Hyperledger Fabric

Understanding Hyperledger Fabric

Hyperledger Overview

DAY 1

Segment 1: Blockchain Basics, Hyperledger Fabric Foundation and Framework overview (120 Minutes)

Break 15 Minutes

Segment 2: Hyperledger Fabric Foundation and Framework Services (120 Minutes)

DAY 2

Segment 3: Hyperledger Fabric Use Cases and Solutions (120 Minutes)

Break 15 Minutes

Segment 4: Hyperledger Fabric Transactions, Development and Certification , Course Close Out (120 Minutes)

Hyperledger Overview

Prerequisites to be successful for exam

- Basic Knowledge of Blockchain
- Basic Knowledge of Computer Networking
- Download from Github materials
- Basic Blockchain knowledge, which can be gained by watching Introducing Blockchain LiveLessons

Hyperledger Overview

Course Expectations

- Engineers/Architects with basic knowledge in programming, IT networking and IT architecture
- Basic understanding of Golang, Java, or Javascript would be helpful but not needed for this foundations course

Recommended preparation:

- Basic Blockchain knowledge, which can be gained by watching Introducing Blockchain LiveLessons

Hyperledger Overview

Course Audience

- *Course is geared towards enterprise customers that are already considering "Permissioned" blockchains such as Hyperledger Fabric*
- *Course is mainly for Architects, Engineers, PreSales.*
- *Some focus for developers and programmers but not a development course or coding course.*

Hyperledger Overview

Resources

- Review Hyperledger.org website
- Review Hyperledger documentation and code
- Join the Community https://www.hyperledger.org/community
- Download from Github additional materials https://github.com/wearetheledger/awesome-hyperledger-fabric

Hyperledger Overview

What you'll learn-and how you can apply it

- What Hyperledger Fabric is and how it benefits your organization
- The advantages of Hyperledger vs other Blockchain
- Hyperledger Blockchain terminology and concepts
- Concepts and use cases of permissioned blockchains
- Design identities, entities, transactions, etc.
- Key services such as Endorsers, orderers, clients, smart contracts
- Hyperledger Composer, Explorer and Playground

Hyperledger Overview

Topics

- What is a Blockchain
- Consensus Algos
- Consensus with Hyperledger Fabric
- Hyperledger Project
- Hyperledger Fabric
- Hyperledger Fabric Nodes
- Hyperledger Fabric Composer and Playground
- Blockchain As a Service (BAaS)
- Planning and deploying Hyperledger

Hyperledger Overview

My Goal--- Demos as much as practical!

Hyperledger Overview

Lets Find out More about the Audience

Hyperledger Overview

Survey Question
What is your current role in your company (Closest)

- IT Infrastructure Manager
- C Level or Director Level
- Blockchain Focused/Dedicated Developer/Architect/Lead
- Cloud Architect/Admin
- Data Engineer/Big Data Engineer
- Developer Other than Blockchain
- IT Security Focused
- Other roles not listed

Hyperledger Overview

Survey Question
What type of company Vertical/Sector are you working for?

- IT Consulting/Professional Services
- IT Vendor/VAR/Reseller
- Financial Sector (Banking/Payments/Investments)
- Energy Sector (Fossil Fuels/Solar/PetroChems)
- Manufacturing (Electronics/Machines/Cars/Planes)
- Government Employment (Federal/State/etc)
- Logistics (Transportation/Shipping)
- Real Estate (Commercial/Residential/Industrial)
- Other Industry Vertical Sector Not Listed

Hyperledger Overview

Survey Question

Who currently is or is planning on using Hyperledger Fabric in their enterprise environments?

- Yes
- No plans to at this time
- Will be in 3 Months
- Will be in 6 Months or more

Hyperledger Overview

Survey Question

What Hyperledger Frameworks are you using in your enterprise environment?

- Fabric
- Burrow
- Sawtooth
- Burrow
- Iroha

Hyperledger Overview

Survey Question

What about Tools Modules in your enterprise?

- Chaintool
- Explorer
- Cello
- Composer

Hyperledger Overview

Survey Question
Is your organization considering any other blockchains?

- Ethereum
- R3 Corda
- Quorum
- Xtrabytes
- Other

Hyperledger Overview

Survey Question
Are you certified in blockchain? What vendor/Association certification?

- Blockchain Training Alliance(CBSA, CBDH,CBDE)
- Blockchain Council (CBE)
- IBM (Developers)
- C4 (CBP)
- Other Blockchain Certifications

Hyperledger Overview

Blockchain Basics Condensed Review

Hyperledger Overview

Blockchain History

History

The Byzantine Generals Problem (1982)

- The problem- A number of generals (from the same Army) have surrounded a walled city on all sides.
- The balance of power is such that if all generals attack at the same time, they will take the city.

History

The Byzantine Generals Problem (1982)

- The challenge -If the generals are not coordinated in their attack, they will lose the city and their campaign.

History

- The Byzantine Generals Problem (1982)
- The solution – Use cryptography to encrypt messages
- Provides mathematical computation power
- Ensures privacy

History

The Byzantine Generals Problem (1982)

- The solution- In 2008 a whitepaper is published by “Satoshi Nakamoto” which outlines a solution to the Byzantine Generals problem
- Bitcoin was started and thus the cryptocurrency launched in 2009

History

The Byzantine Generals Problem (1982)

- Bitcoin uses a Proof of Work(PoW) consensus algo.
- Work must be provided to solve the problem
- Know that Bitcoin was the original solution to the BFT problem.

History

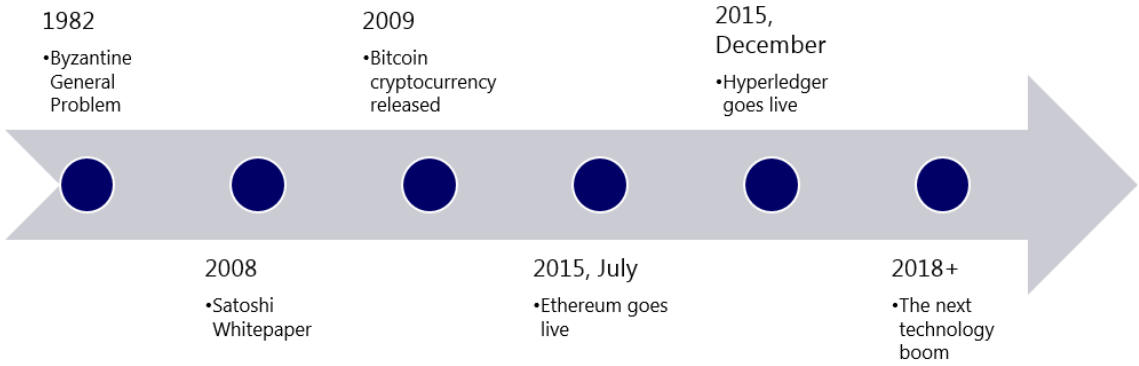
Blockchain release dates

Some release dates for popular blockchains to know

- 2009 Bitcoin
- 2015 Ethereum
- 2015 Hyperledger

History

- History of Blockchain



Hyperledger Overview

Blockchain Basics

Blockchain Basics

- A cryptographically secure, shared, distributed ledger.
- Immutable transactions are written on this distributed ledger on distributed nodes
- Transformational technology in which business and government invest in.
- It’s a decentralized database which stores information in the form of transactions

Blockchain Basics

- Book = Blockchain
- Page = Block
- Page Entry = Blockchain transaction
- Lets think of the blockchain as a book that can be written to but not erased.
- Blockchains can be private or public
- Blockchains are revolutionary way of implementing “trust” into a platform.

Blockchain Basics

- A blockchain is a globally shared data structure, transactional backend database.(In Bitcoin it's generally called a ledger)
- This means that everyone can read entries in the database just by participating in the network.
- If you want to change something in the database, you have to create a so-called "transaction" which has to be accepted by all others.
- The word "transaction" implies that the change you want to make (assume you want to change two values at the same time) is either not done at all or completely applied.

Blockchain Basics

Compare blockchain to other technology

- Telcom network to a telephone
- Databases are centralized where the blockchain is decentralized.
- Blockchains are not built from a new technology. Built from a unique synching of three existing technologies.
- Private or Public

Blockchain Technologies

Built from these technologies.

