

HyperLedger Fabric

Hyperledger fabric Overview

- Purpose: Create permissioned blockchain systems for specific use cases.
- Advantages: Privacy, restricted access, and relevant data sharing tailored to business needs.
- Examples: Business transactions within a specific domain (e.g., a school district or an industry consortium).

Key Features of Hyperledger Fabric:

- Developed under the Linux Foundation's Hyperledger Project (launched in 2015).
- Promotes cross-industry collaboration to advance blockchain adoption.
- Provides tools and frameworks to enable easy development and integration.
- Focus on open-source, safe, reliable, and efficient blockchain technology for enterprises.

Hyper-Ledger Fabric

- Fabric is the first blockchain system to support the execution of distributed applications written in standard programming languages.

Allows them to be executed consistently among many nodes, giving impression of execution on a single distributed blockchain computer

The architecture of fabric follows a novel execute order validate paradigm for distributed execution of untrusted code in a untrusted environment.

It separates the transaction flow into three steps, which may be run on different entities in the system

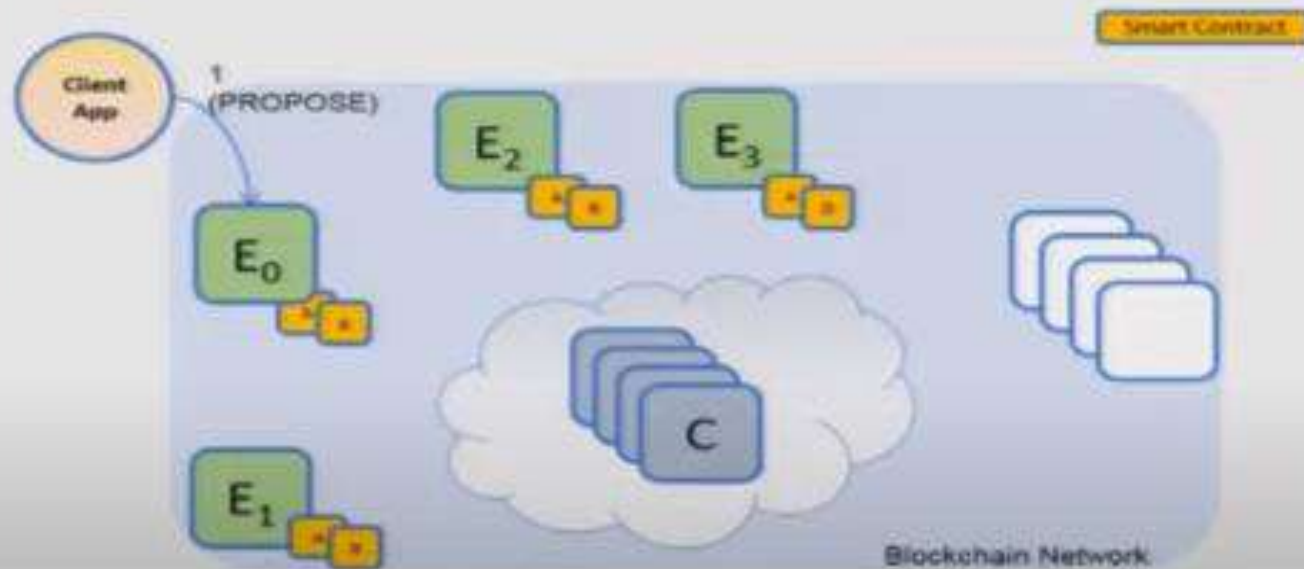
- Executing a transaction and checking its correctness, thereby endorsing it [corresponding to "transaction validation" in other blockchain.
- Ordering through a consensus protocol, irrespective of transaction semantics
- Transaction validation per application specific trust , which also prevents race condition due to concurrency

HyperLedger

- - A permissioned private blockchain.
 - smart contracts in case of Hyperledger is, is can be written in any language.
 - currently, most of the smart contracts for Hyperledger is written in Go language,
 - Hyperledger executes the smart contracts is inside a Docker and they do not use a special EVM for this like Ethereum uses EVM and therefore, it is for a specific language for which bytecode has to be generated.
 - In case of Hyperledger, any general purpose language can be used. But most common is the Go language
 - Hyperledger, the consensus is pluggable in the sense that you can decide what kind of consensus algorithm you need and accordingly you can plug that in into the Hyperledger framework.

