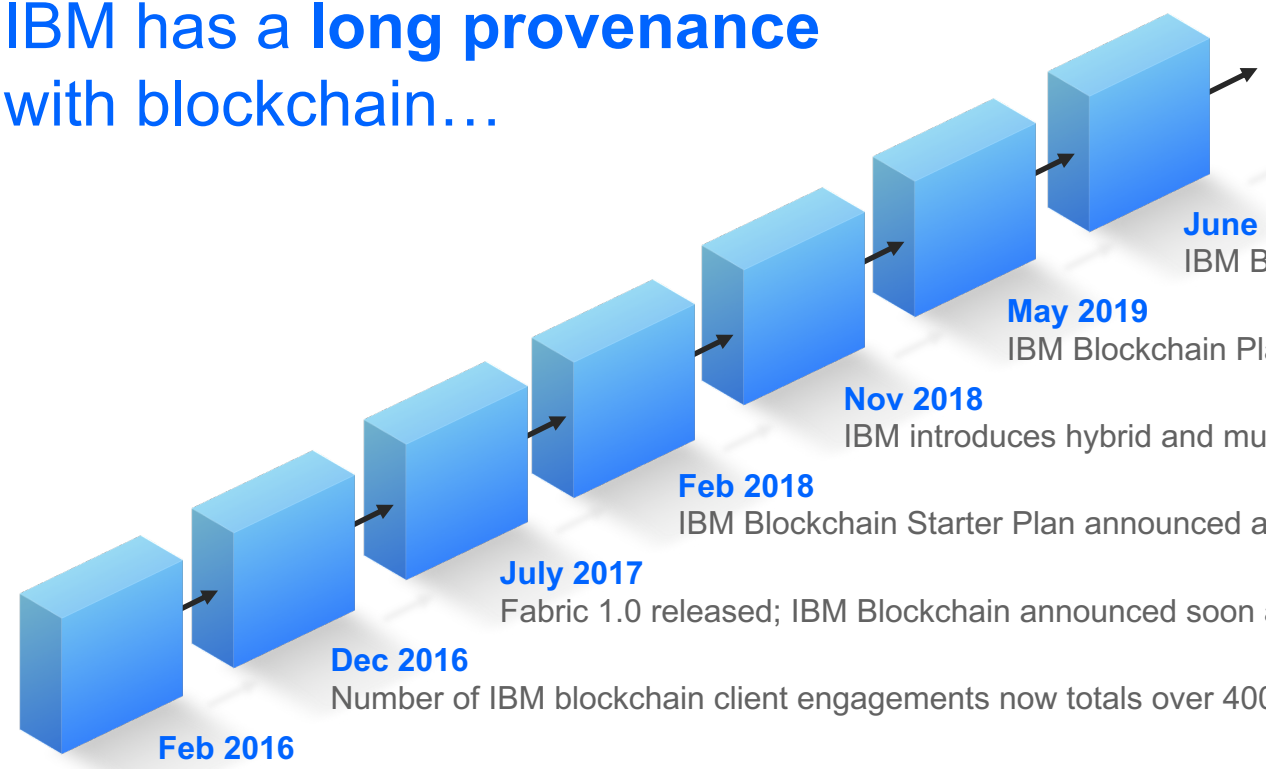
A collaborative effort created to advance cross-industry blockchain technologies for business.

- Open source, open standards and open governance model.
- Currently more than 250 members, IBM is a premier member.
- Premier, General and Associate levels of membership

- **Hyperledger contains currently 7 frameworks and 6 tools.**
  https://www.hyperledger.org/projects
  - IBM is supporting especially the development of the framework *Hyperledger Fabric*
- **Hyperledger Fabric**
  - V1.x available since July 2017
  - Currently over 250 active developers from different organizations.
  - Focus is development of ledger, smart contracts, consensus, privacy, scalability and resilience.

https://www.hyperledger.org/projects/fabric

IBM has a **long provenance** with blockchain…



June 2019
IBM Blockchain Platform for Multicloud released

May 2019
IBM Blockchain Platform for IBM Cloud released

Nov 2018
IBM introduces hybrid and multi-cloud networks

Feb 2018
IBM Blockchain Starter Plan announced at THINK

July 2017
Fabric 1.0 released; IBM Blockchain announced soon after

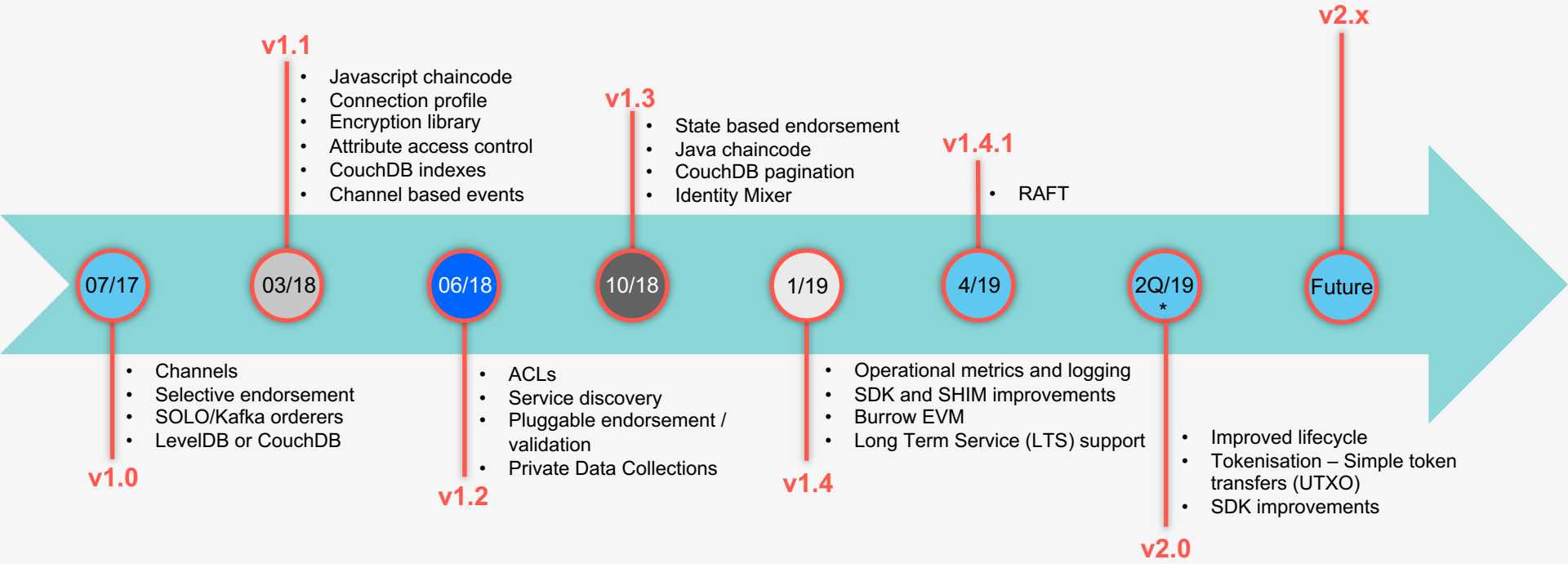
Dec 2016
Number of IBM blockchain client engagements now totals over 400

Feb 2016
IBM becomes a founding member of Linux Foundation Hyperledger; donates code and intellectual property

Aug 2015
IBM starts developing first prototype of blockchain technology (Open Blockchain); first client engagements