- P2P Networks
- Private Key Encryption
- Programs

Blockchain Digital Identity

Digital Identity is established

- Combining a public and private key creates a strong digital identity reference based on possession.
- Private Key
- Public Key

Blockchain Revolution

Blockchain is revolutionary in several ways

- Blockchain is not new technology but a synching of technologies that now make sense.
- Trust is at the center and essentially removes intermediaries. (efficiency)
- Tamperproof public ledger of value.
- Disruptive to the status quo. Legacy is out
- Platform with numerous use cases.

Blockchain Review

Blockchain is complex technology but is a simple concept really. Trust is at the center.

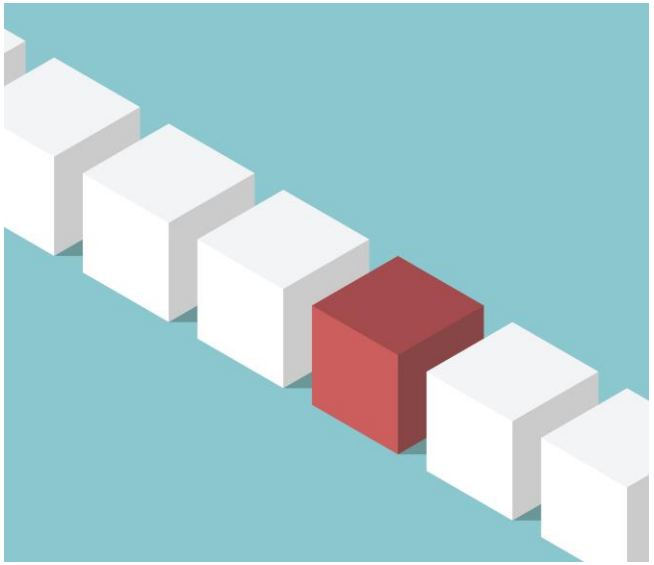
- Blockchains are ledgers shared among computers around the world
- The ledgers in a blockchain are immutable.
- Blockchain is not new technology but a synching of technologies

Hyperledger Overview

Blockchain Ledgers

Blockchain Ledgers

- A ***ledger*** is an append-only record store, where records are immutable and may hold more general information than financial records.

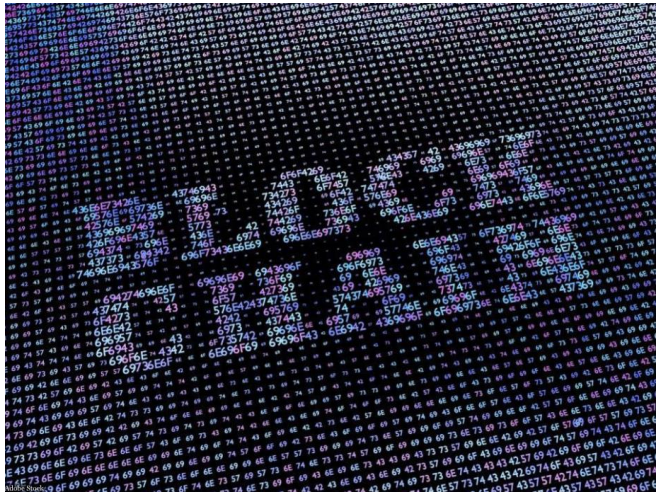


Blockchain Ledgers

- A **distributed *ledger*** is a database that is stored and updated independently by each or node in the blockchain.
- The decentralized and distributed nature is what makes it unique.
- In blockchain they are immutable
- Every single node on the network processes every transaction that occurs.

Blockchain Ledgers

- Consensus is when the distributed ledger has been updated and all nodes maintain their own identical copy of the ledger.
- This architecture allows for a new capacity as a system of recordkeeping that goes beyond being a simple database.



Hyperledger Overview

Blockchain Key Components

Blockchain Components

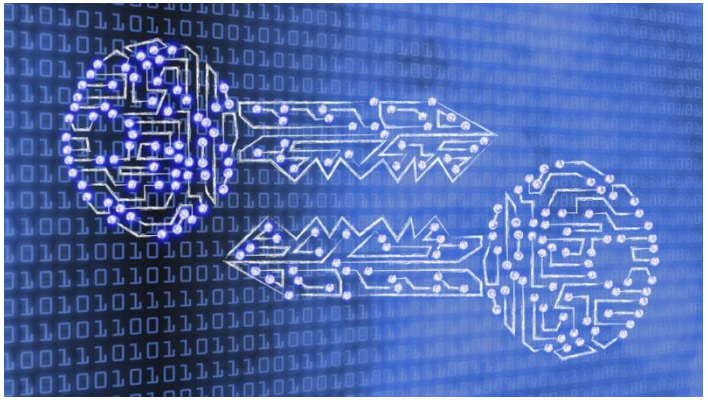
Blockchain Key Components

- Cryptography
- P2P Network
- Shared Digital ledger
- Consensus algorithm
- Validity Rules
- Virtual machine

Blockchain Components

Blockchain Key Components

- Cryptography for transactions.
- Recorded, encrypted and secured between peers in blockchain.
- No need for a centralized authority.



Blockchain Components

Blockchain Key Components

- P2P Network connect the blockchain nodes
- All computers share responsibility on the network
- Workloads are shared

Blockchain Components

Blockchain Key Components

- Shared Digital Ledger is a data structure managed inside the node application.
- Distributed Database held and updated independently by each participant (or node) in a large network.

Blockchain Components

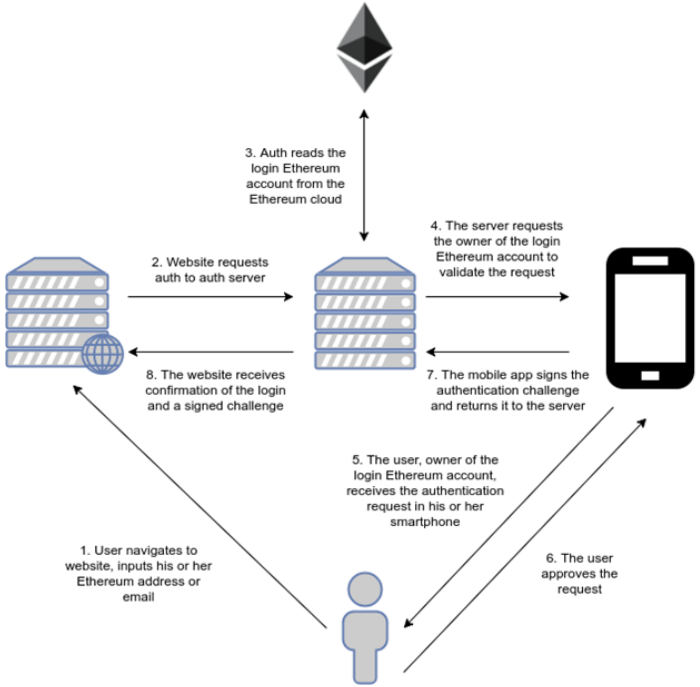
The Consensus algorithm is implemented as part of the node application for how the ecosystem comes to a single view of the ledger.

- Different ecosystems have different methods for attaining consensus
- Determines method for “World State”

Blockchain Components

Blockchain Key Components

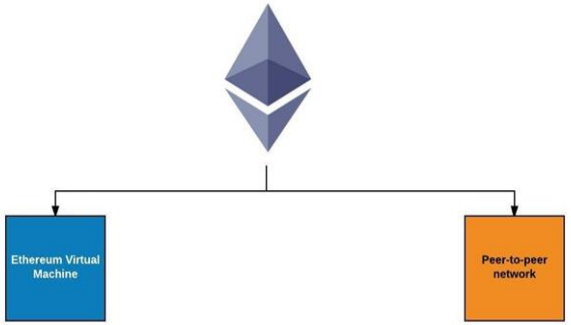
- Validity Rules (validation) state how the user and the transactions will be validated.



Blockchain Components

Blockchain Key Components

- Virtual Machines are a representation of a server created by a computer program and operated with instructions embodied in a language.
- Ethereum and Bitcoin use VMS.
- The virtual machine lives in the Ethereum node applications for example



Hyperledger Overview

Blockchain Architectures

Blockchain Architectures

Blockchains are usually either permissioned or permissionless.

- Were originally developed as permissionless such as BTC
- Permissionless (Public) or Permissioned (Private)
- Enterprises generally favor Permissioned for good reasons.