Example-use Case

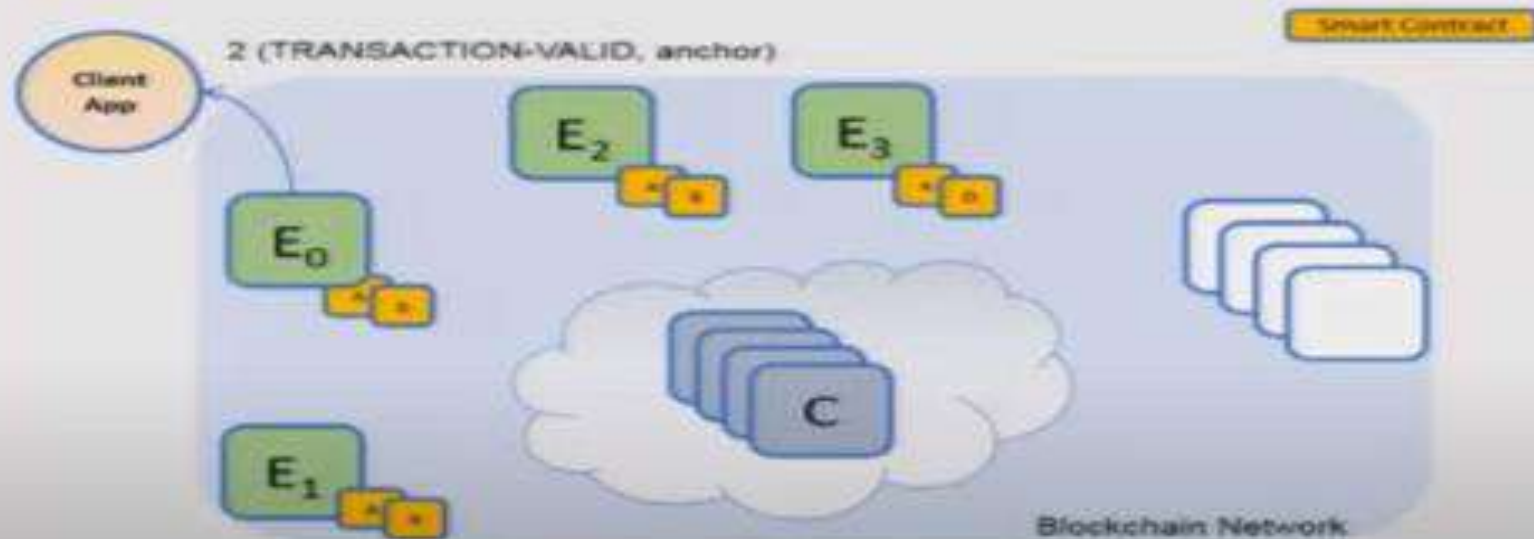
A sample transaction (1/6)



1. The Client App proposes a transaction for **Smart Contract A** to the Endorsing peer E_0 . Endorsement policy: " E_0 , E_1 and E_2 must sign". E_3 is not part of the policy

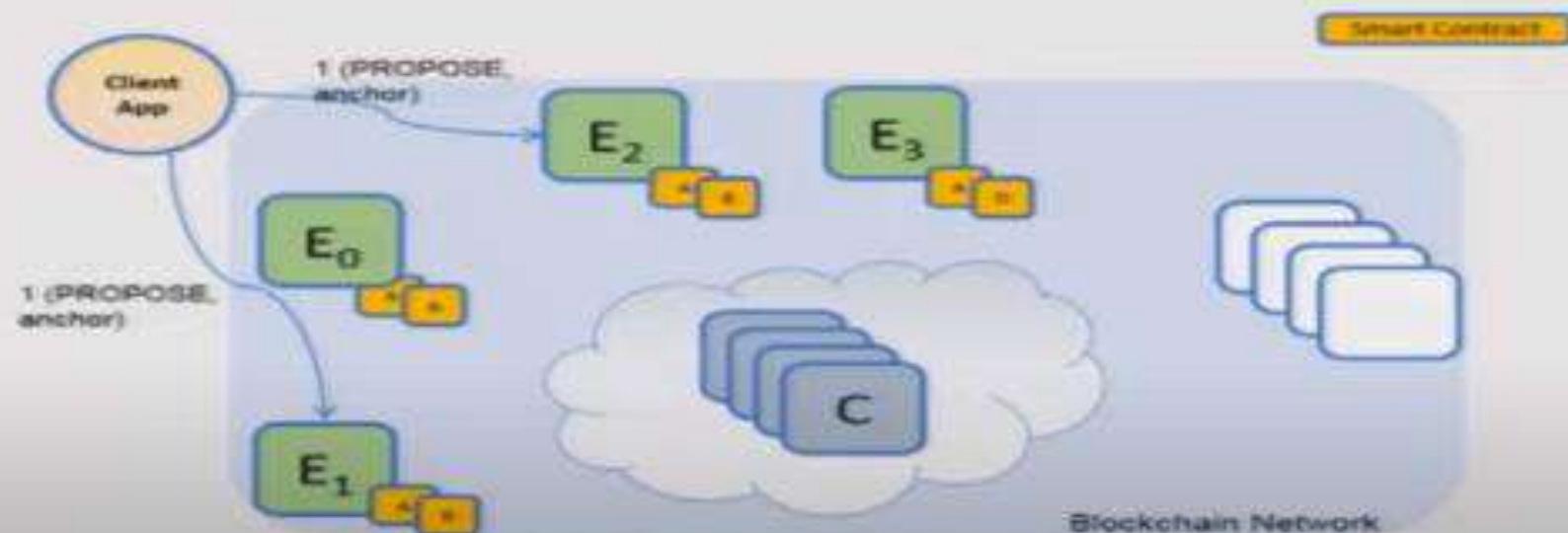
Use Case

A sample transaction (2/6)



