

COMP4137 Blockchain Technology and Applications
COMP7200 Blockchain Technology

Lecturer: Dr. Hong-Ning Dai (Henry)

Lecture 9

Permissioned Blockchains

Outline

- **Permissionless and Permissioned Blockchain**
- Introduction to Hyperledger
- Introduction to Hyperledger Fabric
- Hyperledger Fabric Key Components
 - Membership
 - Ledger
 - Chaincode
 - Privacy
 - Peers
 - Consensus



Permissionless and Permissioned Blockchain

- Blockchain essentially is a **distributed ledger** storing a list of records (Txns)
 - Blocks are cryptographically linked and secured
- The initial implementations of blockchain are cryptocurrencies, such as Bitcoin
 - **Public networks**: anyone can join as a node and start mining
 - **Public transactions**: anyone can query Txns performed by a wallet
- This is not suitable for **business scenarios**
 - Want to use blockchain and keep the transactions **private**
 - For example, logistics company Maersk uses blockchain to track shipping logistics but is unwilling to store confidential business information on a permissionless blockchain
- Hence, **permissioned** blockchains were invented

Similarities

- Common characteristics of these blockchains:
- Both are **distributed ledgers**
 - Multiple versions of the same data will be stored in different places and connected through network
- Both are theoretically **immutable**
 - Stored data cannot be modified without controlling sufficient power over the network
- Both make use of **consensus mechanisms**
 - They have a way for multiple versions of the ledger to reach an agreement on the content

Permissionless Blockchain

- Popular blockchains such as Bitcoin and Ethereum fall under this category
- Also known as **public blockchain**
 - Allow anyone to transact and join as a validator
- Data on blockchain is **publicly available**, and complete copies of the ledger are stored across the network
 - This is what makes it hard to hack these systems
- This blockchain cannot be controlled by anyone, and users can remain **anonymous**
 - Since no need for identifying themselves to get an address and perform transactions

Features

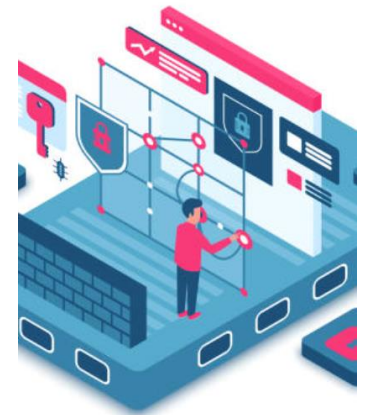
- Digital Assets
 - There is a user-incentivizing **token** that can increase or decrease in value
- Transparency
 - A transparent network could give **users access to all information except private keys**, including addresses, transactions, and the way in which transactions are processed into blocks.
- Decentralization
 - **No central entity** can shut down the network, change its protocols, or edit the ledger.
 - Systems are based on **consensus protocols**. Network changes of any type can be achieved only if 51% of the users agree.

Advantages and Disadvantages

- Advantages:
 - Reliable and security because of the large amount of nodes
 - Anyone can access the ledger and check the correctness of Txs
 - Anyone can use it without creating additional infrastructure
- Disadvantages:
 - Low efficiency
 - High energy consumption
 - 51% attack risk

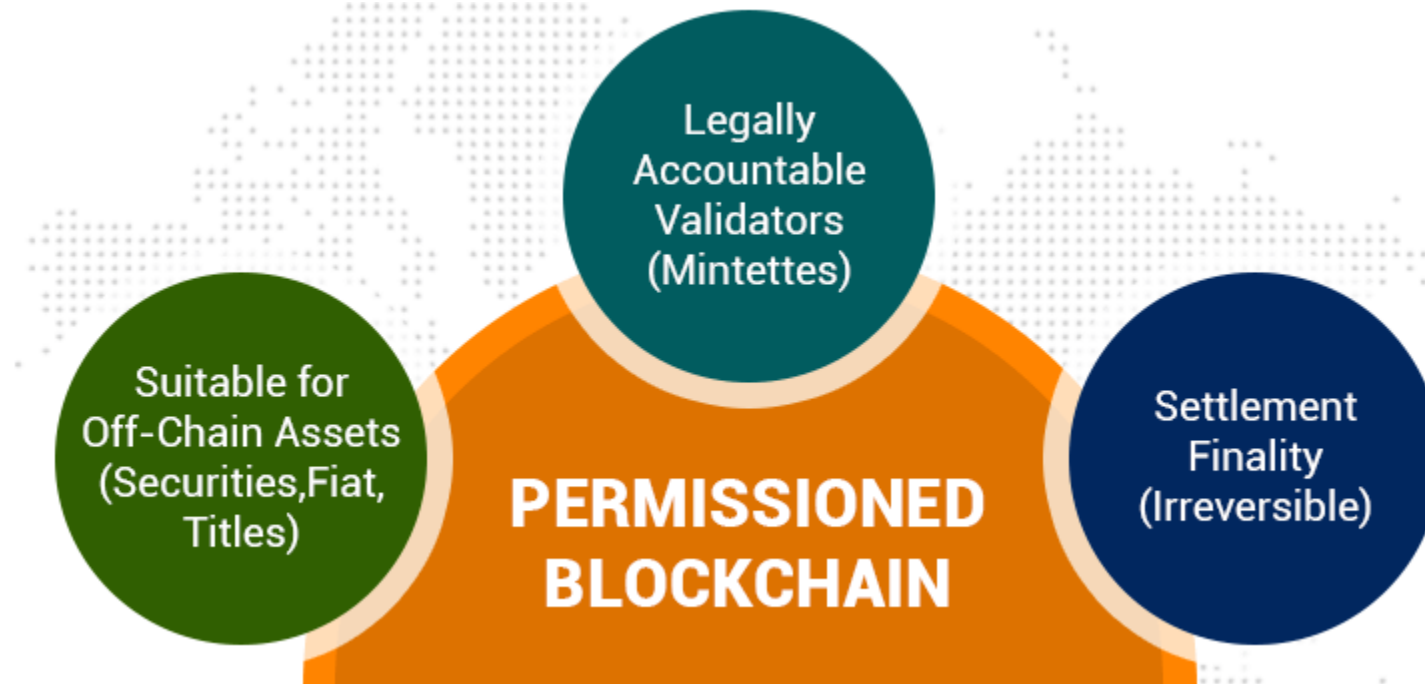
Permissioned Blockchain

- **Permissioned blockchain** provides an **access** control layer on top of the blockchain
 - After getting approval from a central authority, users can join the network to validate Tx's or view data.
- If a permissioned blockchain is deployed only within one organization and used by its various departments, it will be a **private blockchain**
 - For example, a bank may be running a private blockchain operated through some internal nodes (e.g., branches).



Permissioned Blockchain

- Useful for companies, banks, and institutions
 - They are comfortable to comply with the **regulations**.
 - They are concerned about having **complete control of their data**.



Features

- Transparency and Anonymity
 - **Alternative**, depending on the concrete business demands
- Varying Decentralization
 - Members in the network can **negotiate** and decide about the **level of decentralization**.
 - **Private** blockchains can be fully centralized or partially decentralized by choosing suitable consensus algorithms.
- Governance
 - Governance is **decided by the members** in the business network.
 - There are **dynamics** to determine how decisions are made on a central level (can be based on consensus or not).

Advantages and Disadvantages

- Advantages:

- More efficient performance
- More prone to business regulation
- Highly customized
- Access control
- Better scalability

- Disadvantages:

- Security depends on the integrity of its members
- Less transparent
- Vulnerable to hacks and manipulation
- Less anonymous

Outline

- Permissionless and Permissioned Blockchain
- **Introduction to Hyperledger**
- Introduction to Hyperledger Fabric
- Hyperledger Fabric Key Components
 - Membership
 - Ledger
 - Chaincode
 - Privacy
 - Peers
 - Consensus



Hyperledger

Open source
collaborative effort to
advance **cross-**
industry blockchain
technologies

Hosted by **The Linux
Foundation**, fastest-
growing project in the
Foundation's history

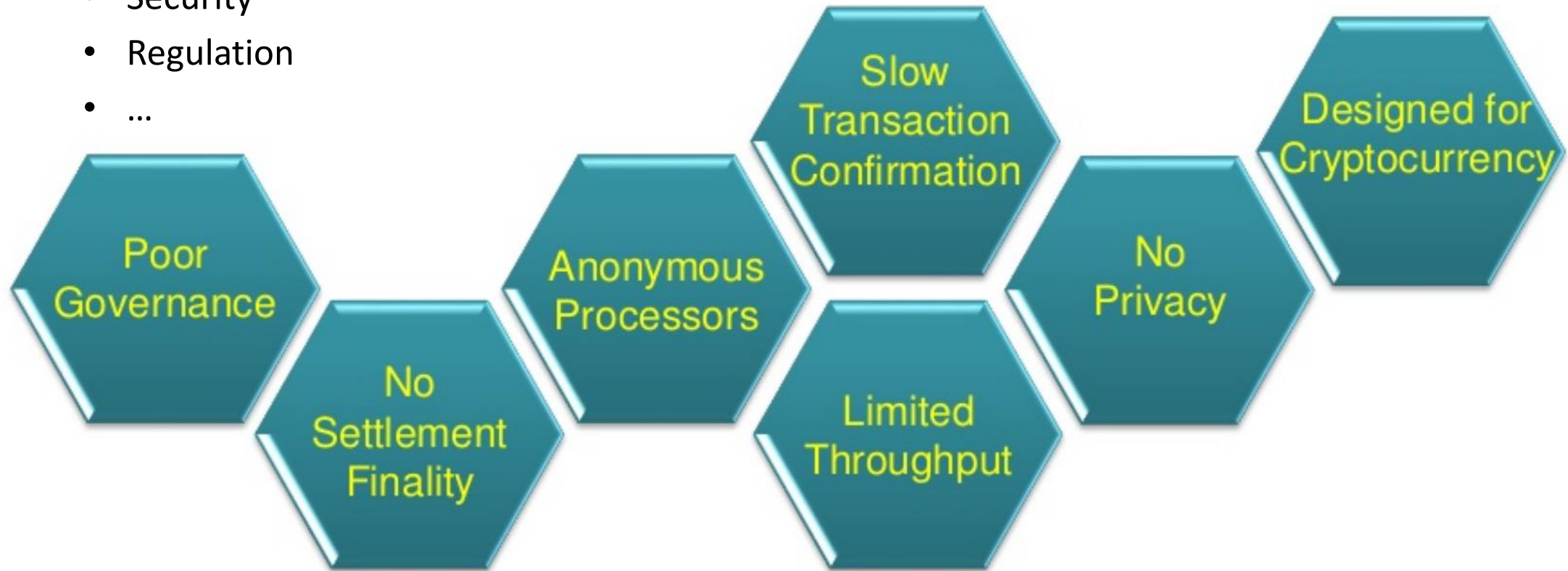
Global collaboration
spanning finance,
banking, IoT, supply
chains, healthcare,
manufacturing,
technology & more

<https://www.hyperledger.org/about>

Overcome Shortcomings

■ Shortcomings of Existing Blockchains

- Performance
- Security
- Regulation
- ...