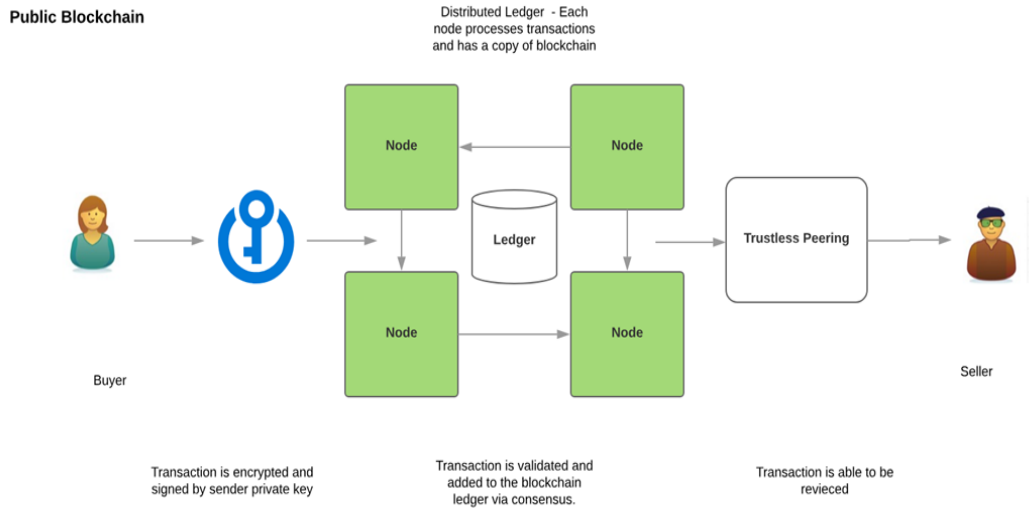
Blockchain Architectures

What is a Public Blockchain?

- Public Blockchains are also referred to as permissionless or Open blockchains.
- Bitcoin was the original permissionless blockchain.
- Transactions are processed by all nodes in the blockchain
- Transactions are publicly viewable(transparent) in the blockchain
- Widely distributed. For example Ethereum has over 16,000 nodes worldwide

Blockchain Architectures

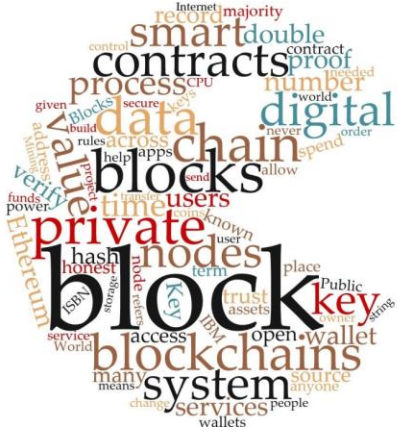
What is a Public Blockchain?



Blockchain Architectures

Public Blockchains Benefits

- Open Read and Write
- Ledger is distributed
- Censorship resistant
- Secure due to mining (51% rule)



Blockchain Architectures

Public Permission-less Blockchains Examples

- Bitcoin
- Ethereum
- Monero

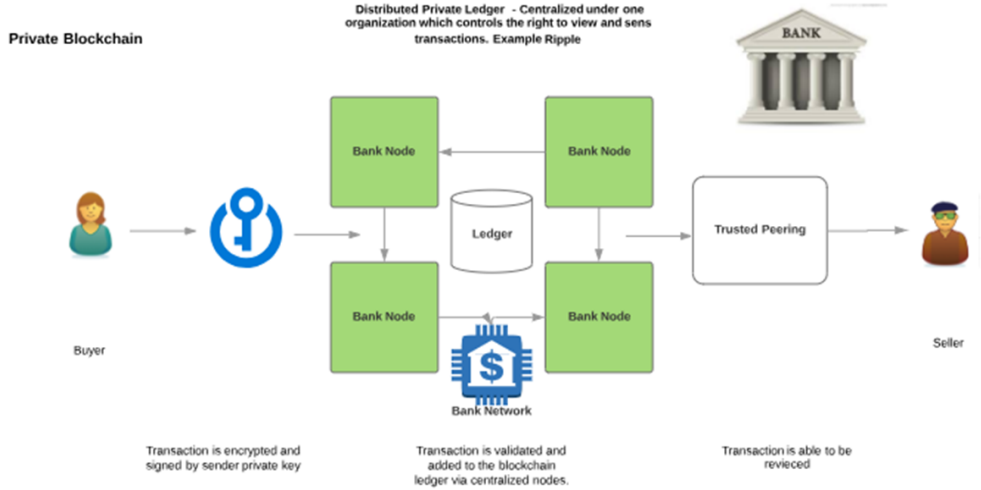
Blockchain Architectures

What are Private Blockchains?

- Private Blockchains are also referred to as permissioned or enterprise blockchains.
- Can be Open Sourced, Consortium or privately developed
- Transactions are processed by select nodes in the blockchain
- Transactions are not publicly viewable(transparent) in the blockchain
- Locally distributed.

Blockchain Architectures

What is a Private Blockchain?



Private Blockchain Benefits

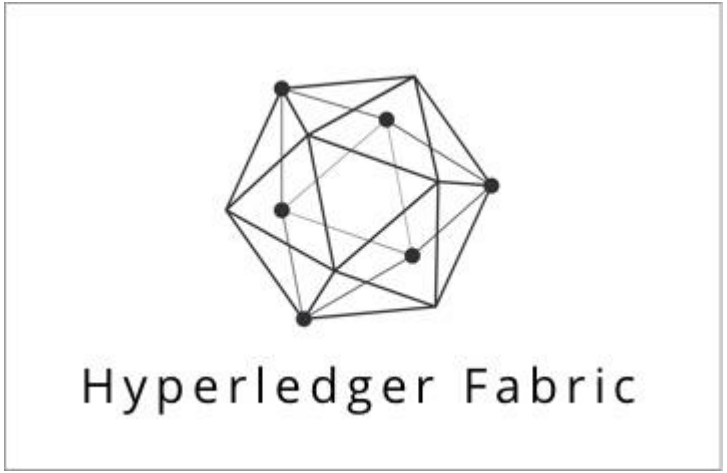
Some Private Blockchains Benefits Are

- Enterprise Permissioned
- Faster Transactions
- Better Scalability
- Compliance Support
- Consensus more efficient (Less nodes)

Blockchain Architectures

Private Permissioned Blockchains Examples are

- Hyperledger Fabric
- R3 Corda
- Quantum
- Ripple



Private vs Public Blockchain

Blockchains

- Permissionless (Public) or Permissioned (Private)

	Public (Permissionless)	Private (Permissioned)
Access to Ledger	Open Read/Write	Permissioned Read/Write
Identity	Anonymous	Known Identities
Security and Trust	Open Network (Trust Free)	Controlled Network(Trusted)
Transaction Speed	Slower	Faster
Consensus	POW/POS	Proprietary or Modular
Open Source	Yes	Depends on Blockchain
Code Upkeep	Public	Consortium or Managed
Examples	Ethereum, Multichain	R3 Corda, Quantum, Hyperledger

Private vs Public Blockchain

Considerations on permissioned vs permissionless blockchain?

- Governance
- Industry Vertical
- Smart Contract Functionality
- Cryptocurrency Requirement
- Consensus Algorithm
- Costing Model
- Integration

Private vs Public Blockchain

Considerations on permissioned vs permissionless blockchain?

- Costing Model
- Integration into enterprise
- Trust
- Transparency
- Privacy
- Security

Hyperledger Overview

Trust Blockchains

Establishing Trust in Blockchain

- Blockchain technology is about storing some kind of data, which are transactions in case of the Bitcoin blockchain.
- Blockchain is essentially transferring trust from an intermediary to technology. (Software Code)
- Storing data in the blockchain is thru cryptographic functions.
- Private Key/Public Key

Trust Blockchains

Establishing Trust in Blockchain

- All transaction data on a chained block is assumed to be trustworthy
- The users base this trust on

1. This data has not been tampered with
2. The Blockchain is immutable

Trustless Blockchains

What is “trustless” in the Blockchain?

- “Trustless” is a model that does not require “trust” to safely interact and transact lets say a “deed” or “purchase”.
- “Trustless blockchains” in reality is a transfer of trust to technology from organizations (Banks, government, corporations).
- Blockchain is built on the premise that “transparent code” essentially removes the need for intermediaries. (technology)
- Smart contracts can essentially reduce the need for accountants, lawyers, bankers, etc essentially trust is transformed.
- “Trustless” in blockchain essentially creates the trust by default.

Trustless Blockchains

Trustless Blockchains

- Financial transparency can result thru the use of blockchains and thus reduces the need for “intermediaries”.
- The blockchain is based on a consensus algo where all nodes agree that the transaction is valid.
- Basically, the ledger acts a trust broker when two parties who don’t trust each other want to trade for example.
- Welcome to the “software driven world”.

