Blockchain + z: Why They're Such a Natural Fit

Volodymyr Paprotski
Blockchain Z Cryptography
IBM Canada

Please Note:

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawl without notice and at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

WHAT DOES Z MEAN TO YOU?

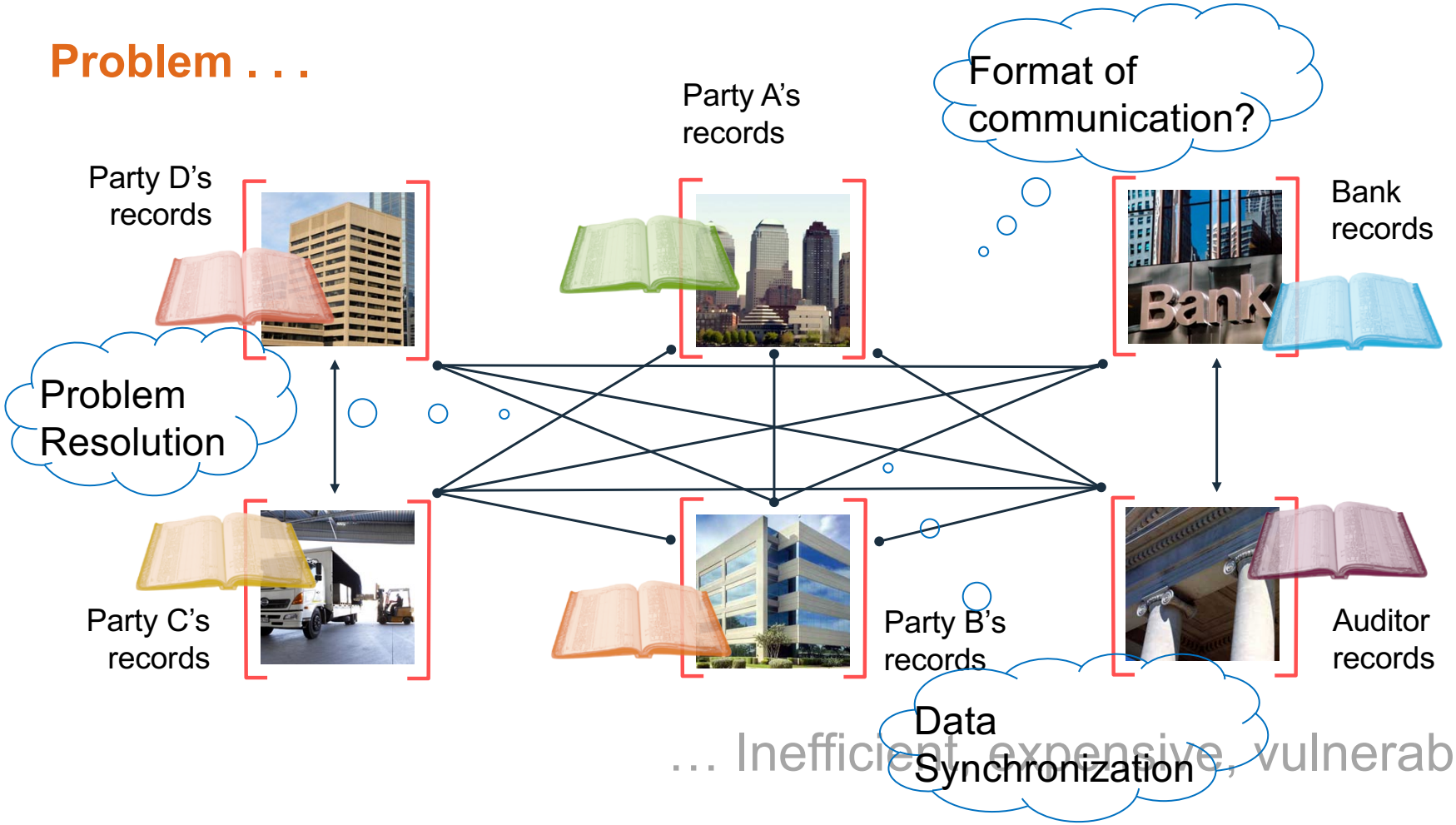
What does Z mean to you?

(Help me out here, I only been on the platform for 11+ years..)

- Transaction Processing
- Data Processing and Storage
- Security
- Availability, Reliability & Consistency
- Performance
- Compliance
- ...

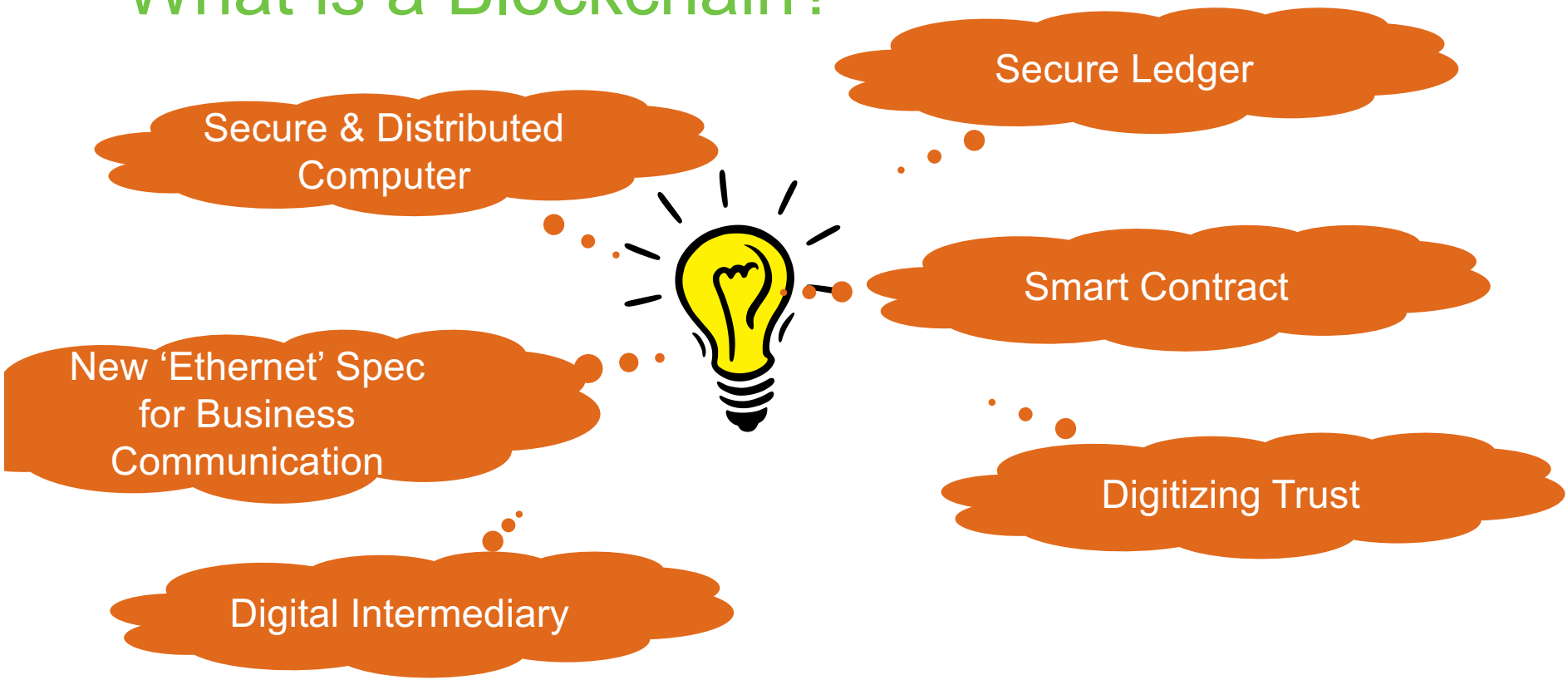
WHAT DOES FABRIC BLOCKCHAIN MEANS TO YOU?

Problem . . .

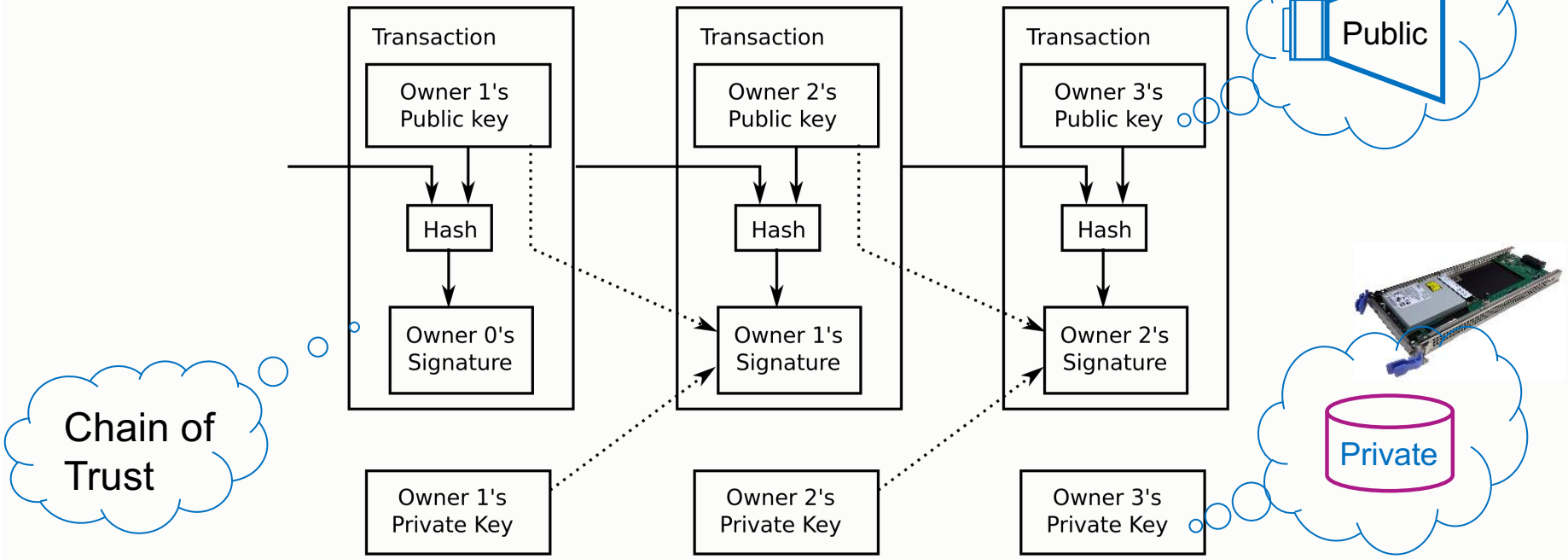


. . . Inefficient, expensive, vulnerable

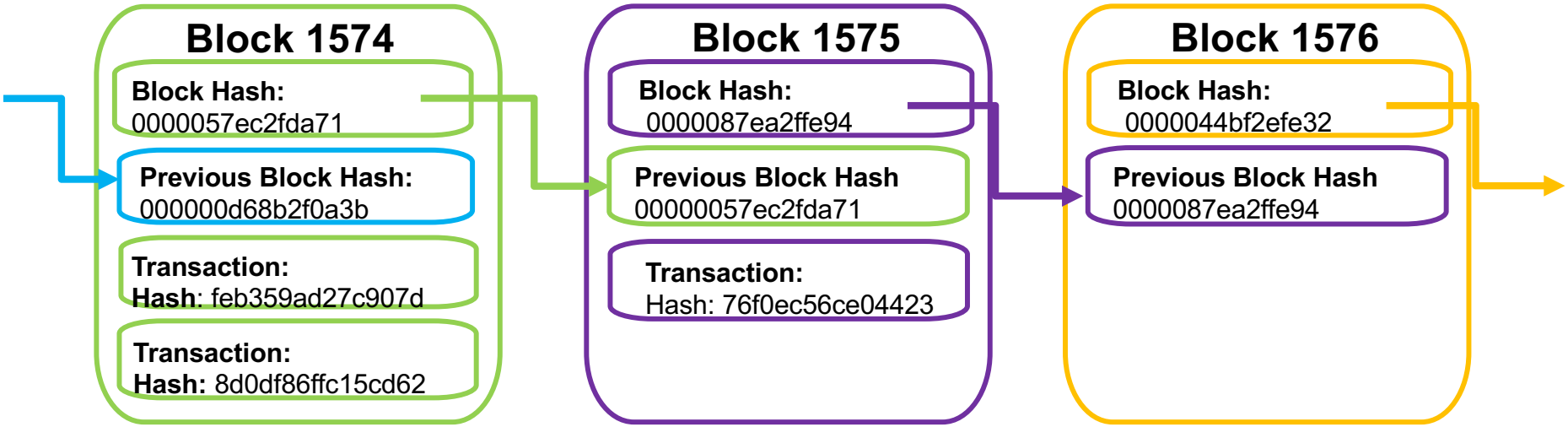
What is a Blockchain?



What is a Blockchain?

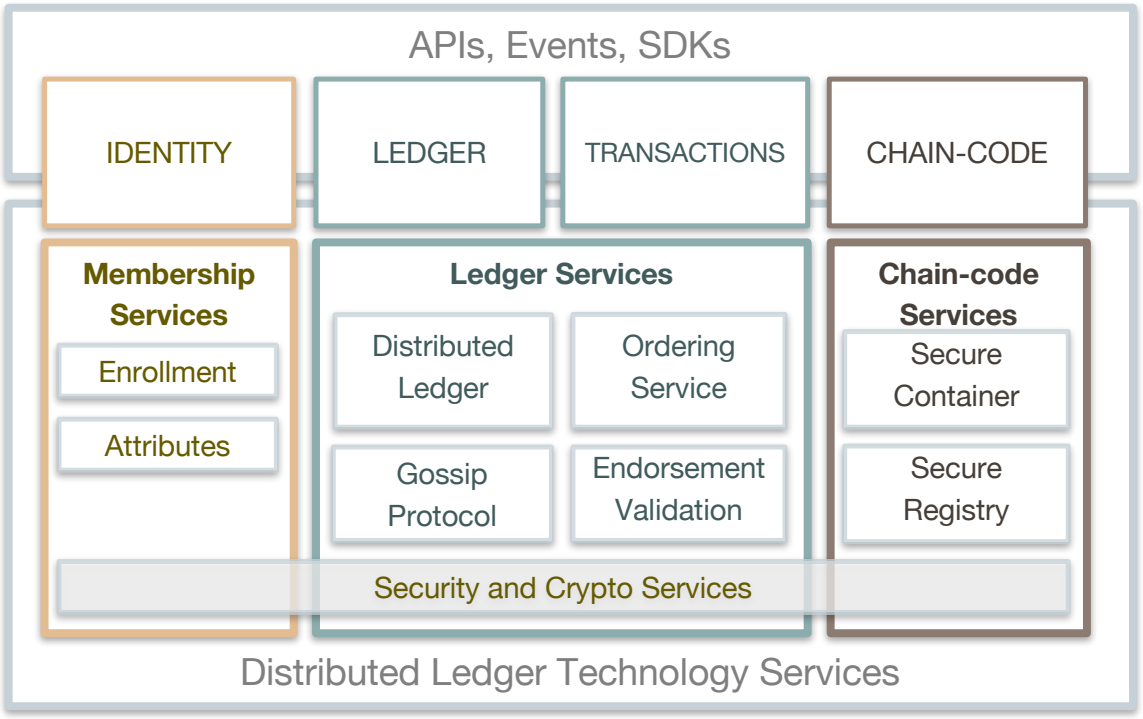


What is a Blockchain?



Made up of a series of blocks added in chronological order

Fabric Reference Architecture



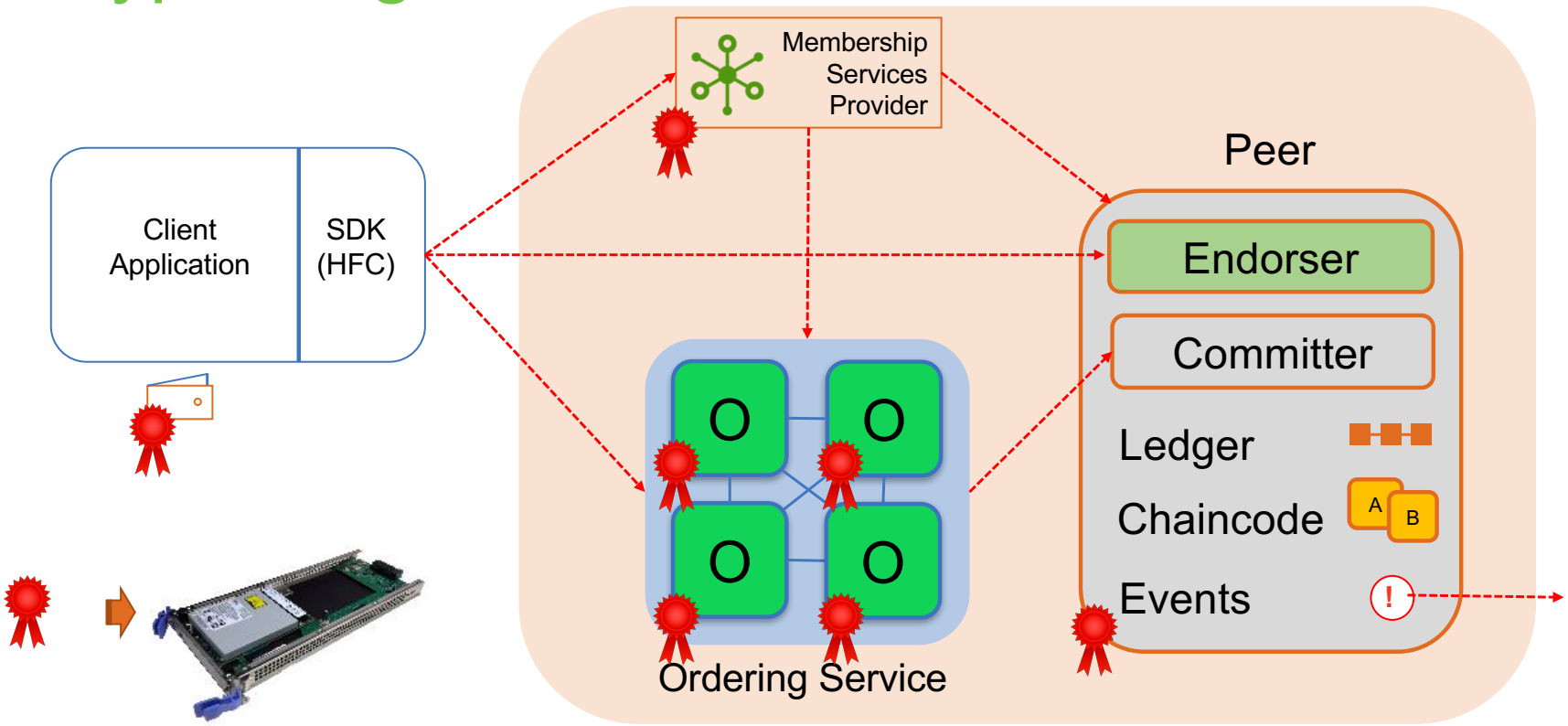
IDENTITY
Pluggable, Membership, Privacy and Auditability of transactions.