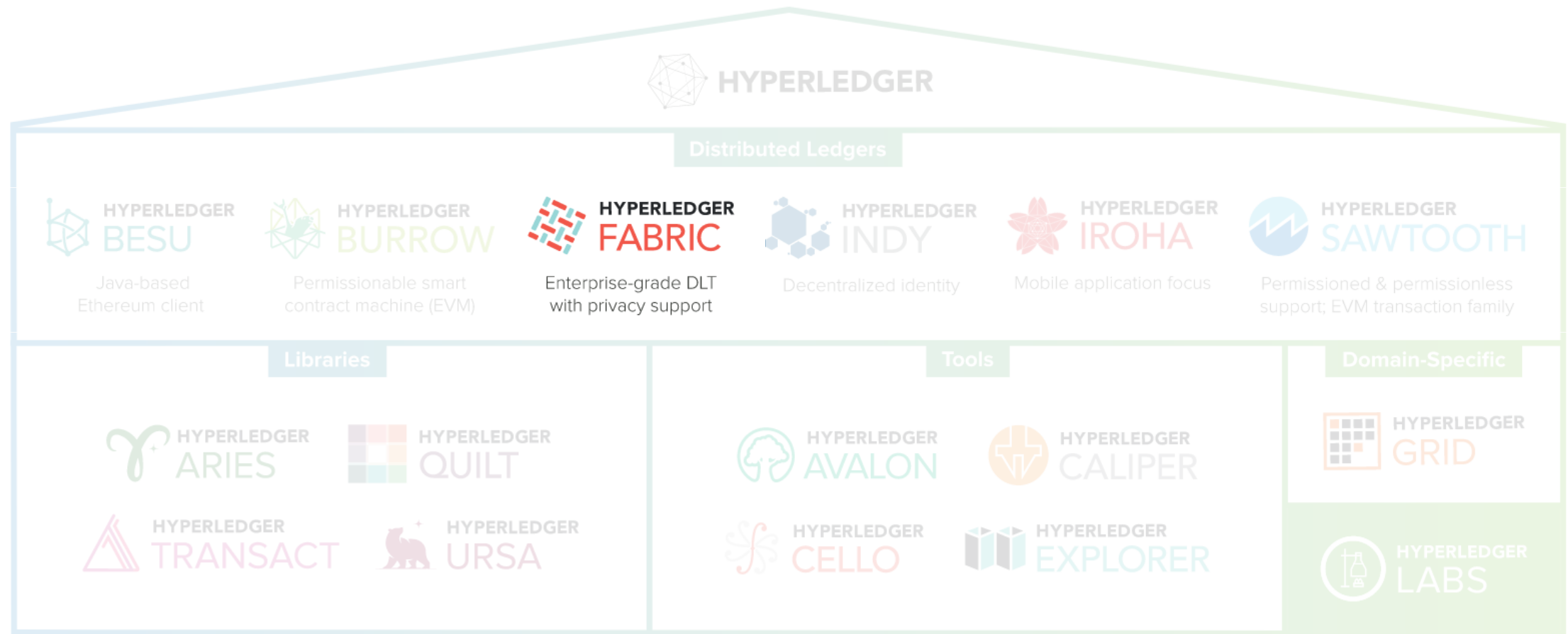


Shared Ledger Database

- Blockchain allows multiple different parties to securely interact with the same universal source of truth.



Hyperledger Modular Umbrella Approach



Benefits

- **Flexible** Modification of Any Component
- **Common** Functional Modules and Defined Interfaces
- **Re-use** of Common Building Blocks
- **Extensible** Codebases
- **Diverse** Developer Community
- **Rapid** Experimentation

Outline

- Permissionless and Permissioned Blockchain
- Introduction to Hyperledger
- Introduction to Hyperledger Fabric
- Hyperledger Fabric Key Components
 - Membership
 - Ledger
 - Chaincode
 - Privacy
 - Peers
 - Consensus



Fabric: Distributed Ledger Technology (DLT)

- A distributed ledger is a type of **data structure**, which resides across **multiple computers, locations, or regions**
 - While distributed ledgers existed prior to Bitcoin, Bitcoin blockchain marks the fusion of a host of technologies, including timestamping of Txns, Peer-to-Peer (P2P) networks, cryptography, and shared computational power, along with a new consensus algorithm.
- DLT generally consists of three basic components:
 - A **data model** that captures the current state of the ledger – blockchain
 - A **language of transactions** that changes the ledger state – smart contracts
 - A **protocol** that builds **consensus** among participants about accepted transactions and their order

Hyperledger Fabric

- Hyperledger Fabric is a platform to develop **distributed applications** with a modular and secure architecture



Permissioned network

Collectively define **membership & access rights** within the business network



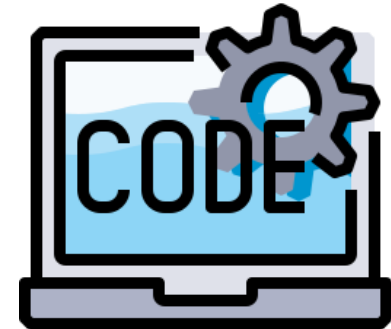
Confidential transactions

Give businesses the flexibility & security to make **transactions visible to select parties** with the correct encryption keys



No cryptocurrency

Require **no mining and expensive computations** to assure transactions



Programmable

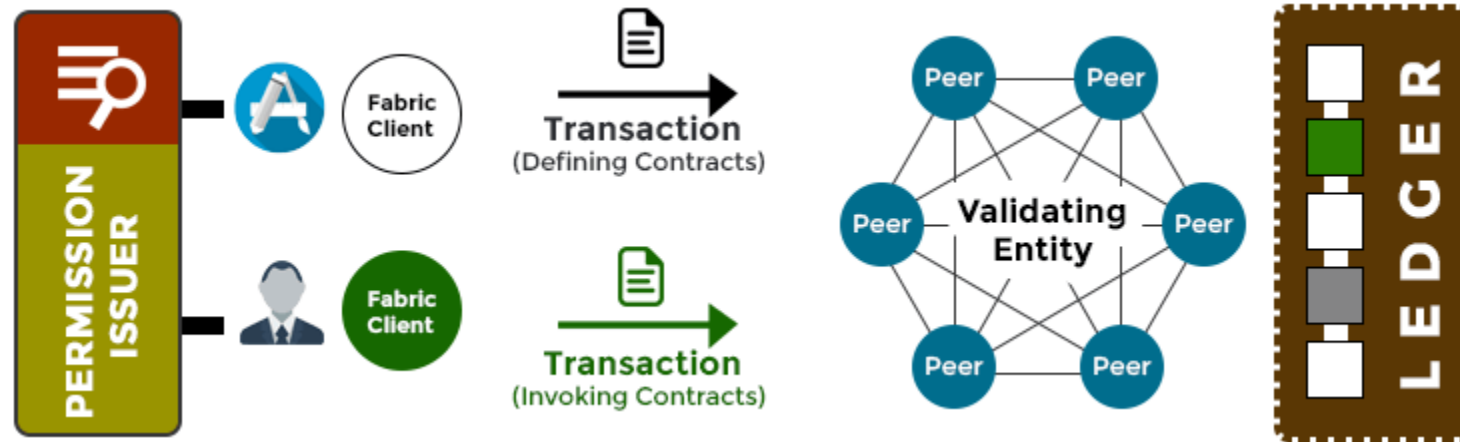
Leverage the embedded logic in **smart contracts** to automate business processes across the network

Cryptocurrency-agnostic

- Independent and agnostic of all alt-coins, cryptocurrencies, and tokens
- Hyperledger exists to create enterprise blockchain software, not to manage any cryptocurrency
- Having said that, the design philosophy includes the capability of creating a token used to manage digital objects, which may represent currencies, although this is not required for the network to operate

Modular and pluggable

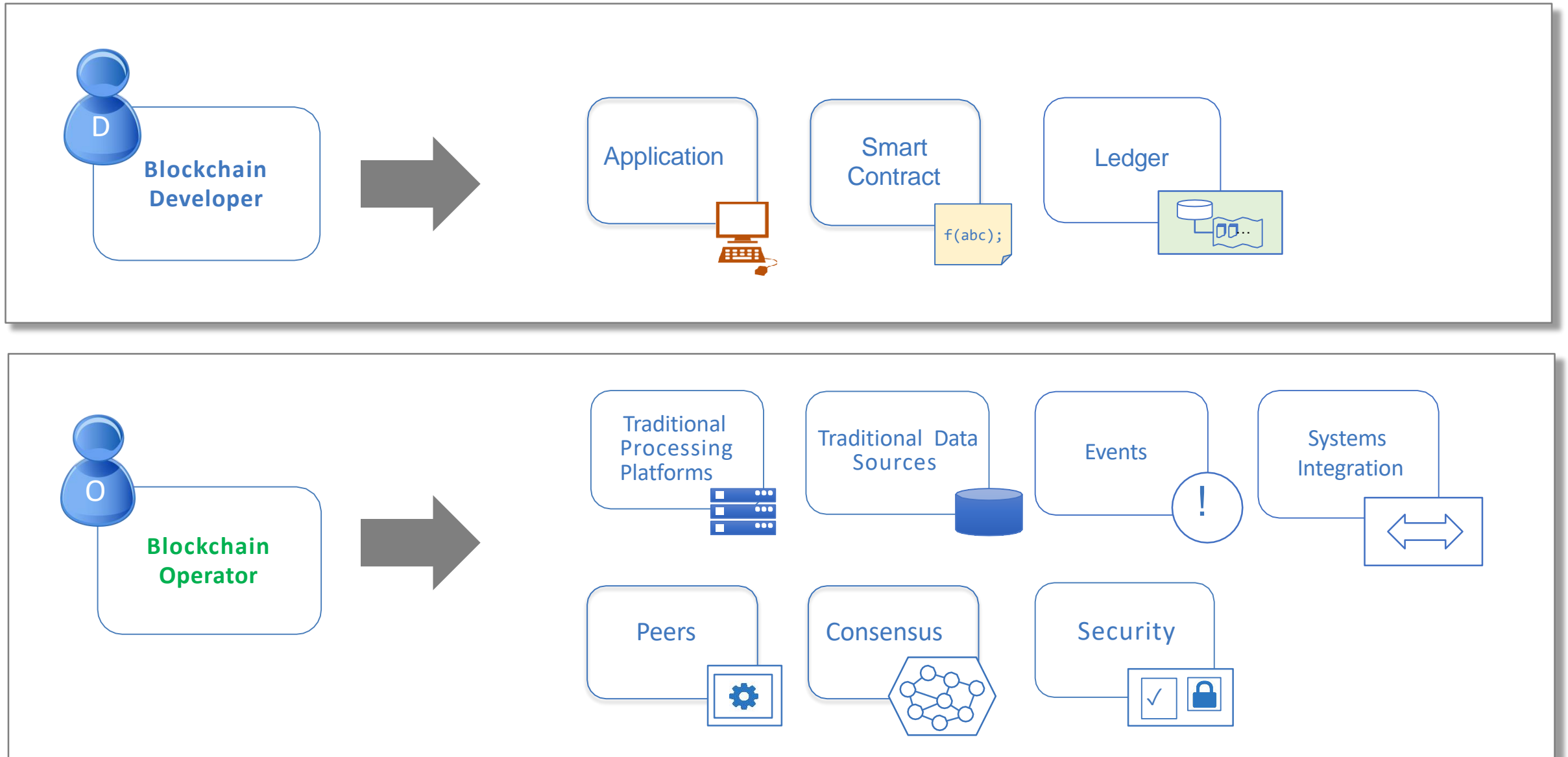
- Hyperledger Fabric offers several pluggable options
 - Ledger data can be stored in multiple formats
 - Consensus mechanisms can be changed
 - Different Membership Service Providers (MSPs) are supported
 - Endorsement and validation policy can be adjusted as need