2. Endorsing peer E_0 endorses a tx and (optionally) "anchors it" with respect to the ledger state version numbers. An "anchor" contains all data read and written by contract that are to be confirmed by other endorsers.

A sample transaction (3/6)



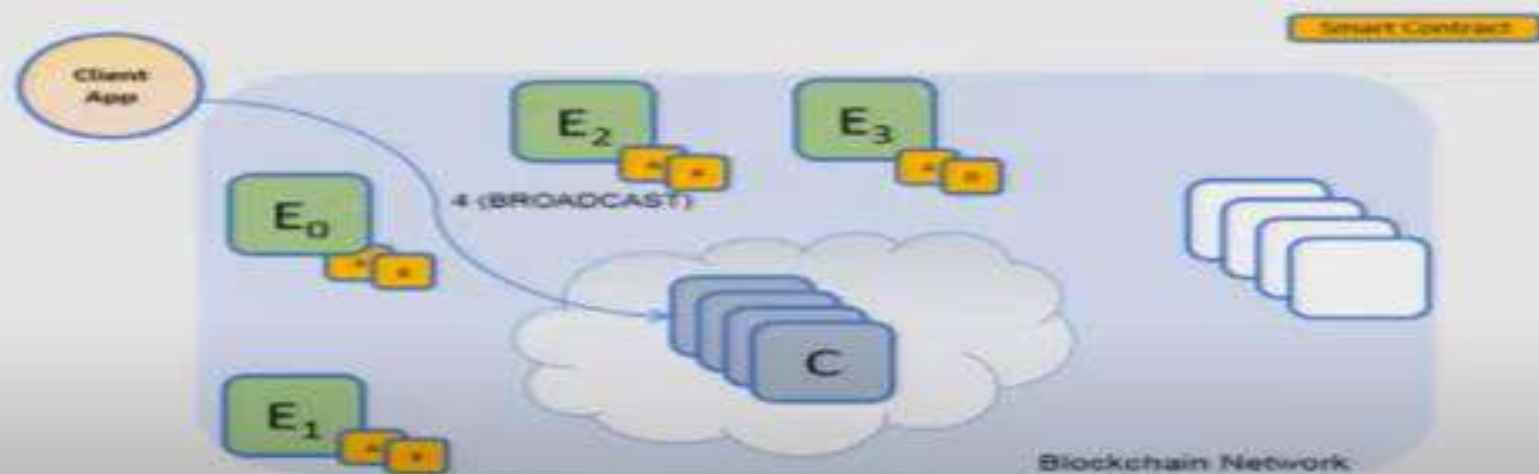
3. The client requests further endorsement from E₁ and E₂. The client may decide to suggest an anchor obtained from E₀ to E₁ and E₂.

A sample transaction (4/6)



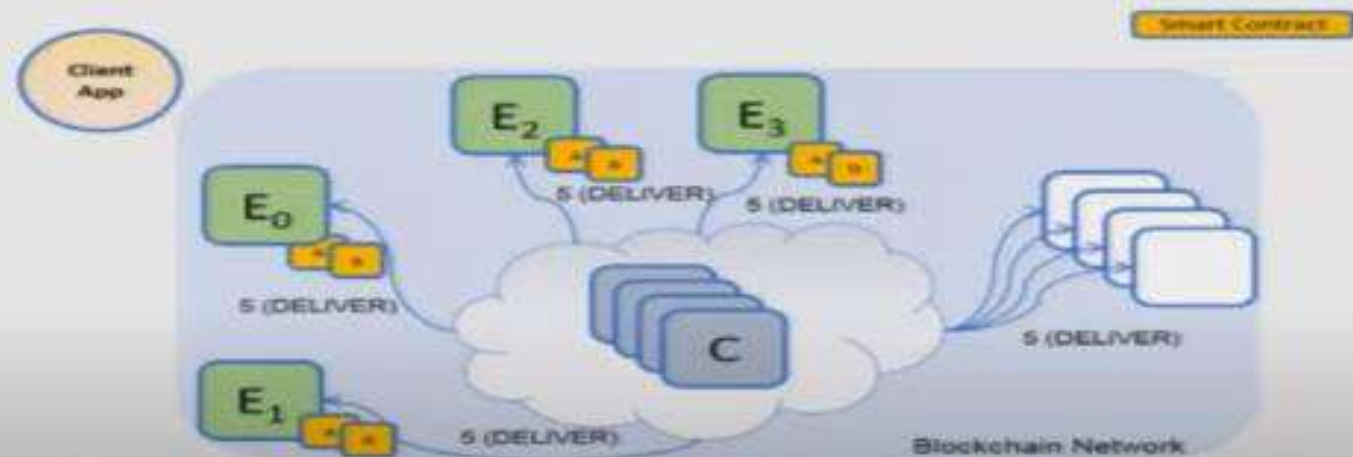
4. The Endorsing peers E₁ and E₂ send the endorsement to client.