HYPERLEDGER FABRIC

Roadmap

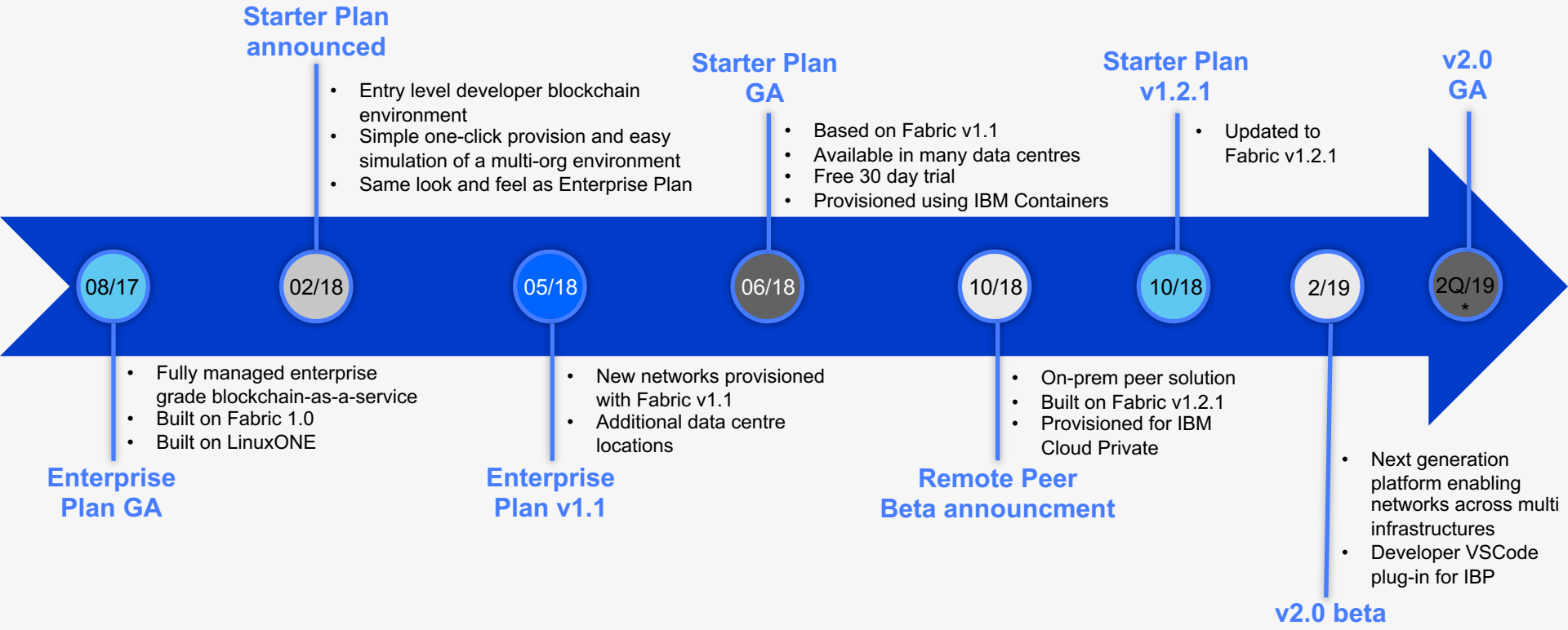


HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

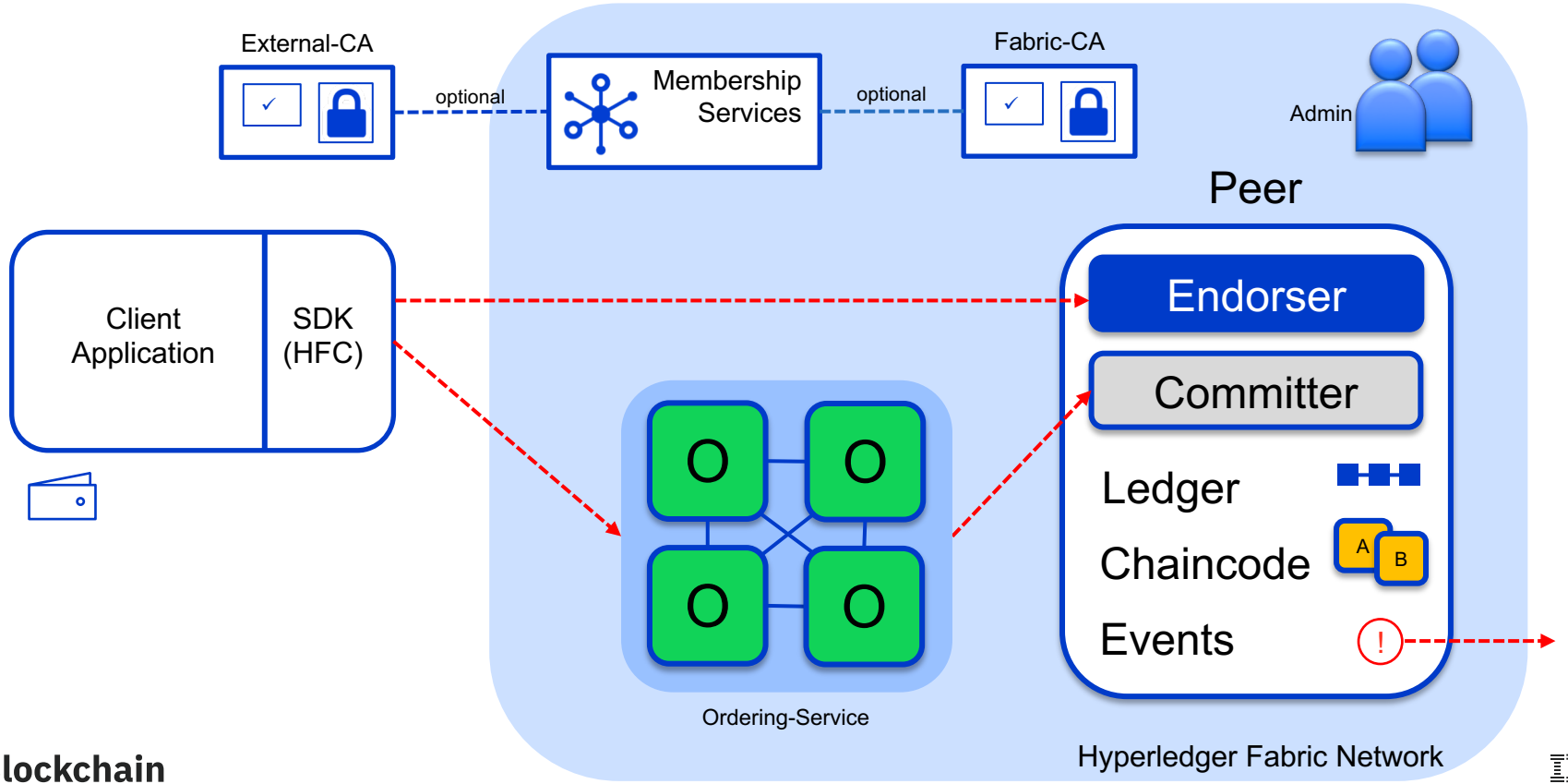
Based on https://wiki.hyperledger.org/projects/fabric/roadmap - Dates determined by the Hyperledger community - (*) Subject to change