Hyperledger Overview

Blockchain Consensus Algos

Blockchains Algos

- Consensus is a dynamic way of reaching agreement in a group.
- While voting just settles for a majority rule without any thought for the feelings and well-being of the minority.
- Consensus makes sure that an agreement is reached which could benefit the entire group as a whole

Blockchains Algos

Blockchain Consensus Algos

- *Proof of Work*
- *Proof of Stake*
- *Delegated Proof-of-Stake (DPoS)*
- *Byzantine Fault Tolerance (BFT)*
- *Directed Acyclic Graphs (DAGs)*
- *Other Algos*

Blockchains Algos

Proof of Stake the blocks aren't created by miners doing work

- Instead by *minters staking* their tokens to "bet" on which blocks are valid.
- In the case of a fork, minters spend their tokens voting on which fork to support.
- Attacks costly but more environmental
- Peercoin and Ethereum will go to.

Blockchains Algos

Proof of Work was the first blockchain consensus algorithm.

- Satoshi Nakamoto created for the Bitcoin blockchain
- PoW, *miners* solve hard problems to create blocks.
- PoW runs on a system of “the longest chain wins.”
- Expensive both cost and environmentally
- Bitcoin, Ethereum, Litecoin

DPOS Consensus

DPoS divides the consensus model in two fundamental parts

1. Electing a group of block producers
2. Scheduling production.

- Not everyone in a DPoS network can produce blocks to validate a transaction

DPOS Consensus

- In the DPoS participants who hold tokens are able to cast votes to elect block producers
- Votes are weighted by the voter's stake, and the block producer candidates that receive the most votes are those who produce blocks.
- In DPoS you can think as stakeholders as notaries and block producers as witnesses

DPOS Consensus

- The benefits
- Separation of concerns
- Stakeholder (token) control
- Scalability
- On Chain Governance
- Avoids the “Nothing at Stake” problem

DPOS Consensus

- Avoids the “Nothing at Stake” problem by addressing the famous Nothing-at-Stake problem in PoS networks in which a small group of validators can take control of the network.
- The fixed of number of token validators in DPoS as well as the dynamic election model prevents this issues from happening.

Byzantine Fault Tolerance

The Byzantine Generals Problem

- A number of generals have surrounded a walled city on all sides.
- The balance of power is such that if all generals attack at the same time, they will take the city.
- Generals must coordinate attack or they could lose the city and their campaign.

Byzantine Fault Tolerance

The Byzantine Generals Problem with computers

- “Byzantine Generals’ Problem” states that no two node on a decentralized network can entirely and irrefutably guarantee that they are displaying the same data.
- Satoshi Nakamoto solved the issue with the BTC POW Consensus algo.

Byzantine Fault Tolerance

The Byzantine Generals Problem with computers

- Essentially a “Byzantine” node is a node that can be rogue by not forwarding packets or perhaps mislead other nodes involved in the P2P Consensus network.
- Hyperledger Fabric out of the box does not provide PBFT, but offers its users to add this consensus mechanism modularly.

Byzantine Fault Tolerance

- IBM backed Hyperledger uses this consensus algorithm.
- (Indy) Plenum Byzantine Fault Tolerance and for Iroha BFT consensus algorithm called Sumeragi
- In PBFT each node maintains an internal storage.
- When a node receives a message, it is signed by the node to verify its format.
- Once enough of the same responses are reached, then a consensus is met that the message is a valid transaction.

Other Consensus Algos

Proof of Elapsed Time

- Created by Intel to run on their trusted execution environment
- Similar to Proof of Work but more energy efficient
- Major Issue – requires trust in Intel, places power back in the hands of a central authority
- Hyperledger Sawtooth (Lottery Based) The current implementation of Hyperledger Sawtooth builds on a Trusted Execution Environment (TEE) provided by Intel's Software Guard Extensions (SGX).

Other Consensus Algos

Proof of Authority

- Uses a set of “authorities” – nodes that are explicitly allowed to create new blocks and secure the blockchain
- Replacement for PoW - Private blockchains only
- Earn the right to become a validator/authority

Other Consensus Algos

Proof of Burn

- Coins are “burned” by sending them to an address where they cannot be retrieved
- The more coins you burn, the better your chances of being selected to mine the next bloc
- Eventually, you must stake more by burning more coins

Other Consensus Algos

Proof of Activity

- Hybrid of PoW and PoS
- Empty template blocks are mined (PoW), then filled with transactions which are validated via Proof of Stake

Other Consensus Algos

Proof of Capacity

- Pay to play with hard drive space
- The most space you ‘stake’ the better your odds of being selected to mine the next block
- Consensus algorithm generates large data sets called ‘plots’ which consume storage
- Major criticism – this method has no real deterrent for bad actors

Hyperledger Overview

Review Questions

Review Questions

A Blockchain ledger is on a distributed network that is _____.

a. shared by all the computers on the network

b. controlled by a central authority

c. controlled by a common administrator

d. shared by other non-Blockchain networks

Review Questions

A public blockchain network is most noted for being _____?

a. very expensive to operate

b. being accessible to anyone

c. used for cryptocurrency exchange

d. all of the above

Review Questions

Permissioned networks are only used by_____?

a. anyone with network permissions

b. participants with a private key

c. authorized persons

d. industry officials

Review Questions

Consensus on the Blockchain defines_____

a. agreement of a valid transaction by all the network nodes

b. basic security of the computer network

c. the labeling of each data block

d. security between two blocks of data

Review Questions

Immutability provides the_____

a. assurance that a transaction cannot be altered

b. network configuration cannot be changed

c. network nodes will never fail

d. guarantee that participants will never put bad information on the Blockchain

Review Questions

When an asset is transferred on the Blockchain it _____

a. guarantees that all network nodes are functioning

b. solves the double-spend accounting problem

c. still belongs to the original owner until paperwork is completed

d. is only recorded by the recipient's bank

Review Questions

The consensus algorithms are used because _____

a. they can include specific rules or conditions to be met

b. they increase the network security from hacking

c. they prevent node failure

d. they increase network speed

Review Questions

Proof of Work is _____

a. fully supported in Hyperledger fabric

b. only used in cryptocurrency trading

c. used to save energy

d. commonly used for gaining consensus

Review Questions

Proof of Stake _____

a. uses more energy than proof of work

b. provides a block award if successful

c. uses mining by a validator

d. does not use mining for consensus

Hyperledger Overview

Blockchain Feature Comparison

Blockchain Feature Comparison

	Ethereum	Hyperledger	Corda	Ripple	Quorum
Industry	Cross	Cross	Financials	Financials	Cross
Governance	Developers	Linux Foundation	R3 Consortium	Ripple Labs	Developers and JP Morgan
Ledger Type	Permissionless	Permissioned	Permissioned	Permissioned	Permissioned
Consensus	PoW/PoS	Pluggable	Pluggable	Probablistic Voting	Majority Voting
Smart Contracts	Yes	Yes	Yes	No	Yes
Crypto $	Ether	NA	NA	XRP	NA
					HFS Research

Hyperledger Overview

Hyperledger Fabric Consensus

Hyperledger Fabric Consensus

Consensus must satisfy two properties to guarantee agreement among nodes: safety and liveness

- Safety - means that each node is guaranteed the same sequence of inputs and results in the same output on each node.
- Liveliness - means that each non-faulty node will eventually receive every submitted transaction. (Assumes that communication do not fail)

Hyperledger Fabric Consensus

- Hyperledger makes use of the permissioned voting-based consensus from the pool of other consensus named the lottery-based consensus. (Kafka in Hyperledger Fabric Ordering Service)
- Voting-based algorithms are advantageous in that they provide low-latency finality.
- More Nodes = More Time to reach Consensus…
- Trade off between Scalabity and Performance

Hyperledger Fabric Consensus

Hyperledger Fabric Consensus is planned out into 3 phases

- Endorsement
- Ordering
- Validation.