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

CHAIN-CODE
“Programmable Ledger”, provide ability to run business logic against the blockchain

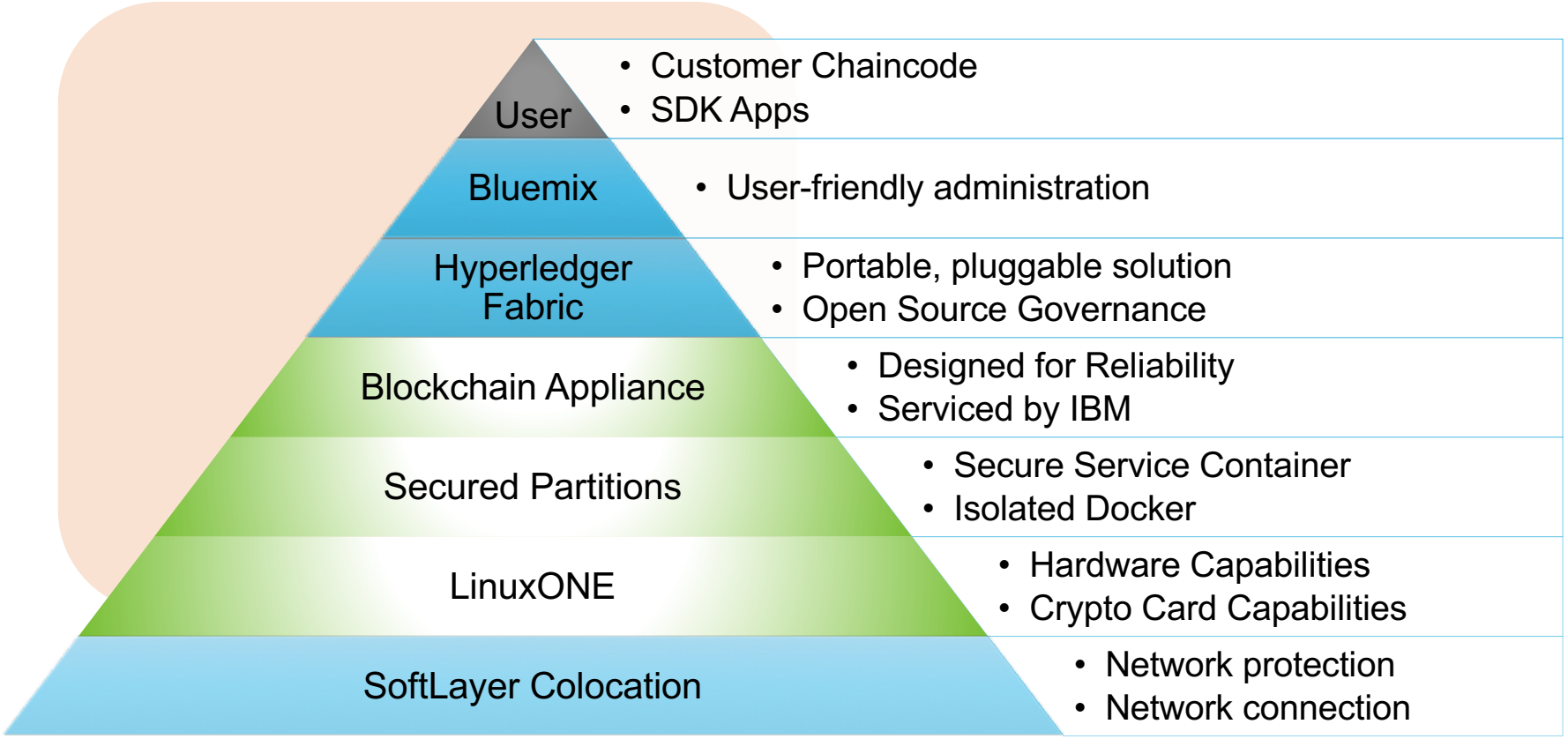
APIs, Events, SDKs
Multi-language native SDKs allow developers to write DLT apps

Hyperledger Fabric V1 Architecture



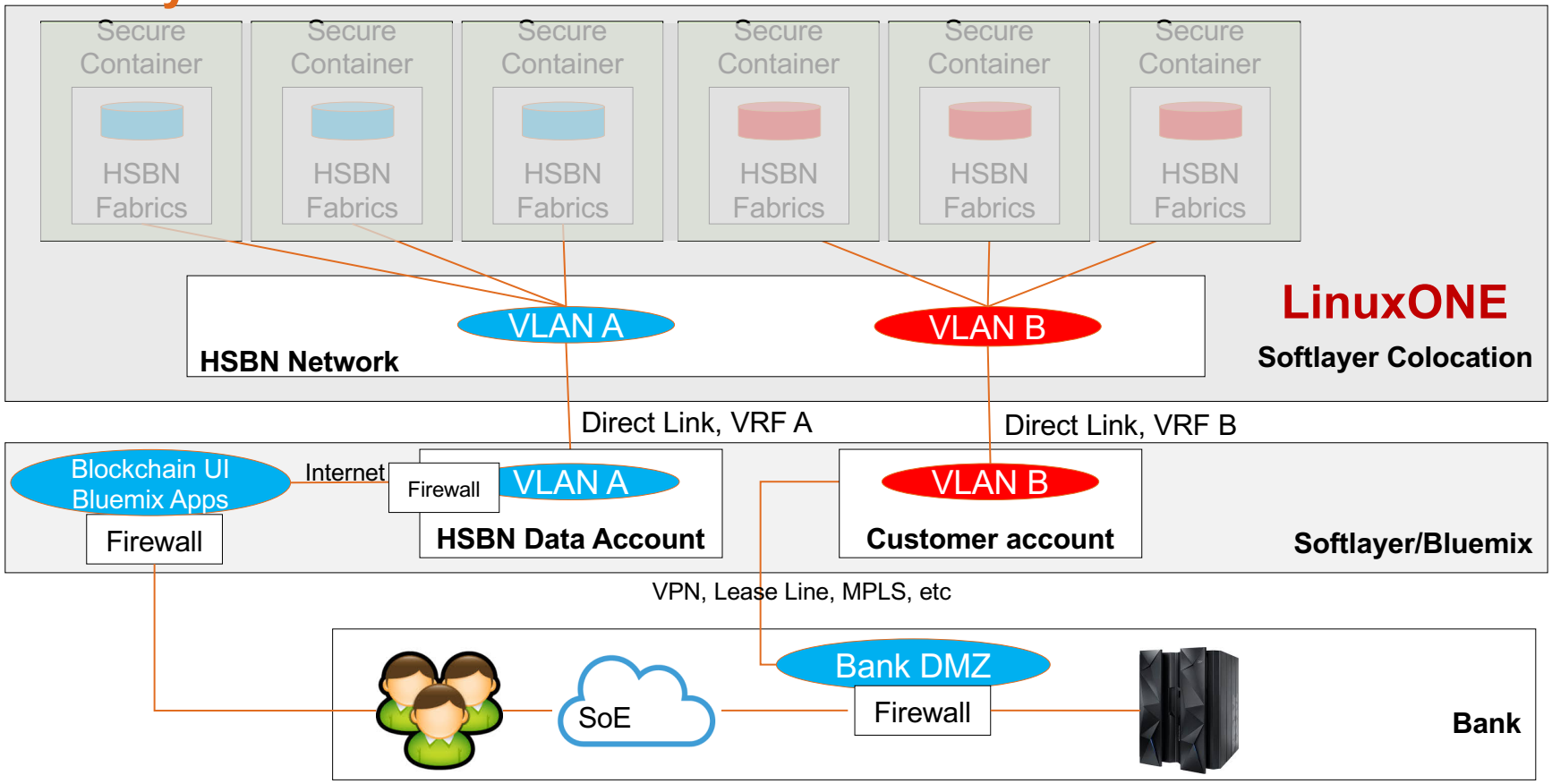
WHAT DOES “IBM BLOCKCHAIN PLATFORM” MEAN TO YOU?

Why IBM Blockchain Platform?



IBM Blockchain Platform Sites

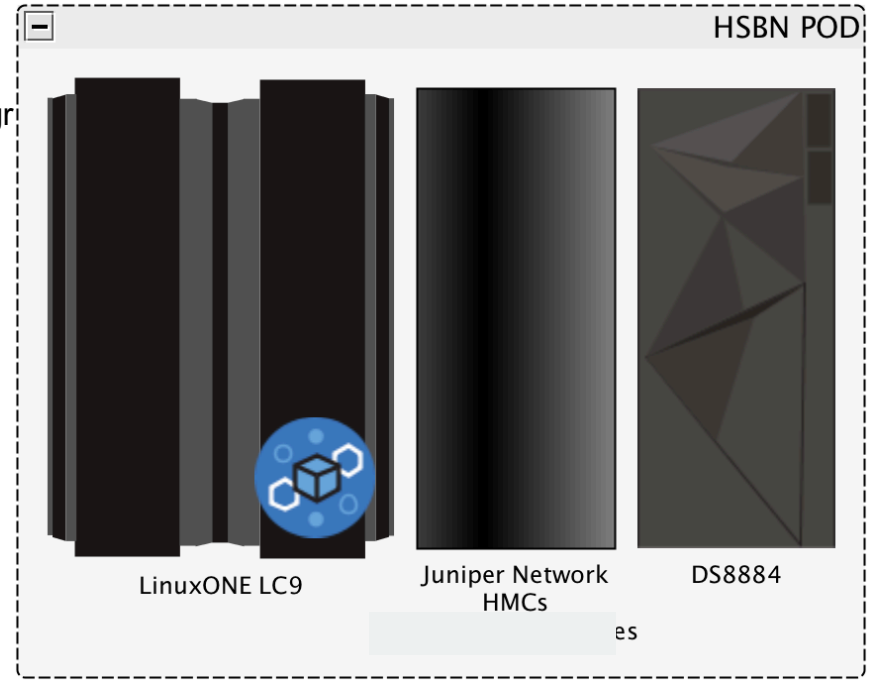
SoftLayer: Access methods



IBP Co-location Pod

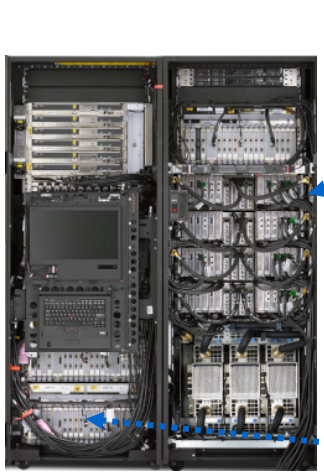
System Support Rack:
- 2x Juniper QFX 5100 Switches
- 2x2x16 IBM Global Console Mgr

LinuxOne – Mod LC 9:
- 4 Drawers, 129 IFLs
- 6 TB Memory
- 16 x OSA cards (mix)
- 10x16Gb Ficon Express
- 4 x Crypto 5S
- Internal Battery Feature
- 2 x Rack Mounted HMC

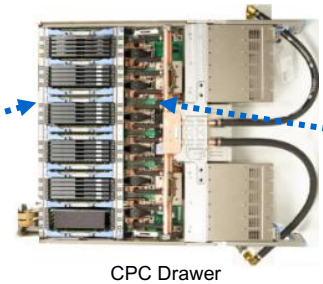


DS 8884 – Mod 984:
- 128GB Proc. Memory
- 4 x 4port 16GB Ficon
- 2.4TB Flash
- 320TB of HDD
- CSM for Back/Restore Flashing

Overview – HW Crypto Support in LinuxONE

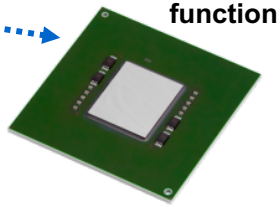


PCIe I/O drawers



CPC Drawer

PU SCM
Each PU is capable of having the **CPACF function**



Crypto Express 5S

Designed to be FIPS 140-2 Level 4

TKE required for management of Crypto Express5S and EP11

Trusted Key Entry (TKE)

Smart Cards

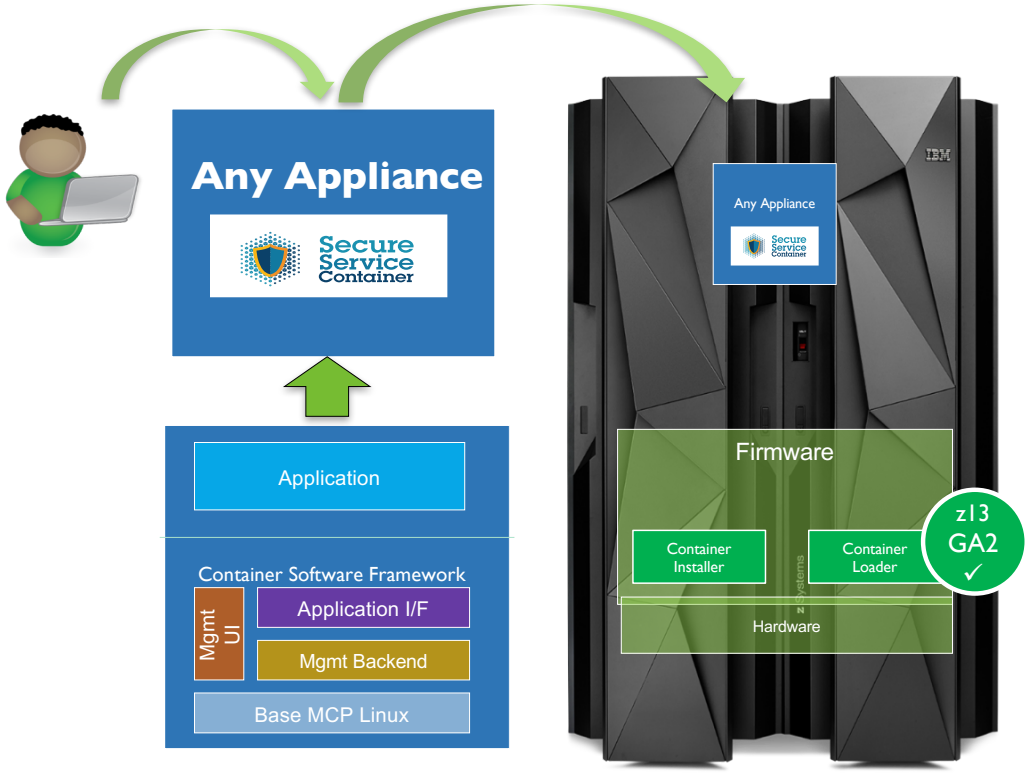
Smart Card Readers

Secure Service Container

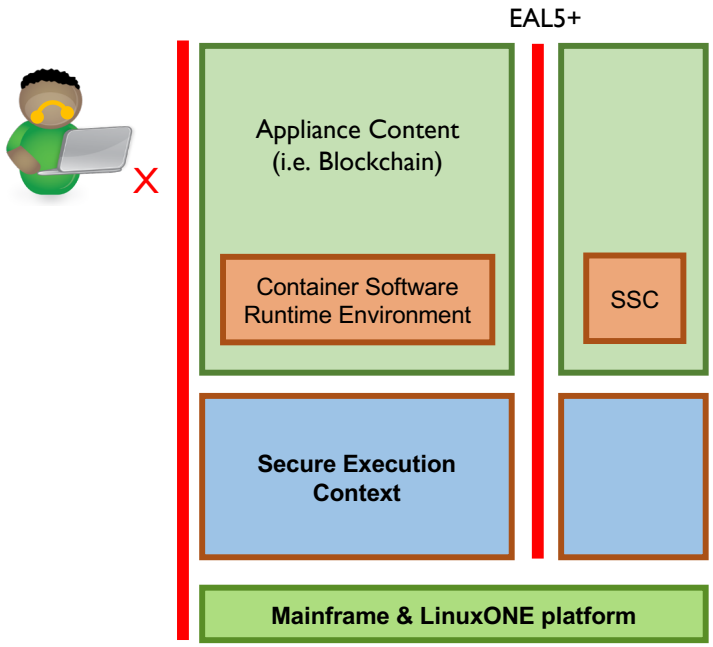
- The Base Infrastructure to Host and Build Software Appliances