IBM Blockchain Platform

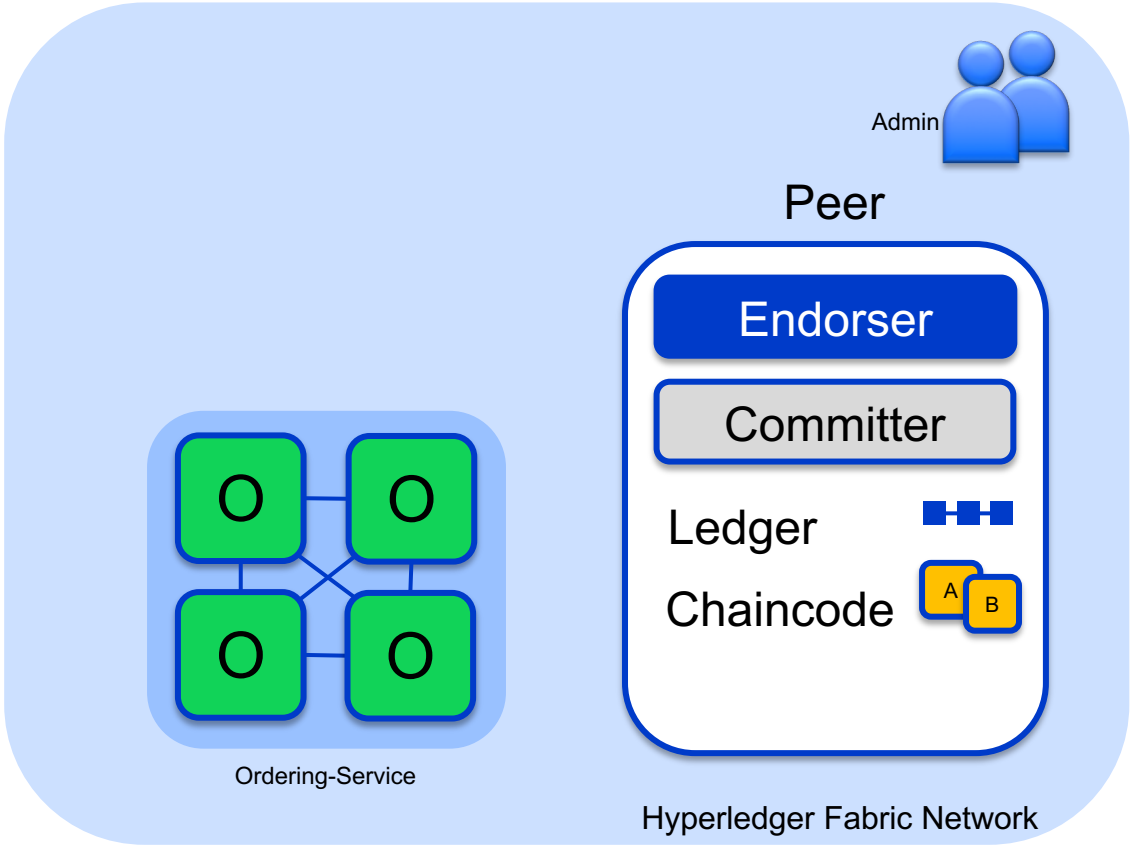
Roadmap



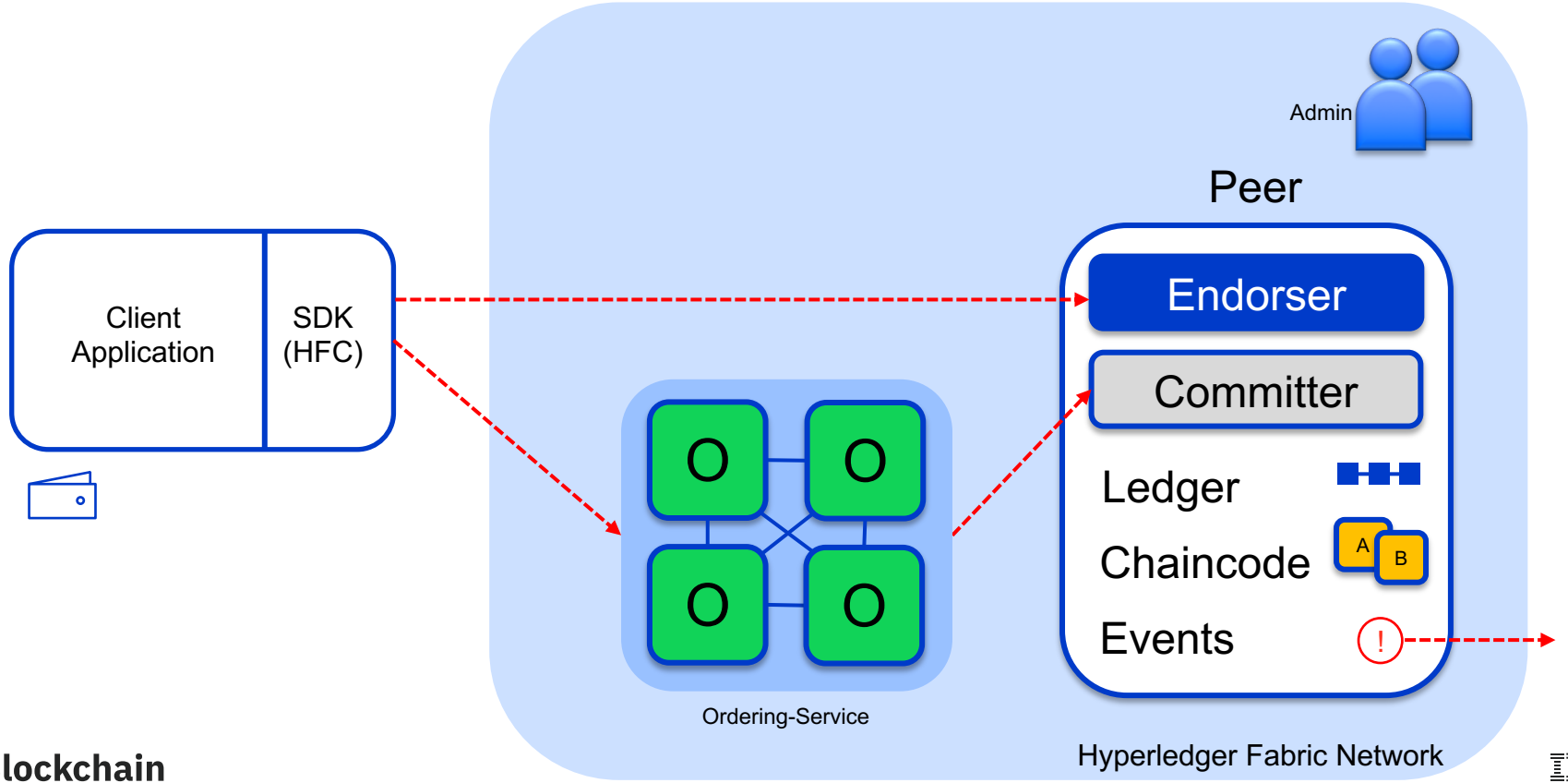
Hyperledger Fabric V1.x Architecture



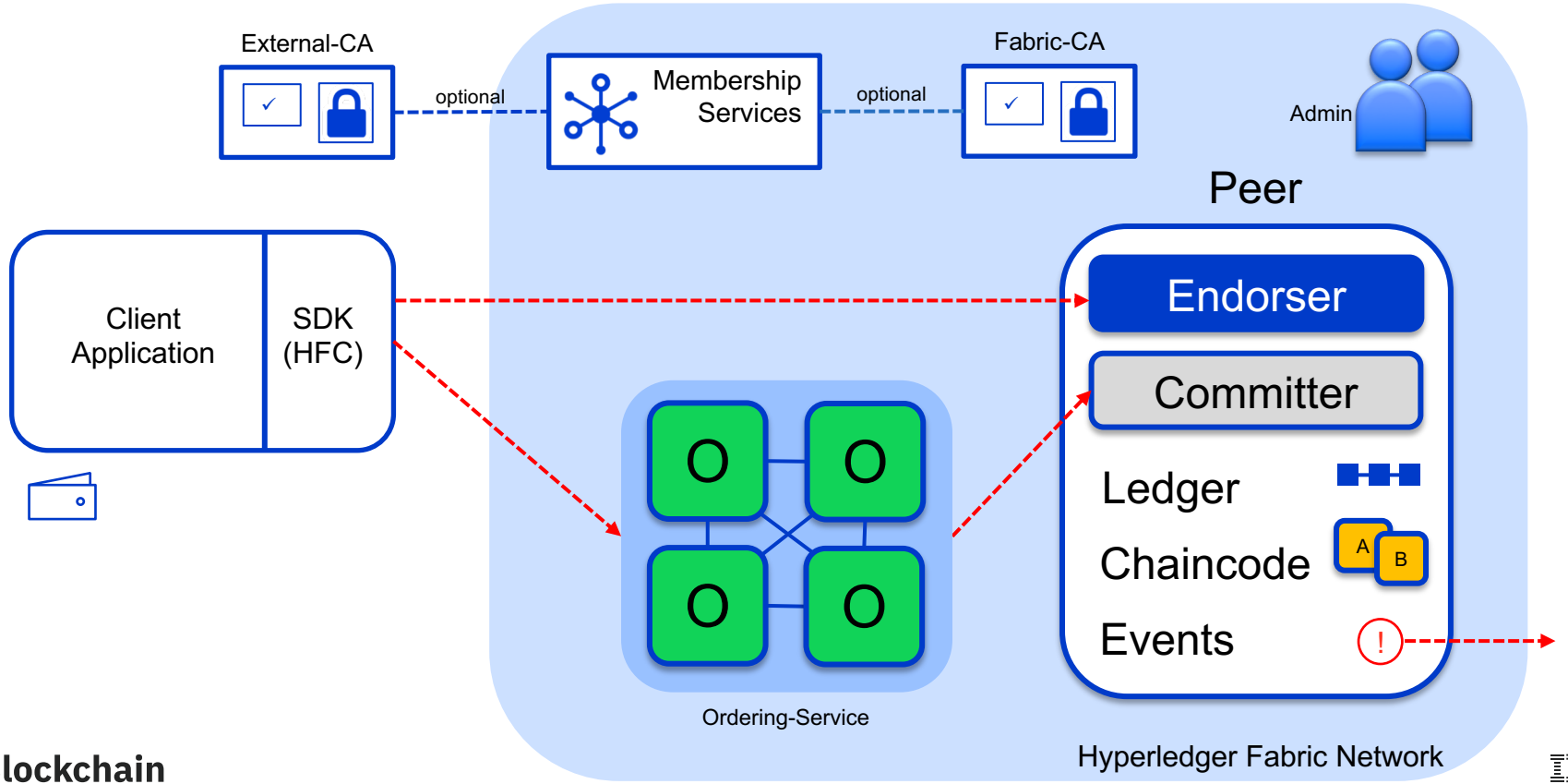
Hyperledger Fabric V1.x Architecture



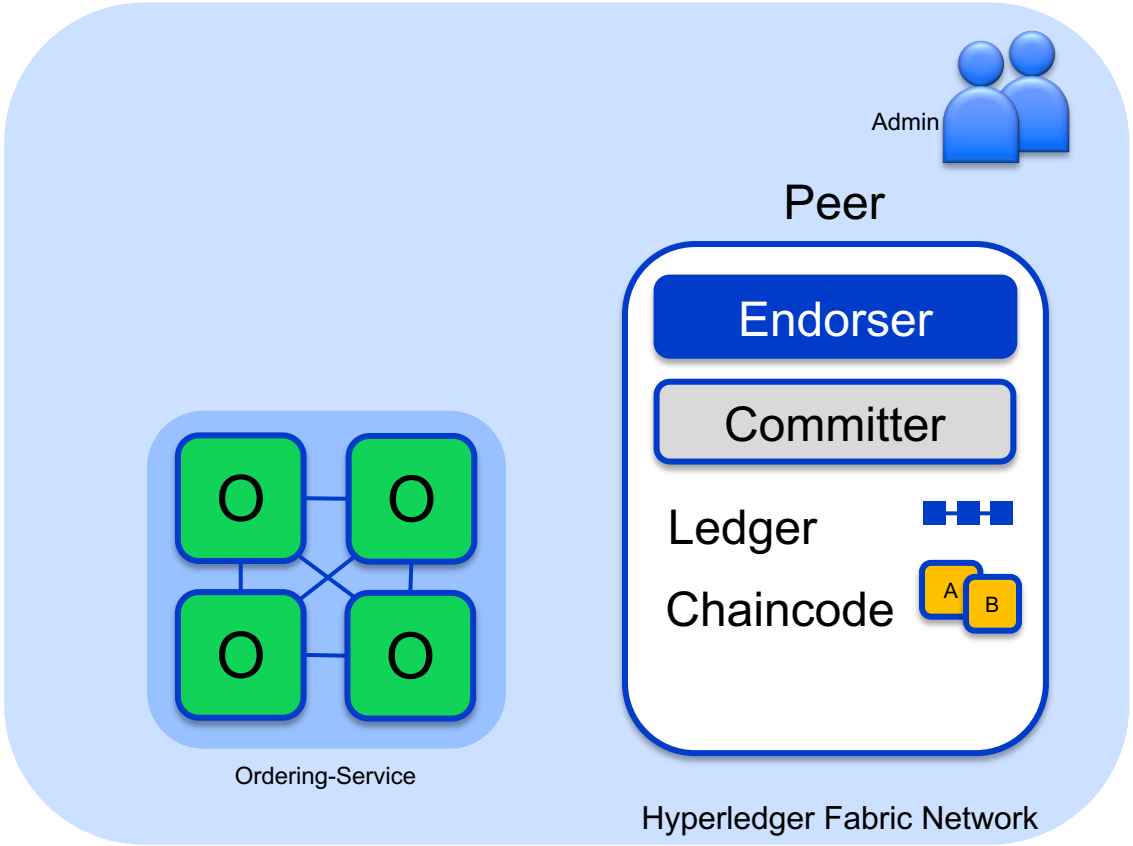
Hyperledger Fabric V1.x Architecture