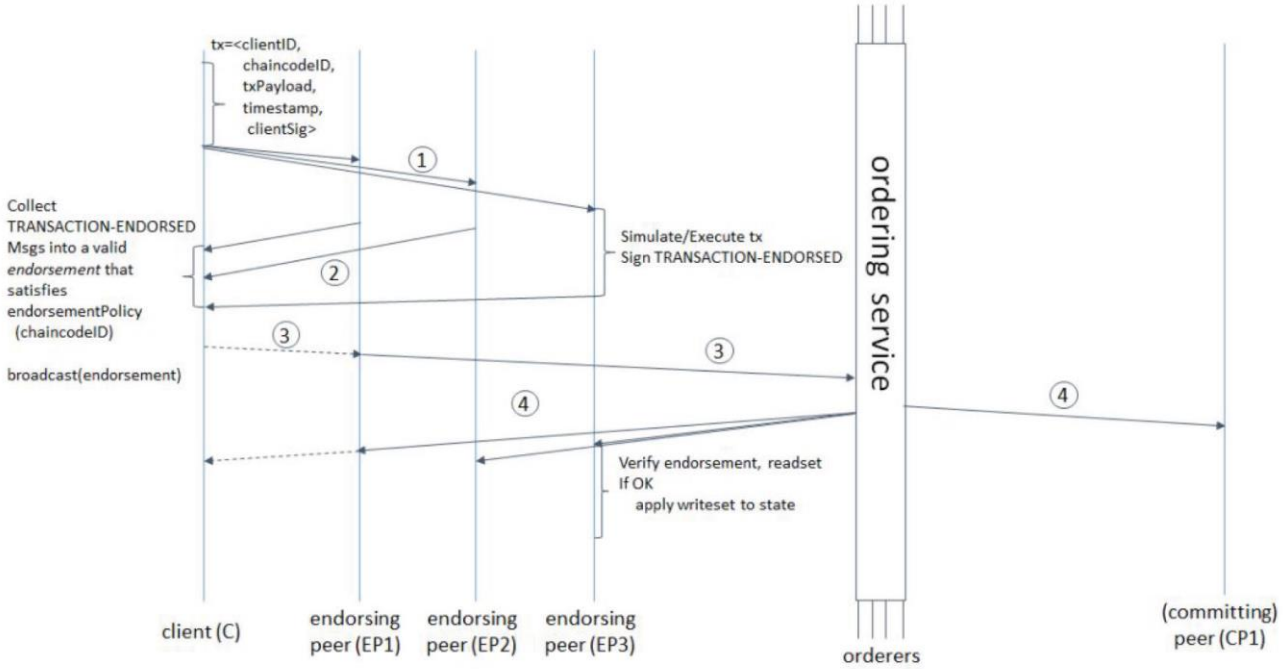
Hyperledger Fabric Consensus

- Endorsement is driven by policy (m out of n signatures) upon which participants endorse a transaction.
- Ordering phase will get the endorsed transaction and agrees to the order to be committed to the ledger.
- Validation takes a block of ordered transactions and validates the correctness of the result.

HYPERLEDGER PROJECT

Hyperledger Fabric Consensus

- Transaction Flow (Linux Foundation)



Review Question

- Voting-based algorithms are advantageous because they provide_____
- A) low-latency finality
- B) high-performance
- C) low-latency consensus
- D) high-security

Review Question

- Hyperledger Fabric Consensus is planned out into 3 phases. Which one is NOT a phase?
- A) Endorsement
- B) Ordering
- C) Validation
- D) Segregation

Hyperledger Overview

Hyperledger Project Overview

Hyperledger Project Overview

- Hyperledger is an open source project that came out of the Linux Foundation and was created in order to help advance cross-industry blockchain technologies.
- It is essentially a global open source collaboration involving leaders from numerous industries.

Hyperledger Project Overview

- Hosted by the Linux Foundation which provides a governance structure and oversight to the Hyperledger community.
- Open Source
- Uses a modular umbrella approach to enterprise blockchains

Hyperledger Project Overview

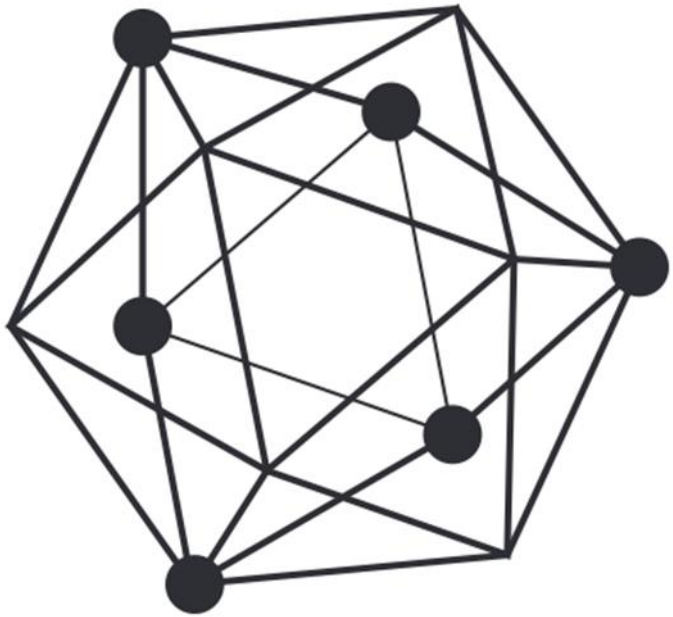
The Hyperledger Project consists of the following

- Infrastructure - Ecosystems that accelerate open development and commercial adoption
- Frameworks – A portfolio of differentiated approaches to business blockchain frameworks developed by a growing community of communities

Hyperledger Project Overview

The Hyperledger Project consists of the following

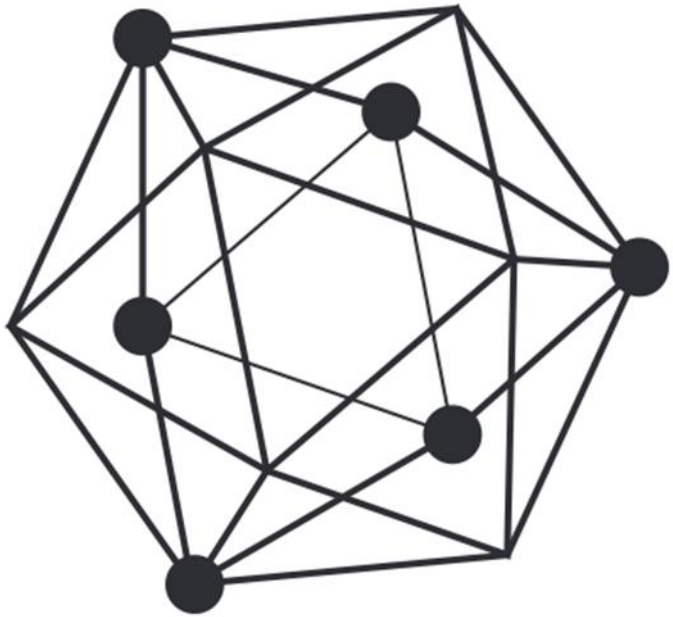
- Tools - Typically built for one framework, and through common license and community of communities approach, ported to other frameworks



Hyperledger Project Overview

The Hyperledger Project consists of the following

- Tools - Typically built for one framework, and through common license and community of communities approach, ported to other frameworks