A sample transaction (5/6)



5. Client formats the transaction and broadcasts it to the consenters for inclusion in the ledger

A sample transaction (6/6)



6. The consensus service delivers the next block in the ledger with the consented transaction.

Hyperledger Fabric

- The Linux Foundation founded Hyperledger in 2015
- Hyperledger Fabric is a platform for distributed ledger solutions in industrial level.
- A modular architecture - Delivers high degrees of confidentiality, resiliency, flexibility and scalability.
- It is designed to support pluggable implementations of different components, and accommodate the complexity and intricacies that exist across the economic ecosystem.
- Breaks from some other blockchain systems is that it is **private** and **permissioned**

Hyperledger Fabric - Cont.

- Like other blockchain technologies, it has a ledger, uses smart contracts, and is a system by which participants manage their transactions.
- Ledger data can be stored in multiple formats, consensus mechanisms can be switched in and out.
- Offers the ability to create **channels**, allowing a group of participants to create a separate ledger of transactions.
- Hyperledger is based on blockchain but its not a crypto currency.
- There is no mining, just order system do it.
- Operational power: 0.5 million operations per minute where as other blockchain does only 1000.

In Summary

- Hyperledger Fabric is enterprise grade distributed ledger based on blockchain technologies that uses smart contracts to enforce trust between parties.
- Hyperledger in general do not enforce any requirements about the hardware, network infrastructures, additional software around it, security models etc.
- No concept of computational power.

Advantages of Hyperledger Fabric

- `Permissioned membership`
- `Performance, scalability, and levels of trust`
- `Data on a need-to-know basis`
- `Rich queries over an immutable distributed ledger`
- `Modular architecture supporting plug-in components`
- `Protection of digital keys and sensitive data`