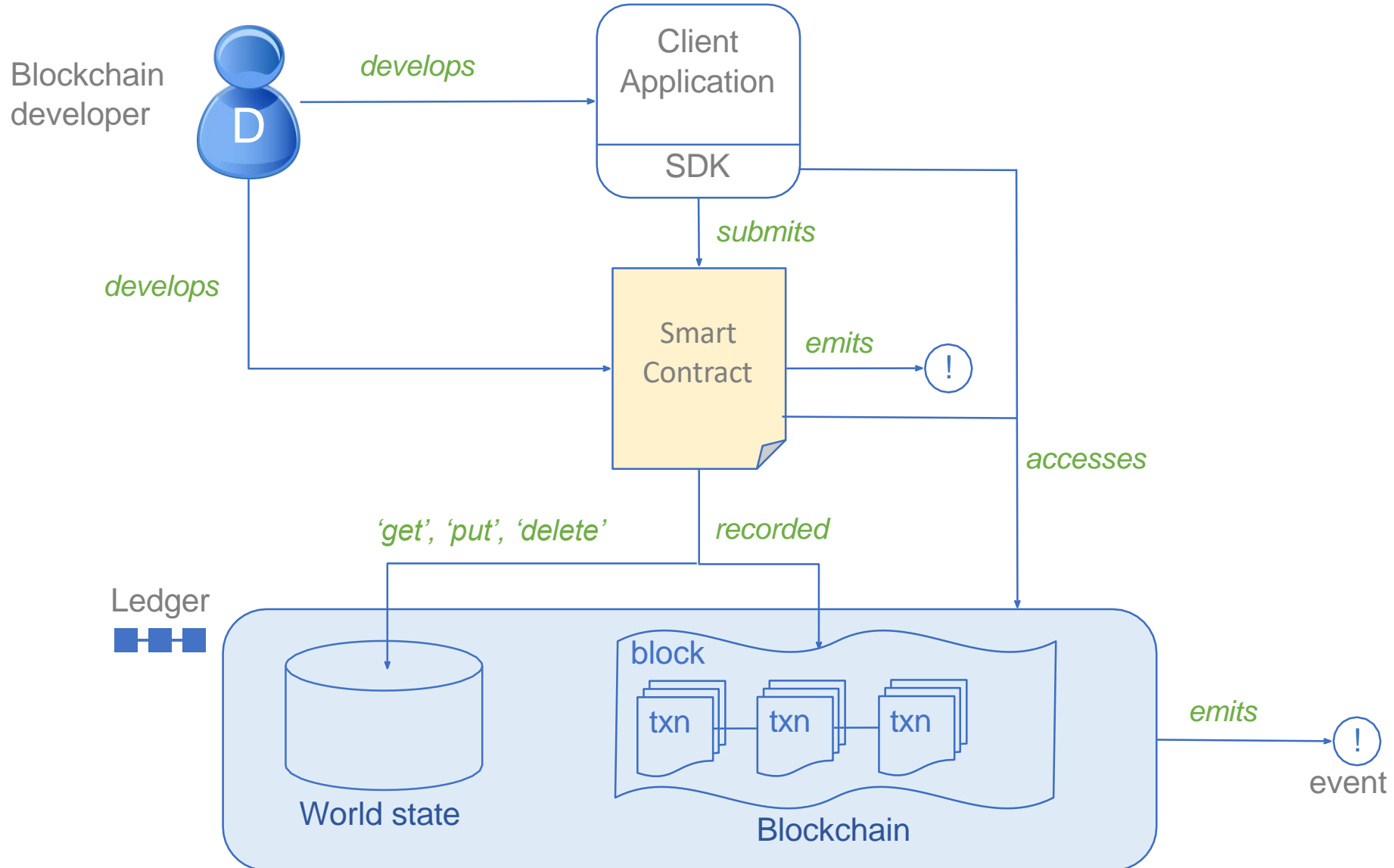
Comparison

	Bitcoin	Ethereum	Hyperledger Frameworks
Cryptocurrency-based	Yes	Yes	No
Permissioned	No	No	Yes
Pseudo-anonymous	Yes	Yes	No
Auditable	Yes	Yes	Yes
Modularity	No	No	Yes
Smart contracts	No	Yes	Yes
Consensus protocol	PoW	PoW/PoS	Various

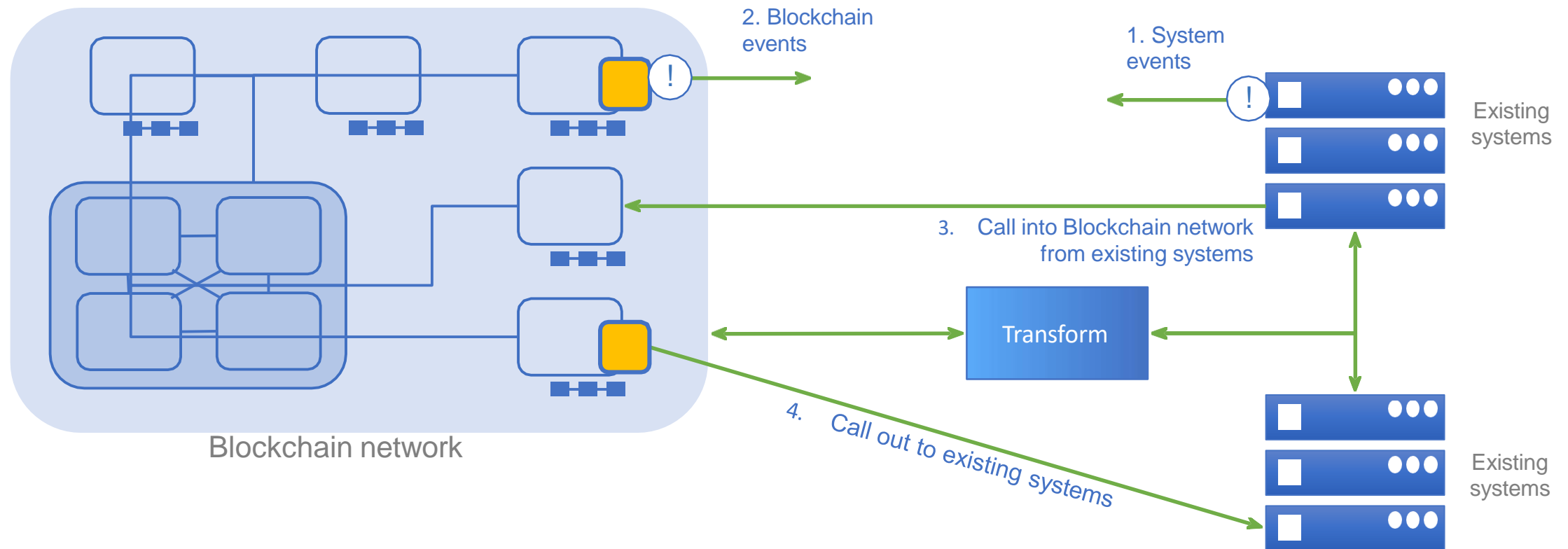
Fabric key actors and their domains



How applications interact with the ledger

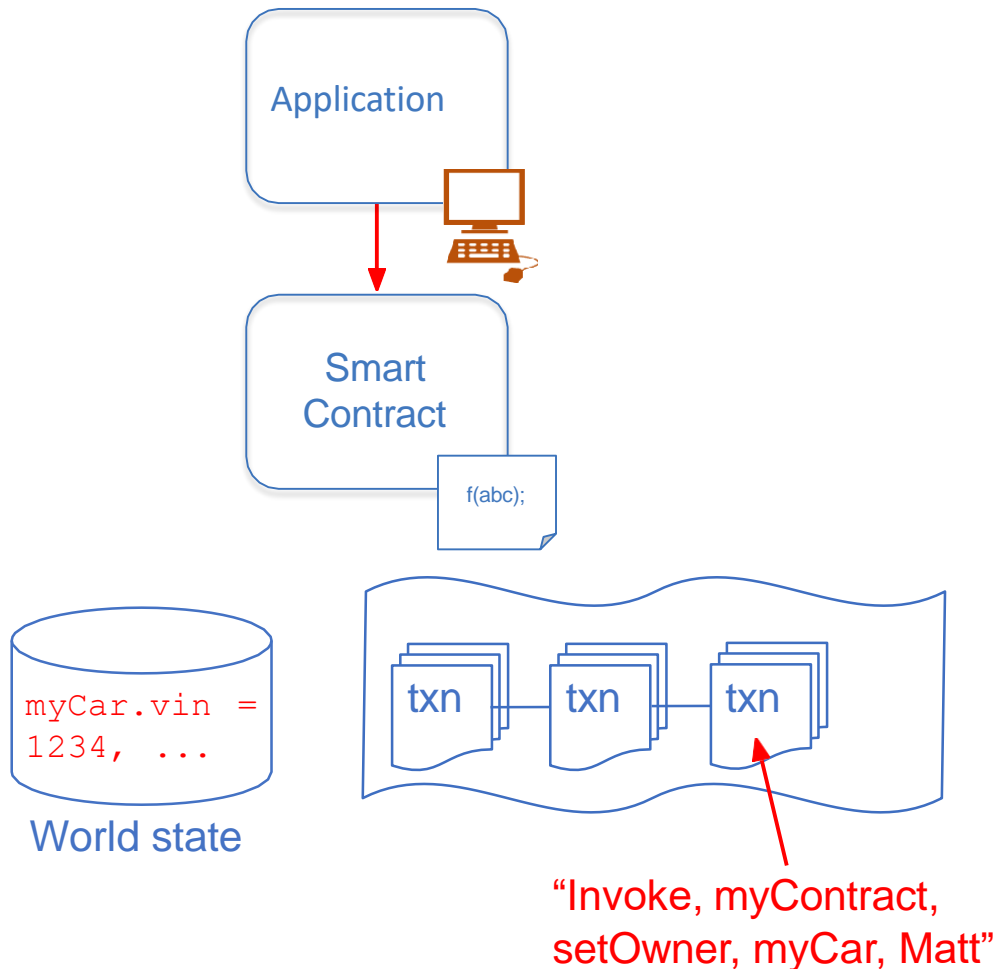


Integrating with existing systems – possibilities



Working with the ledger:

Example of a change of ownership transaction



Transaction input - sent from application

```
invoke(myContract, setOwner,  
       myCar, Matt)  
...
```

Smart contract implementation

```
setOwner(Car, newOwner) {  
    set Car.owner = newOwner  
}
```

World state: new contents

```
myCar.vin = 1234  
myCar.owner = Matt  
myCar.make = Audi  
...
```

Outline

- Permissionless and Permissioned Blockchain
- Introduction to Hyperledger
- Introduction to Hyperledger Fabric
- **Hyperledger Fabric Key Components**
 - Membership
 - Ledger
 - Chaincode
 - Privacy
 - Peers
 - Consensus



Membership

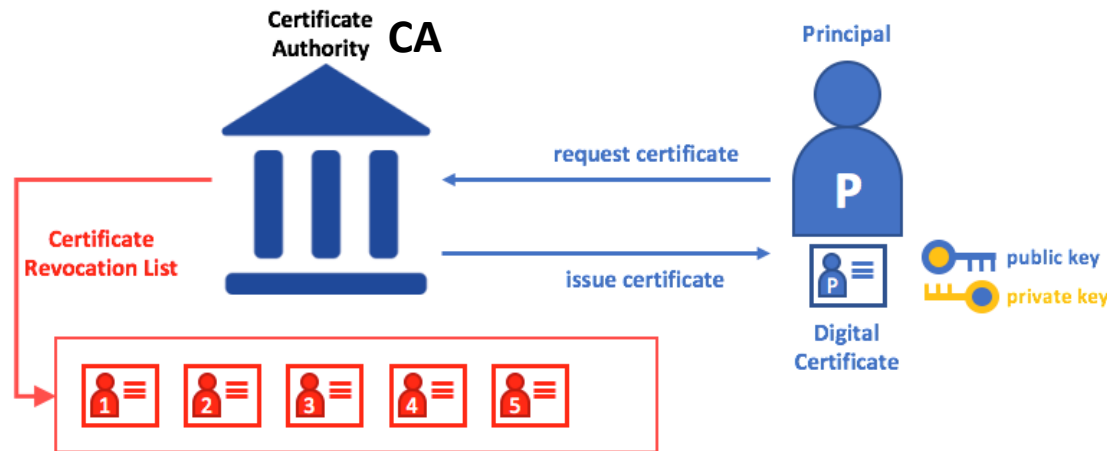
- Most cryptocurrencies use **permissionless** blockchains where **anyone** can join and have **full rights** to use it
 - e.g., anyone can buy BTC or ETH
- On the other hand, business blockchains tend to be permissioned
 - A person needs to meet **certain requirements** to perform **certain actions** on the blockchain
 - Some permissioned blockchains **restrict access** to **pre-verified users** who have already proven who they are
 - Others allow anyone to join, but only let **trusted identities verify transactions** on the blockchain
 - Members of a Hyperledger Fabric network enroll through a trusted **Membership Service Provider (MSP)**

Membership

- Hyperledger Fabric network has different actors
 - e.g. peers, endorser, orderers, anchors, client applications, and administrators
- Each actor has a **digital identity**
 - Identity determines the exact permission over resources and access to information
- A digital identity can have some additional attributes to determine permissions. **Principal** defines a union of an identity and the associated attributes.
 - **Principals** are like groupIDs, but more flexible
 - They can include a wide range of properties of an actor's identity, such as the actor's organization, organizational unit and role
- The permissioned notion can address scenarios where privacy and confidentiality are critical concerns

Membership

- Hyperledger Fabric provides a built-in CA component to create CAs in blockchain network