Hyperledger Fabric V1.x Architecture

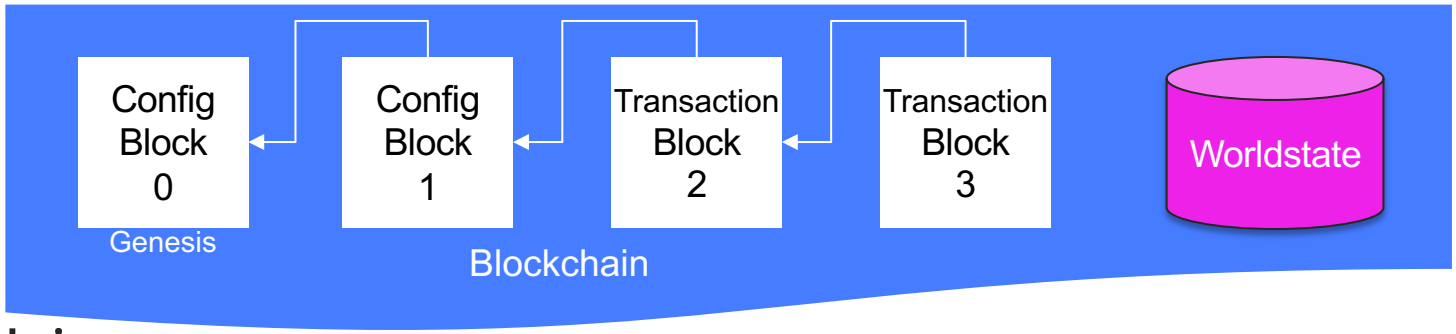


Hyperledger Fabric V1.x Architecture



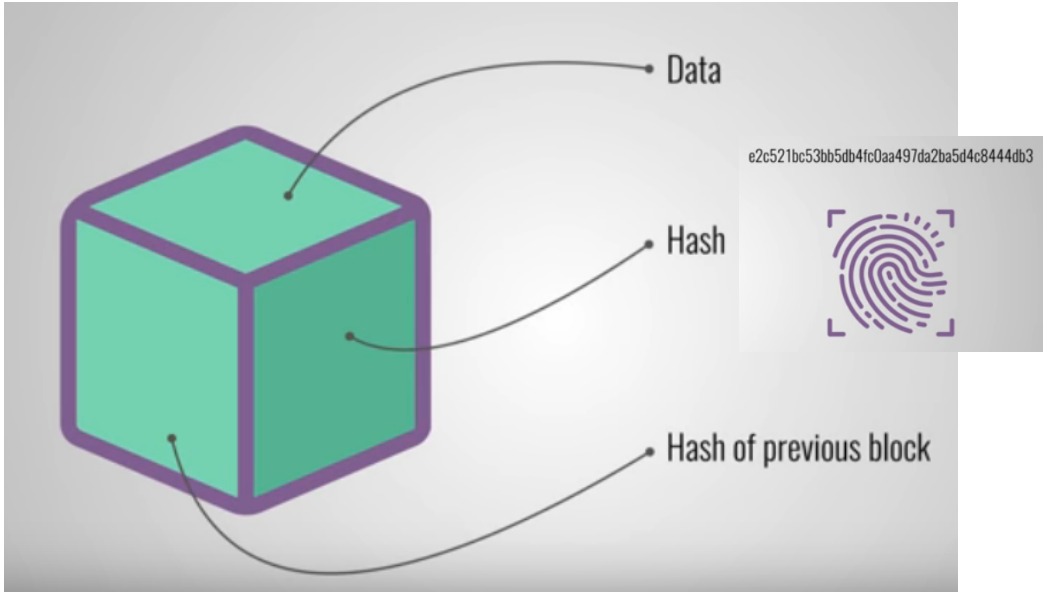
Fabric Ledger

- 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 Contract decides what is written to the worldstate