- **Easy Installation**: Provides simplified mechanism for fast deployment and management of appliance-based solutions
  - O/S, Application, Services packaged as single solution
- **Highly consumable**: Manage the appliance through Remote, RESTful, APIs and web interfaces
- **Secure Runtime**: Provides tamper protection during appliance installation and runtime
- **Data Privacy**: Ensures confidentiality of data and code running within the Appliance – both in-flight and at rest
- **A Software Distribution**: Enables Appliances to be delivered via software distribution channels vs hardware – including maintenance

Secure Service Container Framework Overview

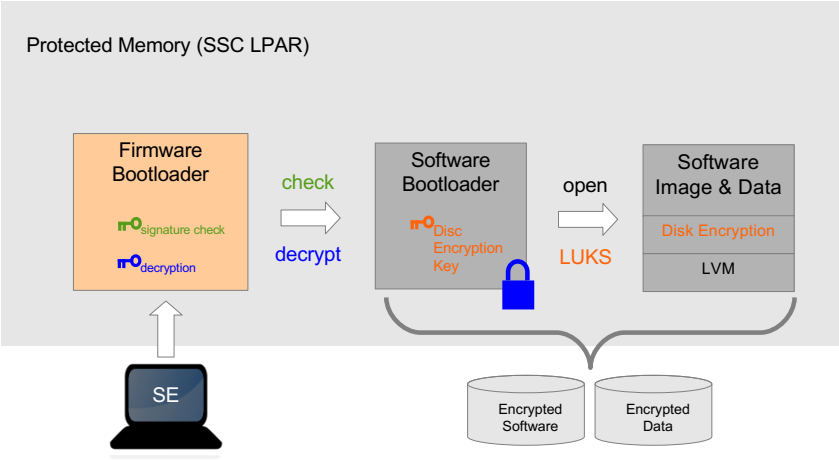


Secure Service Container Protection



- No system admin access
  - Once the appliance image is built, OS access (ssh) is not possible
    - Only Remote APIs available
  - Memory access disabled
  - Encrypted disk
  - Debug data (dumps) encrypted
- Strong isolation between container instances
  - Based on LinuxONE EAL5+ protection profile
  - Requires dedicated HW

SSC Boot Sequence



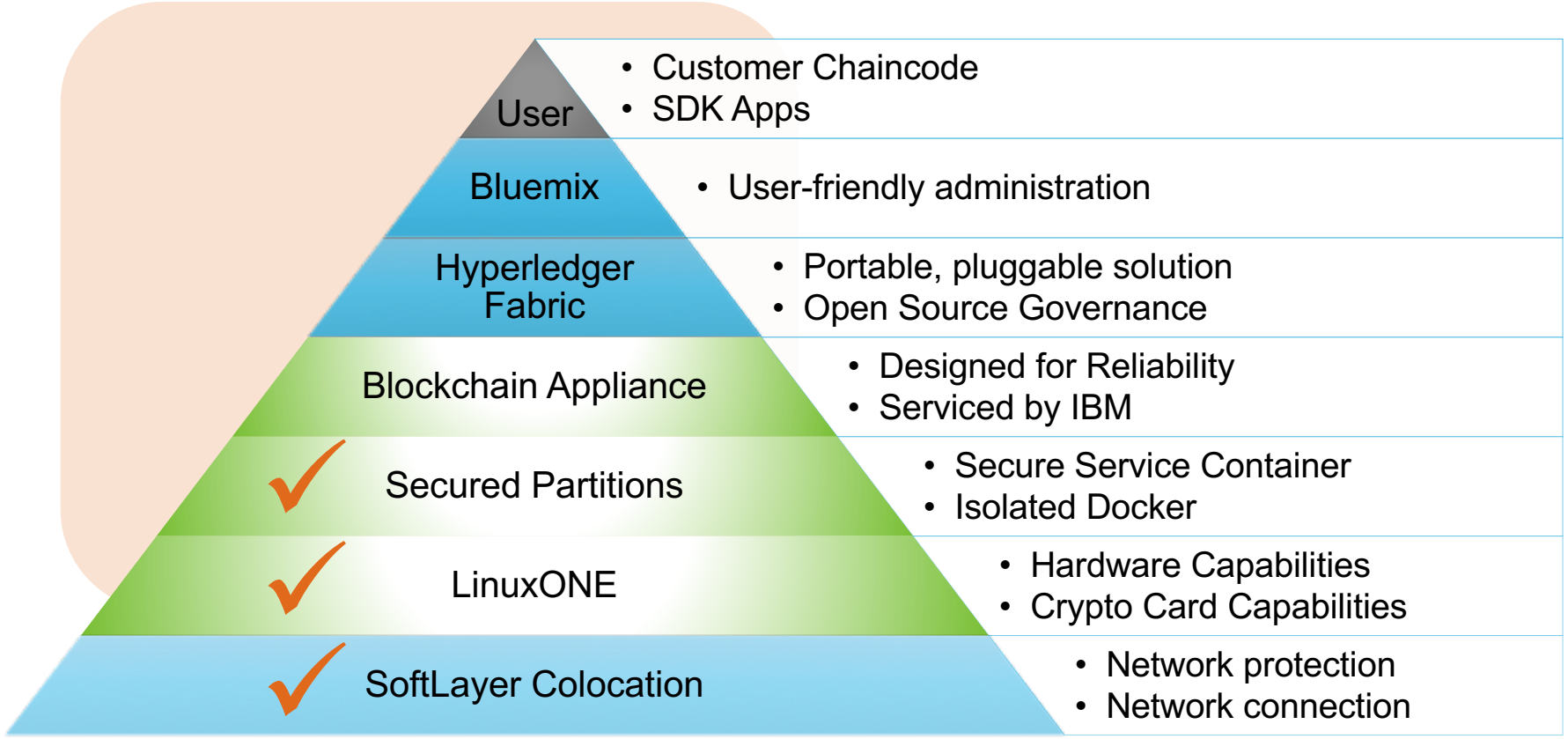
1. Firmware bootloader is loaded in memory

2. Firmware loads the software bootloader from disk
   - Check integrity of software bootloader
   - Decrypt software bootloader

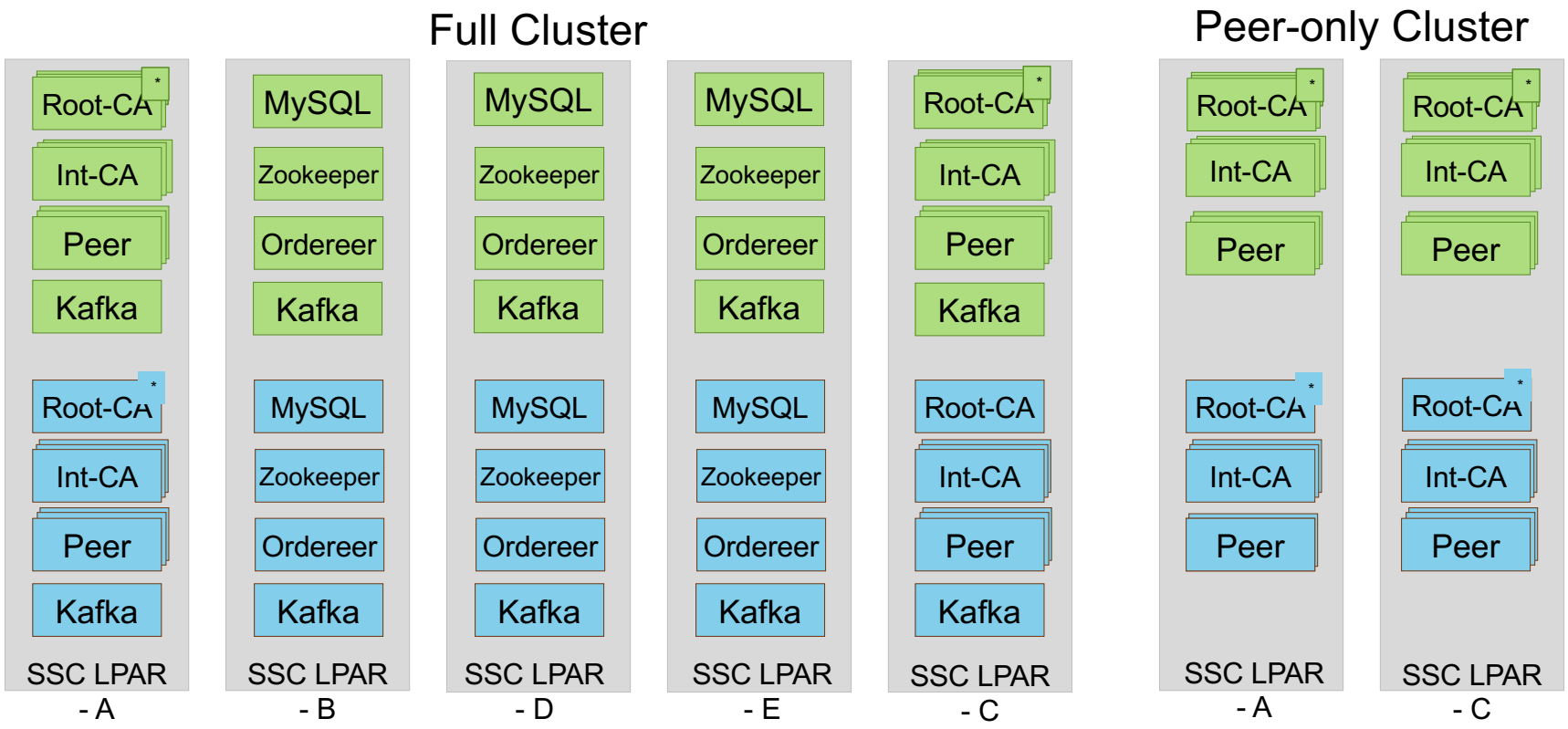
3. Software bootloader activate encrypted disks
   - Key stored in software bootloader (encrypted)
   - Encryption/decryption done on the flight when accessing appliance code and data

4. Appliance designed to be managed by remote APIs only
   - REST APIs to configure Linux and apps
   - No ssh (allowed in dev mode)

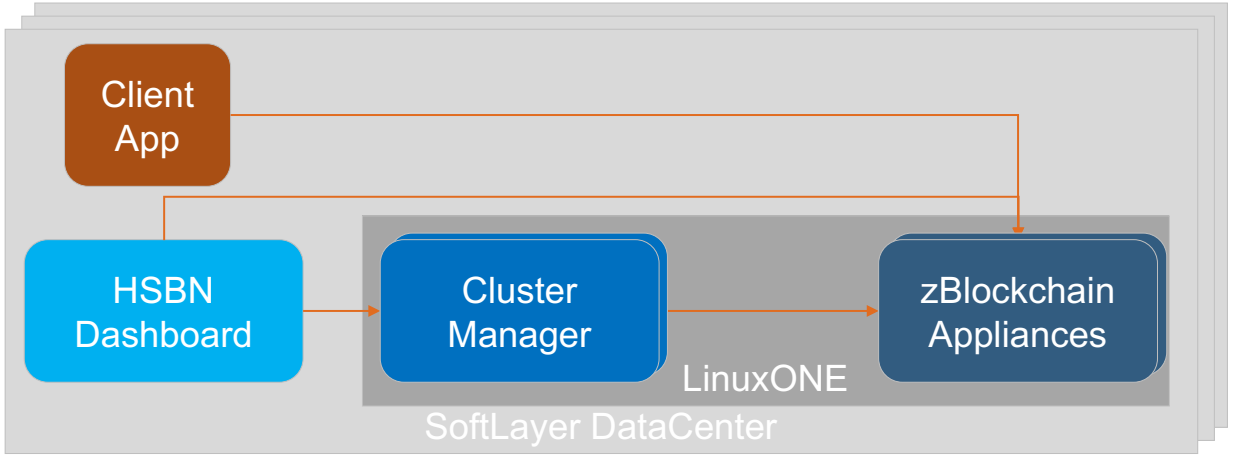
Why IBM Blockchain Platform?



Clusters - Internal topology



Blockchain Appliance Overview



Reliability: Clustering Overview

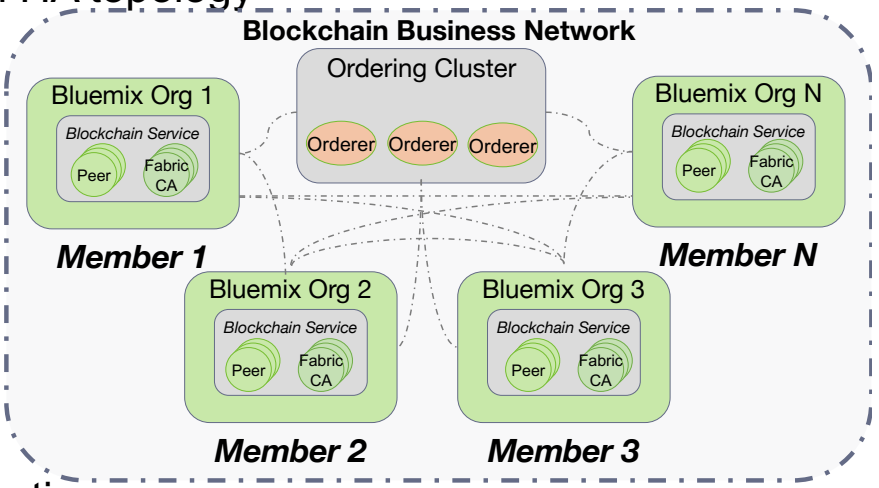
- Distribute nodes over 5 LPARs
  - Remove having one LPAR as single point of failure
  - Create a Fabric Cluster
- Flexible number of nodes
  - Any number of node packs can be added to cluster
  - CA pack (2x nodes), Endorser pack (2x nodes), Ordering Pack (3x nodes)
- Multiple Fabric Clusters:
  - Multiple HSBNs per cluster or Dedicated Cluster
  - Additional HSBN T-Shirt sizing for Peer Nodes

Levels of Data Redundancy

- **The storage unit** (DS8K) uses RAID6 on top of its physical drives in order to provide its logical disks (ECKD volumes) to the LPARs
- All the **SSC disks** are backed up every day via storage flashing
  - Two backups are kept (but can be modified)
- Within the SSC, **each container** will be snapshot on a regular basis
  - Each node can be recovered to a previous state
- A crash of an LPAR does not affect **the fabric**
  - Data is duplicated over the nodes – shared ledger
  - Remaining nodes are enough to operate the fabric

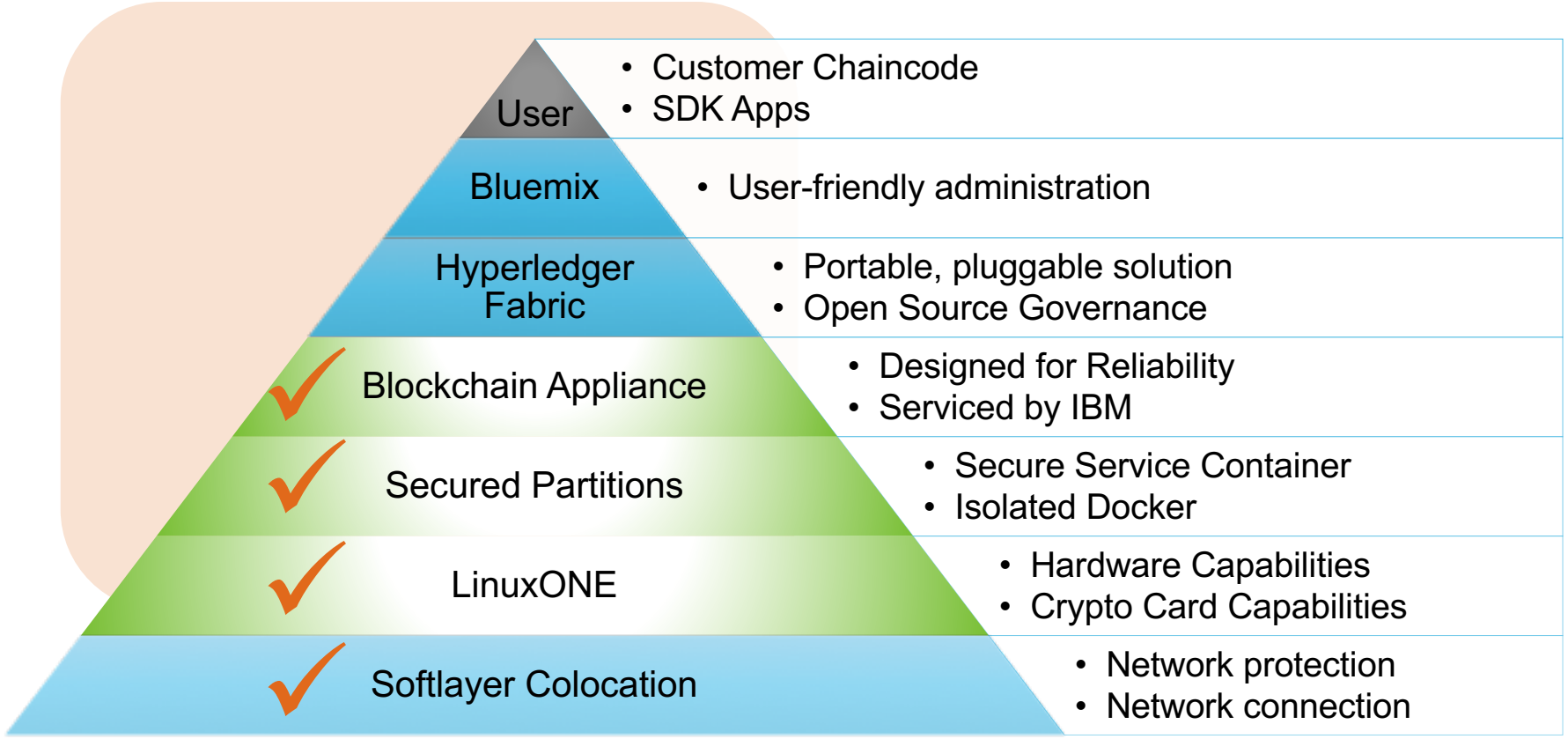
Cluster management

- Create network
  - Call createNode for each node to acomplish HA topology
- Install/Update SSC instaces
- Administrate network
  - Control enrollement of new orgs
  - Manage subchannels
  - Requires using the Hyperledger SDK