Hyperledger Components

- Fabric CA,
- Peer
- Ordering service
- Channel
- Chaincode

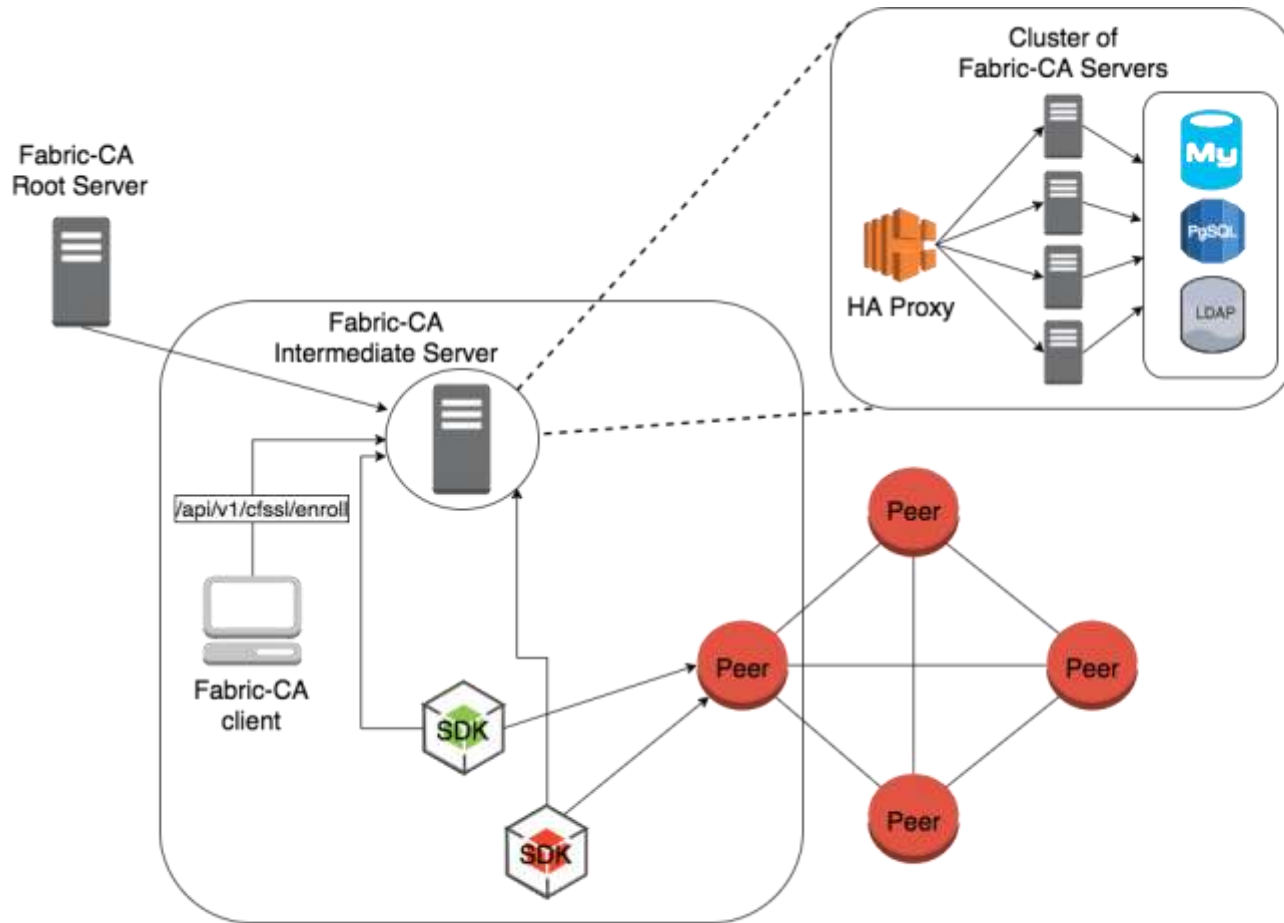
Fabric CA

The Hyperledger Fabric CA is a Certificate Authority (CA) for Hyperledger Fabric.

It provides features such as:

- registration of identities, or connects to LDAP as the user registry
- issuance of Enrollment Certificates (ECerts)
- certificate renewal and revocation
- consists of both a server and a client component.

CA - WorkFlow



Blockchain

CA cont.

- Every single operation that is executed inside hyperledger fabric must be cryptographically signed with this certificate.
- You can add attributes, roles
- Certificates are X.509 standards.
- You can remove the necessity of certificates if you don't need it.
- Chaincodes read this data and make business decisions.

Peer

- Peer is the place where the ledger and the blockchain data is stored.
- You must have more than one peer in production.
- One peer may be part of many channels.
- Every single channel is inside the peer.
- It endorse any update of the ledger.
- You can create backup of the ledger from the peer

Ordering Service

- Ordering service is actually the heart of consensus algorithm and the heart of hyper ledger fabric.
- Main role is to provide the order of operations.
- before committing anything to ledger it must pass through the ordering service.
- it is responsible for verification, security, policy verification etc.

Channel

- **Channel** is a private “subnet” of communication between two or more specific network members.
- A channel is defined by members (organizations), anchor peers per member, the shared ledger, chaincode application(s) and the ordering service node(s) .
- Each peer that joins a channel, has its own identity given by a membership services provider (MSP) .

Channel cont.

- channels are completely isolated,
- they have different ledgers, different height of blocks, policies, stories, rules.
- completely isolated instance of hyper ledger fabric.
- never exchange data.
- outside of a channel , one can't even see that there is a channel.
- you can make a policy who can see the data in the channel and who can make an operation.
- every single party inside a channel must agree about other parties.

Channel configuration properties

- **Versioned:** All elements of the configuration have an associated version which is advanced with every modification. Further, every committed configuration receives a sequence number.
- **Permissioned:** Each element of the configuration has an associated policy which governs whether or not modification to that element is permitted. Anyone with a copy of the previous configtx (and no additional info) may verify the validity of a new config based on these policies.
- **Hierarchical:** A root configuration group contains sub-groups, and each group of the hierarchy has associated values and policies. These policies can take advantage of the hierarchy to derive policies at one level from policies of lower