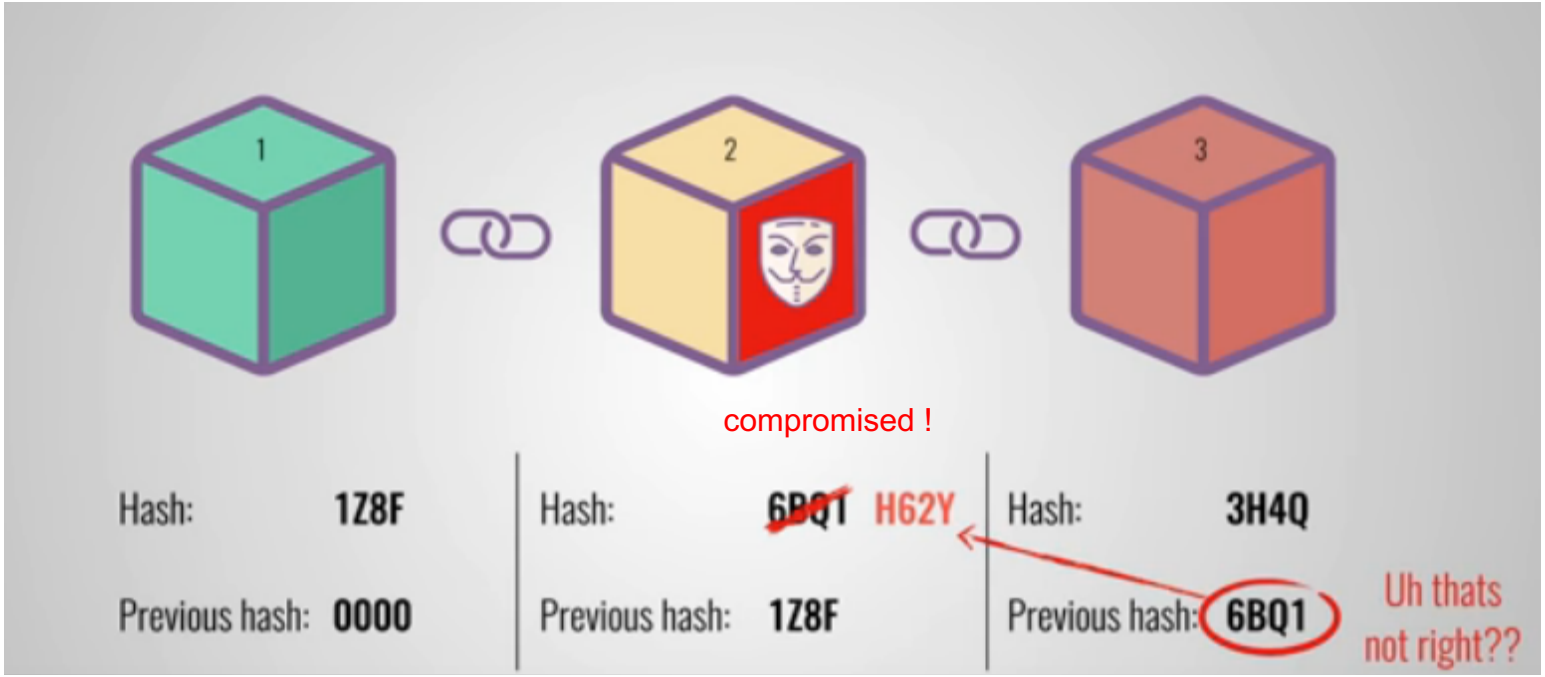


Blocks in the shared ledger

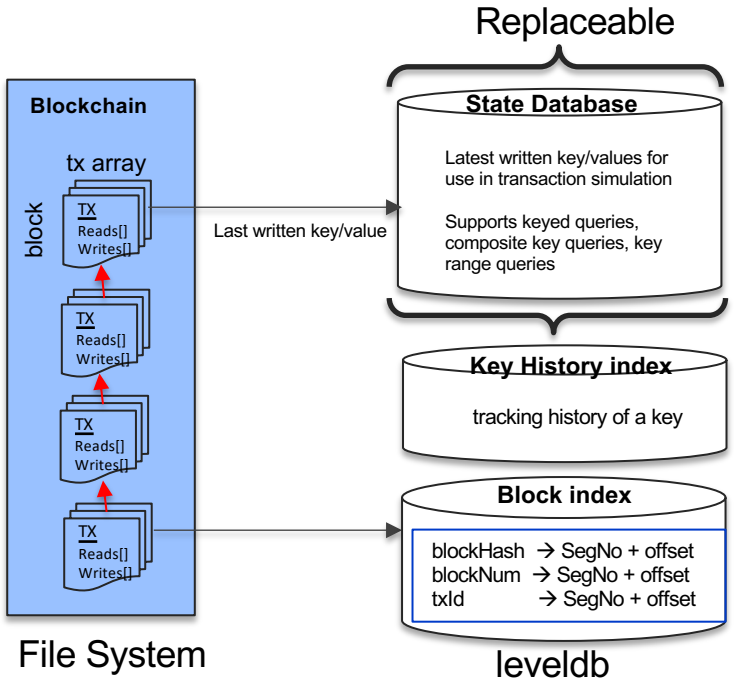
- contain the data to be stored in the blockchain
- the hash of the data
- hash of previous block



Immutability of blocks



Fabric Ledger

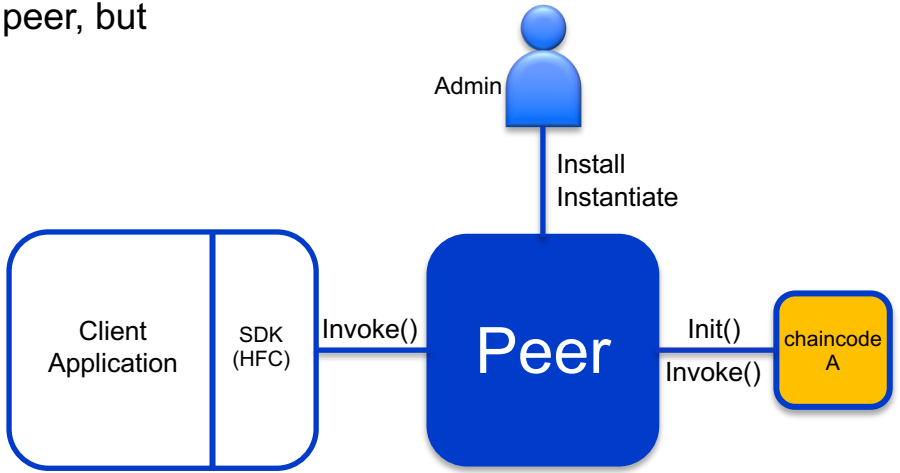


CouchDB (external option) adds rich data queries

Chaincode

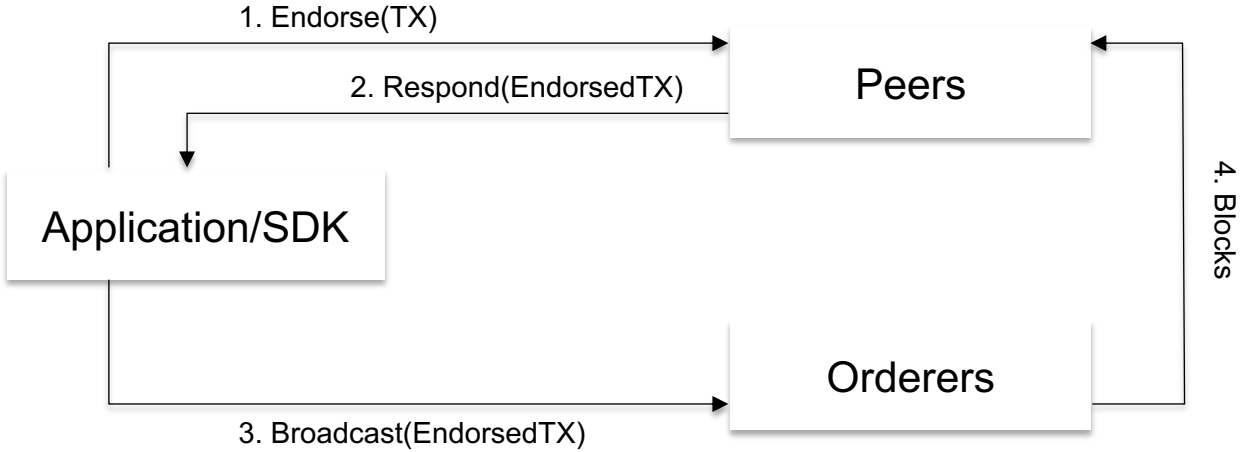
Chaincode (Smart Contract) contains business logic deployed to peers

- Installed on peers and instantiated on channels
- Run in secured docker images separate to the peer
- Interact with the world state through the Fabric shim interface
- Each chaincode has its own scoped world state
- Needs to be present to endorse a transaction from a peer, but does not need to be present to commit transactions
- Language support for:
  - Golang
  - Node.js
  - Java
- Implements:
  - Init() - Called on instantiate and upgrade
  - Invoke() – Called from client application



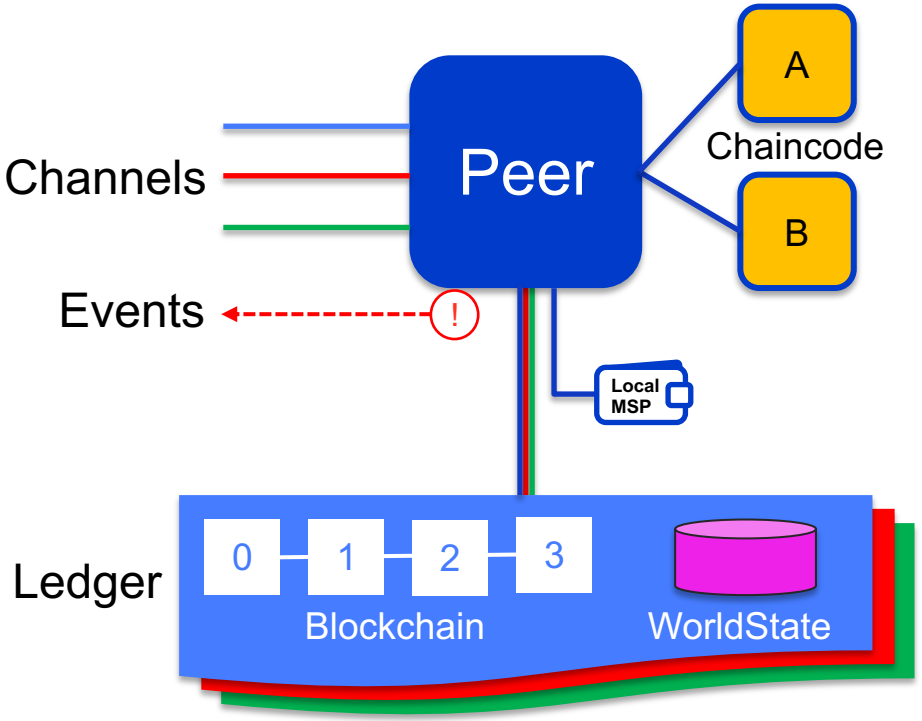
https://jira.hyperledger.org/browse/FAB-8063