- Future:
  - move functions into SSC for additional protection
  - re-utilize for on-prem

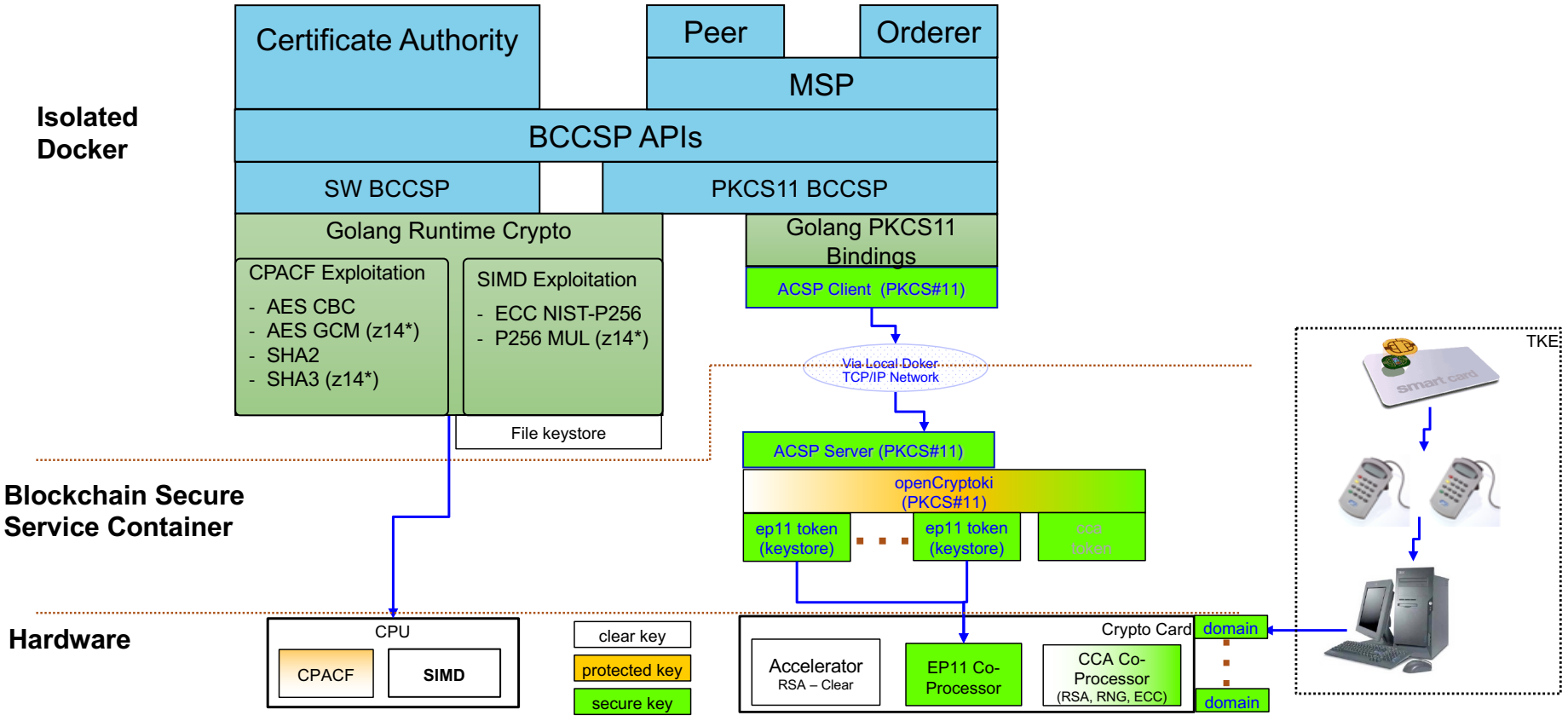
Why IBM Blockchain Platform?



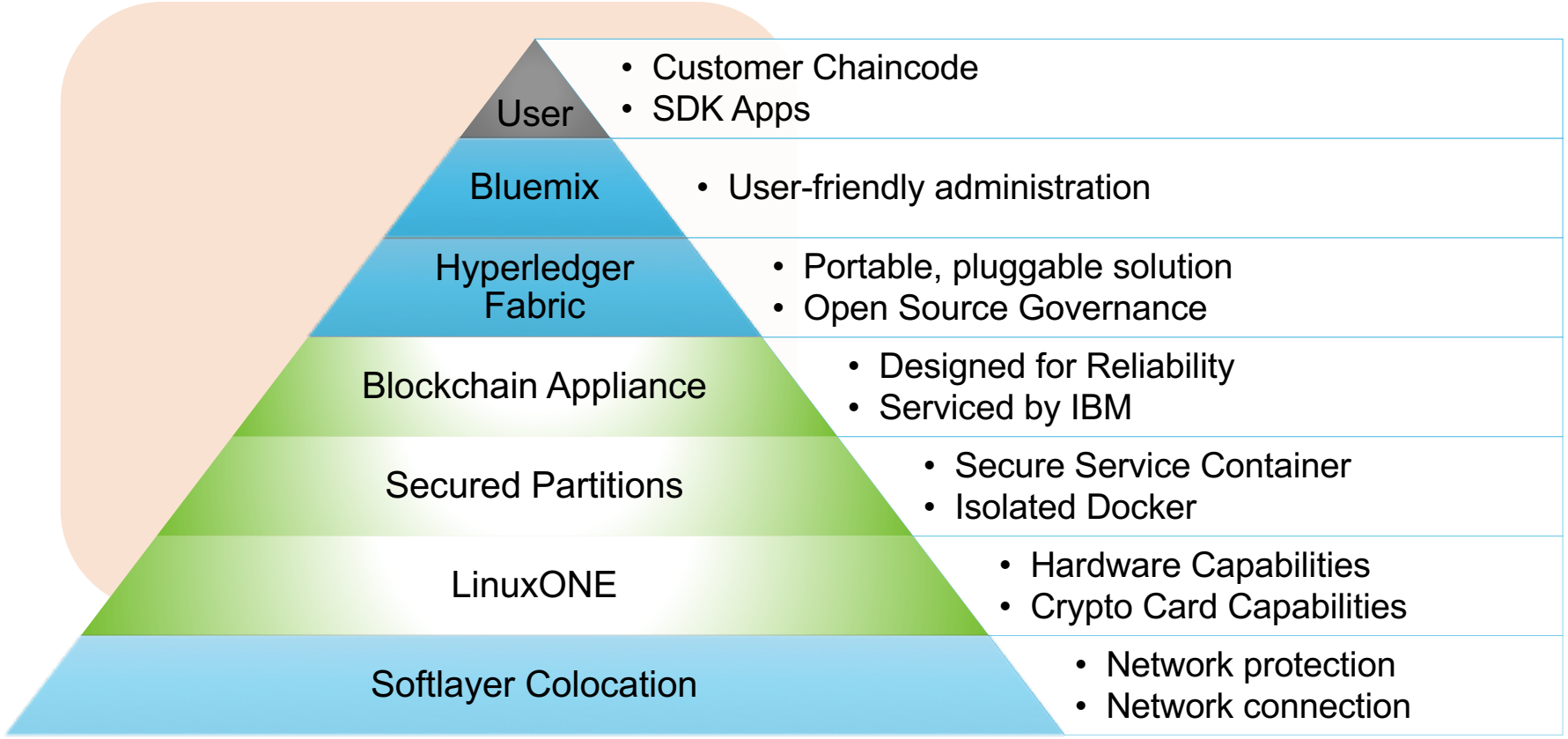
Hyperledger Fabric Security Building Blocks

- Membership Services Provider
  - Identity Management
  - Hyperledger Fabric Certificate Authority (CA)
- TLS
  - Communication
- Channels
  - Policies
  - Admin users
- **Blockchain Crypto Service Provider (BCCSP)**
  - Signature generation and Verification

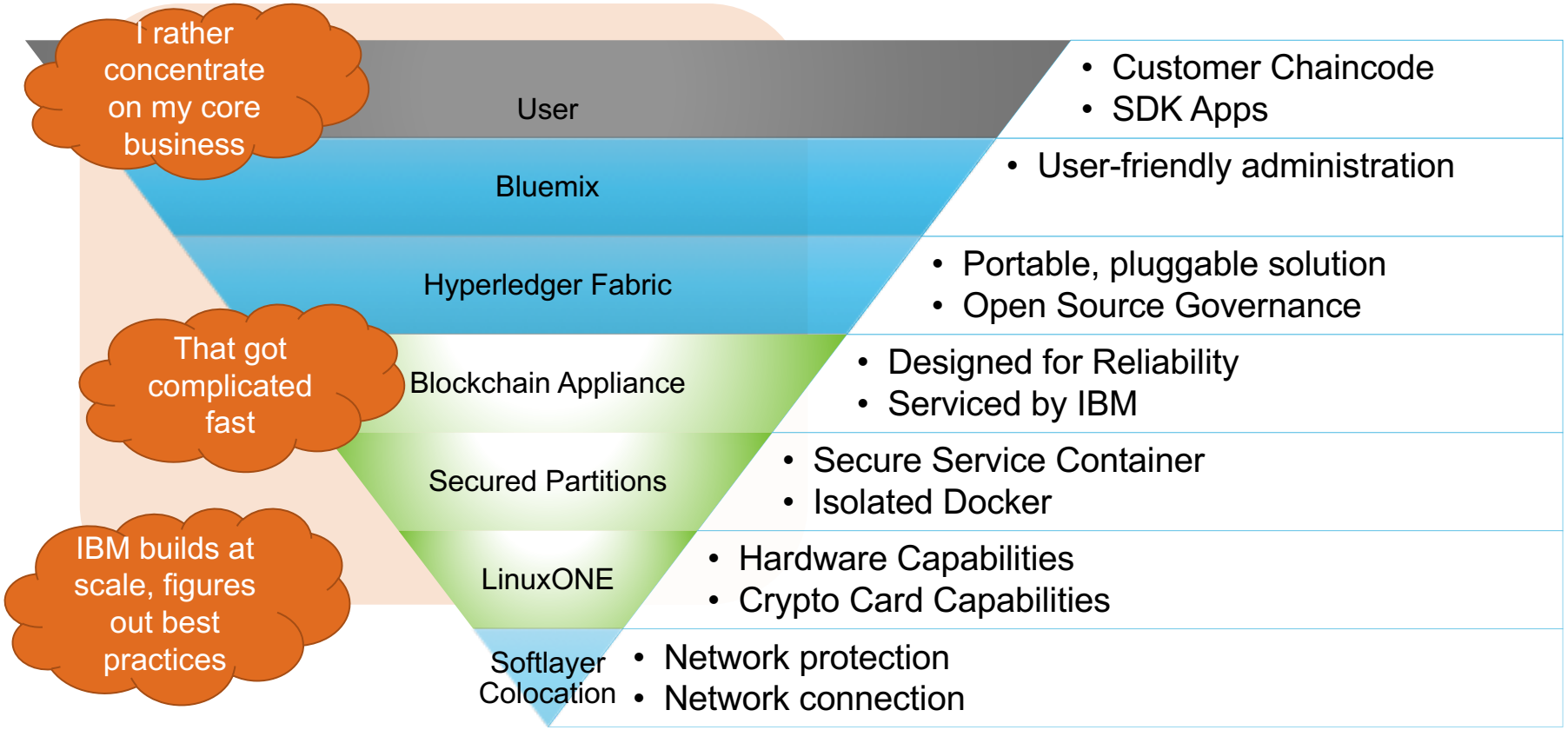
LinuxONE: IBM Blockchain Appliance Crypto Stack



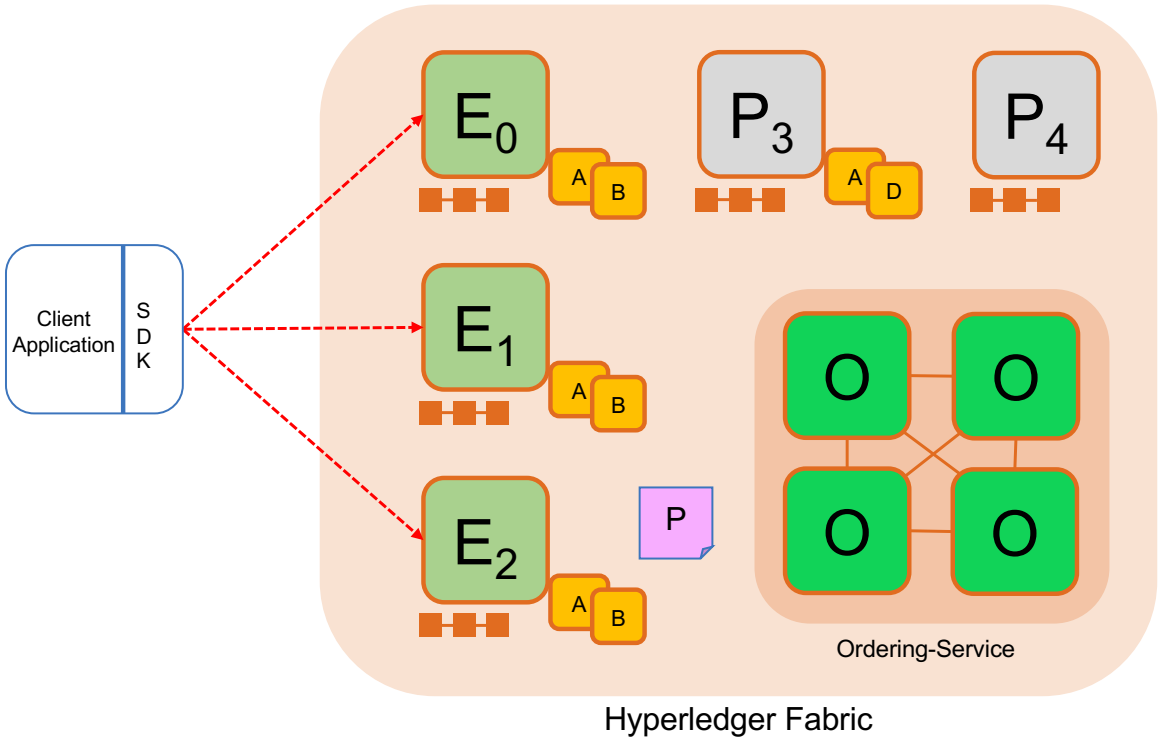
Why IBM Blockchain Platform?



