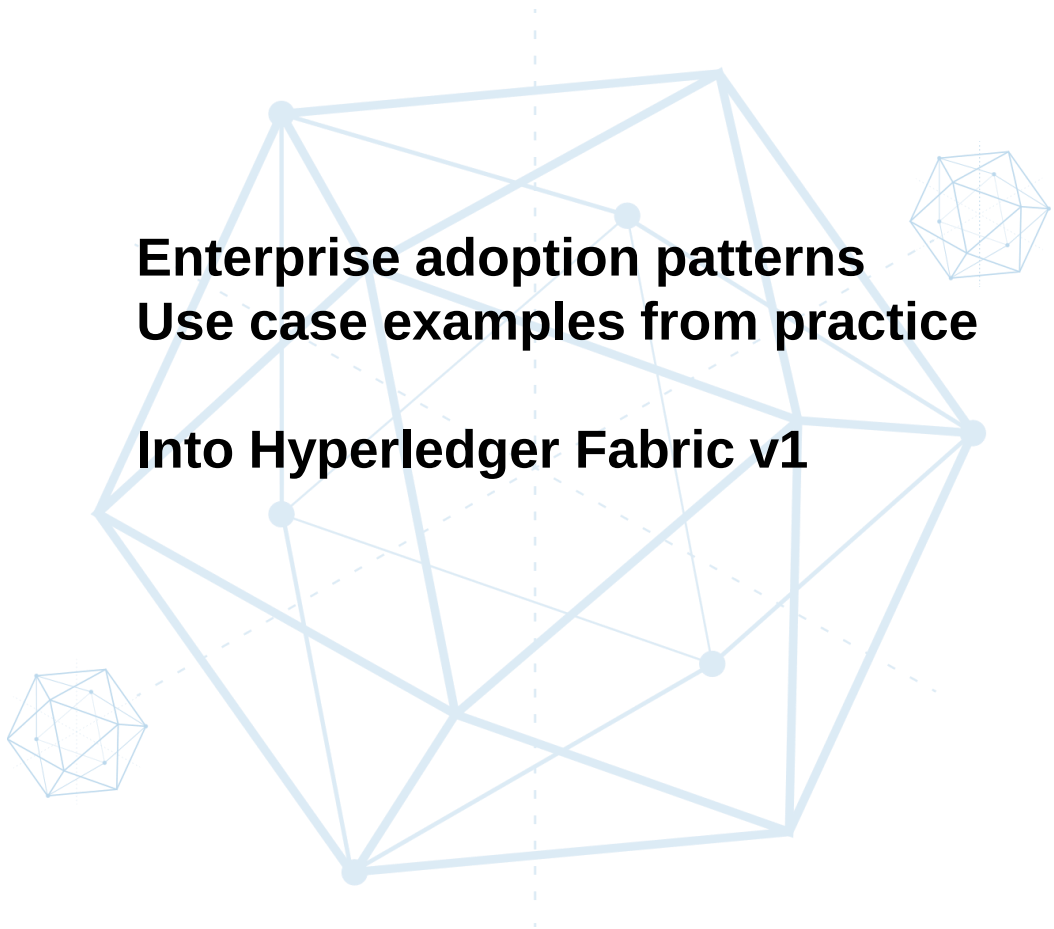
Blockchain for Business



Enterprise adoption patterns
Use case examples from practice

Into Hyperledger Fabric v1

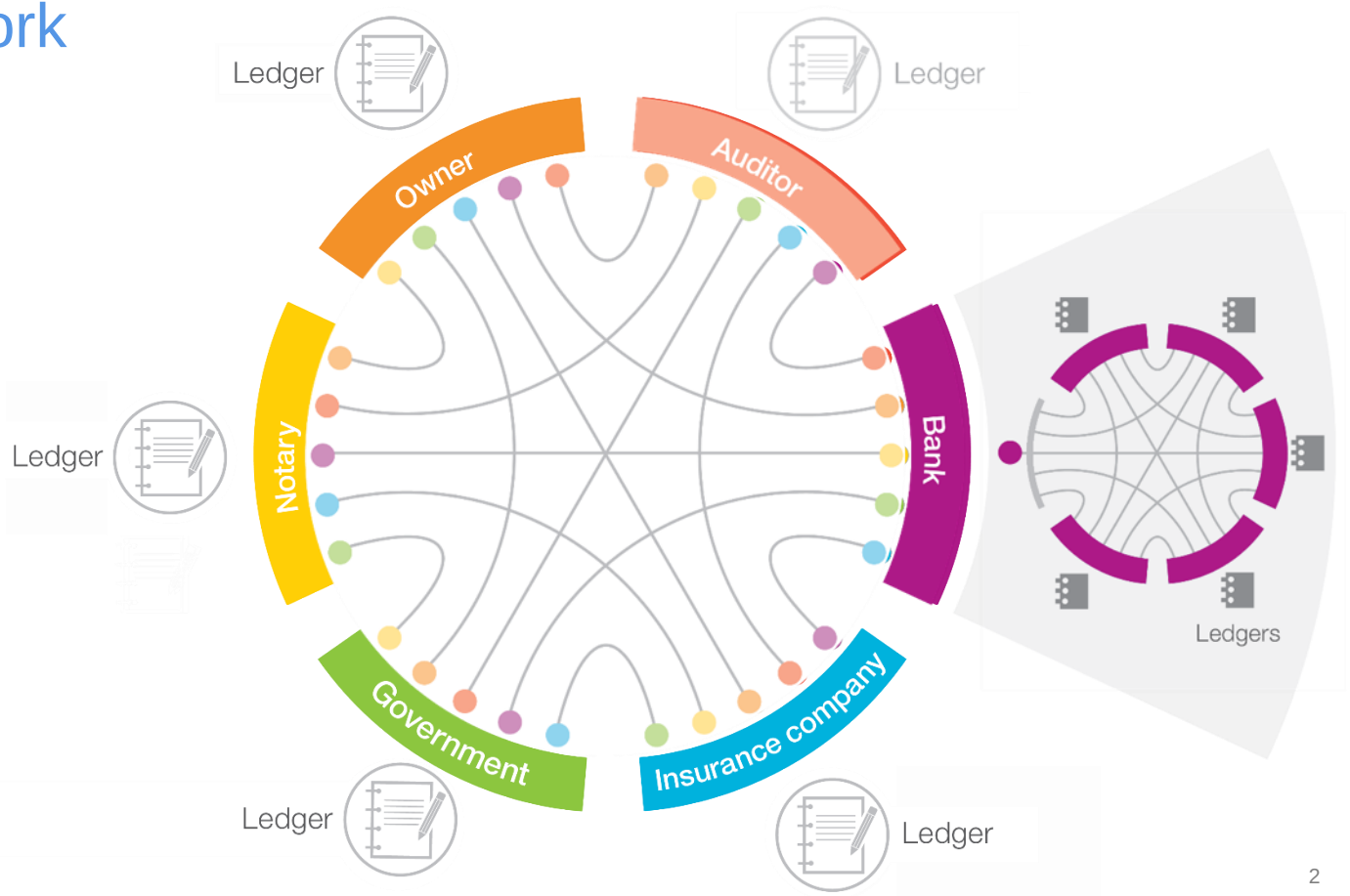
Hyperledger Meetup
Frankfurt, 11. Mai 2017

Recap: Where the problems arise: book keeping across a business network

Every participant keeps their own ledger updated with their transactions

Transactions mostly bilateral message based

Each organization in the network has complex silos that require reconciliation

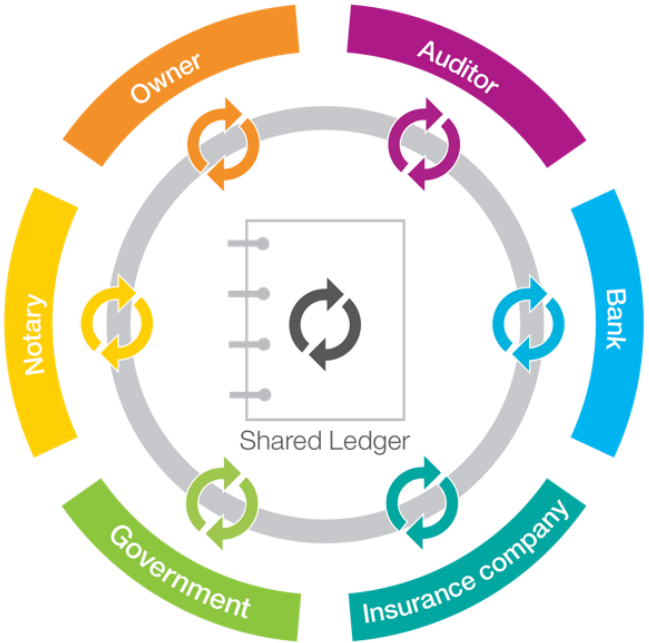


The business solution: Blockchains use a distributed ledger built by all participants