Hyperledger Overview

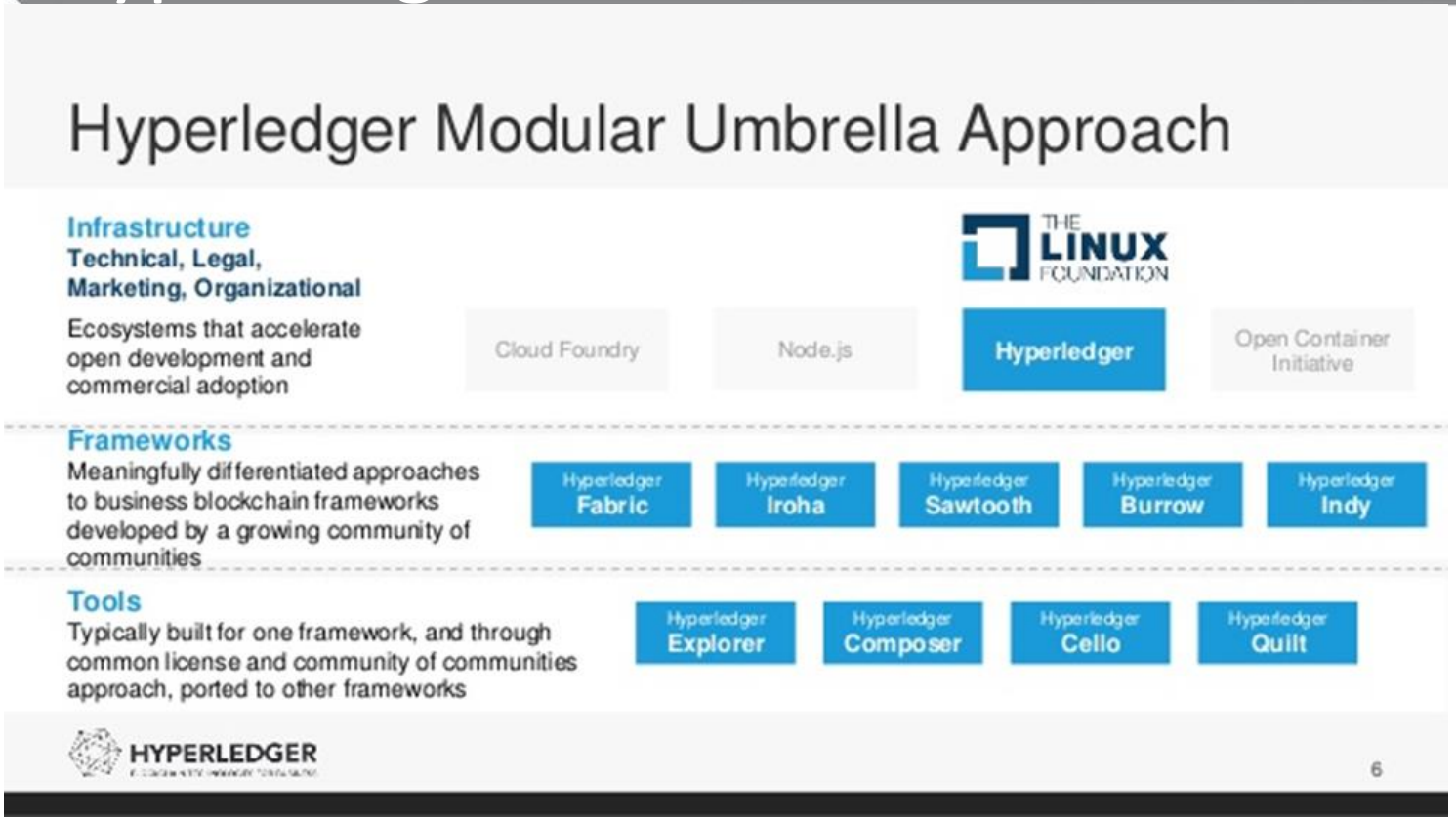
Hyperledger Portfolio

Hyperledger Portfolio

Hyperledger Project has a modular schema.

- Infrastructure
- Frameworks
- Tools

Hyperledger Portfolio



Hyperledger Portfolio

The Infrastructure Module is compromised of

- Leadership
- Governing Board
- Marketing
- Technical Steering Committee

Hyperledger Portfolio

The Framework Module

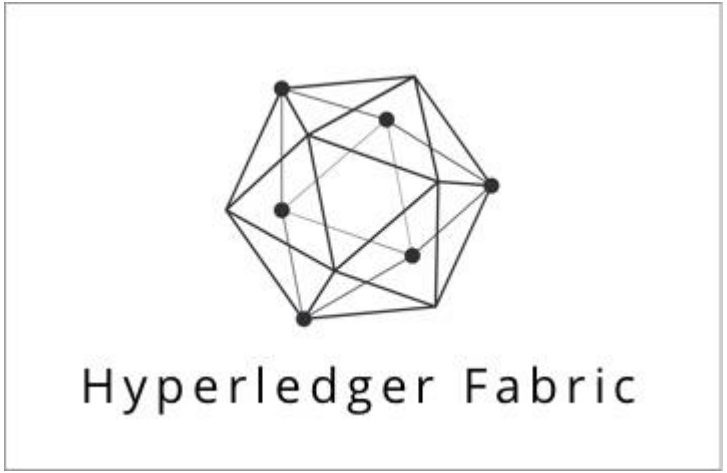
- Hyperledger Indy
- Hyperledger Fabric
- Hyperledger Iroha
- Hyperledger Sawtooth
- Hyperledger Burrow

Hyperledger Portfolio

- **Hyperledger Indy**: Tools, libraries, and reusable components for providing digital identities rooted on blockchains or other distributed ledgers so that they are interoperable across administrative domains, applications, and any other silo

Hyperledger Portfolio

- **Hyperledger Fabric**: Intended as a foundation for developing applications or solutions with a modular architecture
- Hyperledger Fabric allows components, such as consensus and membership services, to be plug-and-play



Hyperledger Portfolio

- **Hyperledger Iroha**: A business blockchain framework designed to be simple and easy to incorporate into infrastructural projects requiring distributed ledger technology

Hyperledger Portfolio

- **Hyperledger Sawtooth**: A modular platform for building, deploying, and running distributed ledgers.
- Hyperledger Sawtooth includes a novel consensus algorithm, Proof of Elapsed Time (PoET), which targets large distributed validator populations with minimal resource consumption

Hyperledger Portfolio

- **Hyperledger Burrow**: A permissionable smart contract machine. The first of its kind when released in 2014
- Burrow provides a modular blockchain client with a permissioned smart contract interpreter built in part to the specification of the Ethereum Virtual Machine (EVM).

Hyperledger Portfolio

The Tools Module

- Hyperledger Quilt
- Hyperledger Composer
- Hyperledger Explorer
- Hyperledger Cello
- Hyperledger Caliper

Hyperledger Portfolio

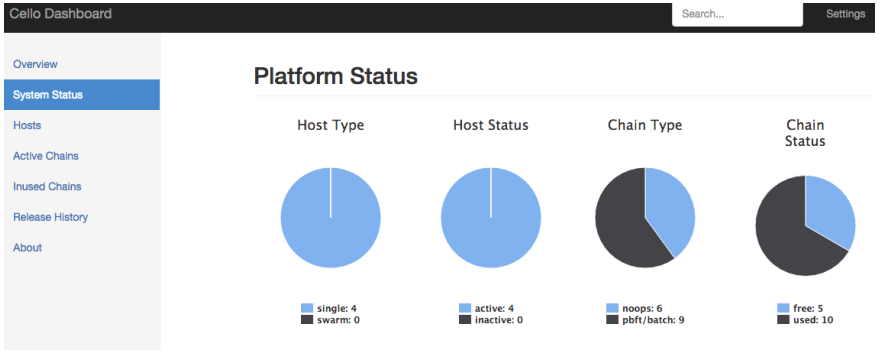
The Tools Module

- Hyperledger Quilt from NTT data and Ripple, is a Java implementation of the inter-ledger protocol by Ripple, which is designed to transfer values across distributed and non-distributed ledgers.

Hyperledger Portfolio

The Tools Module

- Hyperledger Cello was also contributed by IBM.
- It seeks to bring the on demand as-a-service deployment model into the blockchain ecosystem in order to reduce the effort required to create, manage, and terminate blockchains.



Hyperledger Portfolio

The Tools Module

- Hyperledger Composer: (contributed by IBM and Oxchains) is a set of collaboration tools for building blockchain business networks that accelerate the development of smart contracts and blockchain applications, as well as their deployment across a distributed ledger.

Hyperledger Portfolio

The Tools Module

- Hyperledger Explorer: Was originally contributed by IBM, Intel, and DTCC, can view, invoke, deploy or query blocks, transactions and associated data, network information (name, status, list of nodes), chain codes and transaction families, as well as other relevant information stored in the ledger.

Hyperledger Portfolio

The Tools Module

- Hyperledger Caliper: A blockchain benchmark tool that allows users to measure performance of a specific implementation with predefined use cases
- Its in Alpha mode at time of writing.
- Contributed by developers from numerous organizations.

Hyperledger Infrastructure

Hyperledger Infrastructure

Hyperledger Project has a modular schema.