Why IBM Blockchain Platform



Sample transaction: Step 1/7 – Propose transaction

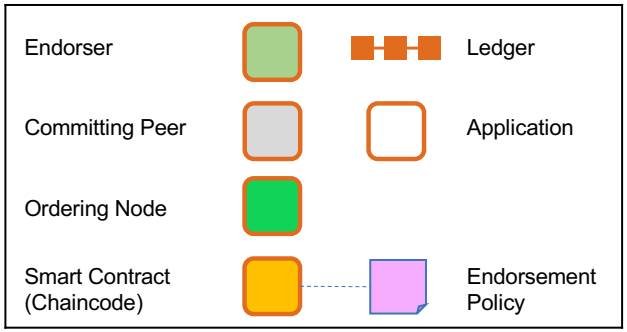


Application proposes transaction

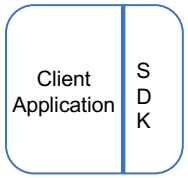
Endorsement policy:
- "E_0, E_1 and E_2 must sign"
- (P_3, P_4 are not part of the policy)

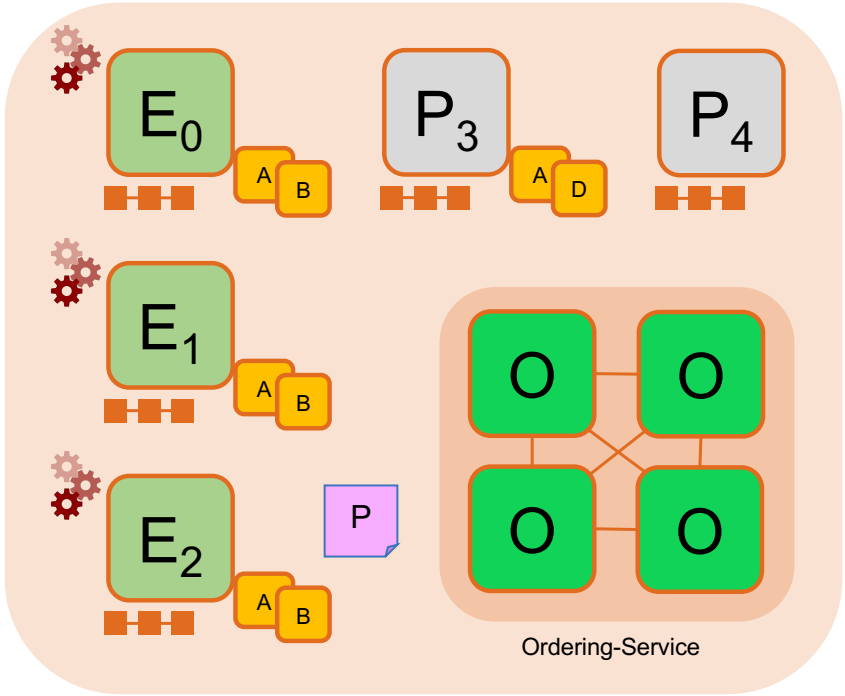
Client application submits a transaction proposal for **Smart Contract A.** It must target the required peers {E_0, E_1, E_2}

Key:



Sample transaction: Step 2/7 – Execute proposal





Hyperledger Fabric

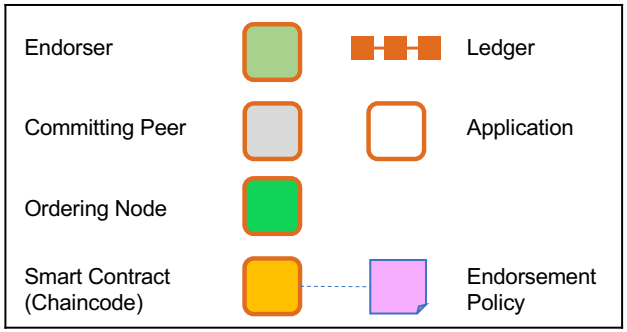
Endorsers Execute Proposals

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

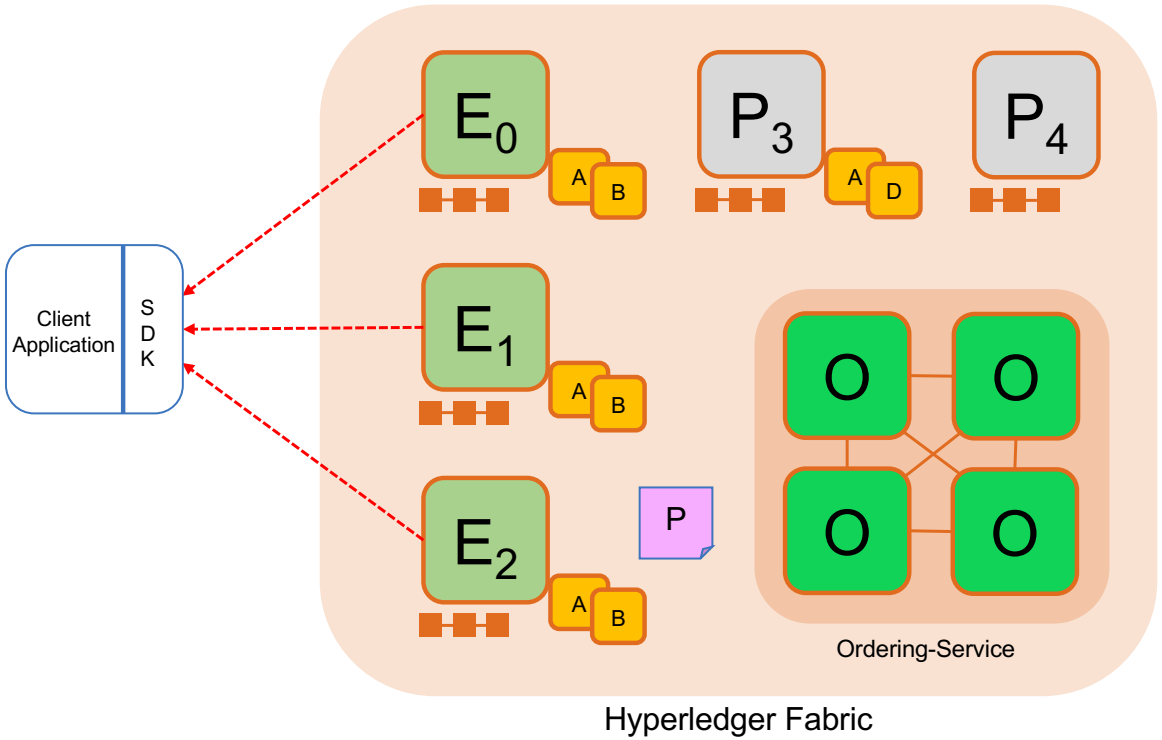
Each execution will capture the set of **R**ead and **W**ritten data, called **RW sets,** which will now flow in the fabric.

Transactions can be signed & encrypted

Key:



Sample transaction: Step 3/7 – Proposal Response



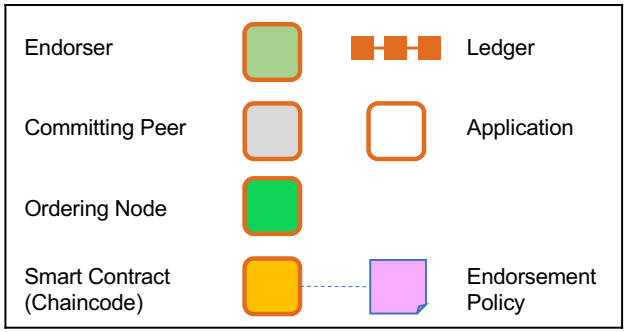
Application receives responses

RW sets are asynchronously returned to application

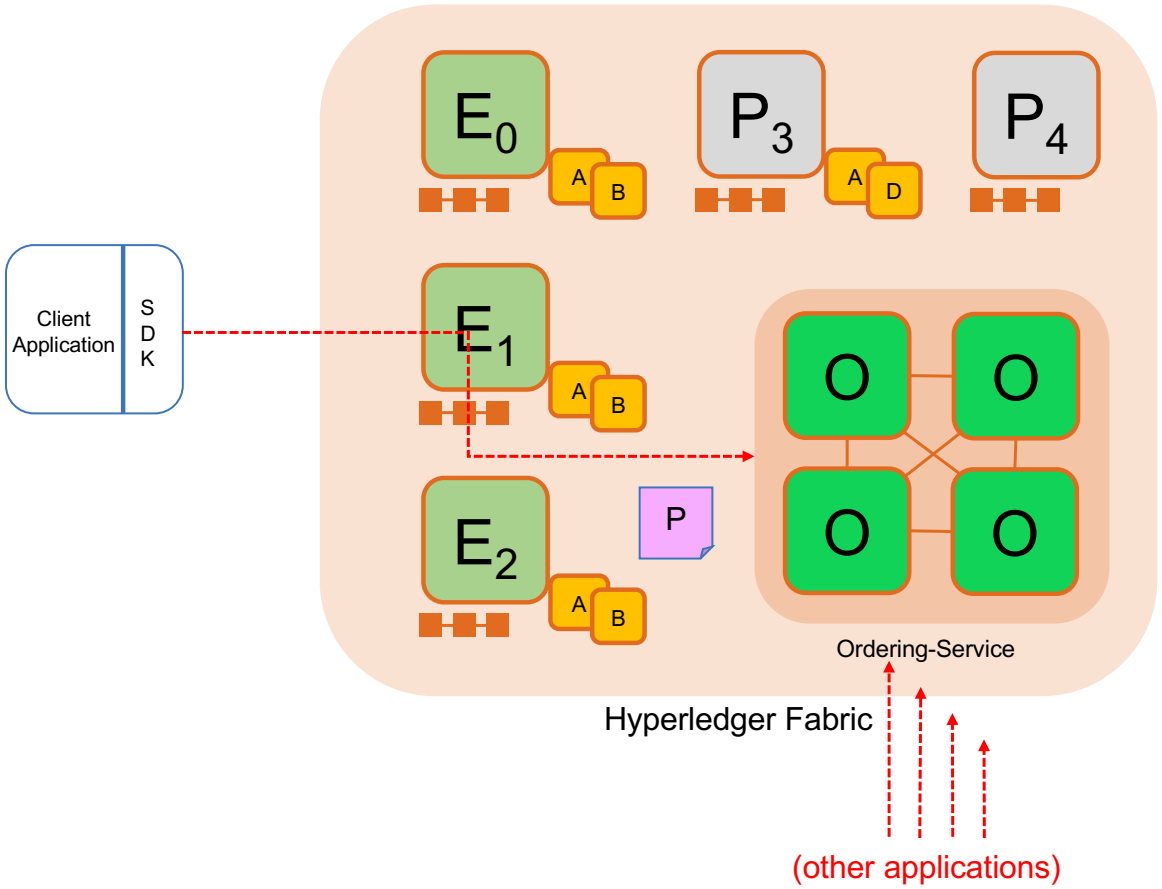
The RW sets are signed by each endorser, and also includes each record version number

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

Key:



Sample transaction: Step 4/7 – Order Transaction

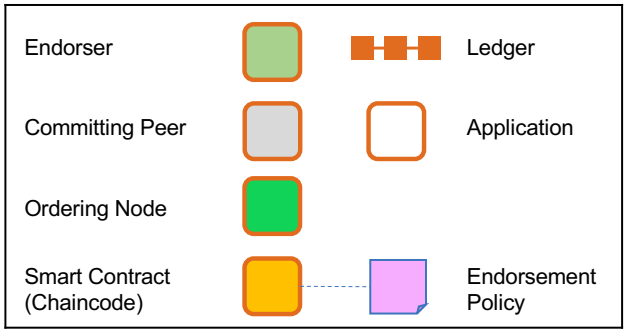


Application submits responses for ordering

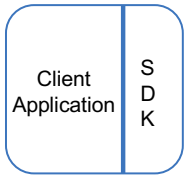
Application submits responses as a **transaction** to be ordered.

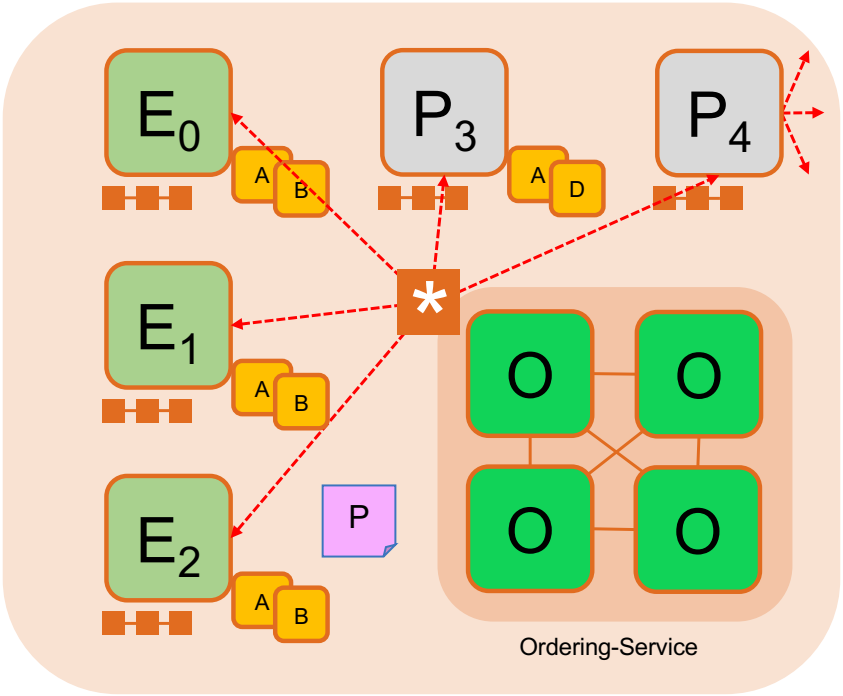
Ordering happens across the fabric in parallel with transactions submitted by other applications

Key:



Sample transaction: Step 5/7 – Deliver Transaction





Hyperledger Fabric

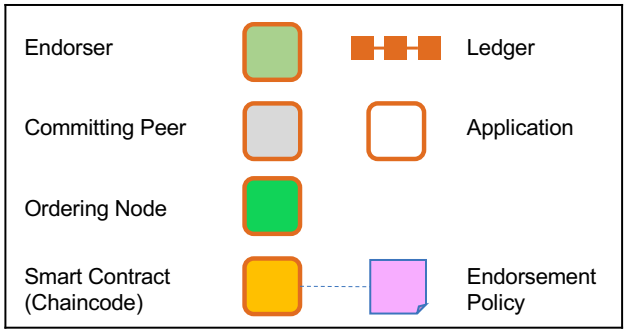
Orderer delivers to all committing peers

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

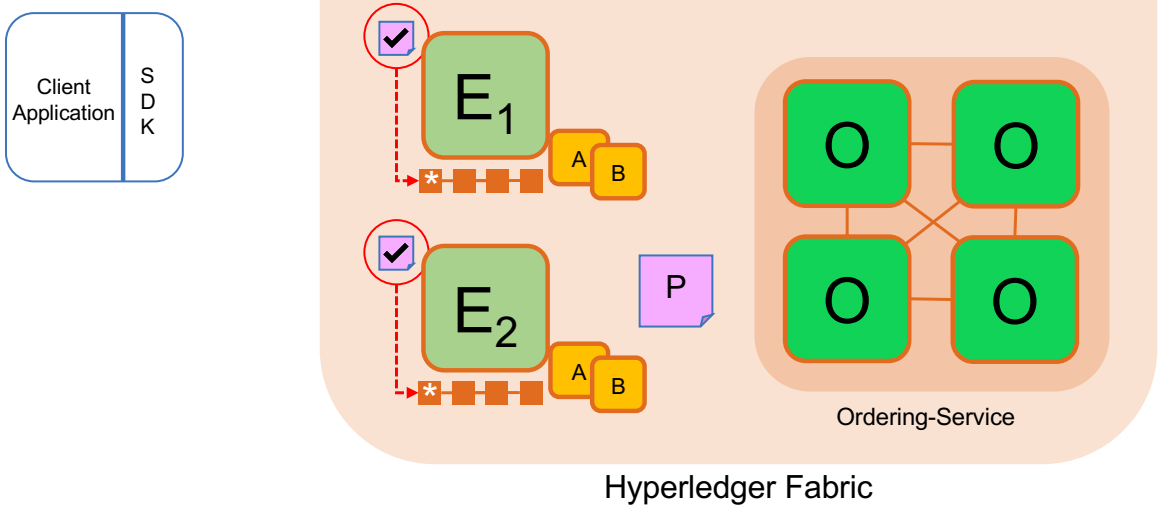
Different ordering algorithms available:

- SOLO (Single node, development)
- Kafka (Crash fault tolerance)

Key:



Sample transaction: Step 6/7 – Validate Transaction



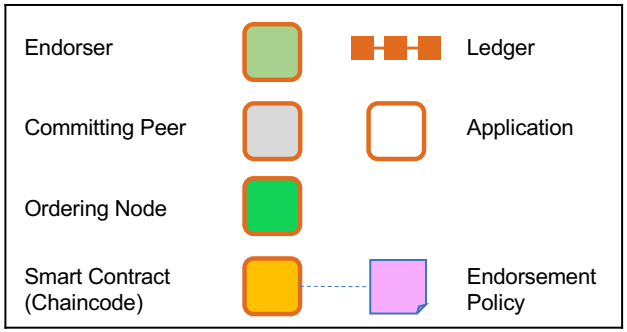
Committing peers validate transactions

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

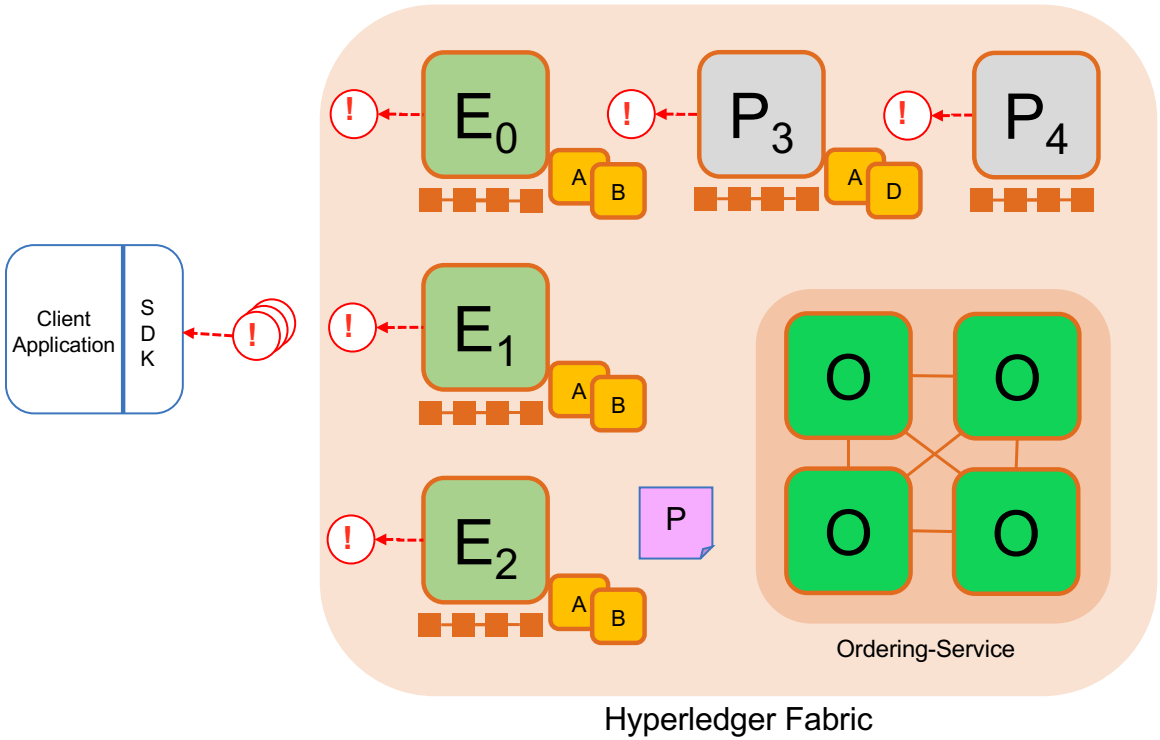
Validated transactions are applied to the world state and retained on the ledger