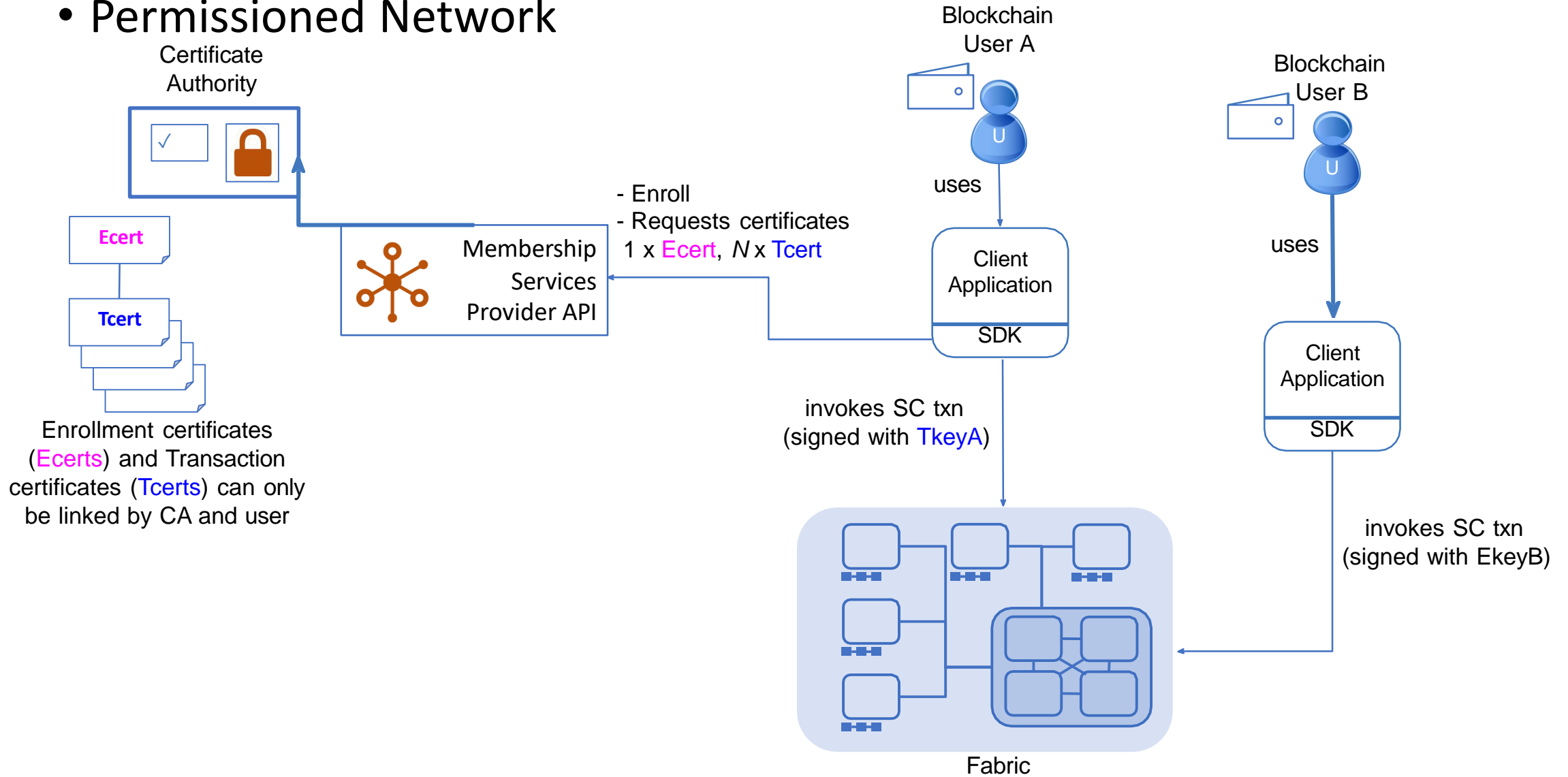


- For an identity to be **verifiable**, it must come from a **trusted authority**
- The Membership Service Provider (MSP) uses a traditional **Public Key Infrastructure (PKI)** to issue certificates

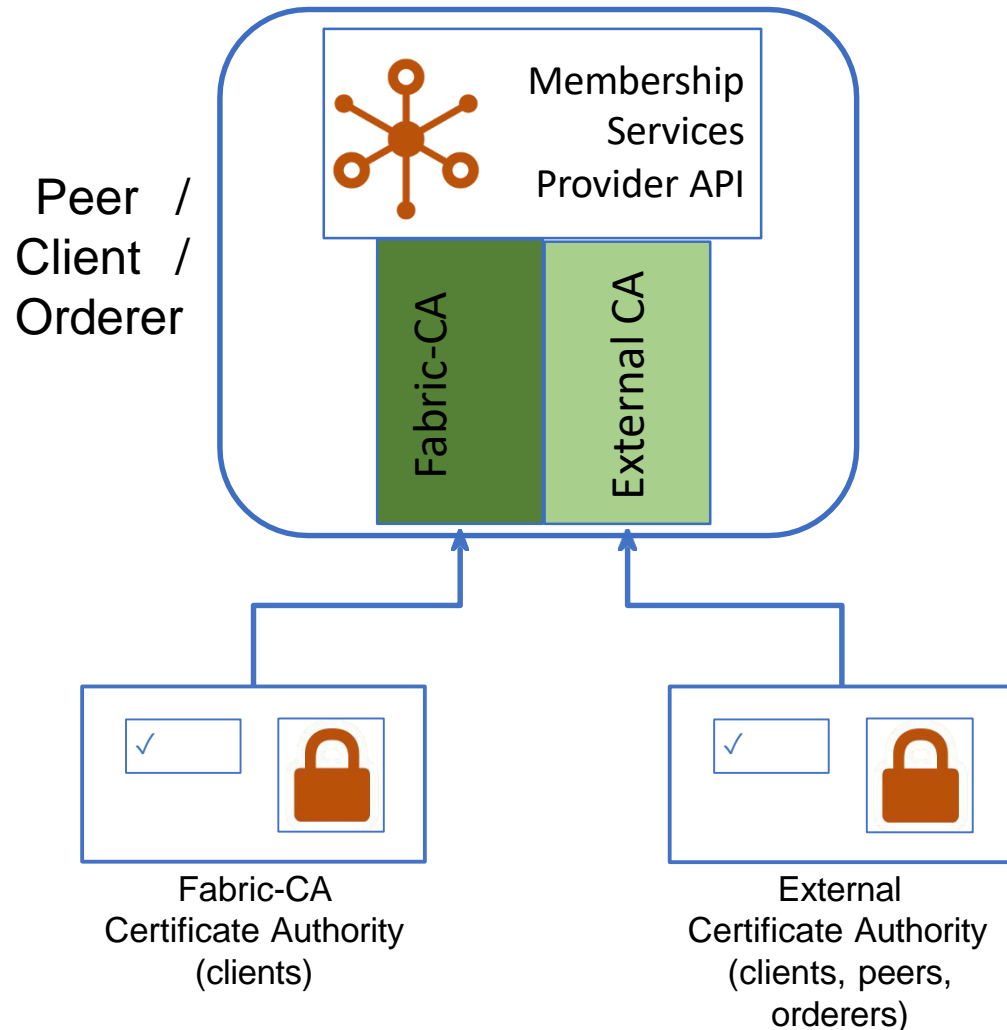
- A PKI is comprised of **Certificate Authorities** who issue digital certificates to actors
- Actors use them to **authenticate** themselves in messages
- A CA's **Certificate Revocation List (CRL)** records certificates that are no longer valid

Membership

- **Permissioned Network**

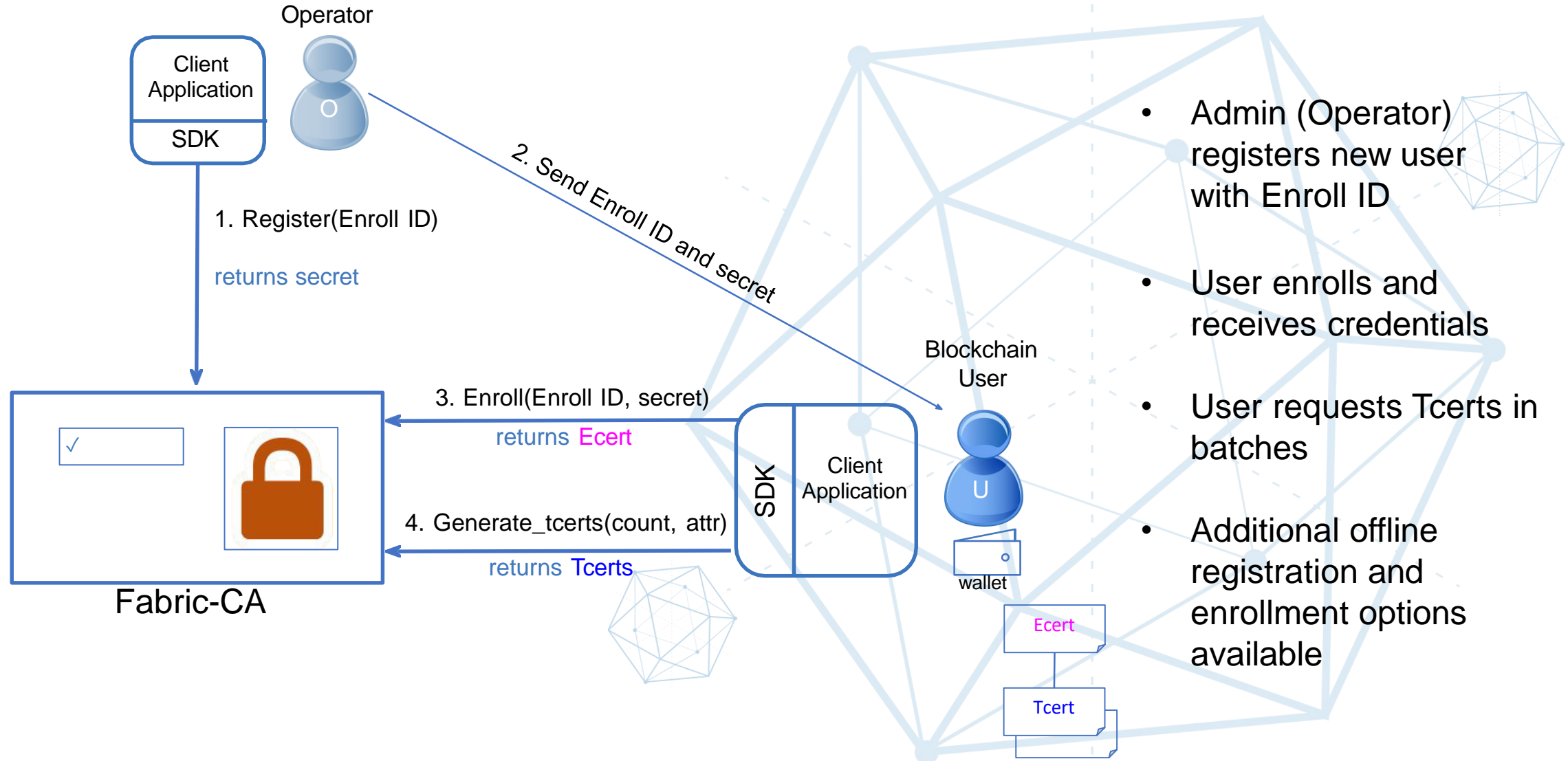


Membership Services Provider API



- Pluggable interface supporting a range of credential architectures
- Governs identity for Peers, Users, and Orderers.
- Provides:
 - Concrete identity format
 - User credential validation
 - User credential revocation
 - Signature generation and verification
 - (Optional) credential issuance

Fabric-CA: New User Registration and Enrollment



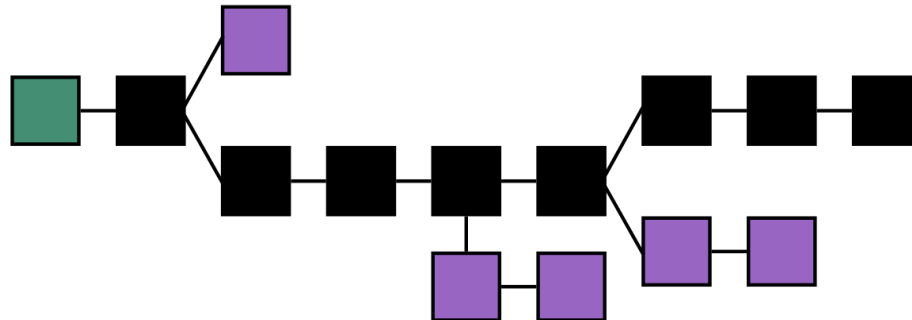
Outline

- Permissionless and Permissioned Blockchain
- Introduction to Hyperledger
- Introduction to Hyperledger Fabric
- **Hyperledger Fabric Key Components**
 - Membership
 - **Ledger**
 - Chaincode
 - Privacy
 - Peers
 - Consensus



Ledger

- A Fabric ledger stores important information about business objects
 - The **current value of the attributes** of the objects
 - e.g., account balance, logistic status, material price
 - The **history of transactions** that resulted in the current values
- While the **current state** of a business object may change, the history of state is **immutable**
 - History can be added to, but cannot be retrospectively changed
- Therefore , a blockchain can be viewed as an **immutable** history of facts about business objects