Transaction data flow



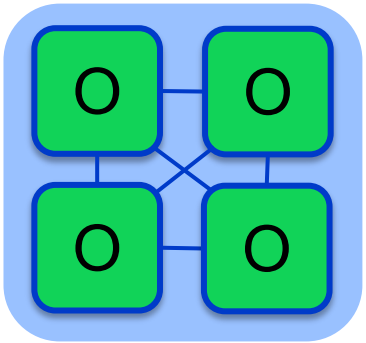
Fabric Peer

- Each peer:
  - Connects to one or more channels
  - Maintains one or more ledgers per channel
  - Maintains installed chaincode
  - Manages runtime docker containers for instantiated chaincode
    - Chaincode is instantiated on a channel
    - Runtime docker container shared by channels with same chaincode instantiated (no state stored in container)
  - Has a local MSP (Membership Services Provider) that provides crypto material
  - Emits events to the client application



Ordering Service

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



Ordering-Service

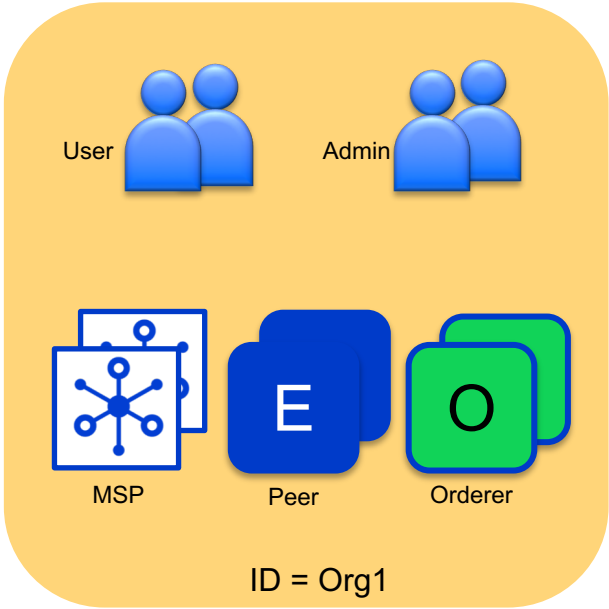
Different configuration options for the ordering service include:

- SOLO
  - Single node for development
- Kafka : Crash fault tolerant consensus
  - Minimum recommendation: 3 Orderer nodes, 4 Kafka nodes, 3 Zookeeper nodes
- Raft: Crash fault tolerant consensus
  - New with Hyperledger Fabric 1.4.1
  - Recommended

Organizations

Organizations define boundaries within a Fabric Blockchain Network

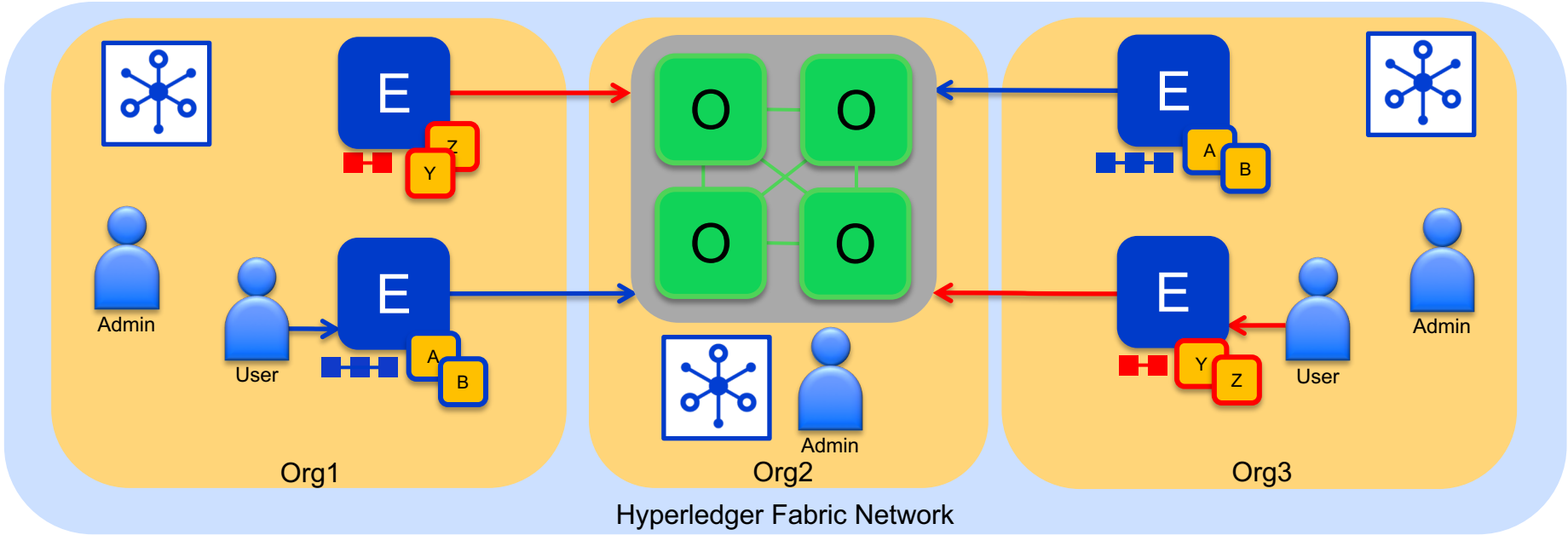
- Each organization defines:
  - Membership Services Provider (MSP) for identities
  - Administrator(s)
  - Users
  - Peers
  - Orderers (optional)
- A network can include many organizations representing a consortium
- Each organization has a unique ID



