

# HyperLedger Introduction

Department of Computer Science and Engineering

Indian Institute of Technology, Jodhpur



Dr. Debasis Das

CSE, IIT Jodhpur

---

# HyperLedger Introduction

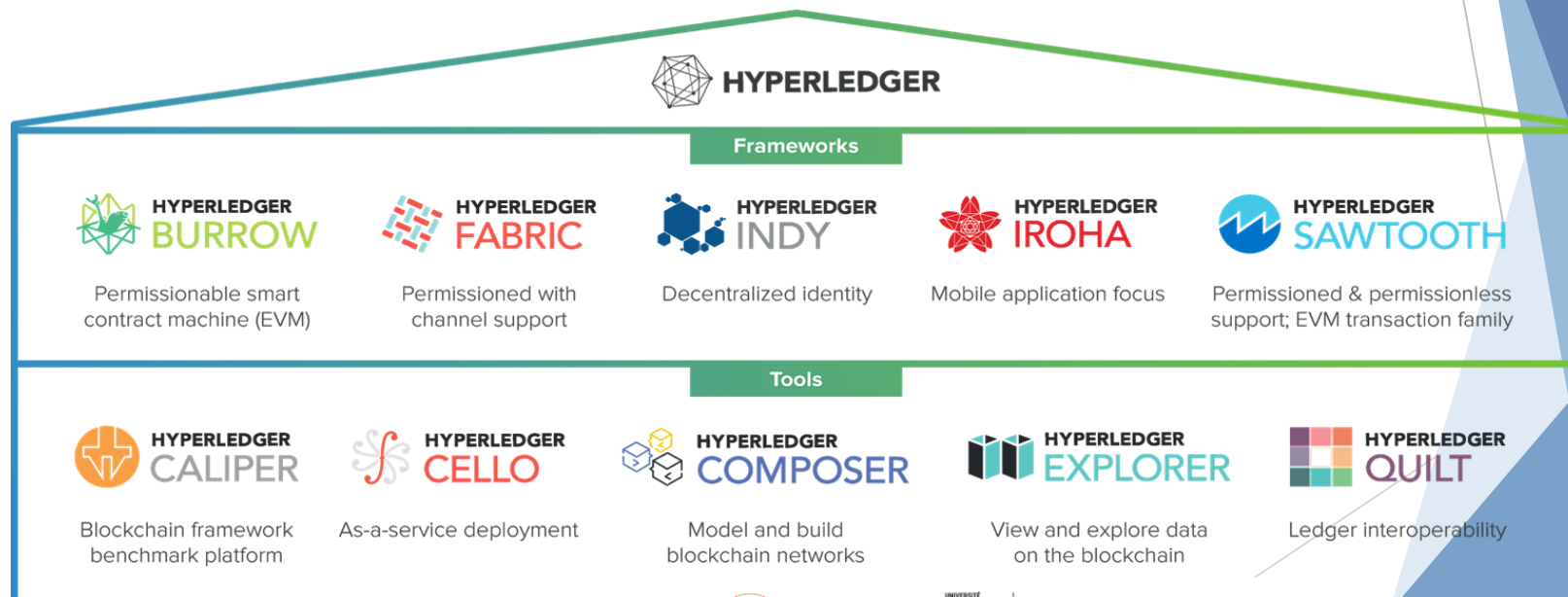
# Hyperledger Fabric

---

- 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**

# What is Hyperledger?

- Incubator of open source Blockchain projects created by The Linux Foundation
- Aims to bring the technology to companies



# 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



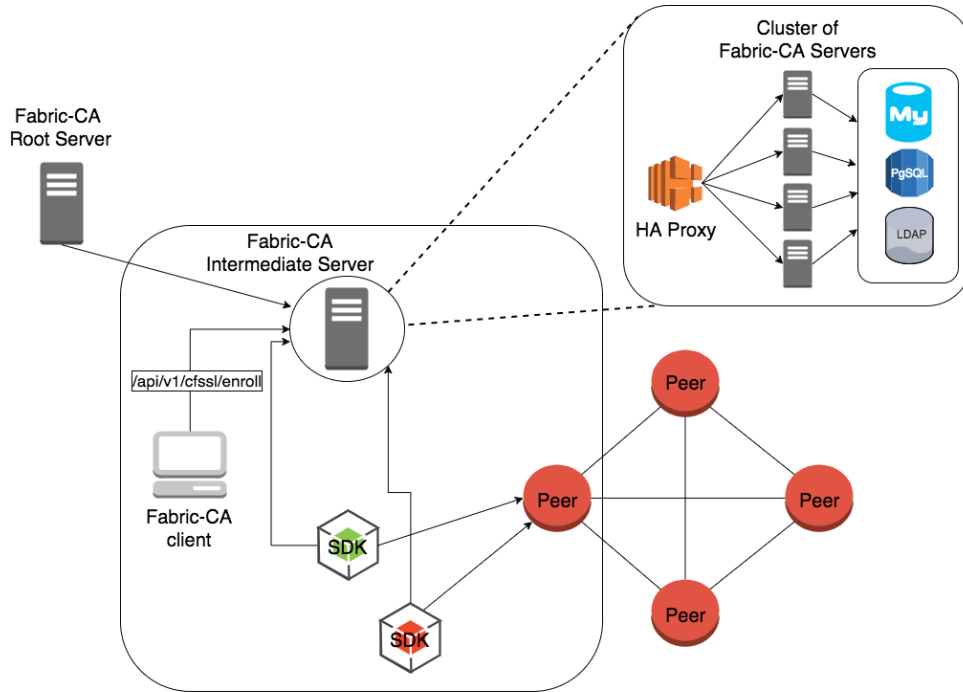
# Fabric CA

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 levels.



# 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.

# 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.**

THANK

YOU