Blockchains shift the paradigm from information held by a single owner to the lifetime history of an asset or transaction distributed across multiple participants in a business network

Instead of messaging-based communications, the new paradigm is state-based

Smart Contracts incorporate business rules for the automation of transactions



Technology Requirements

Append only **Distributed Ledger**

Consensus Protocols for agreeing change to the ledger

Cryptography based security and privacy/ confidentiality

Programable **Smart Contracts**

Hyperledger, a Linux Foundation project

- Announced by The Linux Foundation on December 17, 2015 with 17 founders, now over 130 members
- Hyperledger is an open source and openly governed collaborative effort to advance cross-industry blockchain technologies for business, hosted by The Linux Foundation.
- Hyperledger Fabric is a blockchain framework implementation and one of the Hyperledger projects, intended as a foundation for developing applications/solutions with a modular architecture

Enable adoption of shared ledger technology at a pace and depth not achievable by any one company or industry

Premier

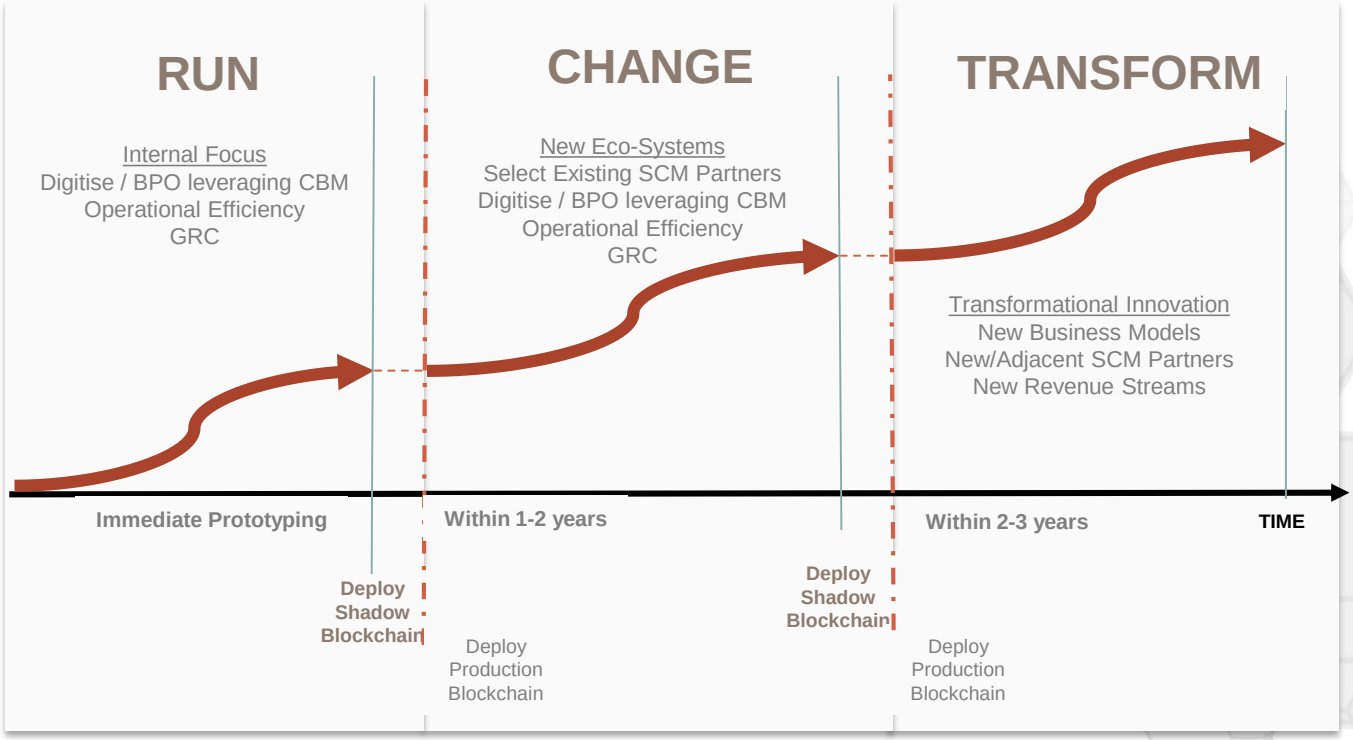
General

Associate

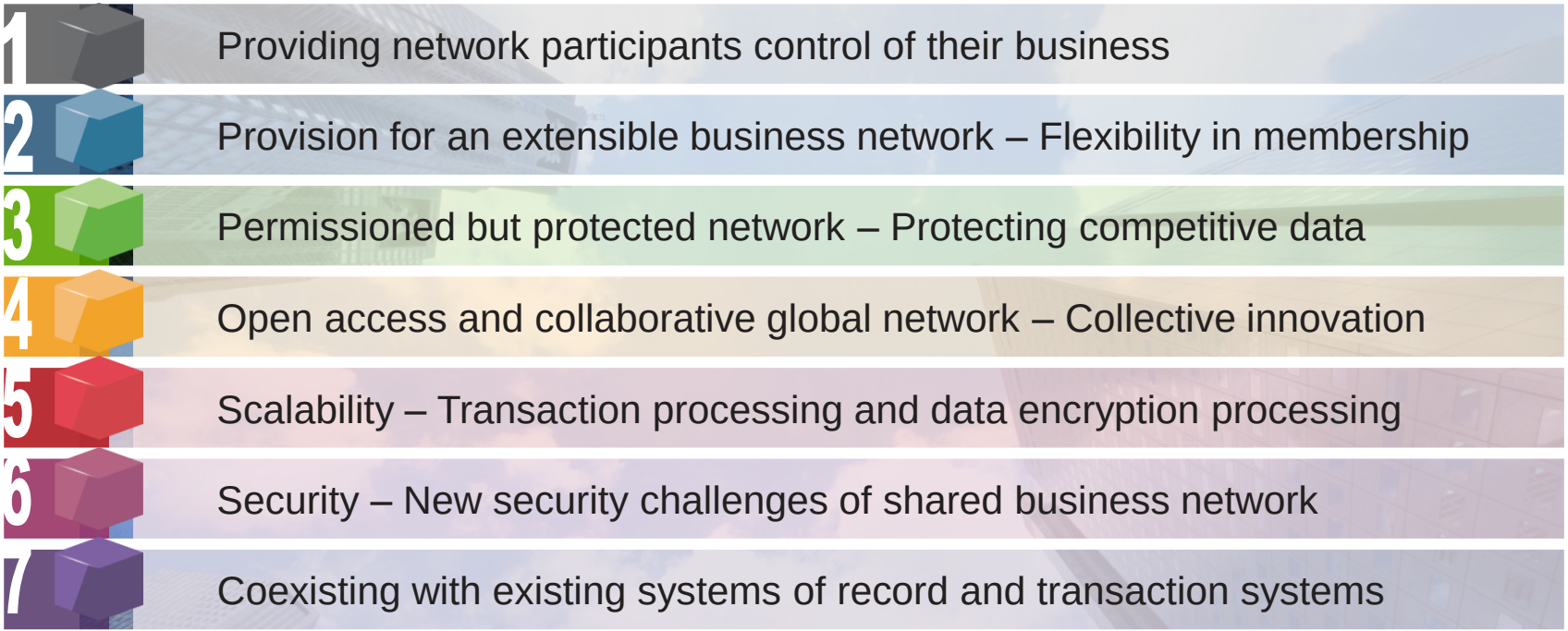
www.hyperledger.org

Strategic Client Blockchain Adoption Journey

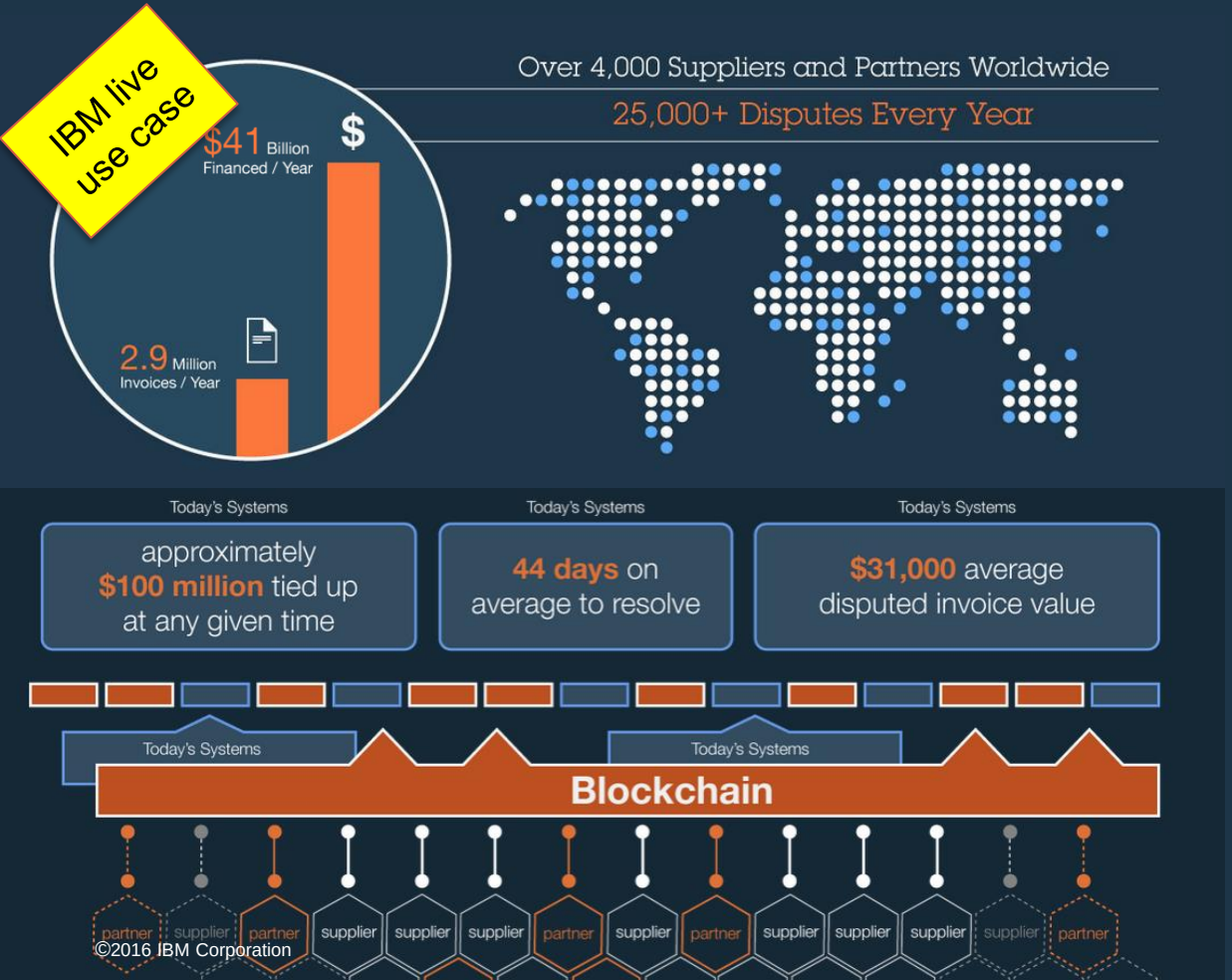
A multi-stage approach, delivering incremental and new business value at each milestone



Seven design principles of sustainable Blockchain business networks



IBM Global Finances's blockchain solution addresses disputes management



Blockchain utilized to significantly improve resolution time for common disputes

- Utilized data available from suppliers to deliver enhanced information to both Suppliers and Business Partners
- Accomplished with no code changes to our core Commercial Financing system using shadow ledger approach
- Integrated Blockchain into existing user interface
- Enhanced data includes key information regarding shipments status which minimizes proof of delivery disputes
- Established a 'platform' for competitive advantage
- Continuing to work with our Suppliers and Business Partners to further expand blockchain capabilities