- Infrastructure
- Frameworks
- Tools

Lets discuss Infrastructure

Hyperledger Infrastructure

The Infrastructure Module is compromised of

- Leadership
- Governing Board
- Marketing
- Technical Steering Committee

Hyperledger Infrastructure

Hyperledger Framework

Framework	Application
Indy	Distributed ledger and utility library
Iroha	DLT, Smart Contract Engine, Utility Libraries (Mobile)
Sawtooth	DLT, Smart Contract Engine
Burrow	Permissioned smart contract application engine
Fabric	DLT, Smart Contract Engine

Hyperledger Framework

Framework	Smart Contract Technology	Smart Contract Type	Language(s) for Writing Smart Contracts
Hyperledger Burrow	Smart contract application engine	On-Chain	Native language code
Hyperledger Fabric	Chaincode	Installed	Golang (> v1.0) or Javascript (> v1.1)
Hyperledger Indy	None	None	None
Hyperledger Iroha[2]	Chaincode	On-chain	Native language code
Hyperledger Sawtooth	Transaction families	On-Chain and Installed	C++, Go, Java, JavaScript, Python, Rust, or Solidity (through Seth)

Hyperledger Framework

- **Hyperledger Indy**: Tools, libraries, and reusable components for providing digital identities rooted on blockchains or other distributed ledgers so that they are interoperable across administrative domains, applications, and any other silo

Hyperledger Framework

- **Hyperledger Fabric**: Intended as a foundation for developing applications or solutions with a modular architecture
- Hyperledger Fabric allows components, such as consensus and membership services, to be plug-and-play

Hyperledger Framework

- **Hyperledger Iroha**: A business blockchain framework designed to be simple and easy to incorporate into infrastructural projects requiring distributed ledger technology

Hyperledger Framework

- **Hyperledger Sawtooth**: A modular platform for building, deploying, and running distributed ledgers.
- Hyperledger Sawtooth includes a novel consensus algorithm, Proof of Elapsed Time (PoET), which targets large distributed validator populations with minimal resource consumption

Hyperledger Framework

- **Hyperledger Burrow**: A permissionable smart contract machine. The first of its kind when released in 2014
- Burrow provides a modular blockchain client with a permissioned smart contract interpreter built in part to the specification of the Ethereum Virtual Machine (EVM).

Hyperledger Infrastructure

Hyperledger Portfolio

The Tools Module

- Hyperledger Quilt
- Hyperledger Composer
- Hyperledger Explorer
- Hyperledger Cello
- Hyperledger Caliper

Hyperledger Portfolio

The Tools Module

- Hyperledger Quilt from NTT data and Ripple, is a Java implementation of the inter-ledger protocol by Ripple, which is designed to transfer values across distributed and non-distributed ledgers.

Hyperledger Portfolio

The Tools Module

- Hyperledger Cello was also contributed by IBM.
- It seeks to bring the on demand as-a-service deployment model into the blockchain ecosystem in order to reduce the effort required to create, manage, and terminate blockchains.

Hyperledger Portfolio

The Tools Module

- Hyperledger Composer: (contributed by IBM and Oxchains) is a set of collaboration tools for building blockchain business networks that accelerate the development of smart contracts and blockchain applications, as well as their deployment across a distributed ledger.

Hyperledger Portfolio

The Tools Module

- Hyperledger Explorer: Was originally contributed by IBM, Intel, and DTCC, can view, invoke, deploy or query blocks, transactions and associated data, network information (name, status, list of nodes), chain codes and transaction families, as well as other relevant information stored in the ledger.

Hyperledger Portfolio

The Tools Module

- Hyperledger Caliper: A blockchain benchmark tool that allows users to measure performance of a specific implementation with predefined use cases
- Its in Alpha mode at time of writing.
- Contributed by developers from numerous organizations.

Hyperledger Infrastructure

Hyperledger Advantages and Use Cases

Hyperledger Advantages

- Innovative architecture which allows Hyperledger Fabric the ability to deliver unique network capabilities such as enhanced privacy and confidentiality, efficient processing, scalability, standard programming languages, and a modular structure that can be customized for individual deployments.
- Such capabilities make Fabric a suitable blockchain platform for businesses.

Hyperledger Advantages

Main Use Cases for Enterprises

- Asset Exchange
- Consortium Ledger Requirements
- High Value Markets
- Compliance Requirements
- Supply Chains and Logistics
- Business Contracts

Hyperledger Infrastructure

Hyperledger Fabric Design

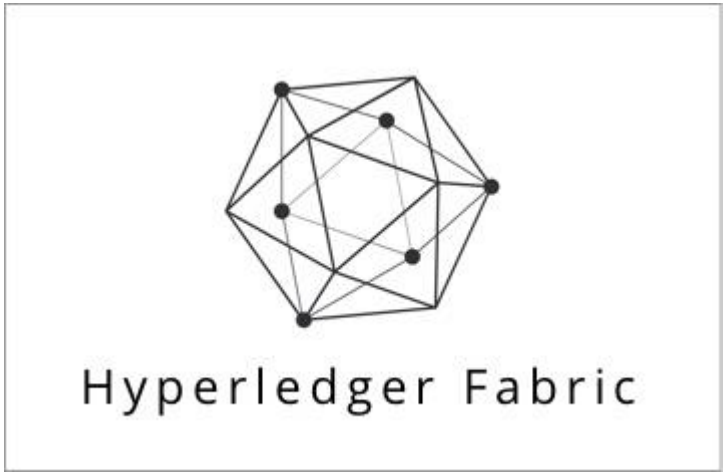
Hyperledger Fabric Design

- Hyperledger Fabric is a blockchain implementation that is designed for deploying a modular and extensible architecture.
- It has a modular subsystem design so that different implementations can be plugged in and implemented over time.
- Hyperledger Fabric is essentially enterprise driven and supports the enterprise fully

Hyperledger Fabric Design

Modular and extensible

- This means modularity in all components of all frameworks.
- Consensus layer
- Smart contract layer
- Communication layer



Hyperledger Fabric Design

Modular and extensible (cont)

- Communication layer
- Data store
- Identity services (root of trust—to identify the participants)
- APIS
- And more components

Hyperledger Fabric Design

Interoperability

- This principle is around backward interoperability and not focused on the interoperability between the various Hyperledger project-powered blockchain systems or business networks.
- API Suite

Hyperledger Design

Secure Solutions

- Enterprise and therefore business network security is paramount.
- The focus is on the interaction between components and the structure that governs the permissioning nature of permissioned blockchains.
- Enterprise prefer permissioned blockchains

Hyperledger Design

Token or CryptoCurrency Agnostic

- Hyperledger projects do not use crypto-assets, cryptocurrency, tokens, or coin-like constructs as incentive mechanics to establish trust systems.
- Behlendorf and the team at Hyperledger remain "sympathetic" to the interest in ICOs, but don't see a future where Hyperledger itself issues a crypto token.

Hyperledger Design

Rich APIS

- The focus here is to ensure that blockchain systems have not only enterprise middleware access but instead access to business networks, existing participants, and new systems without exposing the details of blockchain powered business networks.

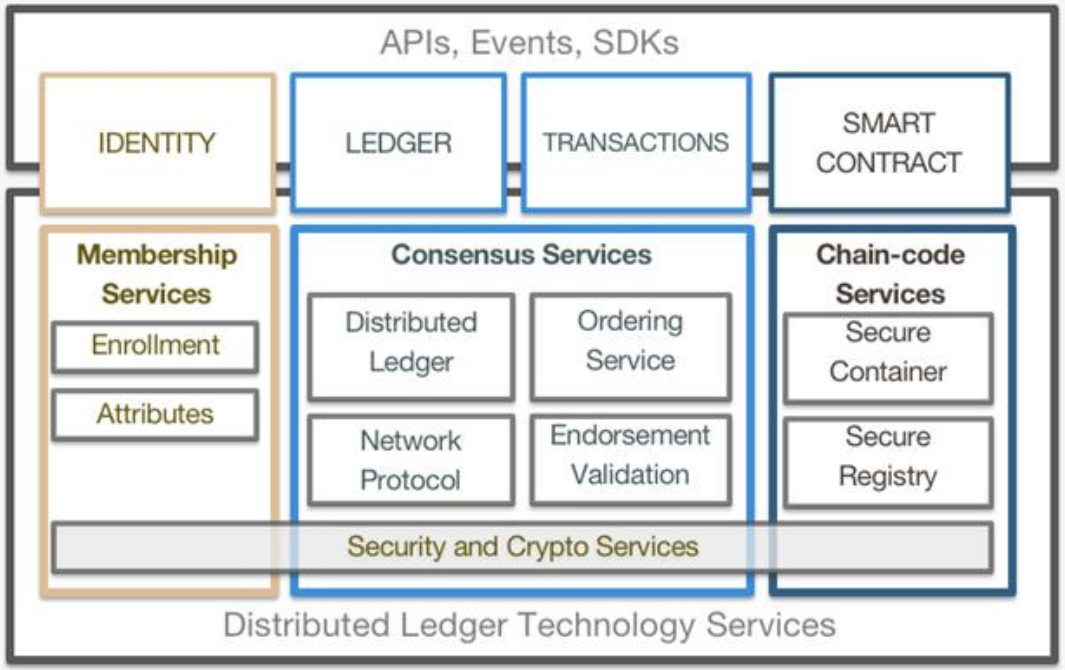
Hyperledger Infrastructure

Hyperledger Fabric Design

- Hyperledger Fabric follows a modular design, and the following are some of the possible components or modules that can be plugged in and implemented.
- Lets discuss some of the main modules