Chaincode

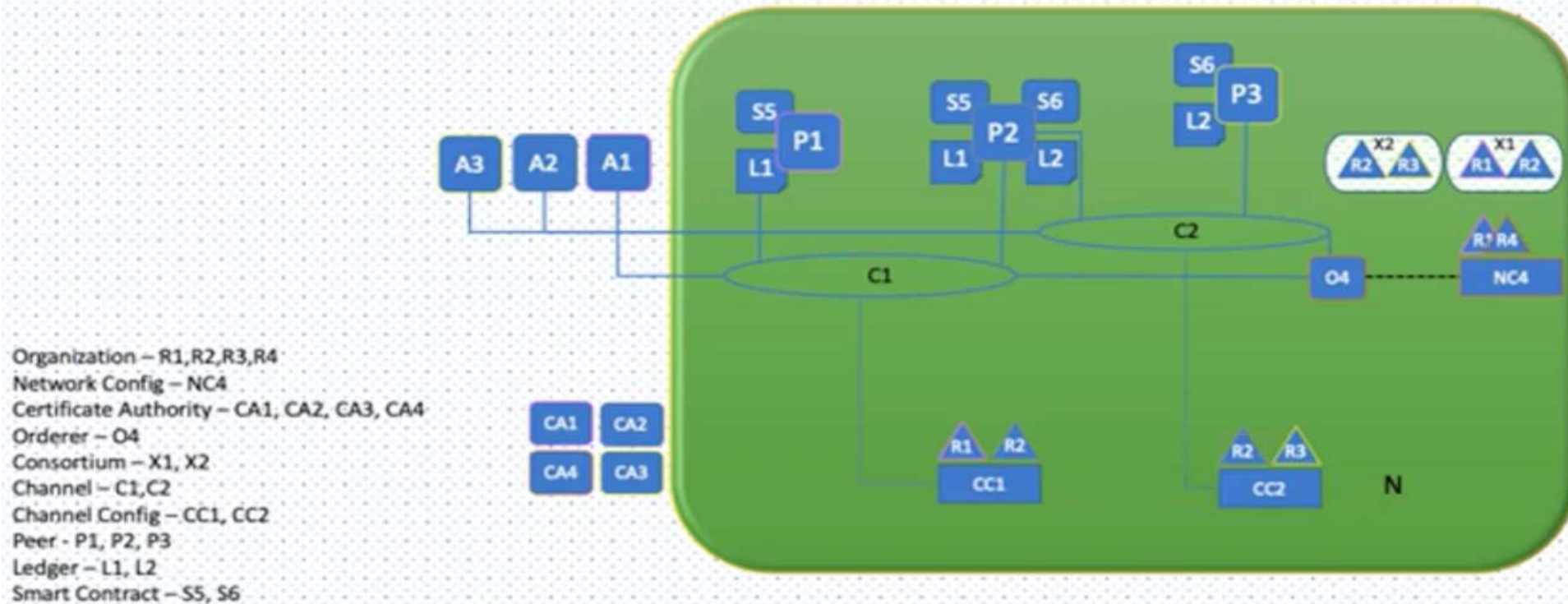
- A chaincode typically handles business logic agreed to by members of the network, so it is similar to a "smart contract".
- All your business logic is inside the chaincode.
- It's written in Go. Implementation of Java and JavaScript are on the way.
- Chaincode must be installed in every peer and channel.
- Policy must be provided.

Hyperledger Composer

- Hyperledger Composer is a set of collaboration tools for building blockchain business networks that make it simple and fast for business owners and developers to create smart contracts and blockchain applications to solve business problems
- Extensive
- Open development toolset and
- Framework to make developing Blockchain applications easier.