IBS intelligence

CLS Group plans pa blockchain technolc

Risk management and FX vendor CLS Gr underpinned by a distributed ledger platfor

The global FX market, the company recko sparsity leads to inconsistent and bespoke intra-day liquidity demands.

Participants in CLS Netting will be able to currencies. They will also have specialisec and its Fabric solution. CLS is also collabc scalable service."

FINANCE MAGNATES

News Intelligen

27/09/2016|09:55 GMT

Photo: FM

CLS Group (CLS) has just announced netting service for buy-side and sell-sic settled outside the CLS settlement ser submit FX instructions for six prod forwards (NDFs), and 24 currencie channels. They will also have the opti

CLS to use Hyperledger Fabric for new payment netting service

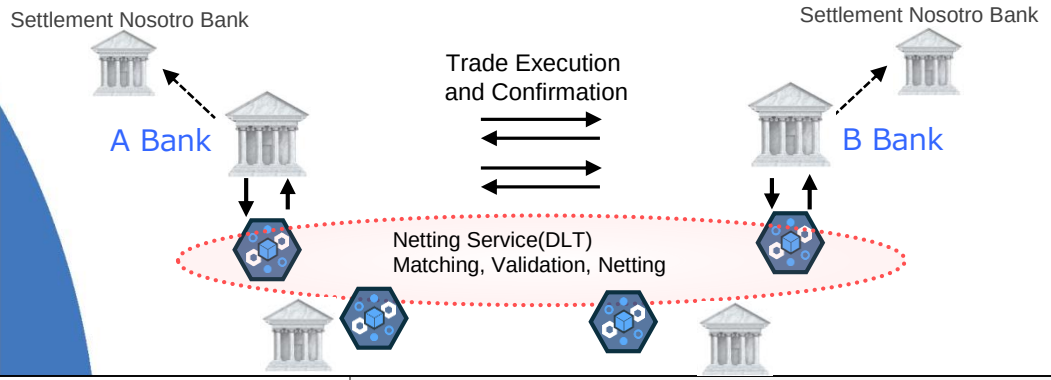
27 September 2016 | 5222 views | 0

Multi-bank foreign exchange counterparty CLS is to build a payments netting service for trades settled outside the core membership using distributed ledger technology based on Hyperledger Fabric.

CLS says it wants to use its position to standardise and expand bilateral payment netting capabilities for the entire FX market, eliminating intra-day liquidity demands caused by inefficient bespoke approaches to netting throughout the market.

The company has signed up 14 banks as early adopters of the proposed service, which will accept FX instructions for six products, including non-deliverable forwards (NDFs), and 24 currencies over existing Swift-based channels.

Example: FX Netting



What

- Lack of standardized payment netting process for trades not settled within current CLS PvP environment
- Institutions intervene manually and inconsistently to complete netting process
- Higher costs and increased intra-day liquidity demands

How

- CLSNet will match FX instructions based on the same matching principles as the CLS core settlement service and will send a match notification to each counterparty. Allows all counter-parties to have the same validated record of transaction and fulfillment

Benefits

1. It will enable them to submit FX instructions for six different products
2. When the product launches CLS will support 24 different currencies vs the 18 currencies it already offers