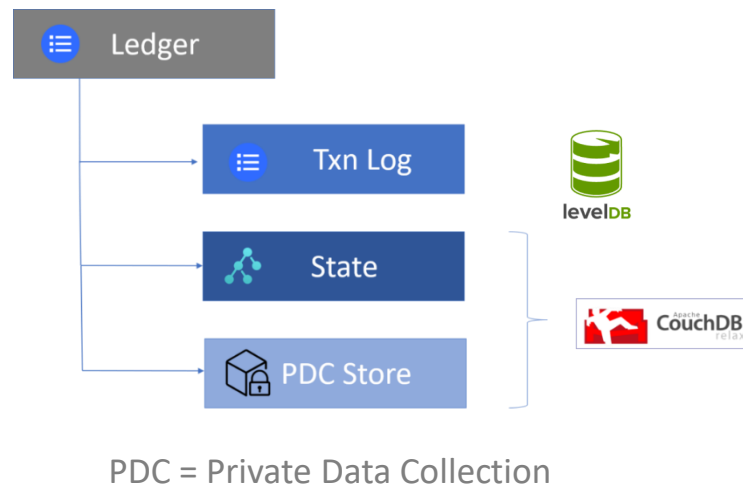


Ledger

- The ledger is a combination of the **world state** database and the **transaction log** history
 - Each participant has a copy of the ledger to every Hyperledger Fabric network they belong to
- The **world state** describes the state of the ledger at a given point in time
 - It is the database of the ledger
- The **transaction log** records **all transactions**, which have resulted in the current value of the world state
 - It is the update history for the world state

Ledger

- A Hyperledger Fabric ledger is a blockchain storing the immutable, sequenced transactions and the current state



- The ledger has a **replaceable data store for the world state**
- By default, the world state is a **LevelDB** database containing **key-value pairs**
 - **Production** system normally uses **CouchDB**
- The transaction log records the history values of the ledger database

LevelDB



-
- The LevelDB database (at Google by Sanjay Ghemawat and Jeff Dean) is a fast database that allows you to store key-value pairs.
 - LevelDB key-value pairs are stored in arbitrary byte arrays.
 - LevelDB provides support for only Key, Key range, and composite key queries, i.e., Put(key, value), GET(key, value), and Delete(key).
 - LevelDB is a NoSQL database
 - There is no support for SQL queries, indexes, or relational data models.

CouchDB CouchDB

- The CouchDB database is an open-source, document-oriented database implemented in Erlang that collects and stores data in JSON format.
- Hyperledger Fabric users can replace their default LevelDB world state database with a CouchDB database
- CouchDB is a NoSQL database that allows for rich queries of the stored JSON content. The default query language in a CouchDB database is JavaScript
- CouchDB allows for JSON queries and indices, which makes it easier to fulfill your Hyperledger Fabric network auditing and reporting requirements than in LevelDB.

Outline

- Permissionless and Permissioned Blockchain
- Introduction to Hyperledger
- Introduction to Hyperledger Fabric
- **Hyperledger Fabric Key Components**
 - Membership
 - Ledger
 - **Chaincode**
 - Privacy
 - Peers
 - Consensus



Chaincode

- Chaincode defines assets and business logic in the form of transaction instructions for modifying the assets
- A ledger records current and historical state of business objects, while a **smart contract** defines the **executable logic** that generates new states to be added to the ledger
- A **smart contract** is packaged into a **chaincode** which will be deployed to a blockchain network
- Chaincodes execute against the current state database and are initiated through a **transaction proposal** and result in a set of key-value writes that can be submitted to the network and applied to the ledger on all peers

Chaincode

- Assets enable the **exchange** of almost anything with monetary value over the network
 - **Tangible**: real estate and currency
 - **Intangible**: access right, intellectual property
- Fabric provides the ability to define assets using chaincode and transaction instructions for modifying the assets
- Assets are represented as a collection of **key-value** pairs, with state changes recorded as transactions on a ledger
- Chaincode can be implemented using several **programming languages**
 - Golang, Node.js, and Java are supported

Outline

- Permissionless and Permissioned Blockchain
- Introduction to Hyperledger
- Introduction to Hyperledger Fabric
- **Hyperledger Fabric Key Components**
 - Membership
 - Ledger
 - Chaincode
 - **Privacy**
 - Peers
 - Consensus



Privacy

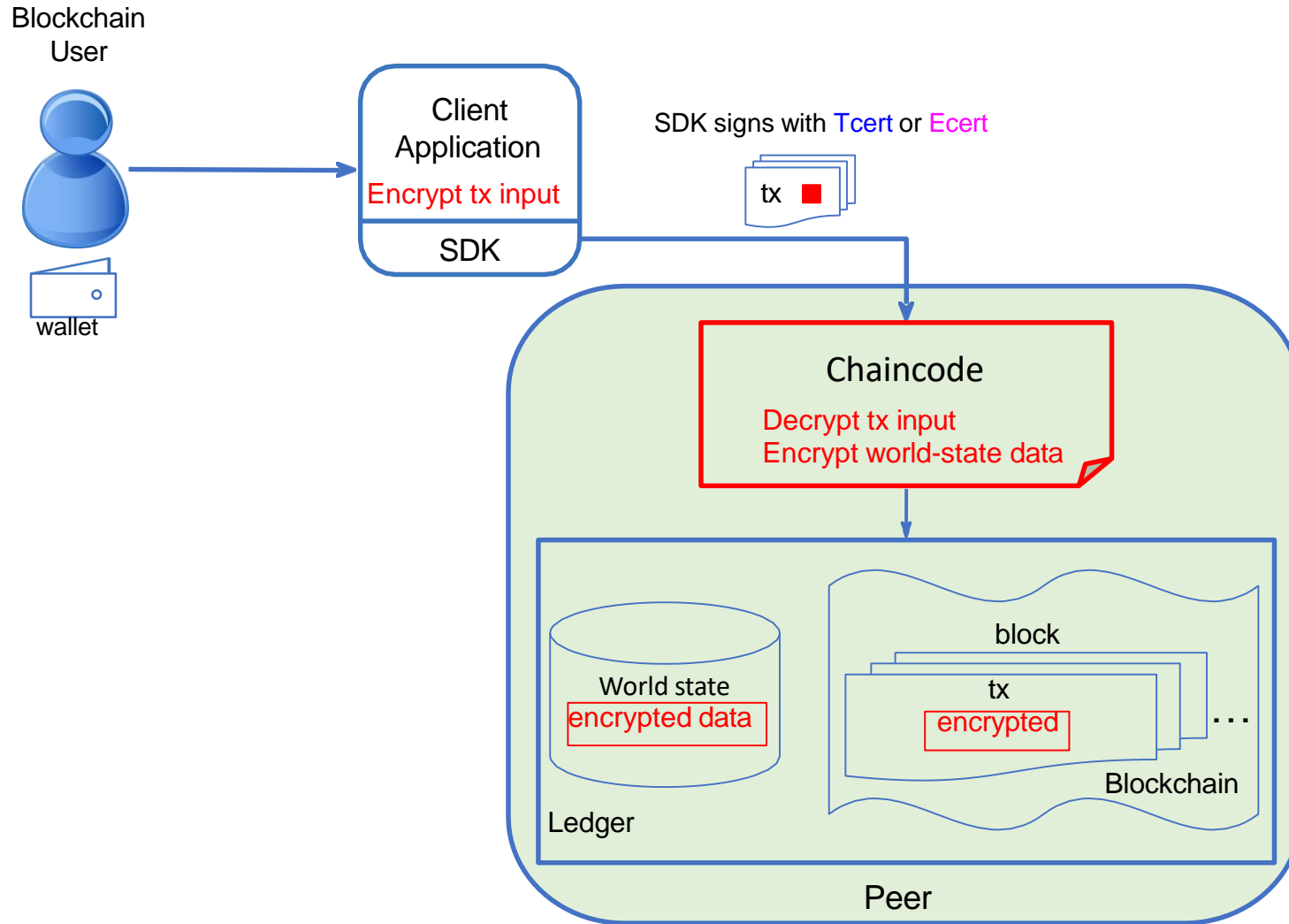
- Fabric offers the ability to create channels, which are used to enforce privacy and control information sharing
- A **channel** allows a group of participants to create a **separate blockchain network**
 - If three participants form a channel, then those participants - and no others - have copies of the ledger for that channel
- This is an important option for networks where **some participants** might be **competitors** and do not want every transaction they make to be shared
 - A special price they are offering to some participants and not others

Privacy

- Multiple channels can be created in the network and each channel has an isolated ledger
- A **ledger** exists in the scope of a **channel**
 - It can be shared across the entire network (if all participants are operating on one common channel)
 - It can be private by including only a specific set of participants
- In the latter scenario, these **participants** can create a separate channel and **isolate their ledger**

Privacy

- Contract Confidentiality through encryption



Handled in the application domain.

Multiple options for encrypting:

- Transaction Data
- Chaincode*
- World-State data

Chaincode optionally deployed with cryptographic material, or receive it in the transaction from the client application using the *transient* data field (not stored on the ledger).

*Encryption of application chaincode requires additional development of system chaincode.

Privacy

- Multiple subsets of an organization can be created in the network and each subset can have its own data collection
- When a subset of organizations on that channel need to keep their Tx data confidential, a private data collection (collection) is used
- A collection is logically separate from the channel ledger, accessible only to the authorized subset of organizations
- Thus, channels keep transactions private from the broader network, while collections keep data private between subsets of organizations on the channel
- To further protect data, values within chaincode can be encrypted using cryptographic algorithms (e.g., AES) before sending transactions to the ordering peers
 - Once encrypted data has been written to the ledger, it can be decrypted only by a user holding the corresponding decryption key

Outline

- Permissionless and Permissioned Blockchain
- Introduction to Hyperledger
- Introduction to Hyperledger Fabric
- **Hyperledger Fabric Key Components**
 - Membership
 - Ledger
 - Chaincode
 - Privacy
 - **Peers**
 - Consensus



Peers and roles



Committing Peer: Maintains ledger and state. Commits transactions. May hold smart contract (chaincode).



Endorsing Peer: Specialized committing peer that receives a transaction proposal for endorsement, responds granting or denying endorsement. Must hold smart contract