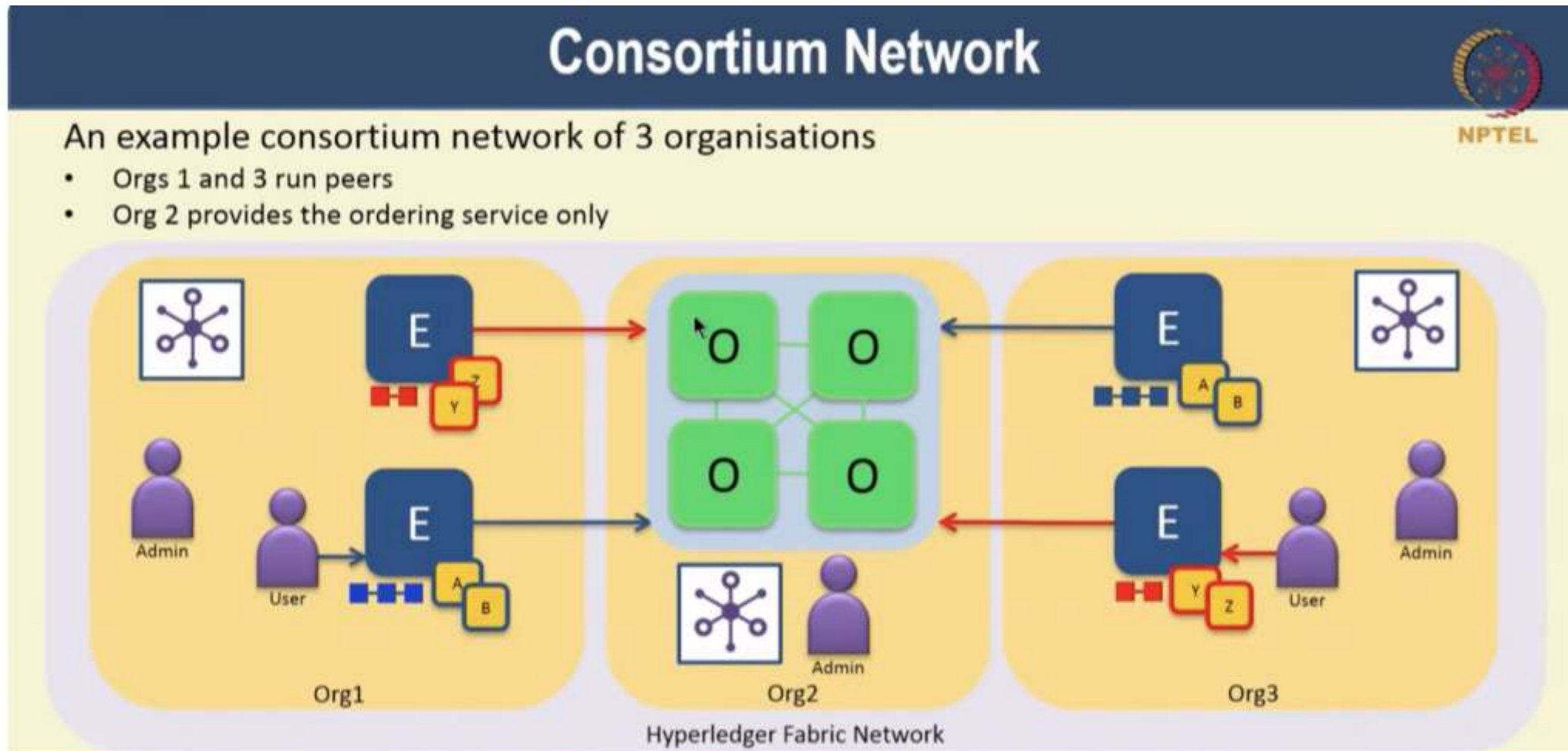
Design- Hyperledger Network

Lets DESIGN the network



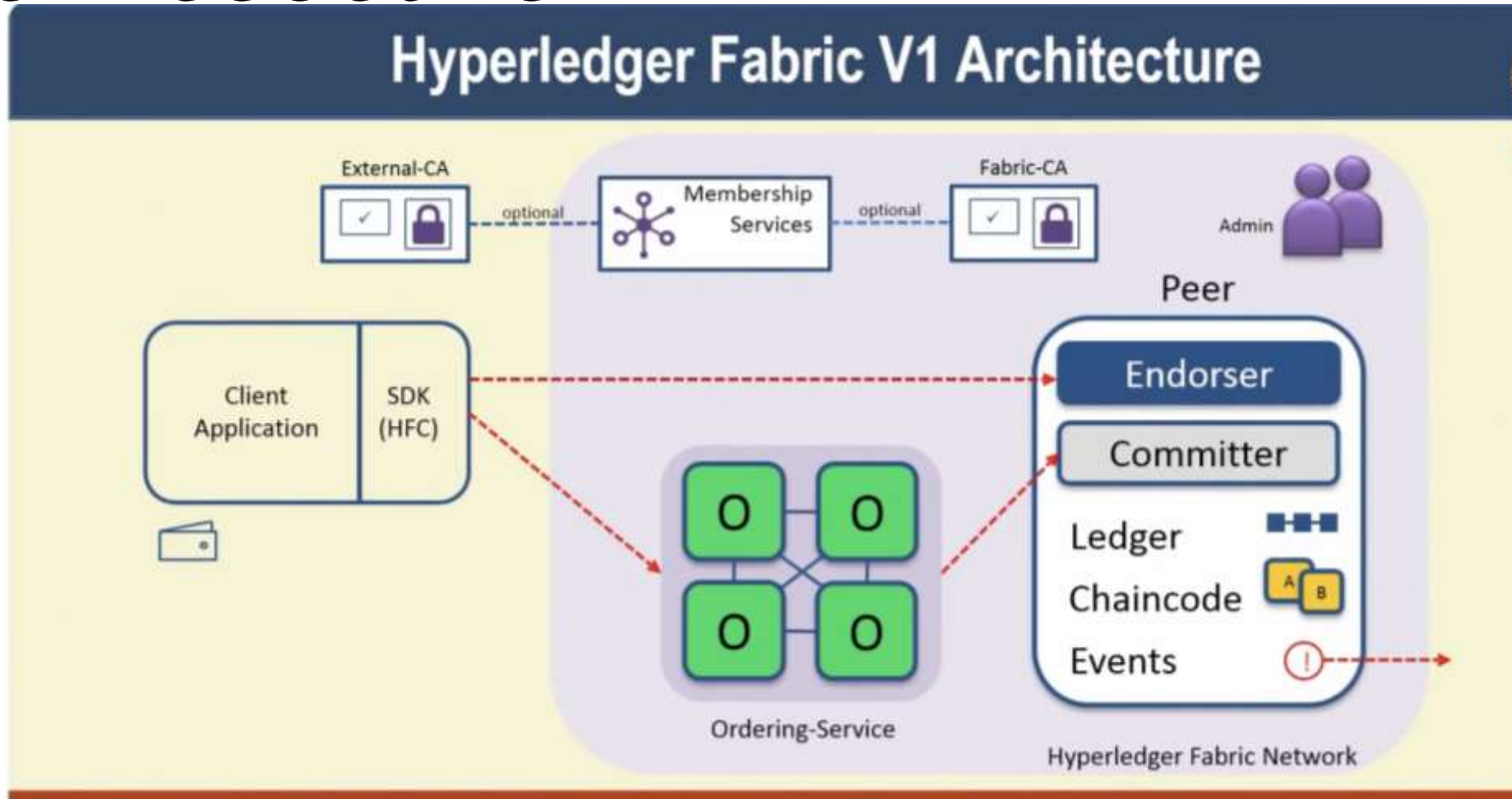
Source : hyperledger-fabric.readthedocs.io