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

Key:



Sample transaction: Step 7/7 – Notify Transaction

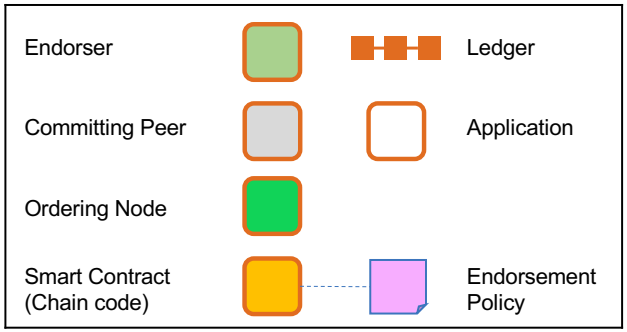


Committing peers notify applications

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

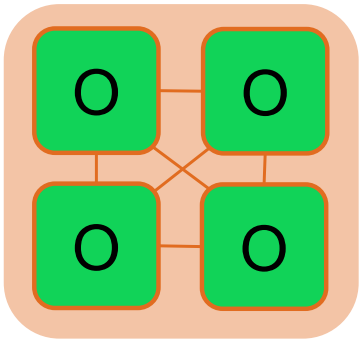
Key:




IBM
Channels and Ordering Service

Ordering Service

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



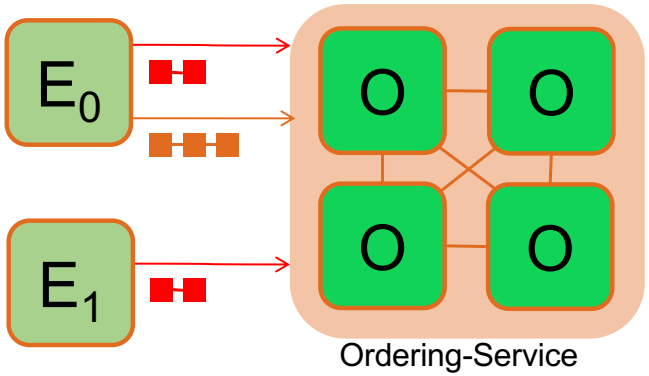
Ordering-Service

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

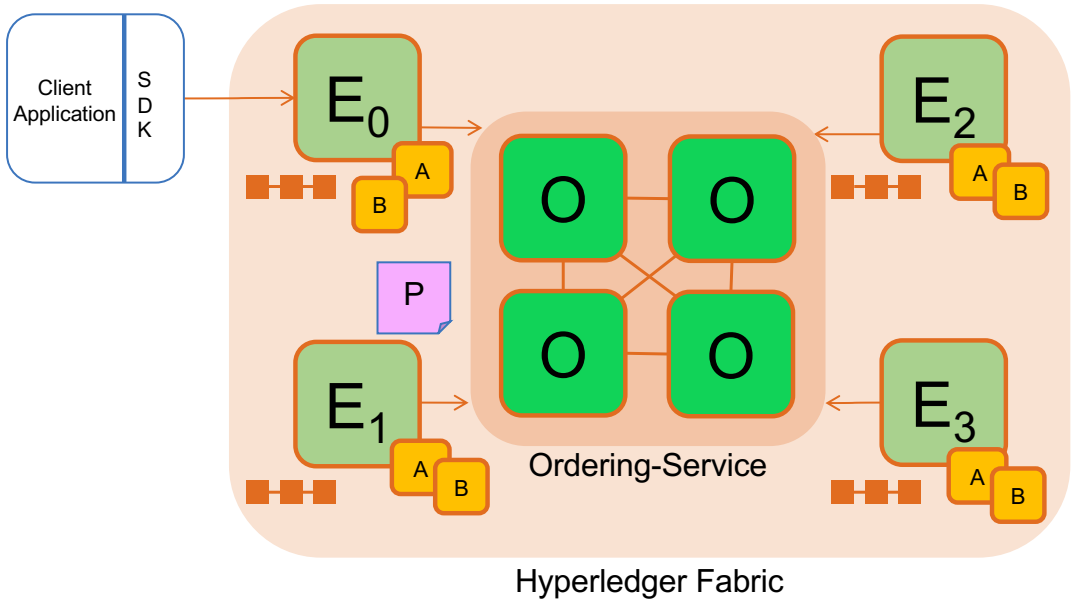
Channels

Separate channels isolate transactions on different ledgers

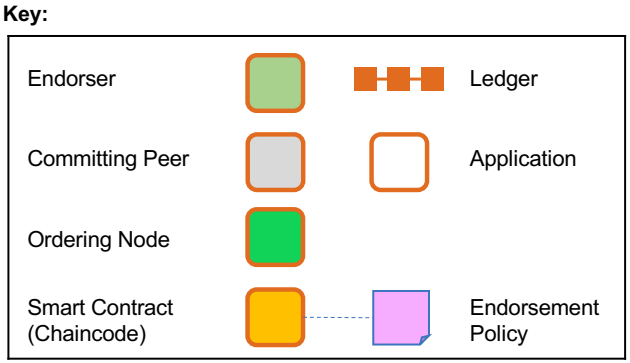


- Chaincode is installed on peers that need to access the worldstate
- Chaincode is instantiated on specific channels for specific peers
- Ledgers exist in the scope of a channel
  - Ledgers can be shared across an entire network of peers
  - Ledgers can be included only on a specific set of participants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

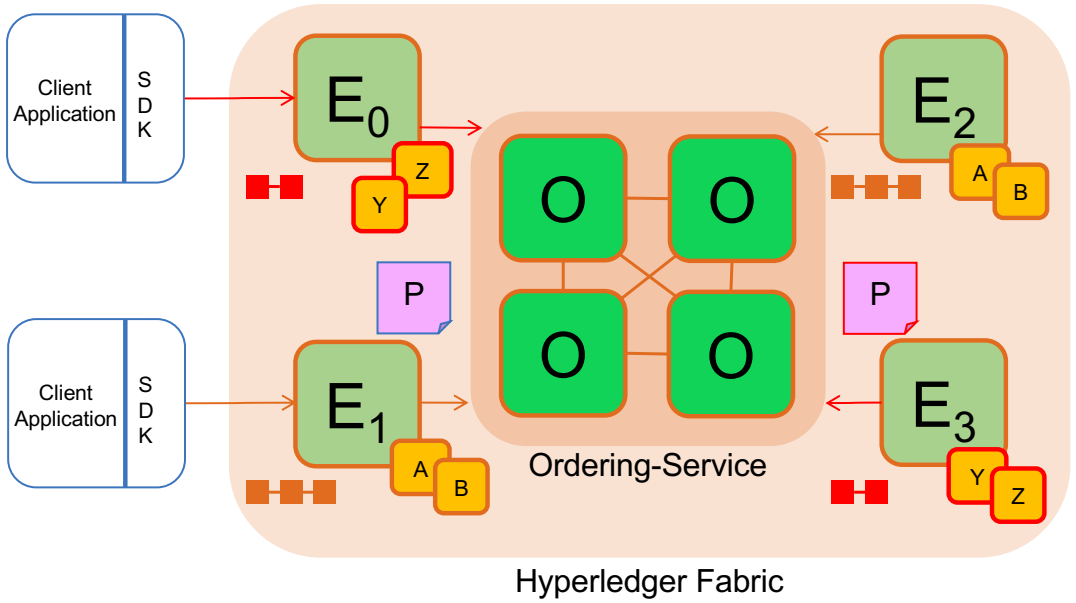
Single Channel Network



- Similar to v0.6 PBFT model
- 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

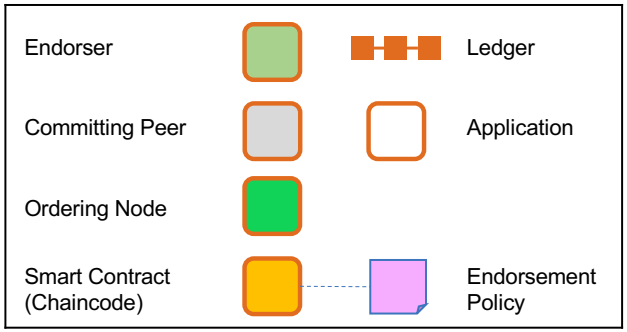


Multi Channel Network

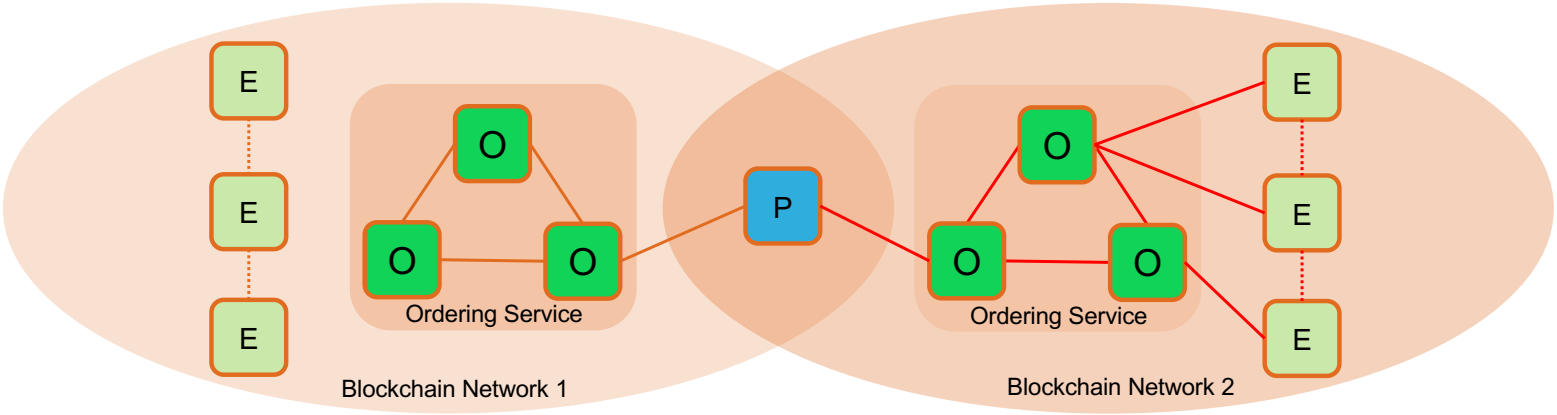


- Peers E_0 and E_3 connect to the red channel for chaincodes **Y** and **Z**
- Peers E_1 and E_2 connect to the blue channel for chaincodes **A** and **B**

Key:



Future 2 network topology



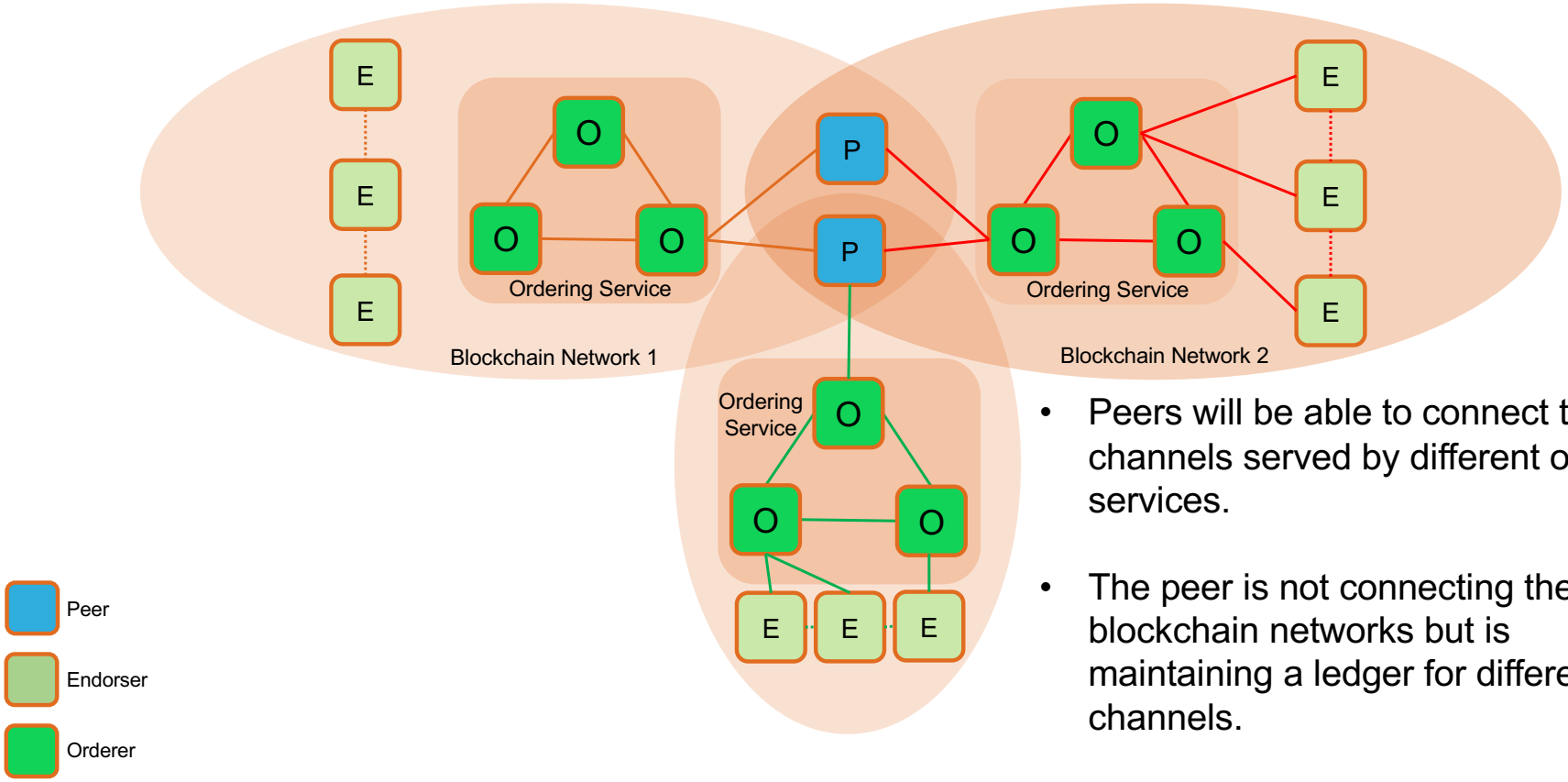
- Peers will be able to connect to channels served by different ordering services.
- The peer is not connecting the three blockchain networks but is maintaining a ledger for different channels.