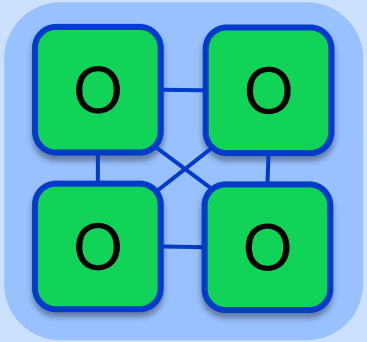
Consortium Network

An example consortium network of 3 organizations

- Orgs 1 and 3 run peers
- Org 2 provides the ordering service only



Bootstrap Network (1/6) - Configure & Start Ordering Service



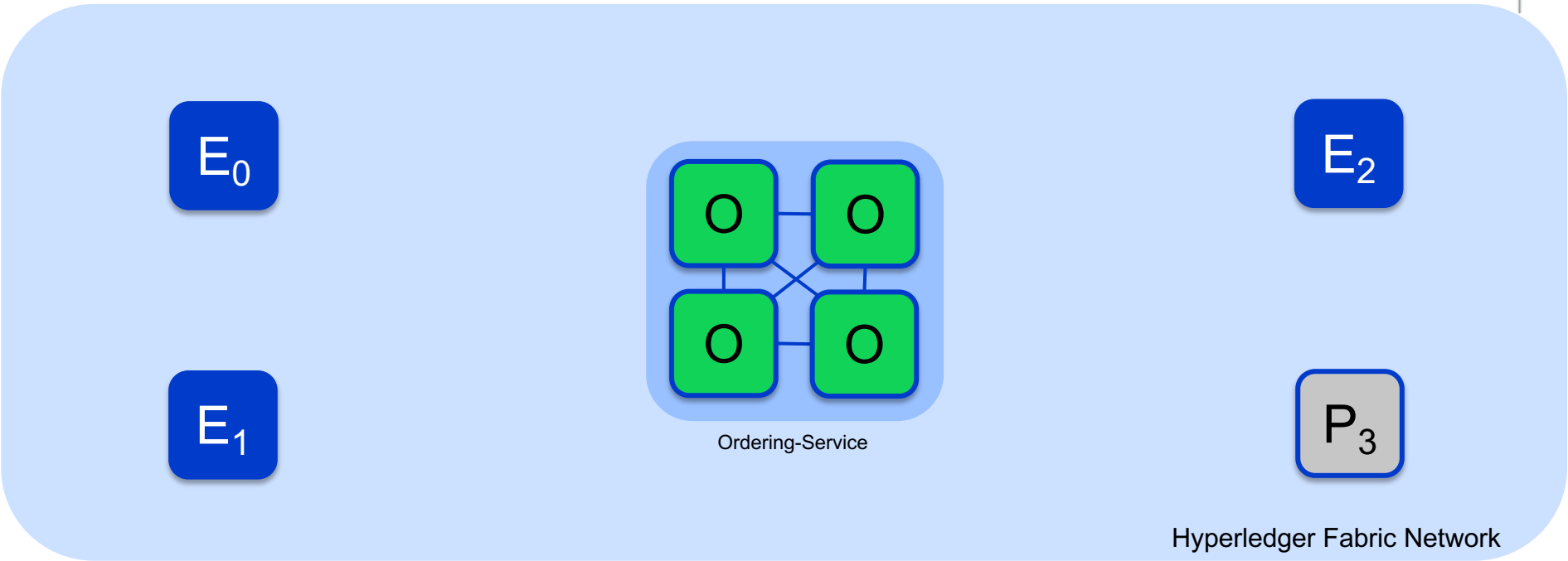
Ordering-Service

Hyperledger Fabric Network

An Ordering Service is configured and started for the network:

$ docker-compose [-f orderer.yml] ...

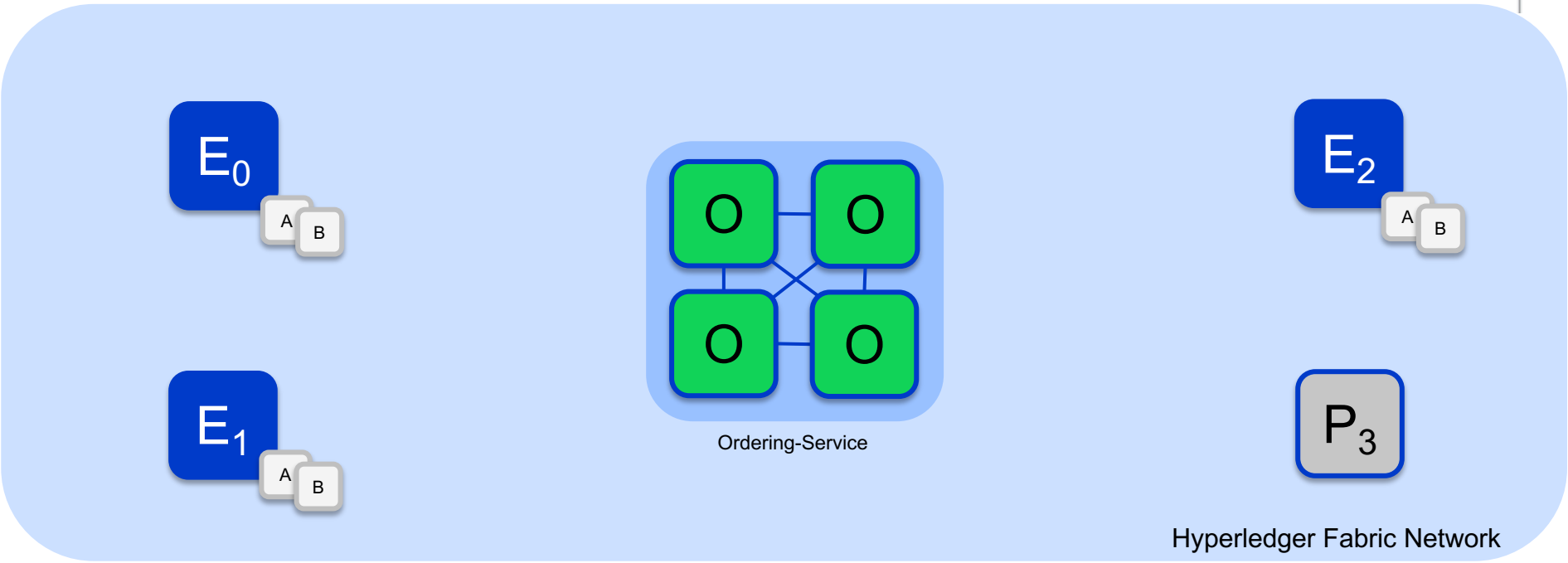
Bootstrap Network (2/6) - Configure and Start Peer Nodes



A peer is configured and started for each Endorser or Committer in the network:

$ peer node start ...

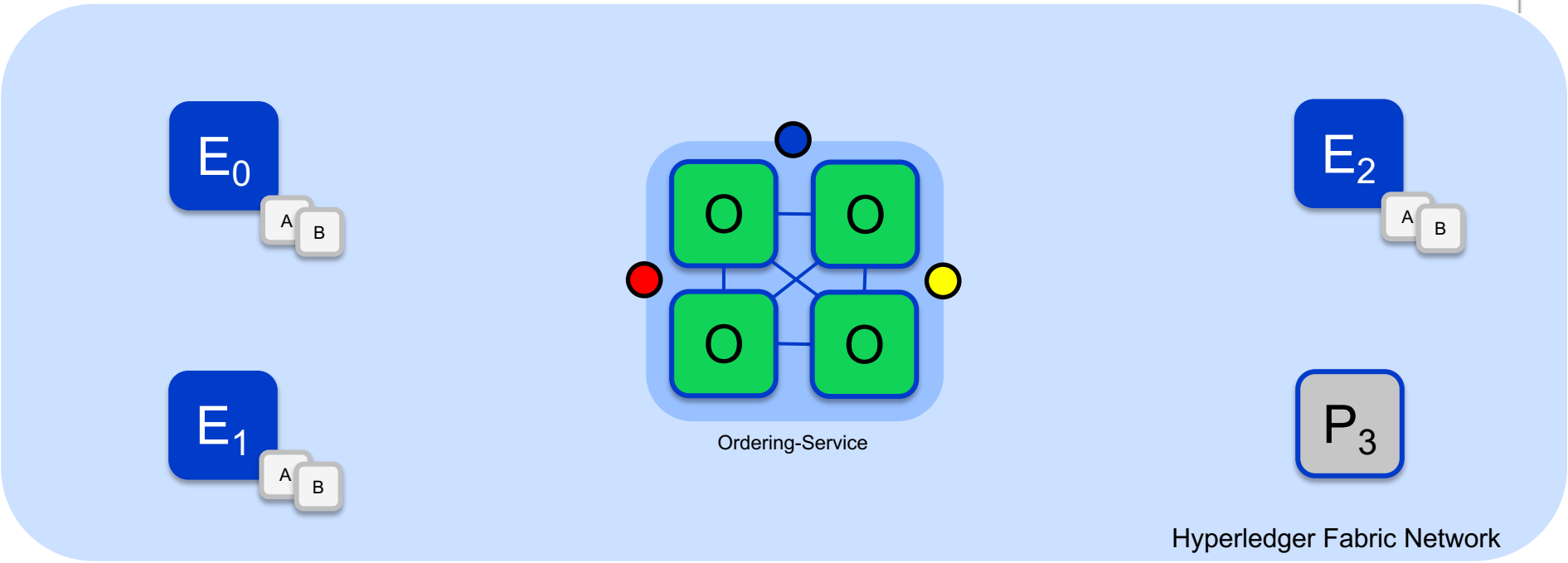
Bootstrap Network (3/6) - Install Chaincode



Chaincode is installed onto each Endorsing Peer that needs to execute it:

$ peer chaincode install ...