Consortium Network



Source figure -NPTEL

Hyperledger Fabric Architecture



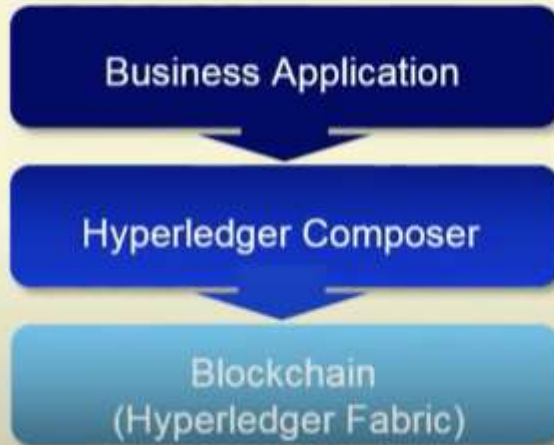
Source figure -NPTEL

Hyperledger Composer

Hyperledger Composer: Accelerating Time to Value

<https://hyperledger.github.io/composer>

- A suite of high level application abstractions for business networks
- Emphasis on **business-centric vocabulary** for quick solution creation
- Reduce risk, and increase understanding and flexibility



- Features
 - Model your business networks, test and expose via APIs
 - Applications invoke transactions to interact with business network
 - Integrate existing systems of record
- Fully open and part of Linux Foundation Hyperledger
- Try it in your web browser now:
<http://composer-playground.mybluemix.net/>

Reading Material

Fun Reading

- Certificate Authority, Wikipedia article: https://en.wikipedia.org/wiki/Certificate_authority
- Fabric Architecture Deep Dive: <http://hyperledger-fabric.readthedocs.io/en/release-1.0/arch-deep-dive.html>
- Fabric CA Documentation: <http://hyperledger-fabric-ca.readthedocs.io/en/latest>

Environment setup

Installing the pre-requisites

- Operating Systems: Ubuntu Linux 14.04 / 16.04 LTS (both 64-bit), or Mac OS 10.12
- Docker Engine: Version 17.03 or higher
- Docker-Compose: Version 1.8 or higher
- Node: 6.x (note versions 7 and higher are not supported)
- npm: v3.x or v5.x
- git: 2.9.x or higher
- Python: 2.7.x
- nvm and Apple Xcode (for Mac)
- Hyperledger Composer Extension for VSCode.

Implementation

- http://hyperledger-fabric.readthedocs.io/en/release-1.0/build_network.html

youtube.com

hyperledger fabric blockchain

Create

Actors in a Blockchain Solution

Blockchain Architect	Responsible for the architecture and design of the blockchain solution
Blockchain User	The business user, operating in a business network. This role interacts with the Blockchain using an application. They are not aware of the Blockchain.
Blockchain Regulator	The overall authority in a business network. Specifically, regulators may require broad access to the ledger's contents.
Blockchain Developer	The developer of applications and smart contracts that interact with the Blockchain and are used by Blockchain users.
Blockchain Operator	Manages and monitors the Blockchain network. Each business in the network has a Blockchain Network operator.
Membership Services	Manages the different types of certificates required to run a permissioned Blockchain.
Traditional Processing Platform	An existing computer system which may be used by the Blockchain to augment processing. This system may also need to initiate requests into the Blockchain.
Traditional Data Sources	An existing data system which may provide data to influence the behavior of smart contracts. to remember are the architect the developer So, some of you who are students, who are

Blockchain Components and Concepts

NPTEL-NOC IITM 517K subscribers

36K views 5 years ago

No description has been added to this video.

...more

Blockchain Architecture Design and Use Cases

- Blockchain Components and Concepts
- Blockchain for Enterprise - Overview
- Fabric - Membership and Identity Management
- Hyperledger Fabric Details
- Hyperledger Fabric - Transaction Flow
- Blockchain Architecture Design and Use Cases - I
- Fabric Demo on IBM Blockchain Cloud - I