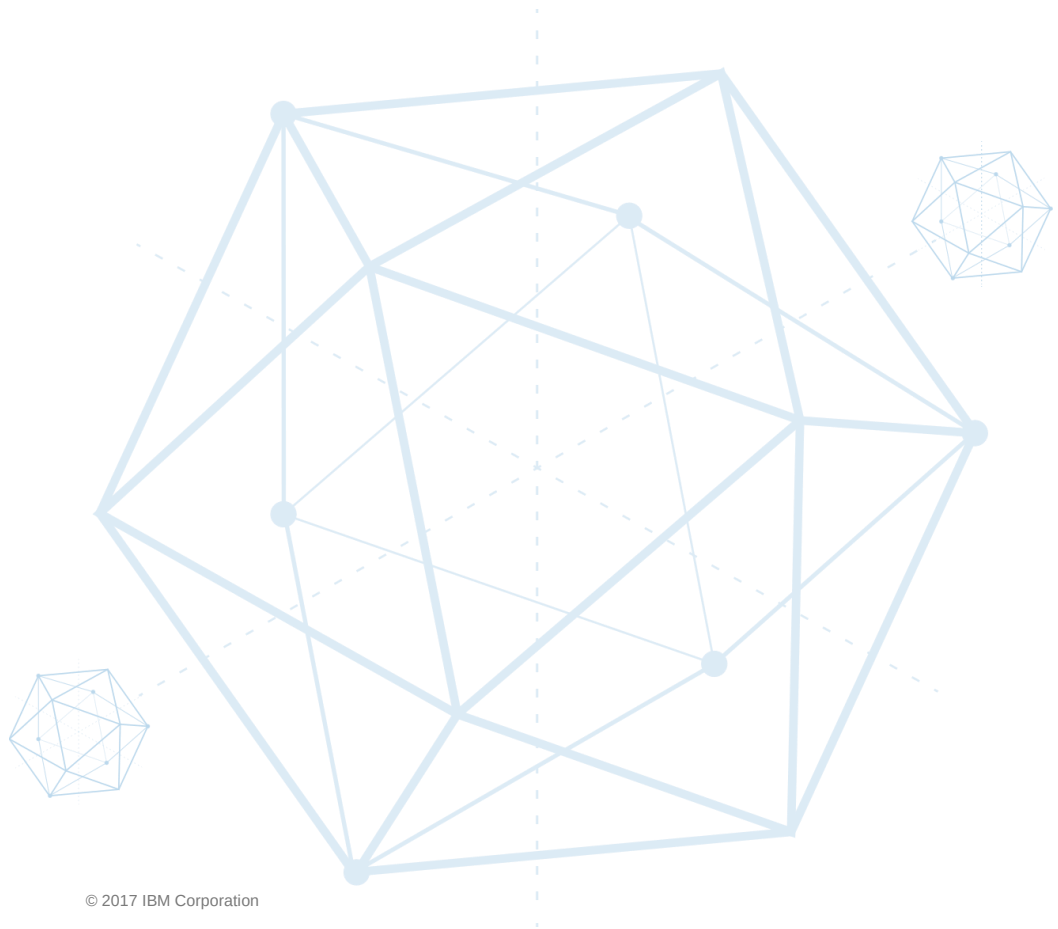
Current business issues in the global distribution market

Banks	Importers and Exporters	Carriers	Forwarders	Ports	Authorities
Manual, paper-based processes Lack of Real-Time information	Excess Inventory Manual, paper-based processes Duplication of Administrative Process	No single version of ”the Truth” Manual, paper-based processes	Manual Data Collection Manual, paper-based processes	Collection and Delivery Black Holes Sub-optimal stack placement Manual Data Collection	False Positives Lack of visibility pre-manifest Lack of visibility into land move-ment before / after ocean transport

Root Causes:

Multiple data formats *Point-to-point interactions* *Absence of messaging standards*

Embedded video: MAERSK Blockchain can be found at https://www.youtube.com/watch?v=tdhpYQCWnCw



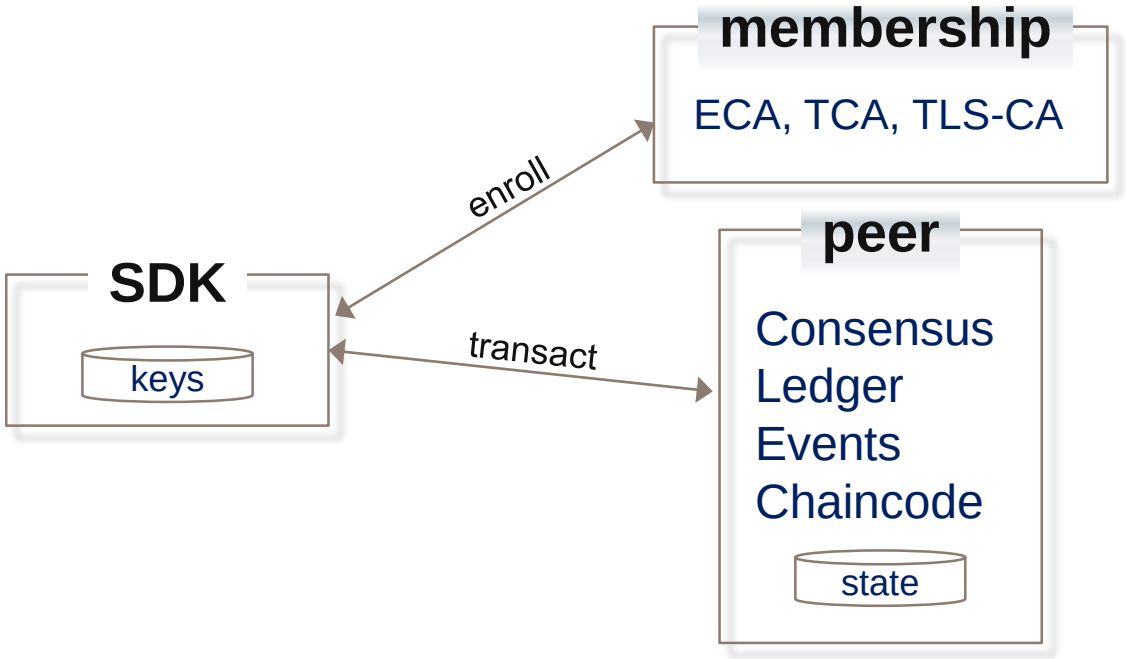
Hyperledger Fabric

Into v1

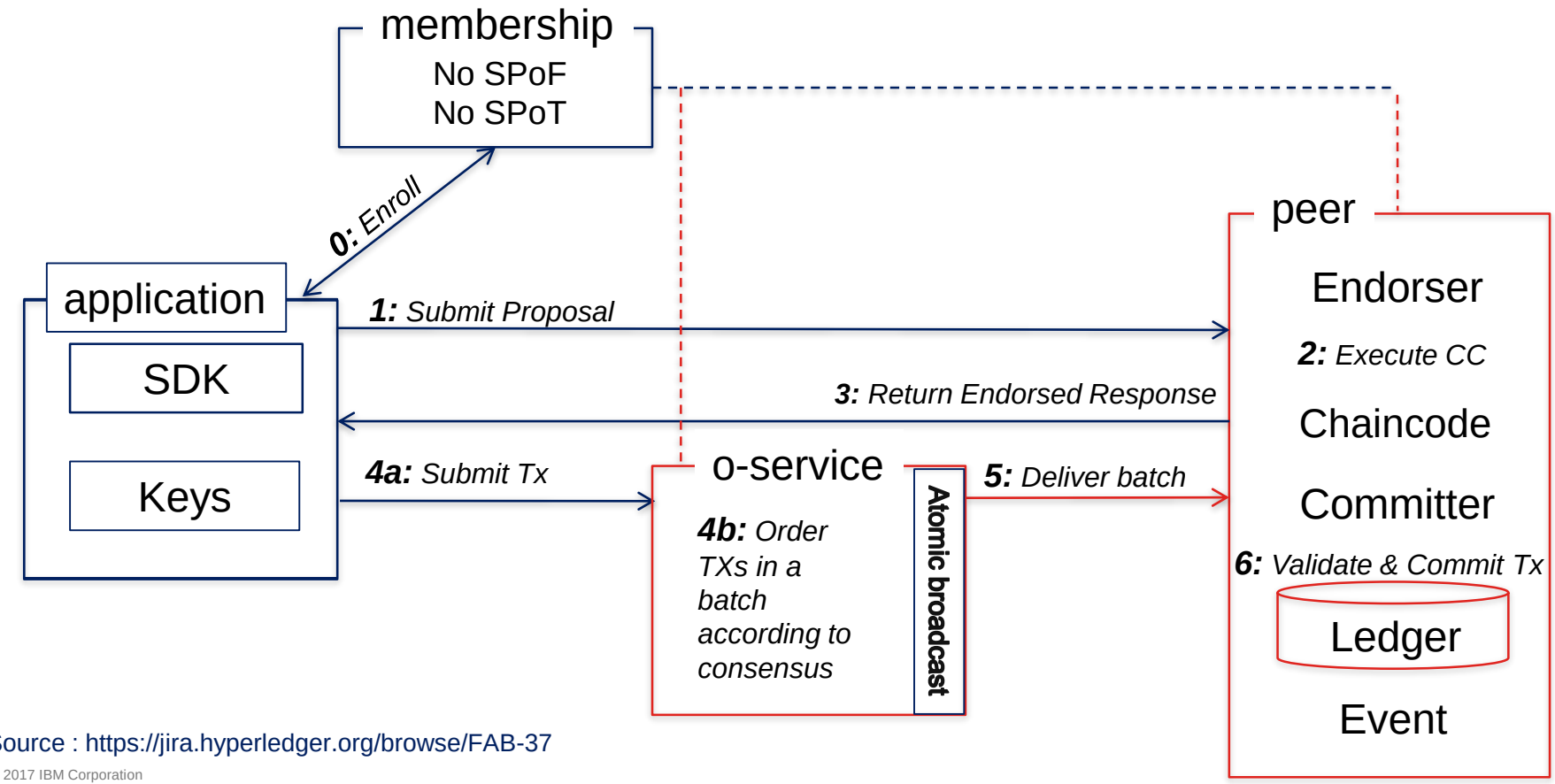
Embedded Video Hyperledger v0.6 Lessons Learned Video can be found at