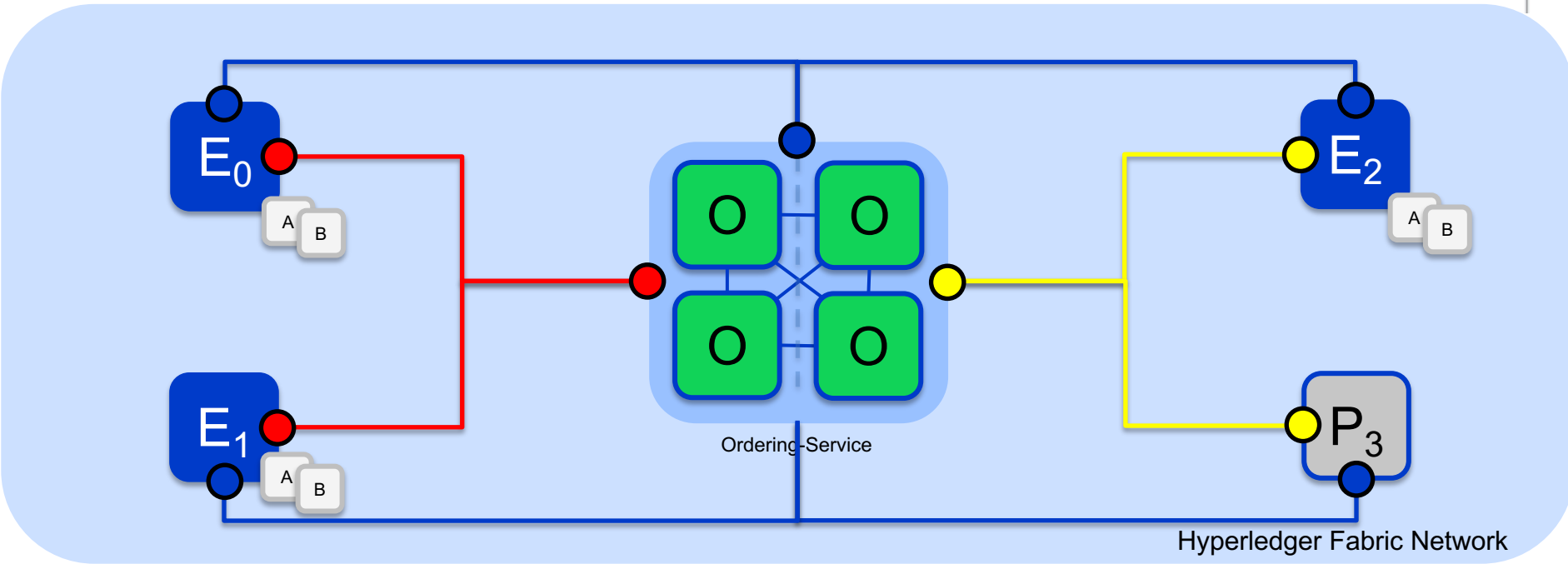
Bootstrap Network (4/6) – Create Channels



Channels are created on the ordering service:

$ peer channel create –o [orderer] ...

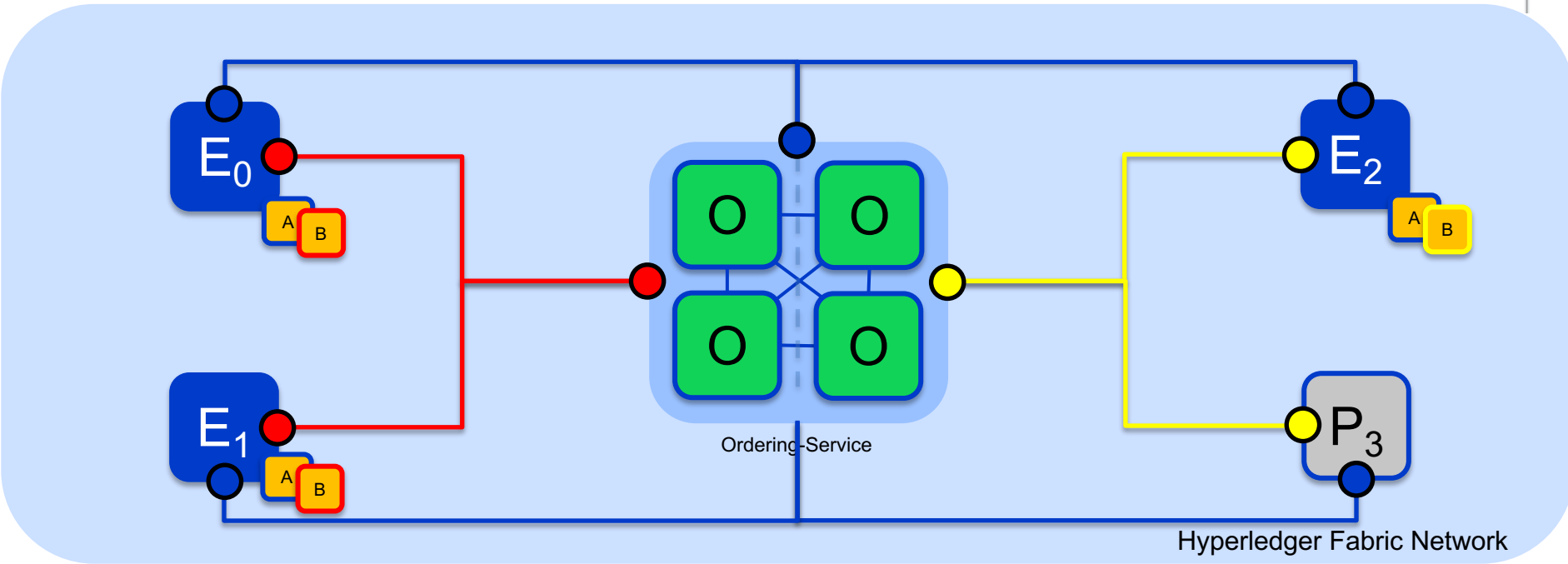
Bootstrap Network (5/6) – Join Channels



Peers that are permissioned can then join the channels they want to transact on:

$ peer channel join ...

Bootstrap Network (6/6) – Instantiate Chaincode



Peers finally instantiate the Chaincode on the channels they want to transact on:

$ peer chaincode instantiate ... –P ‘policy’

An Endorsement Policy is specified and once instantiated chaincode can process transactions.

Thank you

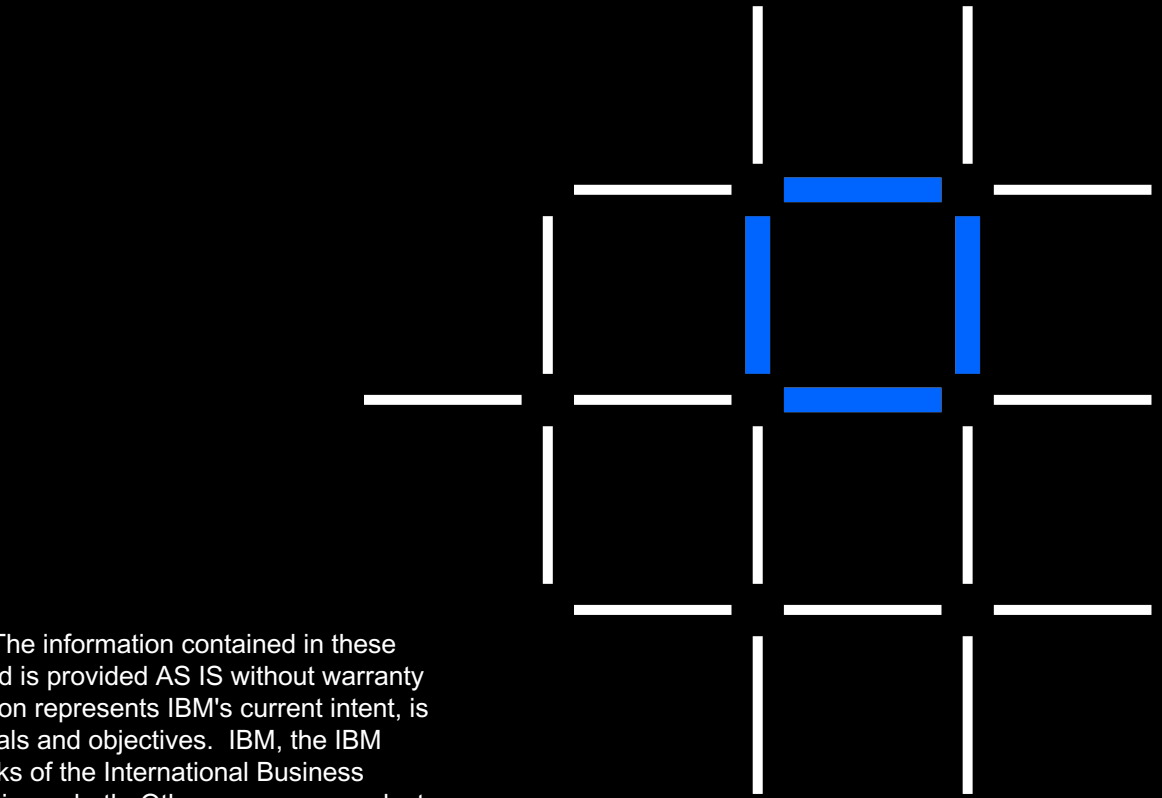
Barry Silliman
silliman@us.ibm.com

IBM **Blockchain**

www.ibm.com/blockchain

developer.ibm.com/blockchain

www.hyperledger.org



IBM


IBM