Peer

Endorser

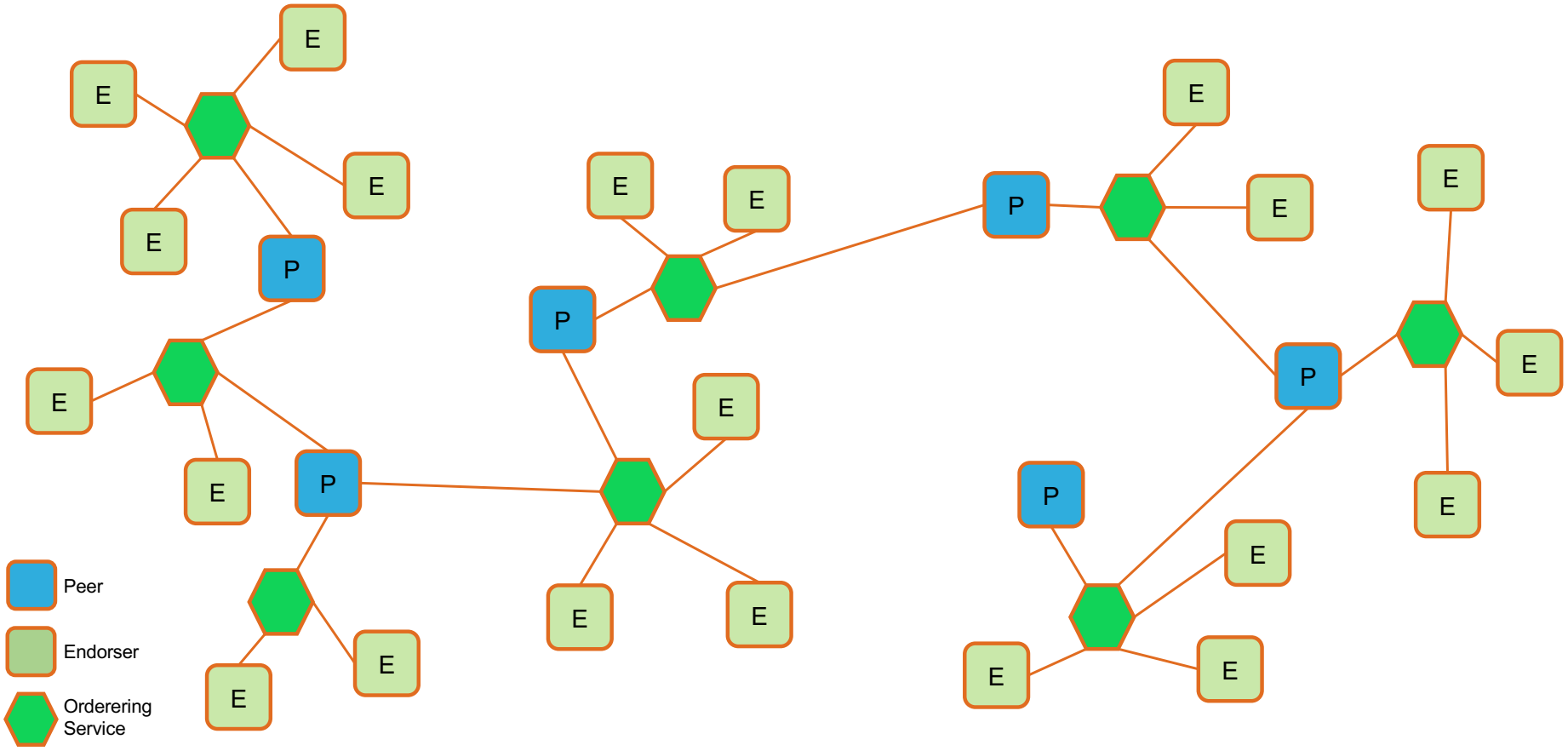
Orderer

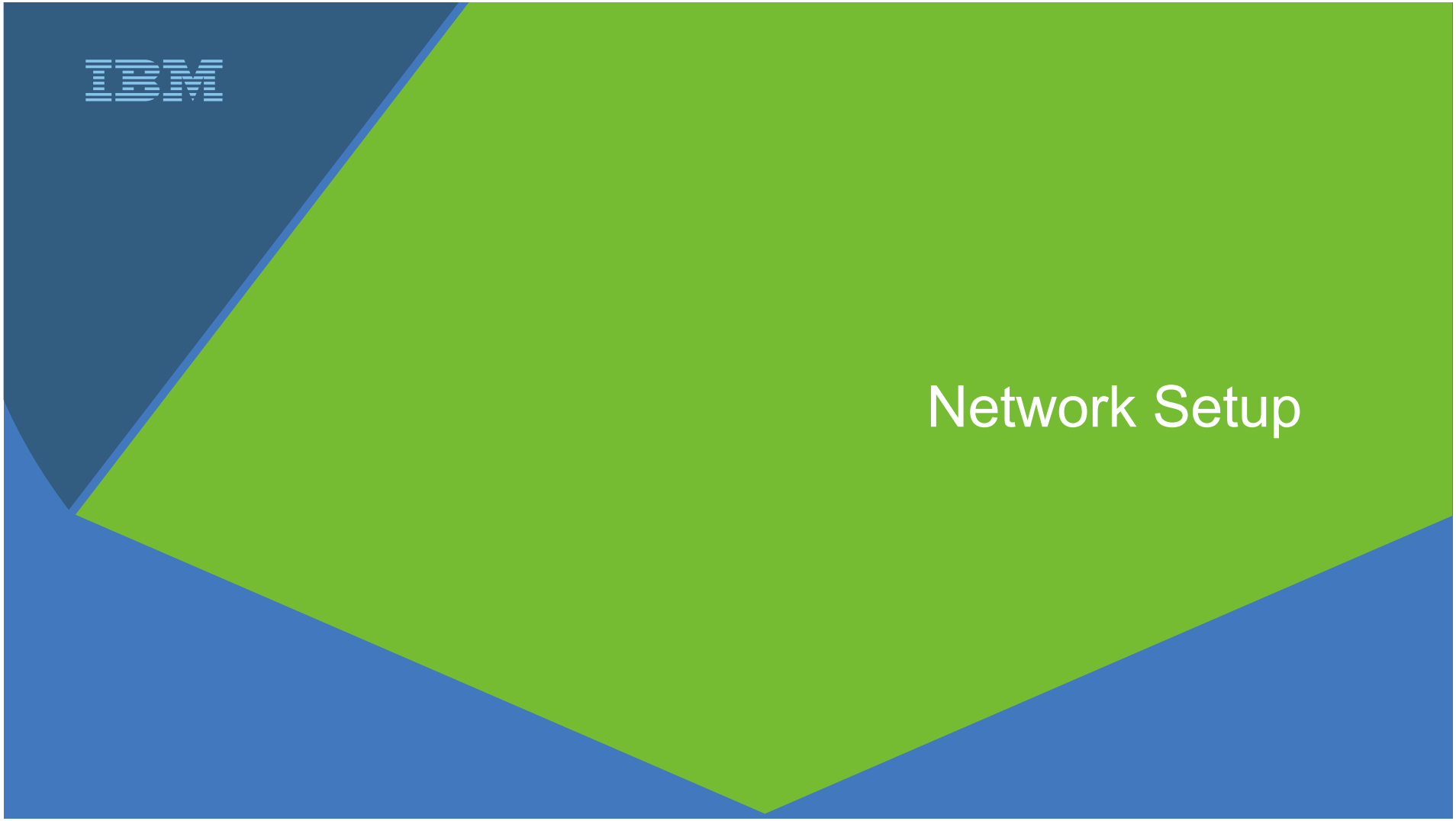
Future 3 network topology



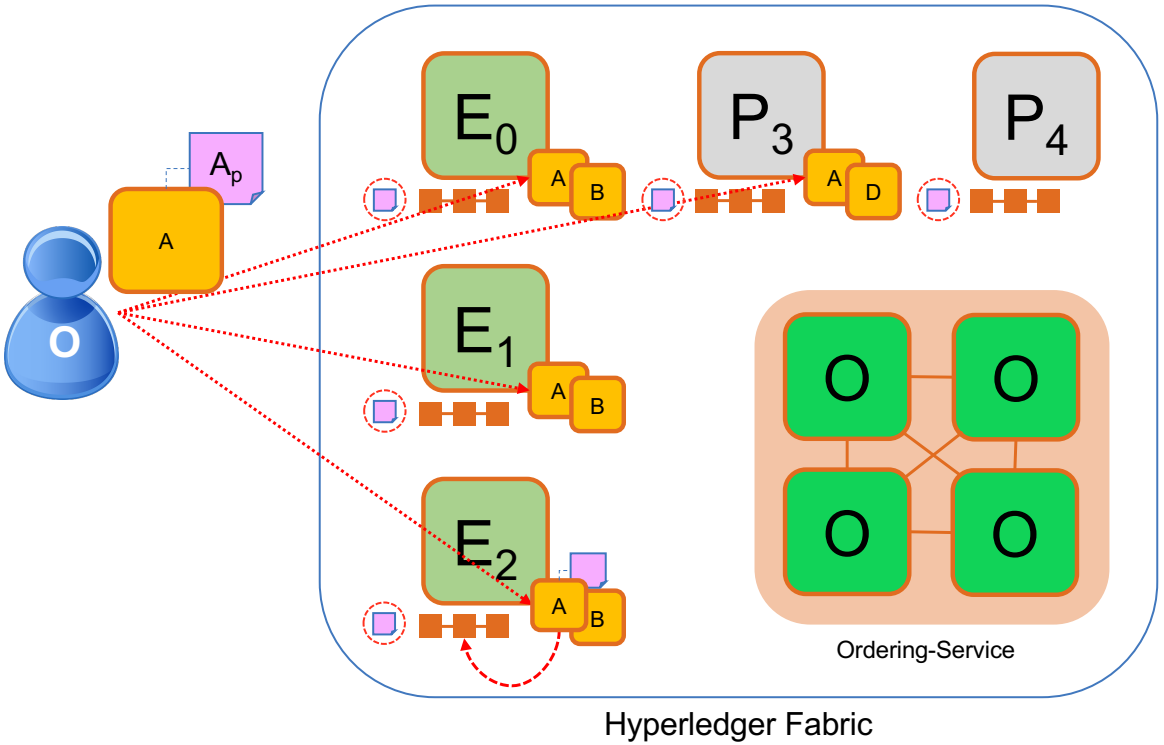
- Peers will be able to connect to channels served by different ordering services.
- The peer is not connecting the three blockchain networks but is maintaining a ledger for different channels.

Future Network of Networks




Network Setup

Installing and instantiating smart contract Chaincode



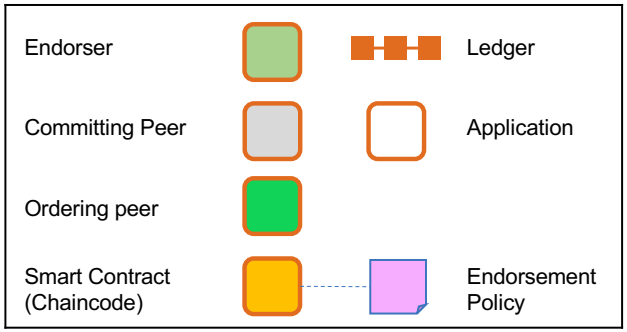
Operator installs then instantiates

Operator **installs** smart contracts with endorsement policies to appropriate peers: **E_0**, **E_1**, **E_2**, **P_3**, and not **P_4**

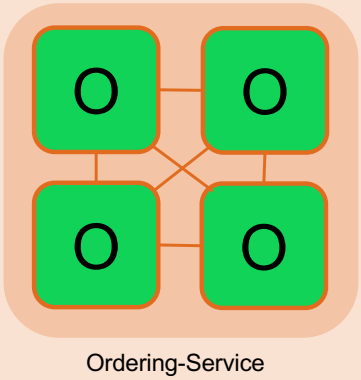
Operator **instantiates** smart contract on given channel. One-time initialization

Policy subsequently available to all peers on channel, e.g. including **P_4**

Key:



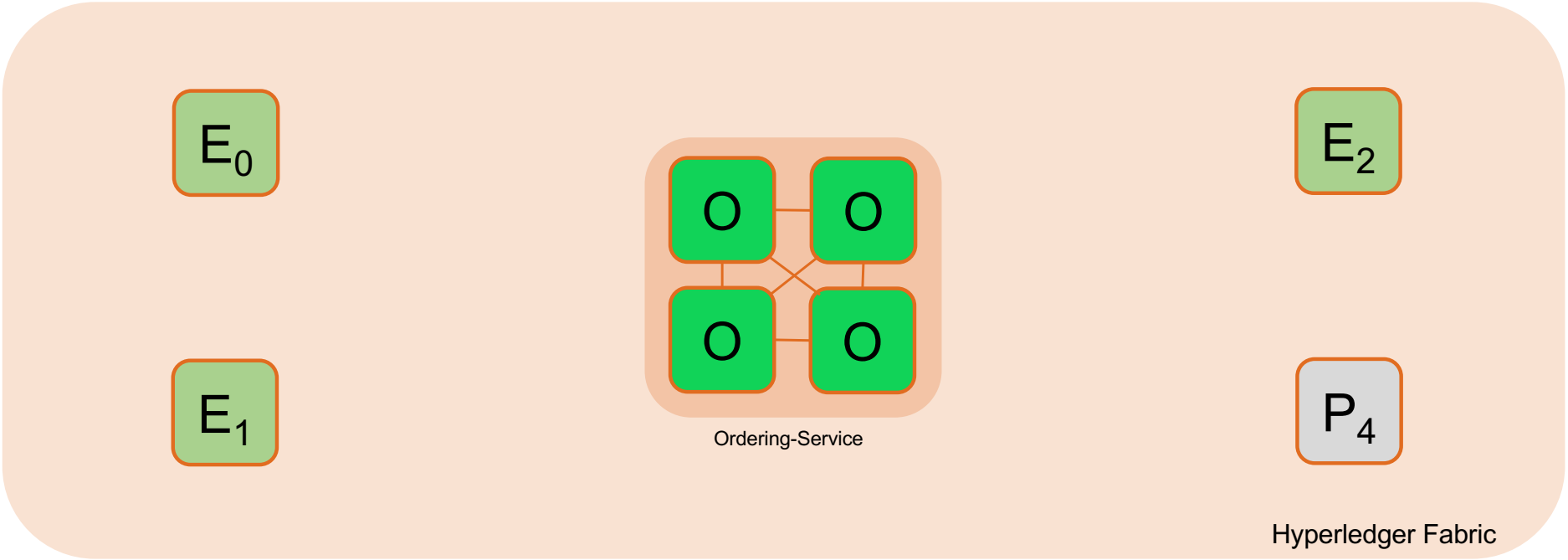
Bootstrapping the Network (1/6) – Configure & start Ordering Service



Hyperledger Fabric

- An Ordering Service is **configured** and started for other network peers to use
  `$ docker-compose [-f orderer.yml] ...`

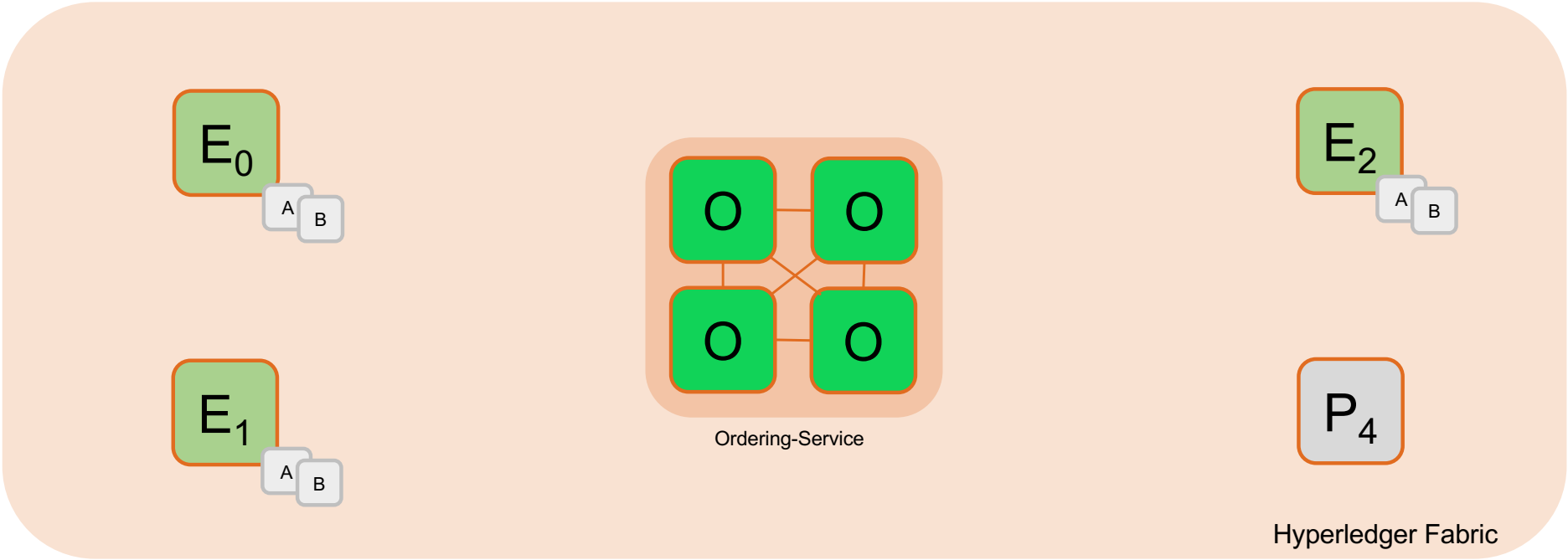
Bootstrapping the Network (2/6) – Configure and Start Peer Nodes



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

```
$ peer node start ...
```

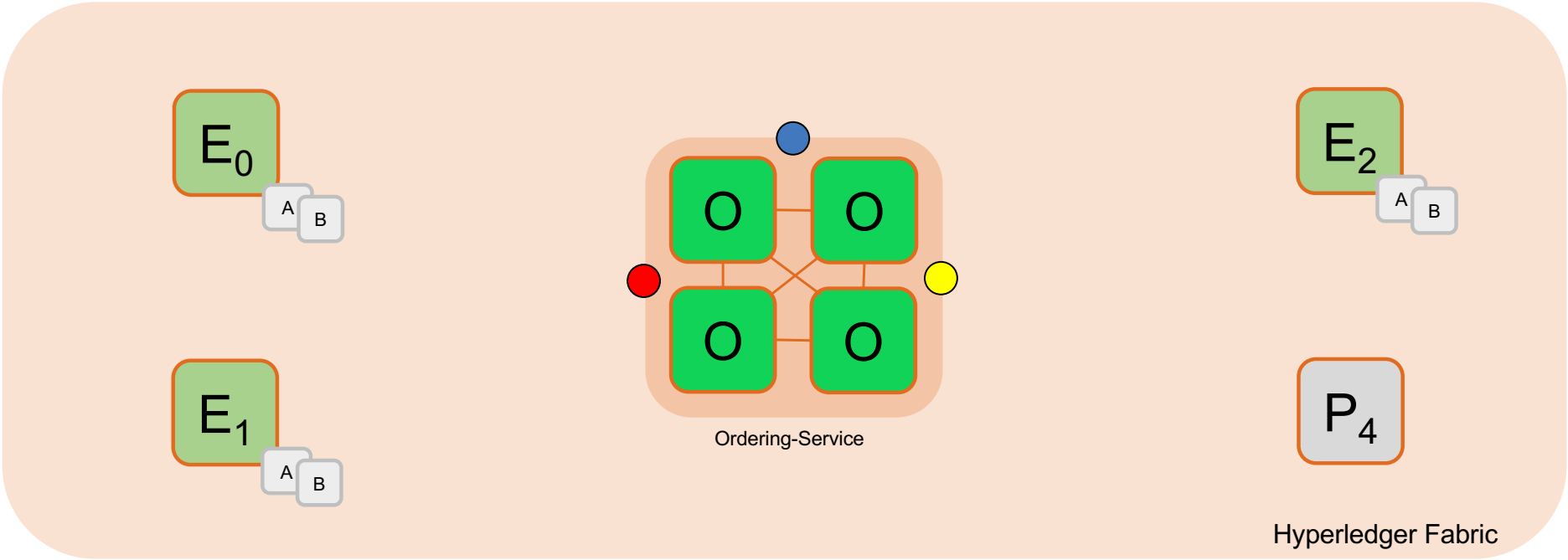
Bootstrapping the Network (3/6) – Install Chaincode