https://www.youtube.com/watch?v=EKa5Gh9whgU

Architecture of Hyperledger Fabric v0.6

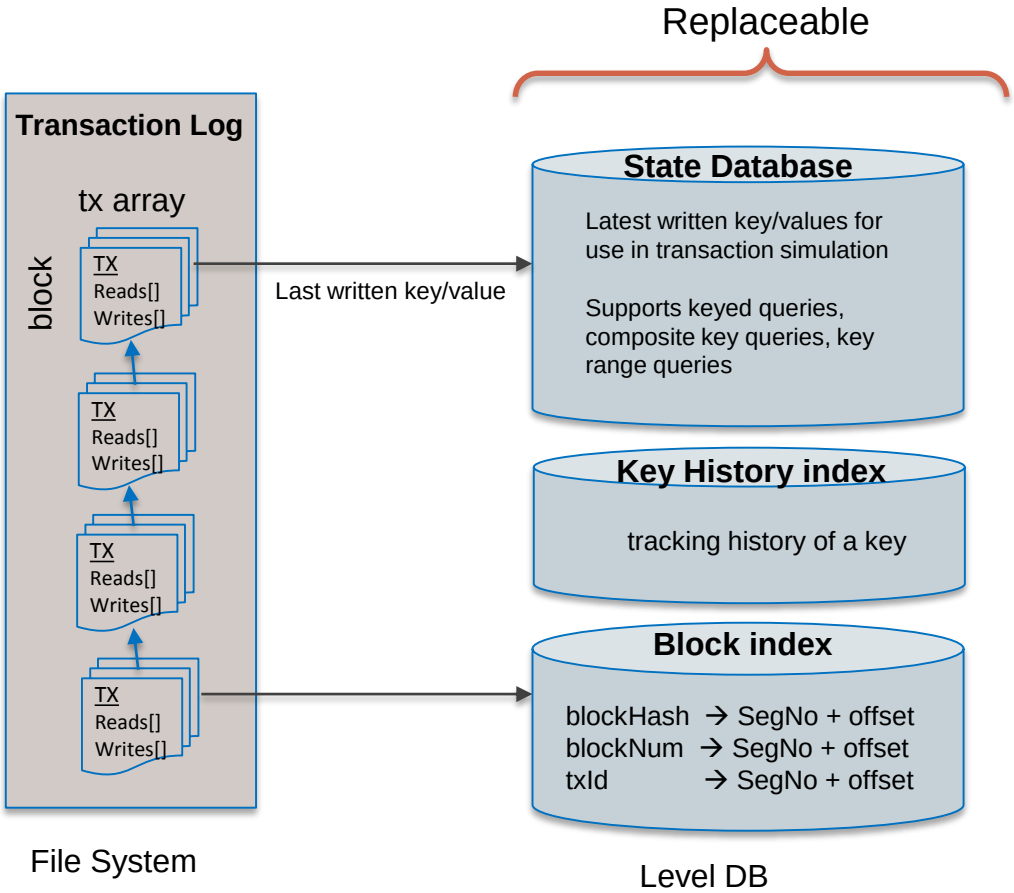


Architecture of Hyperledger Fabric v1



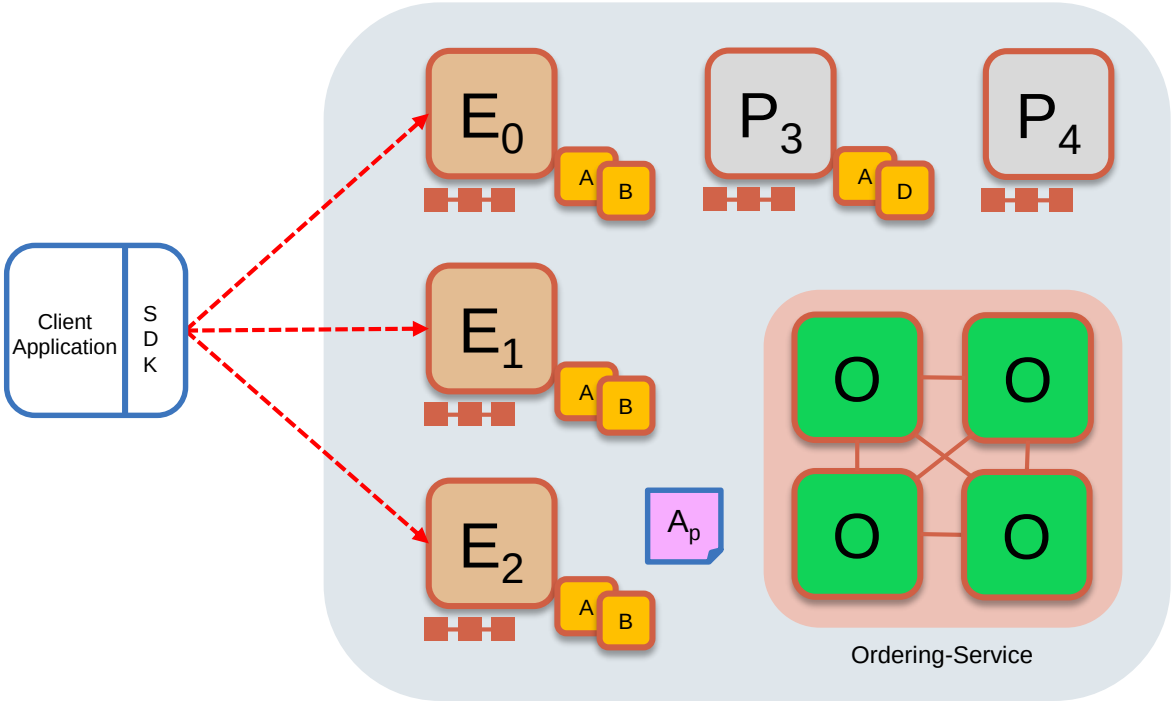
Source : https://jira.hyperledger.org/browse/FAB-37

Ledger representation



CouchDB (external option) supports keyed queries, composite key queries, key range queries, plus full data rich queries (beta in 1.0)

Sample transaction: Step 1/7 – Propose transaction

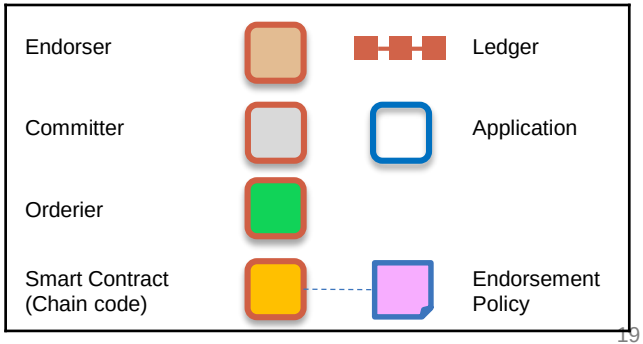


Application proposes transaction

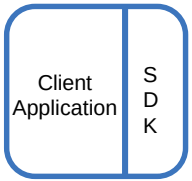
Endorsement policy:
- "**E_0**, **E_1** and **E_2** must sign"
- (**P_3**, **P_4** are not part of the policy)