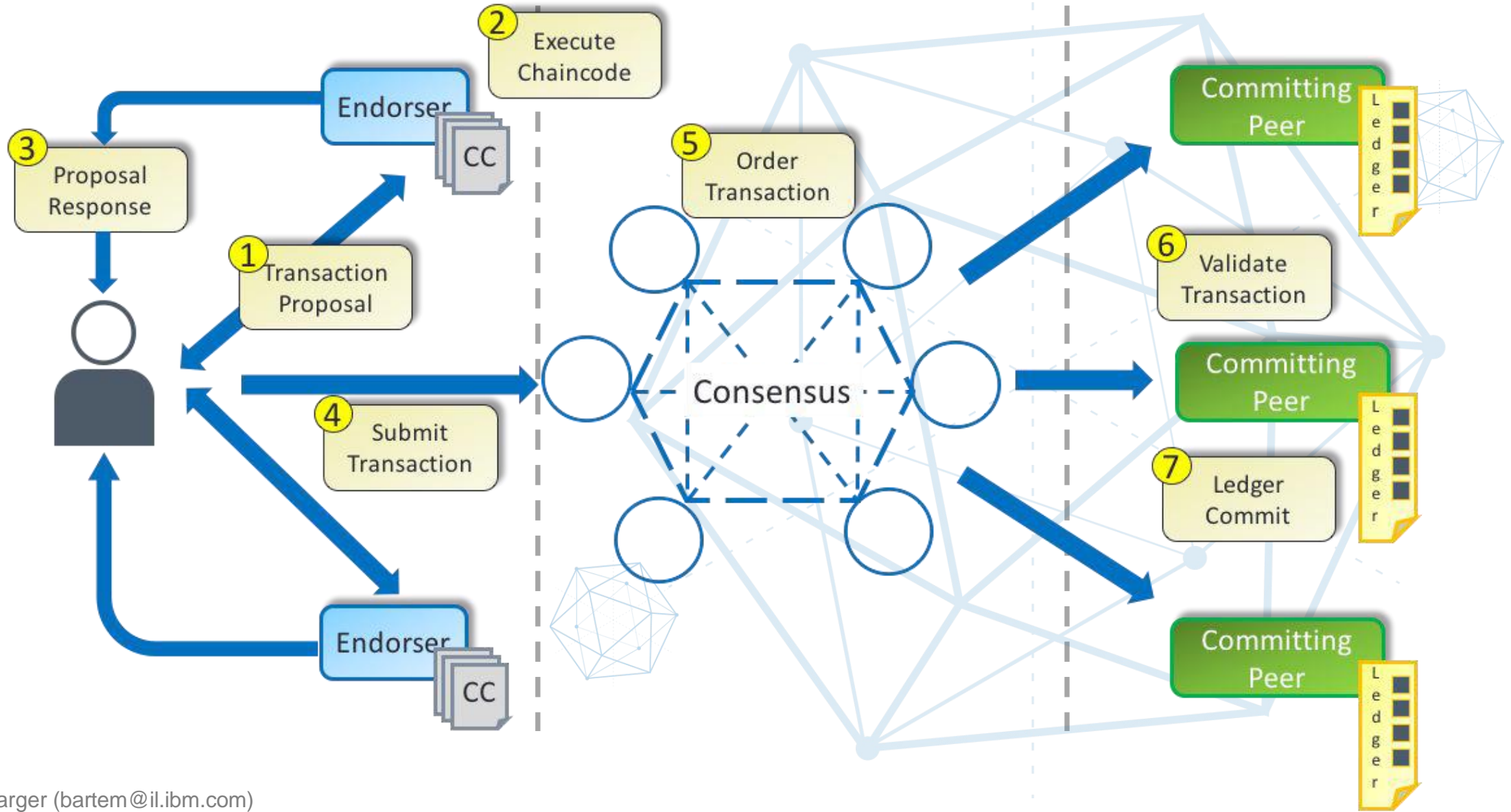


Ordering Nodes (service): Approves the inclusion of transaction blocks into the ledger and communicates with committing and endorsing peer nodes. Does not hold smart contract. Does not hold ledger.

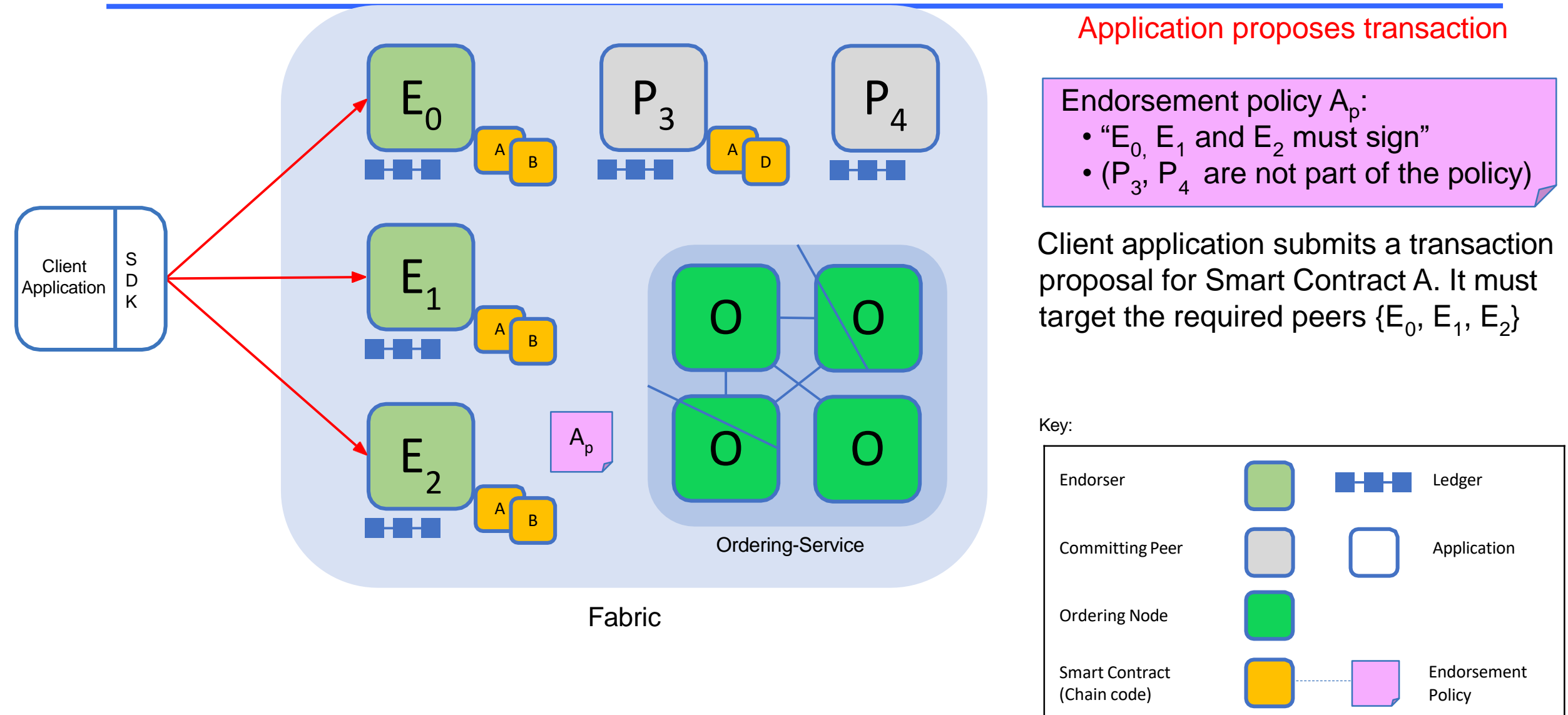
What is ordering?

- Permissionless blockchains, such as Ethereum and Bitcoin allow transactions to be ordered and bundled into blocks
- These systems rely on *probabilistic* consensus algorithms, which eventually guarantee the consistency to a high degree of probability while they may still suffer from divergent chains (i.e., forks)
- Hyperledger Fabric works differently
 - Relying on a node named an orderer to order transactions
 - It is deterministic consensus (no forks)
 - separating the endorsement of chaincode execution (at peers) from ordering to improve the scalability

Hyperledger Fabric



Sample transaction: Step 1/7 – Transaction Proposal



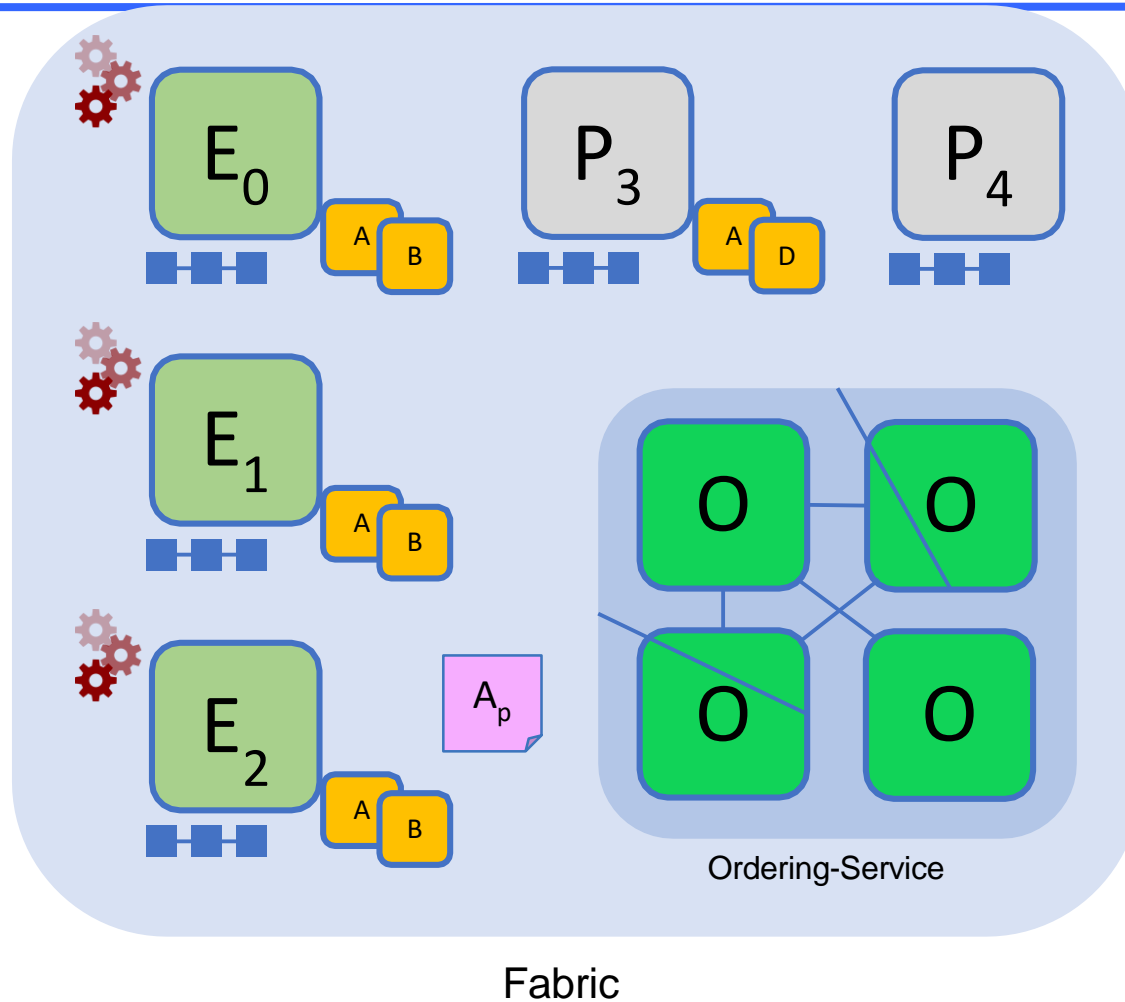
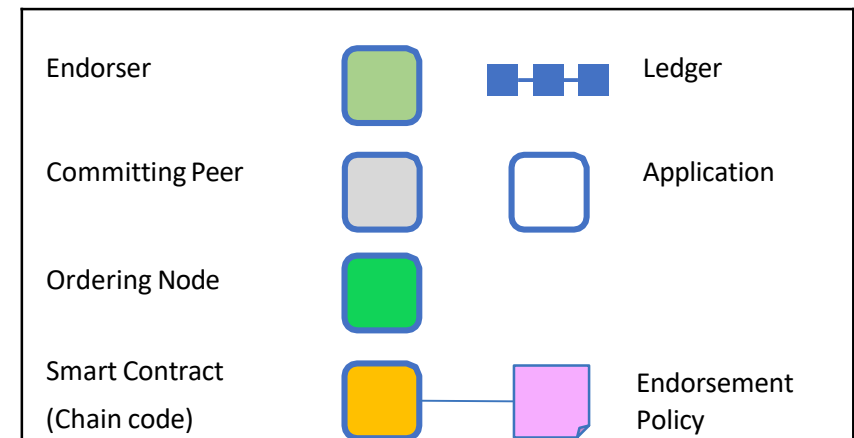
Sample transaction: Step 2/7 – Execute Chaincode