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

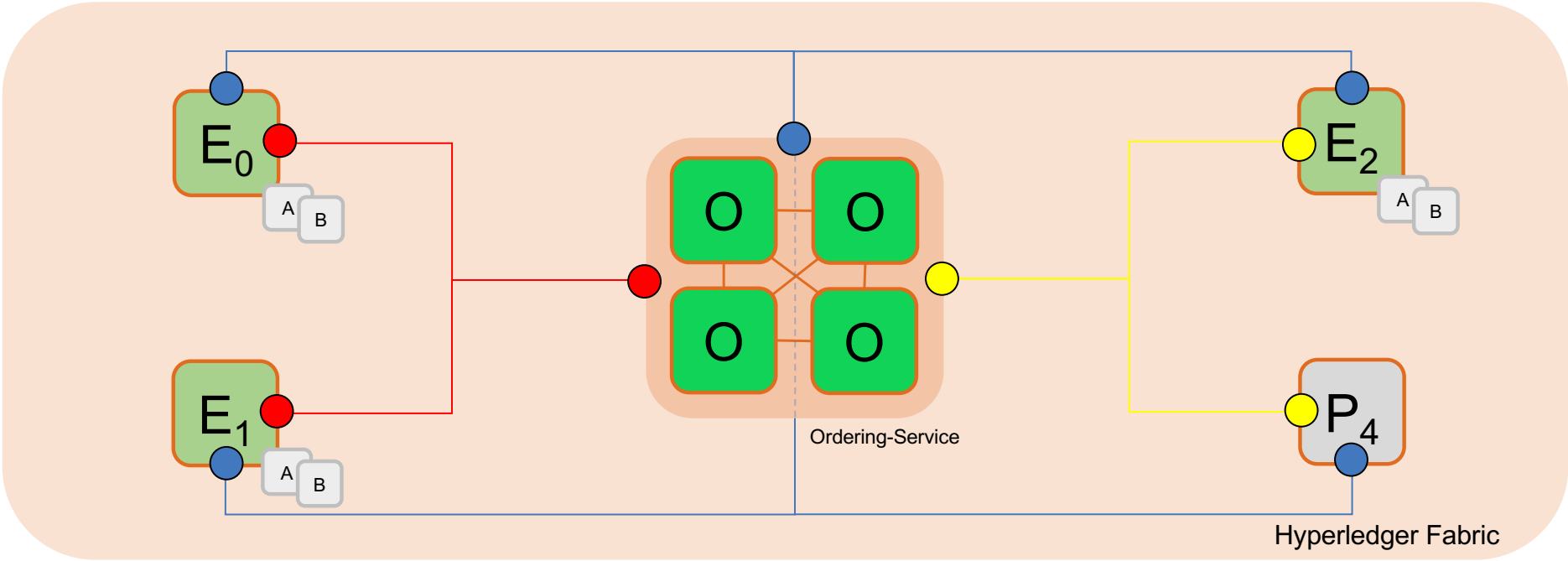
```
$ peer chaincode install ...
```

Bootstrapping the Network (4/6) – Create Channels



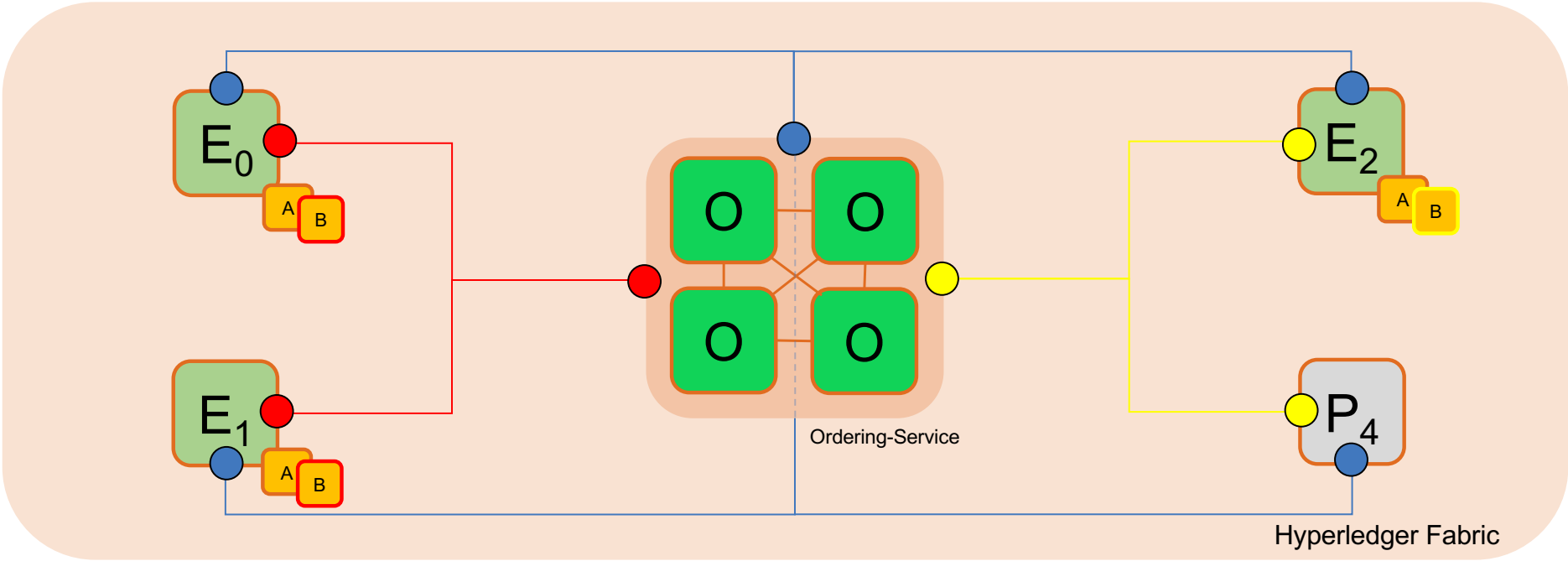
- Channels are **created** on the ordering service
  `$ peer channel create -o [orderer] ...`

Bootstrapping the Network (5/6) – Join Channels



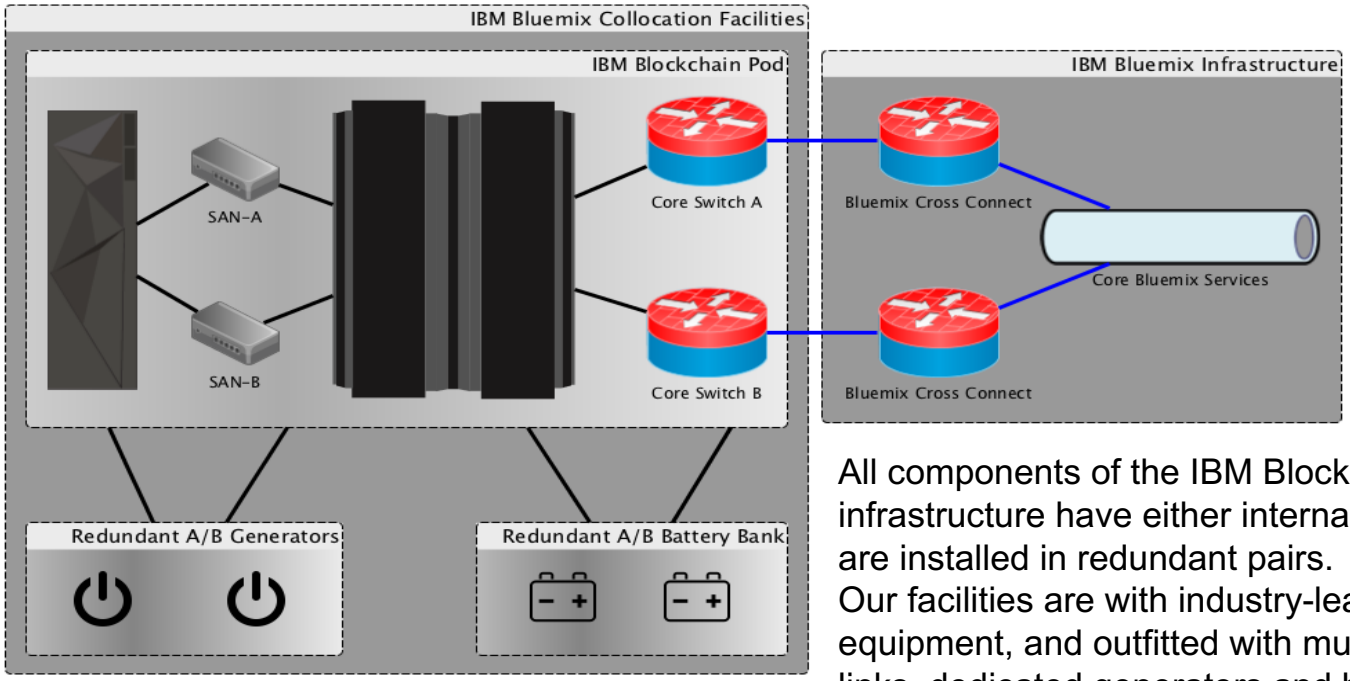
- Peers that are permissioned can then **join** the channels they want to transact on
  `$ peer channel join ...`

Bootstrapping the Network (6/6) – Instantiate Chaincode



- Peers finally **instantiate** the Chaincode on the channels they want to transact on
  `$ peer channel instantiate ... -P 'policy'`
- Once instantiated a Chaincode is live and can process transaction requests
- Endorsement Policy is specified at instantiation time

Hardware HA Approach - Single Site



The IBM Bluemix Infrastructure

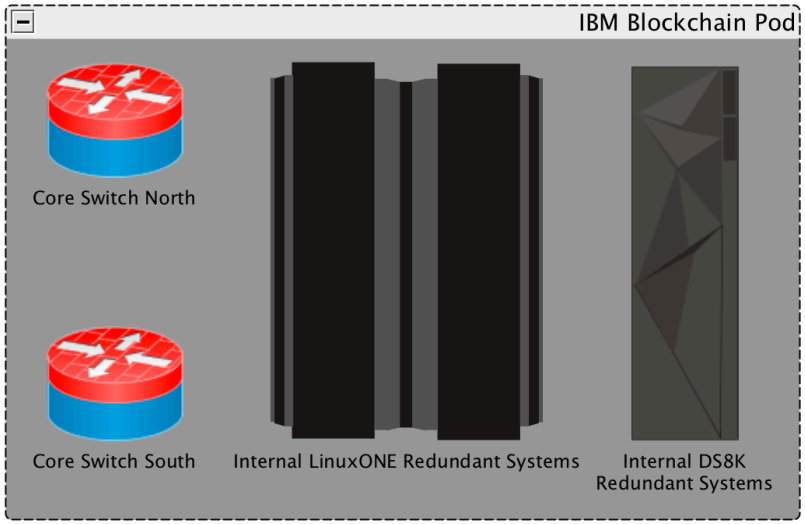
DS 8884 – Mod 984:
- 128GB Proc. Memory
- 4 x 4port 16GB Ficon
- 2.4TB Flash
- 320TB of HDD
- CSM for Back/Restore Flashing

LinuxOne – Mod LC 9:
- 4 Drawers, 129 IFLs
- 6 TB Memory
- 16 x OSA cards (mix)
- 10x16Gb Ficon Express
- 4 x Crypto 5S
- Internal Battery Feature
- 2 x Rack Mounted HMC

All components of the IBM Blockchain Platform infrastructure have either internally redundant systems or are installed in redundant pairs.
Our facilities are with industry-leading hardware and equipment, and outfitted with multiple power feeds, fiber links, dedicated generators and battery backup.
Redundant n+1 power and cooling resources are regularly inspected to guarantee stability.

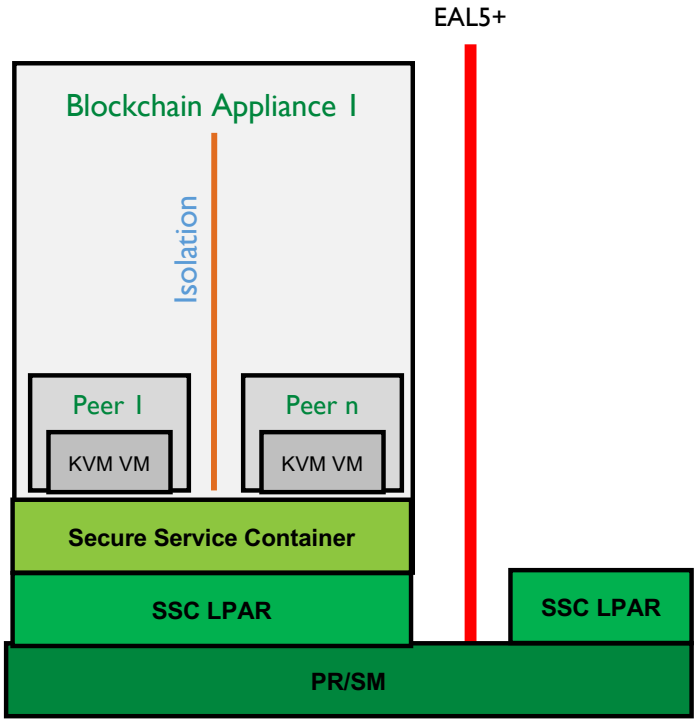
IBM Blockchain Platform High Availability

- IBP Pods are deployed in collocation with IBM Bluemix Data Centers in key locations around the world
- The Pods are designed to leverage the intrinsic HA capabilities of the IBM LinuxONE and DS8000 Storage systems
- All components lacking internal redundancy are deployed in North/South pairs



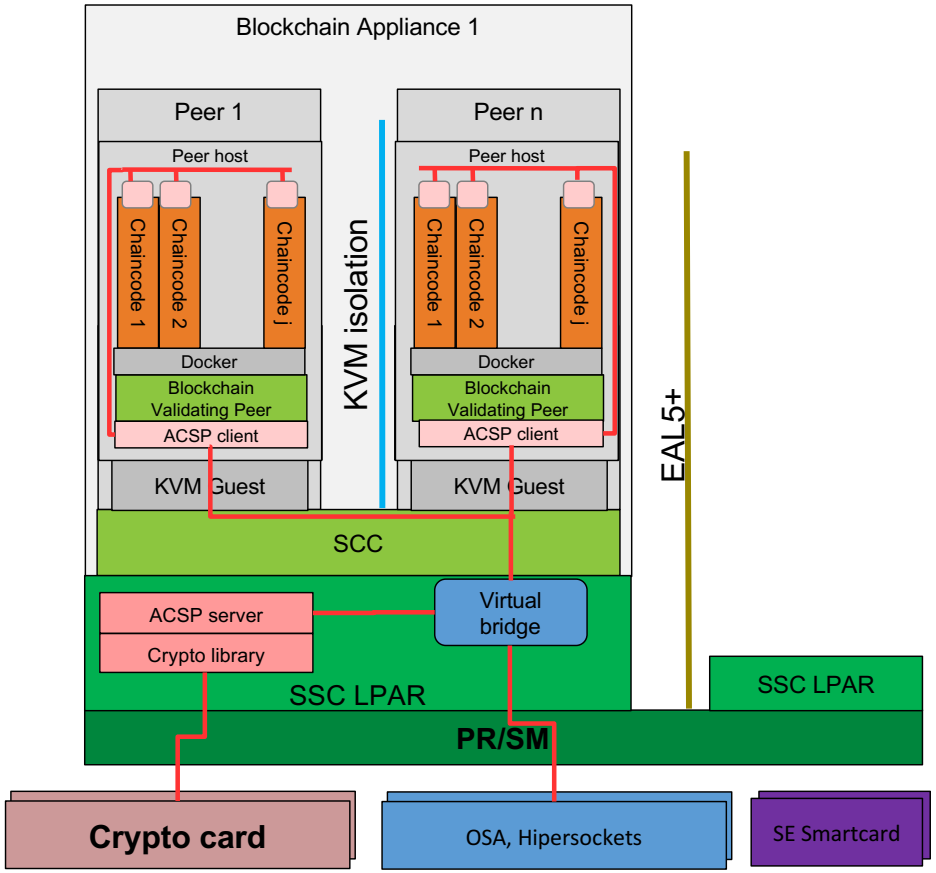
- In addition to core pod redundancy, IBM Blockchain utilizes core redundancy features of the IBM Bluemix Infrastructure platform. Including network redundancy.

IBM KVM Based Blockchain Appliance



- First create LPARs for SSC's
- Install SSC Blockchain appliance
- KVM (virtualization manager) is used to deploy blockchain peers as VM's
  - All within the SSC, providing peer isolation
  - KVM/VMs are not visible (exposed)
  - Blockchain ports for peer access are open for external access
- Multiple peers peer system
- Advantages
  - Only SSC and Blockchain API's are exposed

zBlockchain Appliance



- ❑ First create LPARs for SSCs
- ❑ Install SSC Blockchain appliance
- ❑ KVM (virtualization manager) is used to deploy blockchain peers as VM's
  - All within the SSC, providing peer isolation
  - KVM/VMs are not visible (exposed)
  - Blockchain ports for peer access are open for external access
- ❑ Multiple peers peer system
- ❑ Advantages
  - Only SSC and Blockchain APIs are exposed