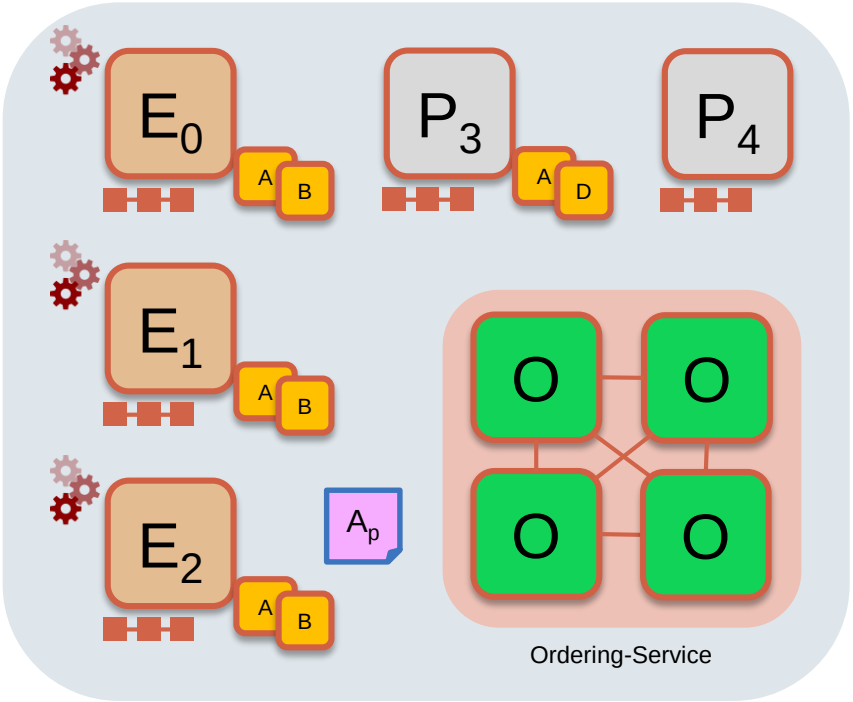
Client application submits a transaction proposal for **chaincode A.** It must target the required peers {**E_0**, **E_1**, **E_2**}

Key:



Sample transaction: Step 2/7 – Execute proposal



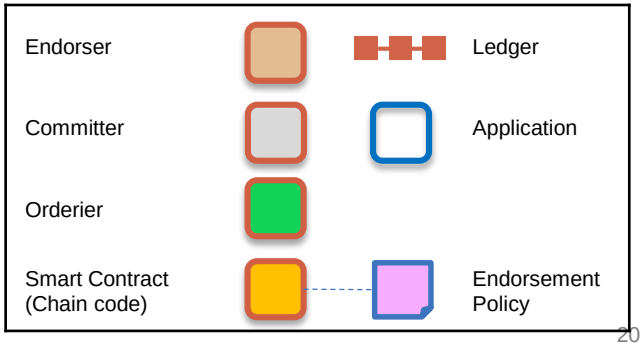


Endorsers Execute Proposals

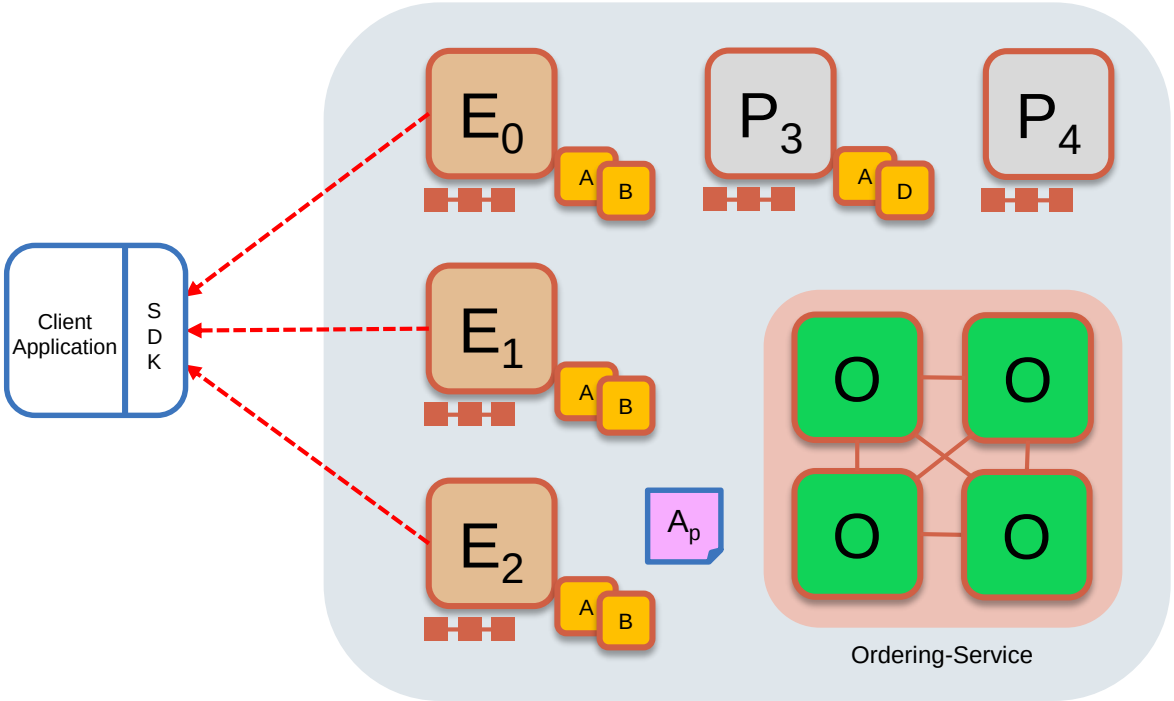
E_0, **E_1** & **E_2** will each execute the *proposed* transaction. None of these executions will update the ledger

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

Key:



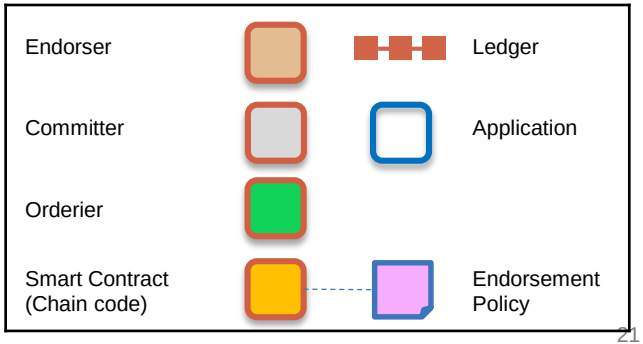
Sample transaction: Step 3/7 – Proposal Response



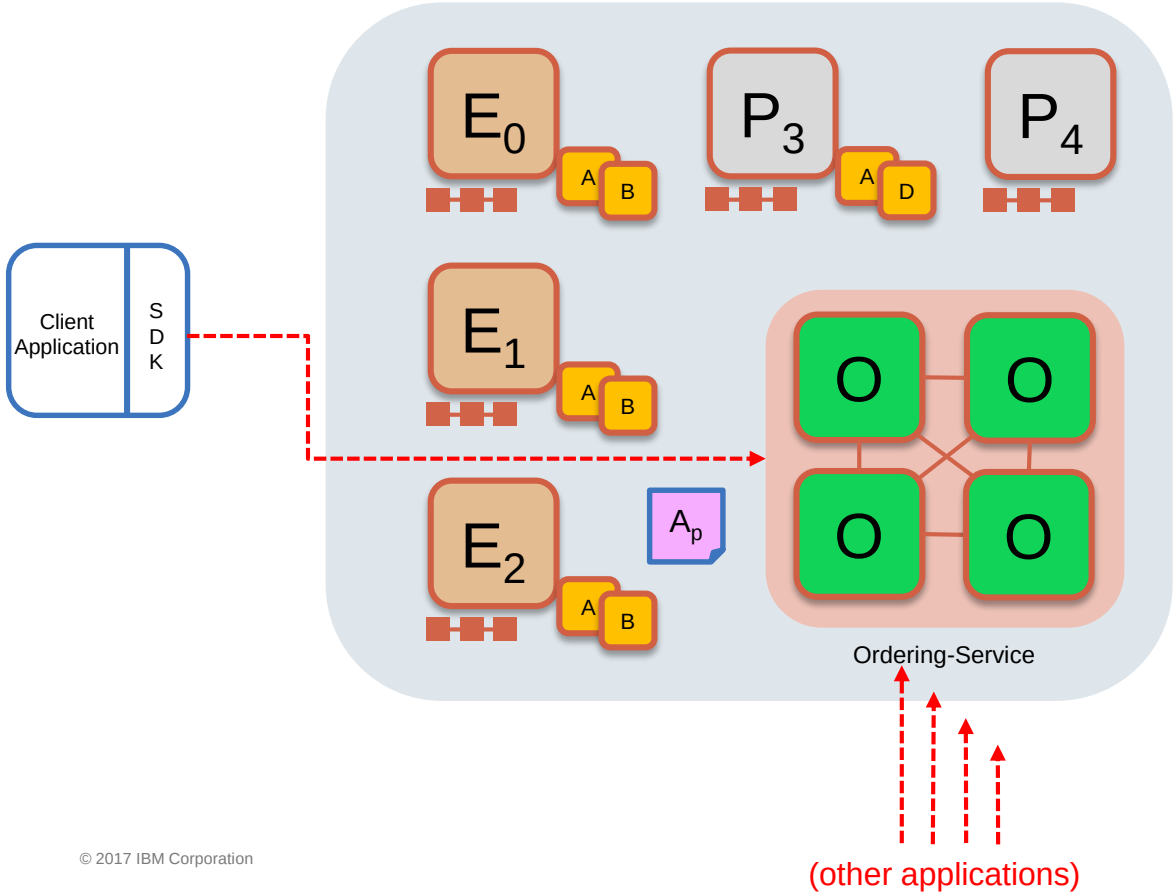
Application receives responses

The RW sets are signed by each endorser and returned to the application

Key:



Sample transaction: Step 4/7 – Order Transaction

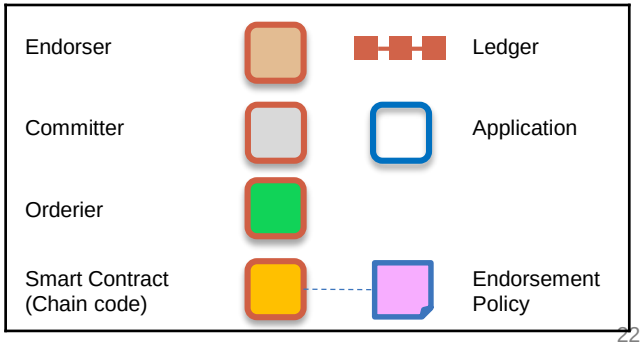


Application submits responses for ordering

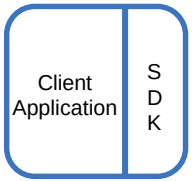
Application submits responses as a **transaction** to be ordered.

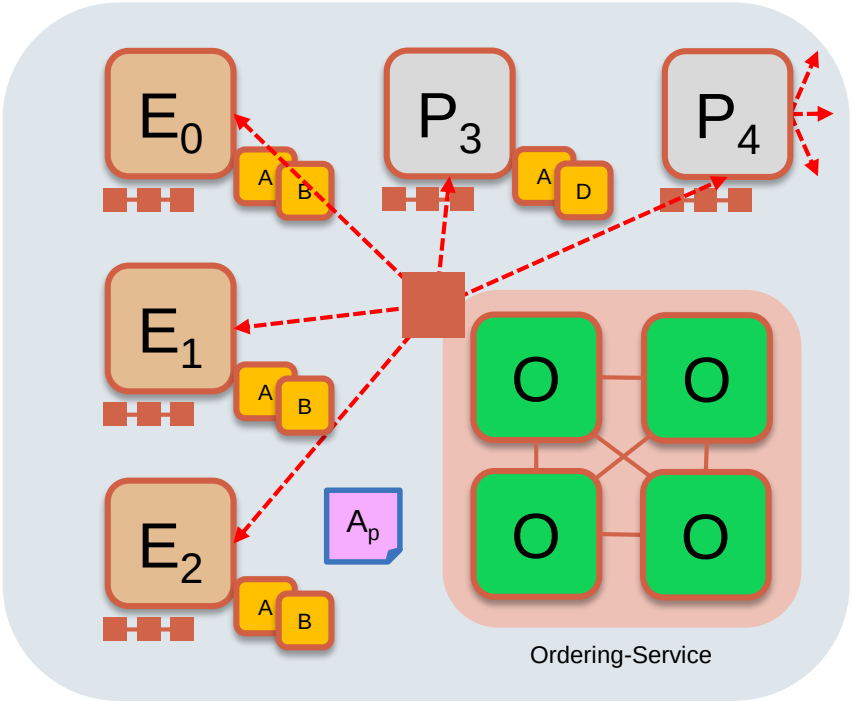
Ordering happens across the fabric in parallel with transactions submitted by other applications

Key:



Sample transaction: Step 5/7 – Deliver Transaction





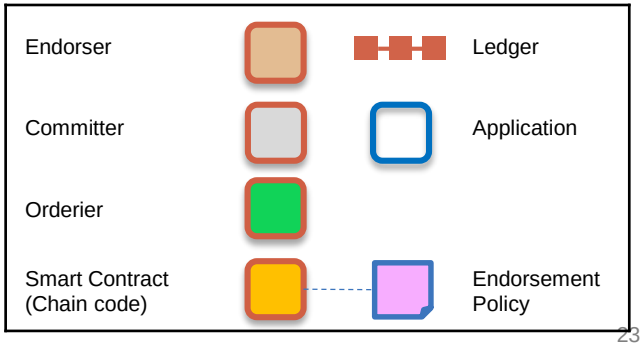
Orderer delivers to all committing peers

Ordering service collects transactions into blocks for distribution to committing peers. Peers can deliver to other peers using gossip (not shown)

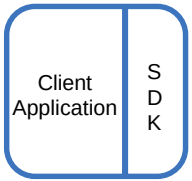
Different ordering algorithms available:

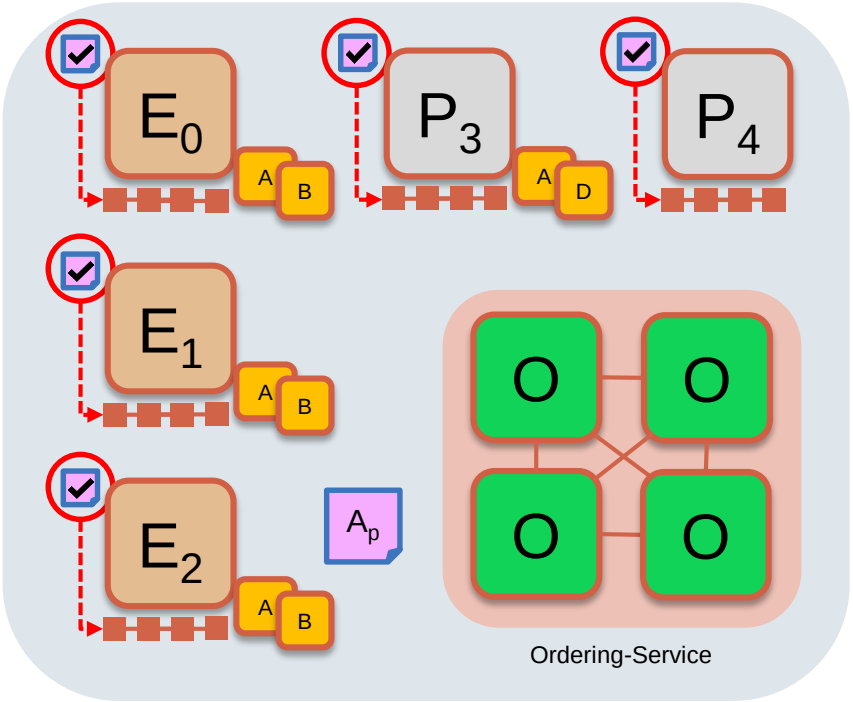
- SOLO (single node, development)
- Kafka (blocks map to topics)
- SBFT (tolerates faulty peers, future)

Key:



Sample transaction: Step 6/7 – Validate Transaction



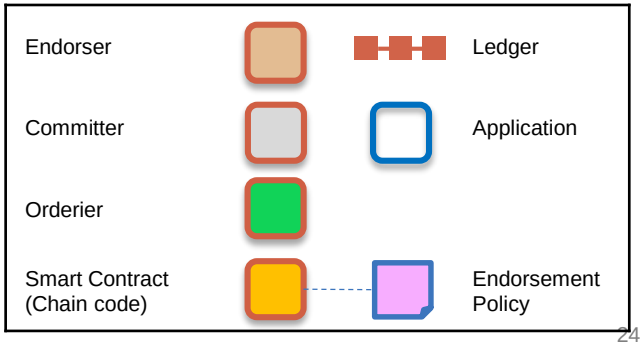


Committing peers validate transactions

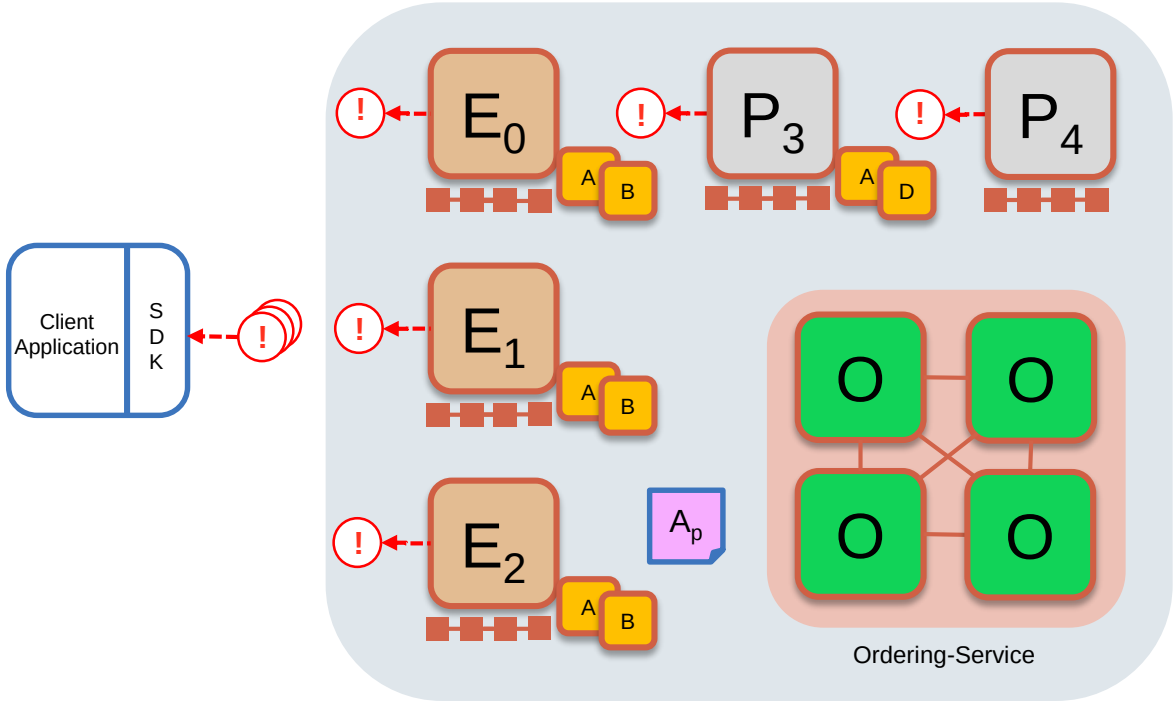
Every committing peer validates against the endorsement policy. Also check RW sets are still valid for the current state

Transactions are written to the ledger and update caching DBs with validated transactions

Key:



Sample transaction: Step 7/7 – Notify Transaction

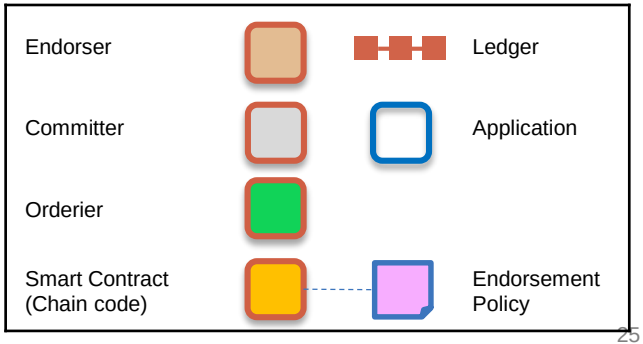


Committing peers notify applications

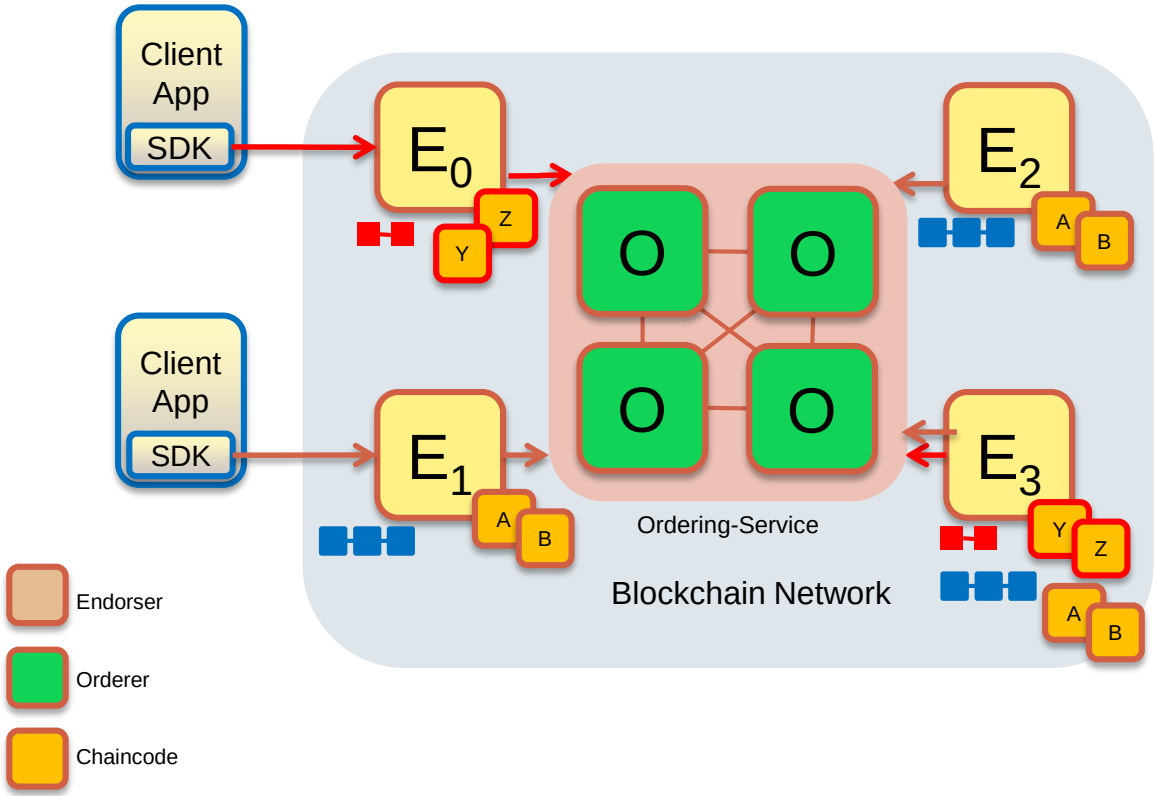
Applications can register to be notified when transactions succeed or fail, and when blocks are added to the ledger

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

Key:



Example of Multi-Channel with Endorsement



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

Where to Ask Questions

Hyperledger Community has moved off Slack to RocketChat.

Go to chat.hyperledger.org and register.

You will be required to have a linux foundation ID however. If you aren't registered with the Linux Foundation, get an ID from https://identity.linuxfoundation.org/

For questions on Version 1.0, go to the fabric-questions channel.

Also every day, the docker build status is posted when passing the continuous integration tests will be posted on fabric-ci (only posted when tests pass)

Useful Information To Get You Started

- **Documentation actively getting updated as we progress:** http://hyperledger-fabric.readthedocs.io/en/latest/

- **Support for Docker images for easy deployment for Hyperledger-fabric 1.0.** Docker images will be available for all major components to run a network (peers, solo orderer, CLI, CA, Kafka, CouchDB). A “Getting started” section will be available in the Hyperledger-fabric publications. Getting started will help a developer or user to start the network, run a simple application , and learn the basics of running v1. See: http://hyperledger-fabric.readthedocs.io/en/latest/

- **Support for a tool that helps bootstrap a network.** The bootstrap network tool is available and called the Configuration Transaction Generator (configtxgen). The tool is designed to configure the network with organizations included in the ordering service genesis block and generates the configuration transaction artifacts used for channel creation.

Thank You!

Thomas Hartmann
Expert Consultant

IBM Global Business Services

IBM Deutschland GmbH
Wilhelm-Fay-Straße 30-34
65936 Frankfurt

Mobile: +49 171 2234 153
Phone: +49 7034 274 0608

Mail: thomas.hartmann@de.ibm.com