Endorsers Execute Proposals

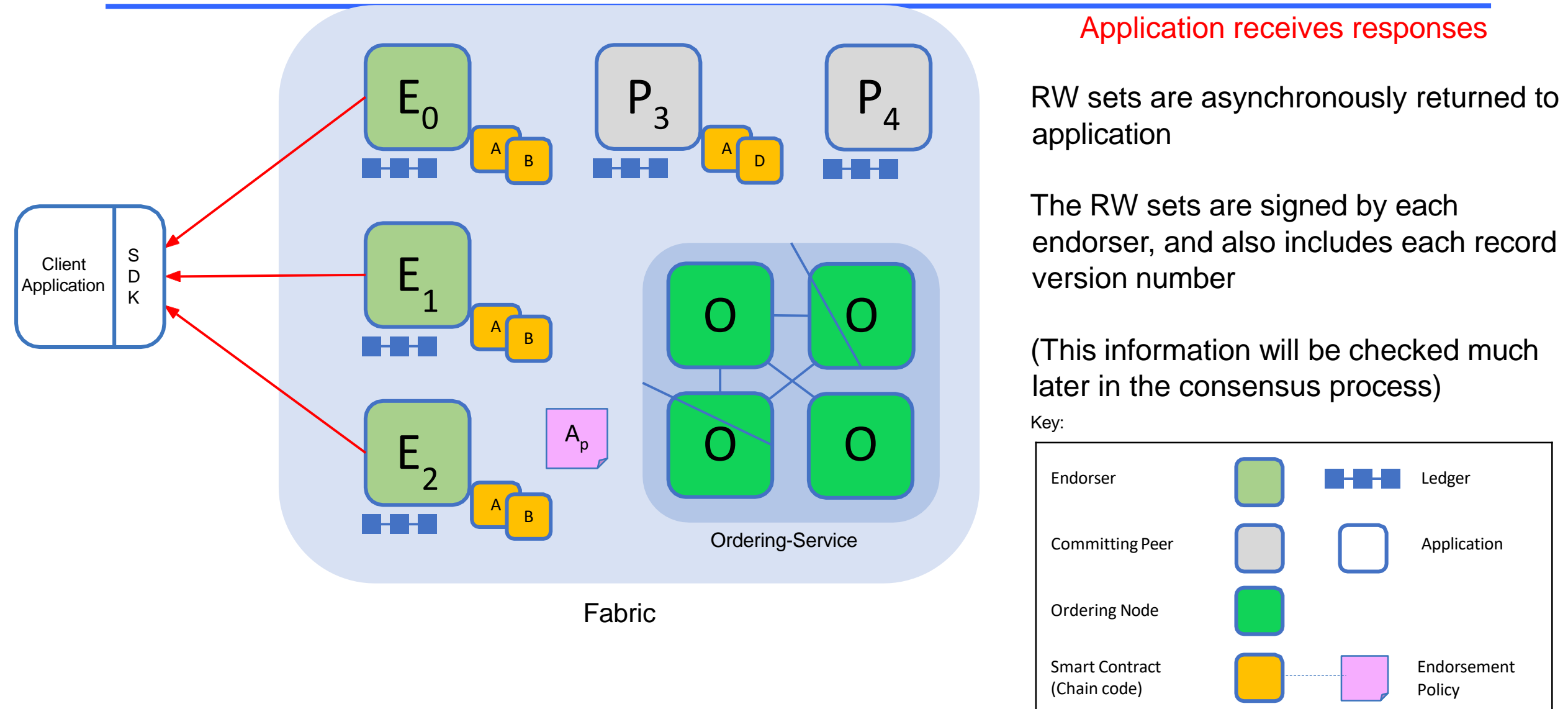
E_0 , E_1 & E_2 will each execute the *proposed* transaction (chaincode). None of these executions will update the ledger

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

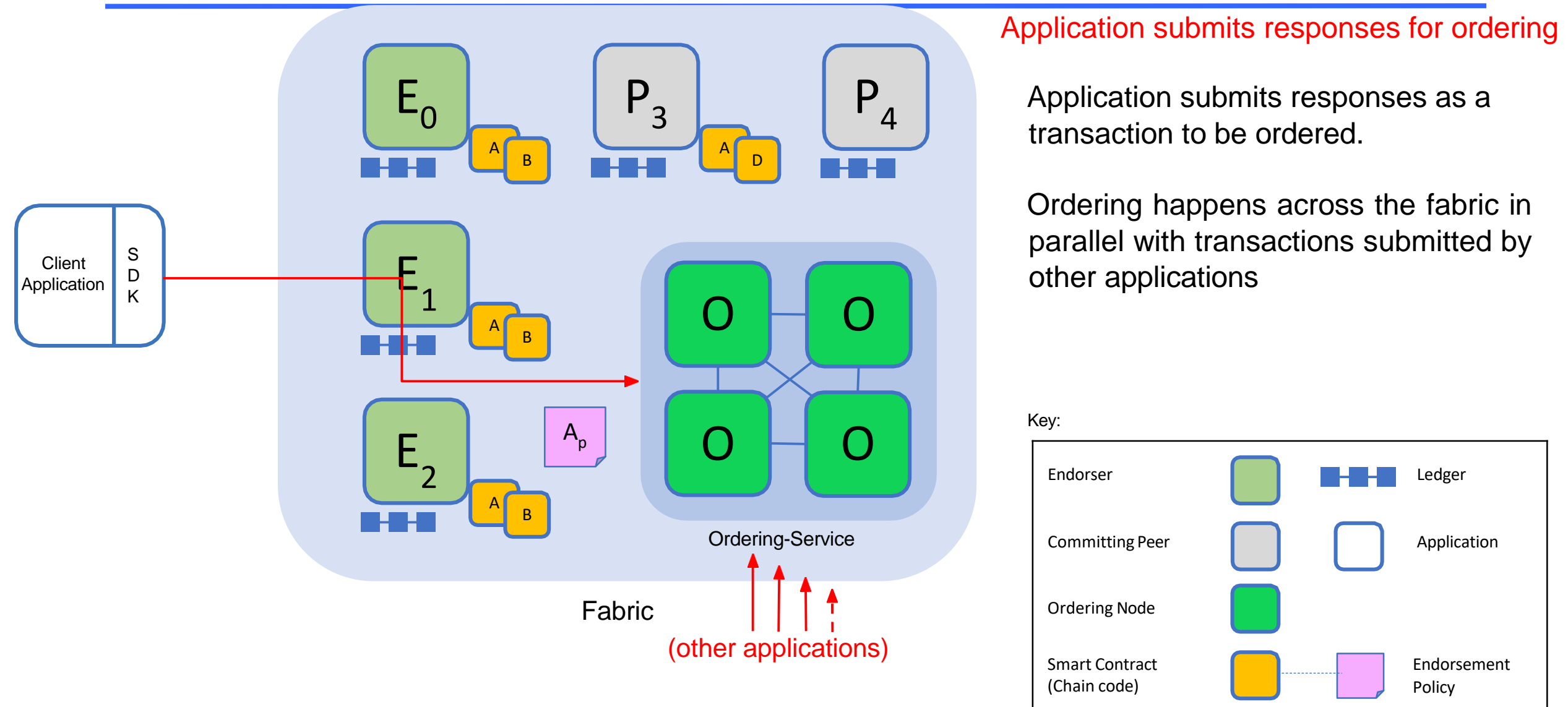
Transactions can be signed & encrypted
Key:



Sample transaction: Step 3/7 – Proposal Response



Sample transaction: Step 4/7 – Submit Transaction



Sample transaction: Step 5/7 – Order Transaction

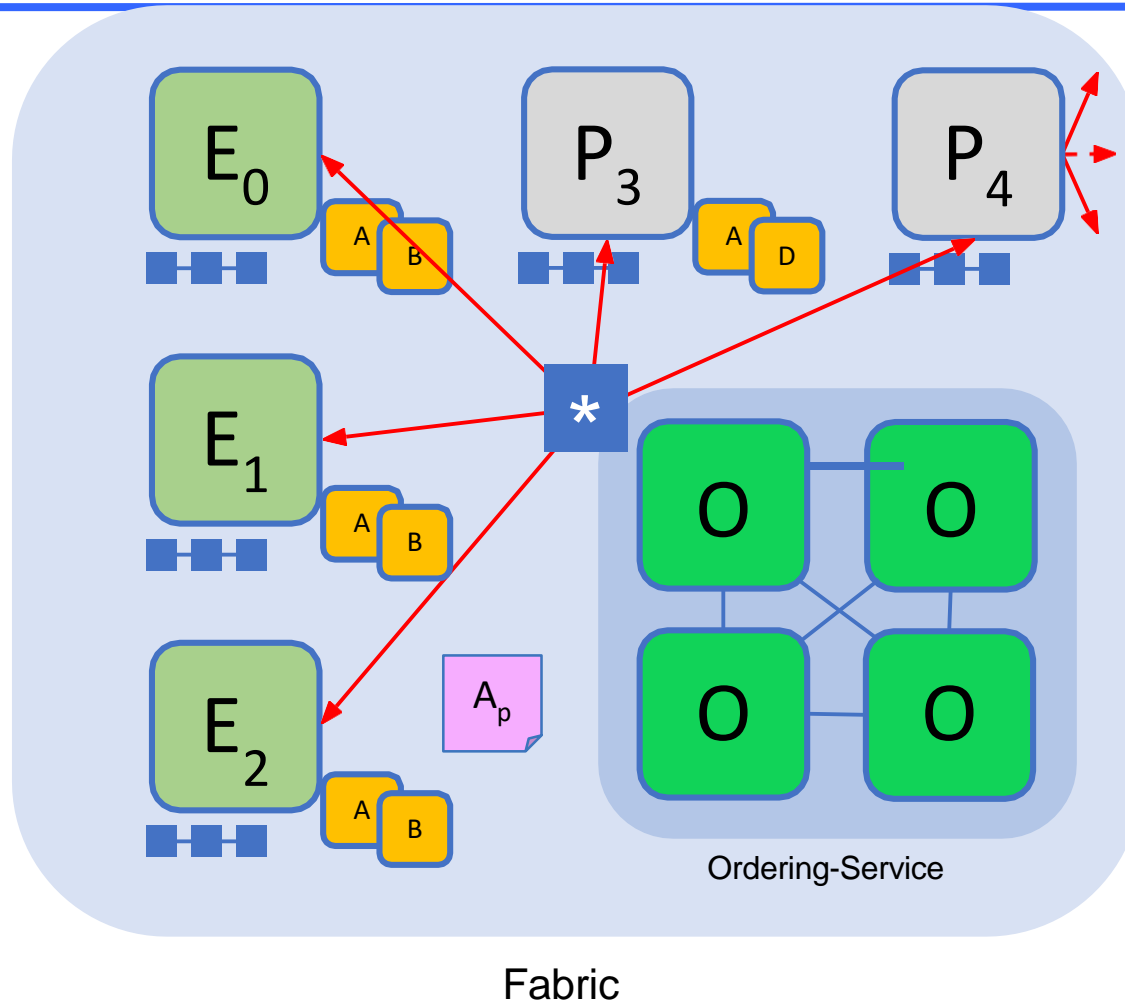
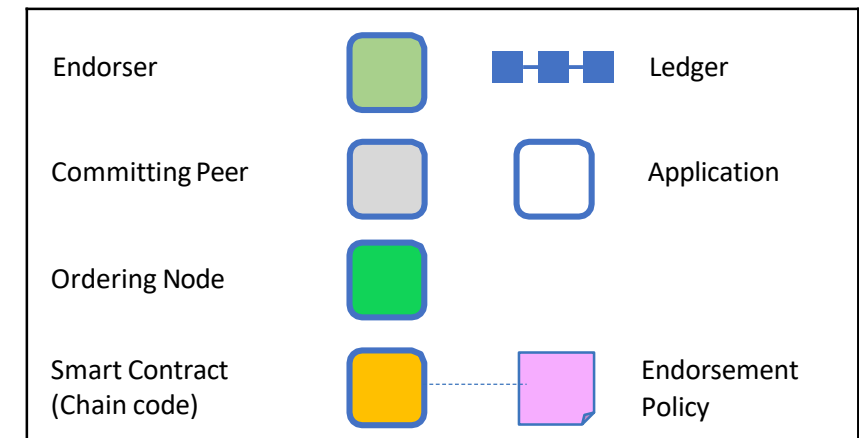
Orderer delivers to all committing peers

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)
- SBFT (Simplified Byzantine fault tolerance)

Key:



Sample transaction: Step 6/7 – Validate Transaction

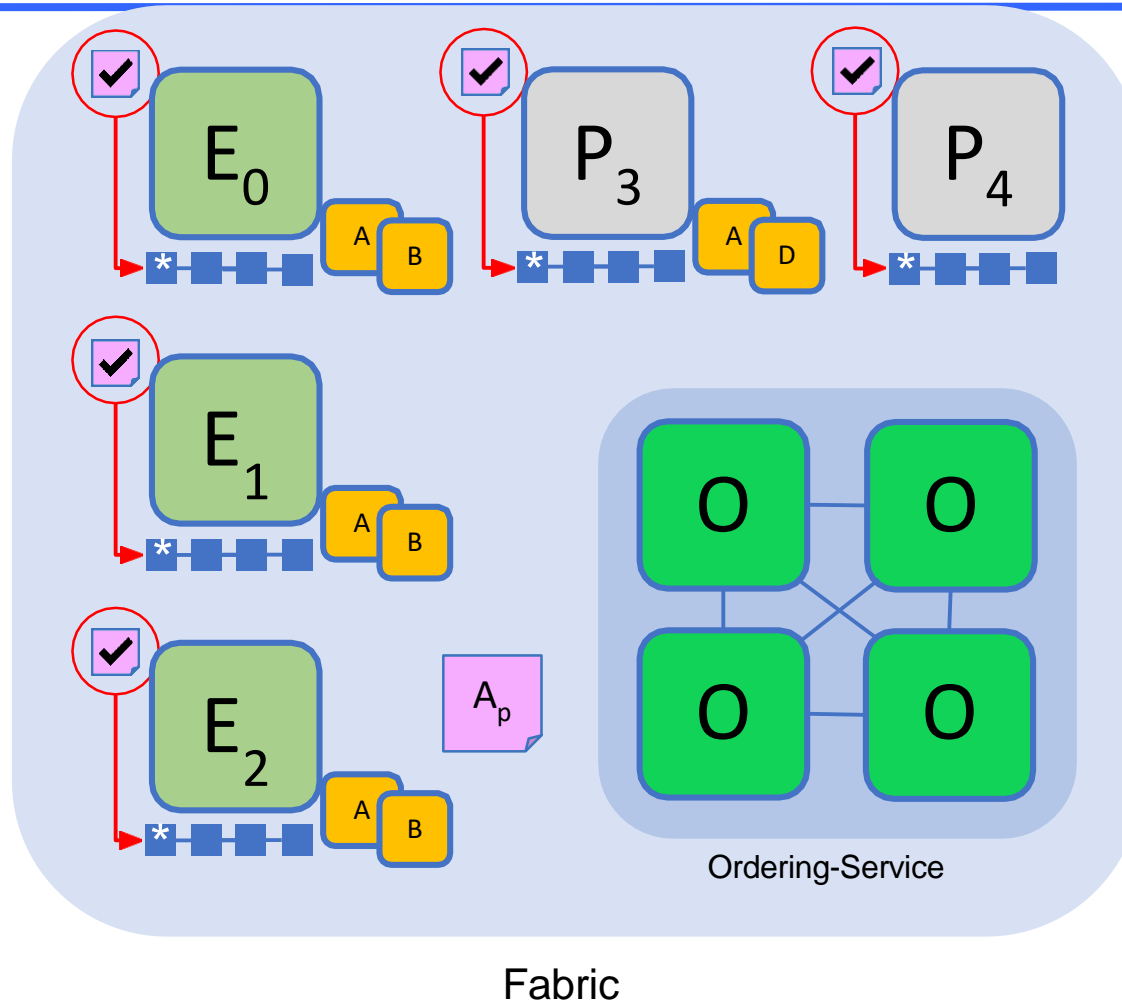
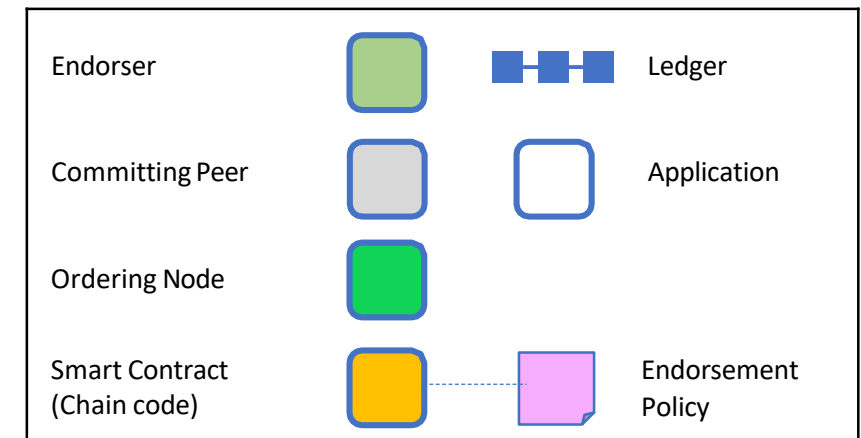
Committing peers validate transactions

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:

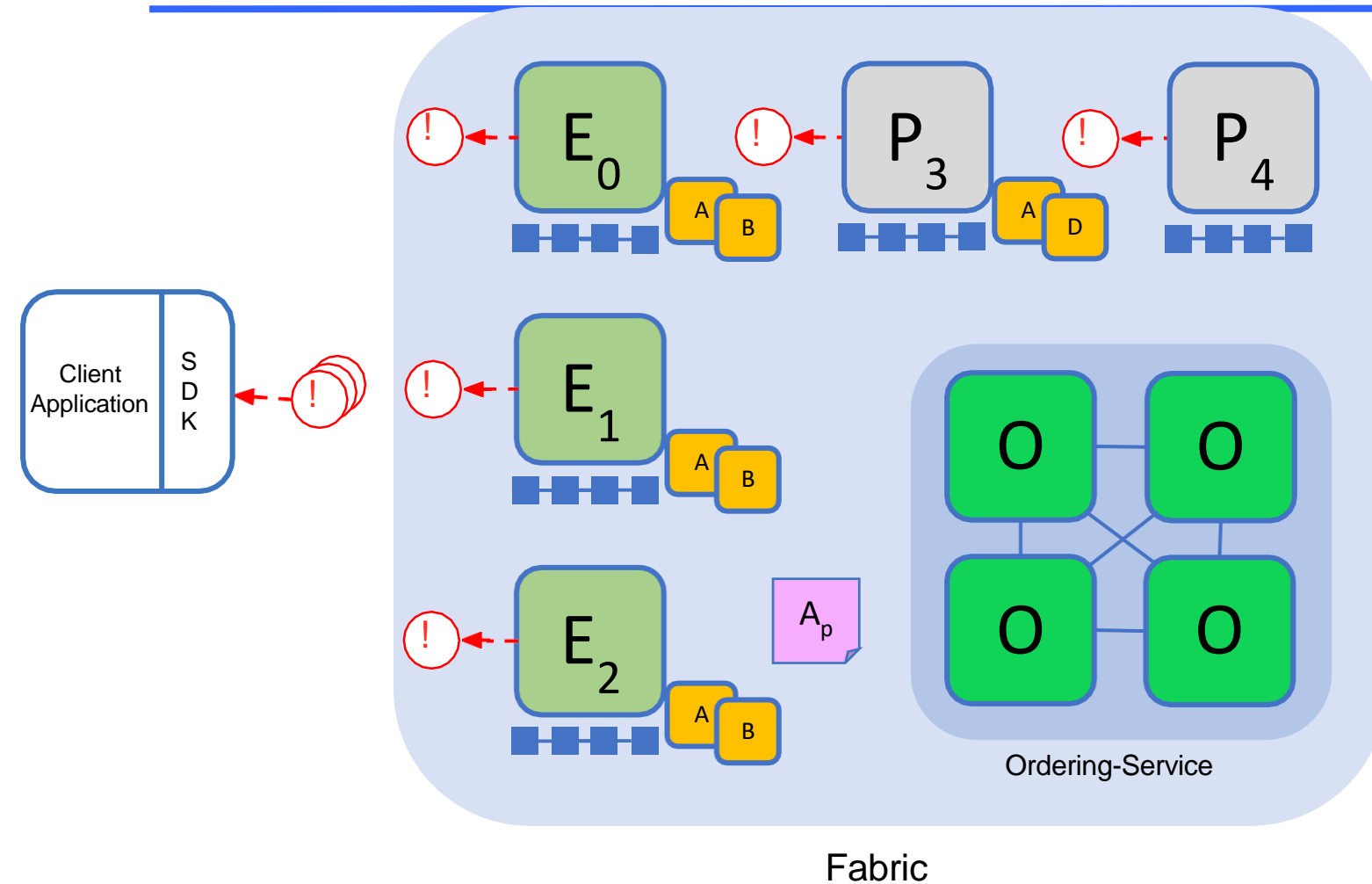


Sample transaction: Step 7/7 – Ledger Commit








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 (Chain code)			Endorsement Policy

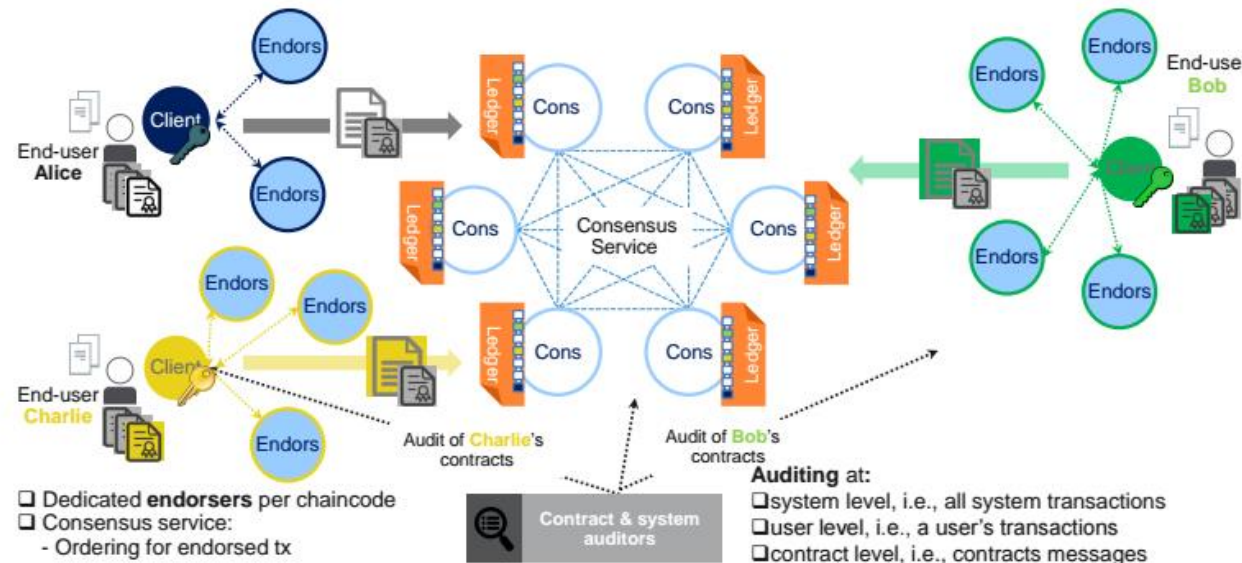
Outline

- Permissionless and Permissioned Blockchain
- Introduction to Hyperledger
- Introduction to Hyperledger Fabric
- **Hyperledger Fabric Key Components**
 - Membership
 - Ledger
 - Chaincode
 - Privacy
 - Peers
 - **Consensus**



Consensus

- Consensus is defined as the **full verification cycle** of Tx's and BLKs
 - Proposal, Endorsement, Ordering, Validation, Commitment
- Consensus is achieved when the order and results of a block's transactions have met the **policy criteria checks**
 - e.g., which members endorse a certain Tx class, how many endorsements needed, system version, identity verification, ...
 - Prevent threats (e.g., double spending) and protect data integrity



Summary

- Permissioned blockchains provide an additional level of security over typical blockchain systems like Bitcoin, as they require an **access control** layer.
- These blockchains are favored by individuals who require **security**, **identity**, and **role definition** within the blockchain.
- Hyperledger is an open source community focused on developing a suite of stable frameworks, tools and libraries for enterprise-grade blockchain deployments.

Summary

- **Fabric** (a project in Hyperledger) is a platform for developing distributed applications with a modular & secure architecture.
- The **modular design** satisfies a broad range of industry use cases and enables performance at scale while preserving privacy.
- In Hyperledger Fabric workflow, Endorser peers validate Txns, Orderer peers order Txns and generate blocks, Anchor peers broadcast blocks to general peers.
- Hyperledger Fabric includes many **key components**, such as membership, ledger, chaincode, privacy, peers, consensus.

References

- Glossaries
 - <https://hyperledger-fabric.readthedocs.io/en/release-1.4/glossary.html>
 - <https://openblockchain.readthedocs.io/en/latest/glossary/>
 - <https://fabrictestdocs.readthedocs.io/en/latest/glossary.html>
 - <https://hyperledger.github.io/composer/v0.19/reference/glossary>
 - <https://hackernoon.com/hyperledger-fabric-the-20-most-important-terms-made-simple-2753f925db4>
- Demystifying Hyperledger Fabric (Part 1 of 3): Fabric Architecture
 - <https://www.serial-coder.com/post/demystifying-hyperledger-fabric-fabric-architecture/>
- Demystifying Hyperledger Fabric (Part 2 of 3): Private Data Collection
 - <https://www.serial-coder.com/post/demystifying-hyperledger-fabric-private-data-collection/>
- Demystifying Hyperledger Fabric (Part 3 of 3): Network Traffic Handling, Service Discovery, and Operations Service
 - <https://www.serial-coder.com/post/demystifying-hyperledger-fabric-network-traffic-handling-service-discovery-and-operations-service/>