Hyperledger Fabric Design

Reference Architecture



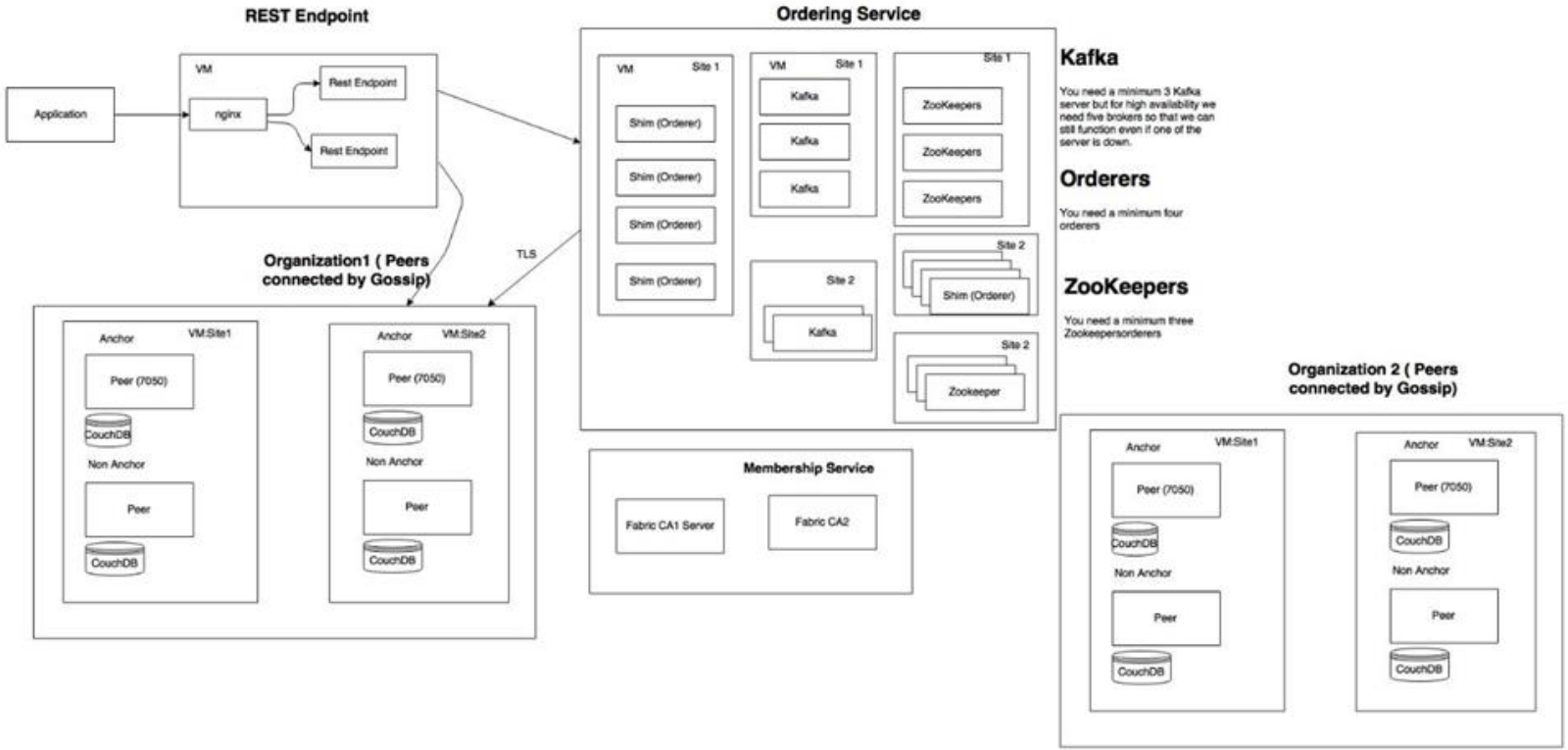
IDENTITY
Pluggable, Membership, Privacy and Auditability of transactions.

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

SMART CONTRACT
"Programmable Ledger", provide ability to run business logic against the blockchain (aka smart contract)

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

Hyperledger Fabric Design



Hyperledger Fabric Design

Membership services

- This module is essentially a permissioning module and acts as a vehicle to establish a root of trust during network creation, but this is also instrumental in ensuring and managing the identity of members.

Hyperledger Fabric Design

Transactions

- A transaction is a request to the blockchain to execute a function on the ledger.
- The function is implemented by a chaincode.

Hyperledger Fabric Design

Smart contract /chaincode services

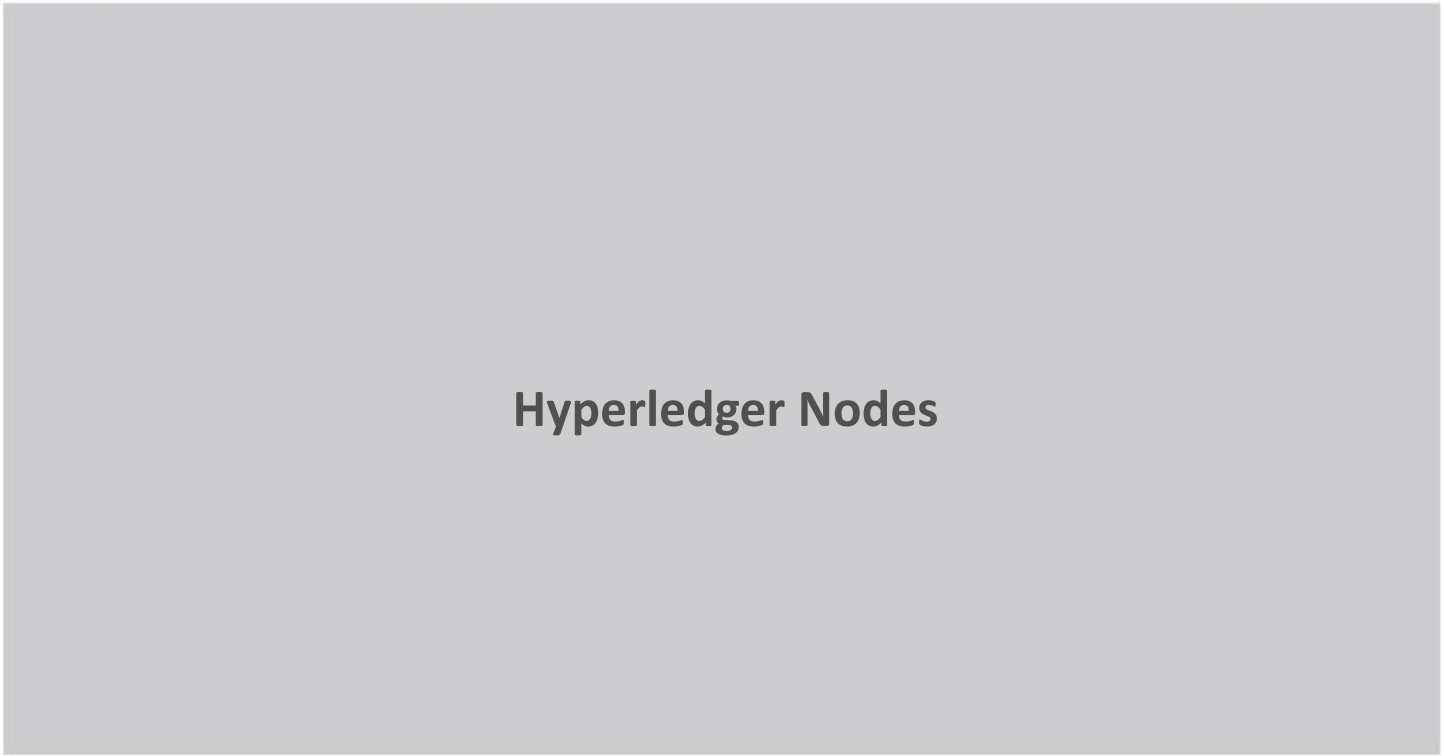
- Chaincode is an application-level code stored on the ledger as a part of a transaction.
- Chaincode runs transactions that may modify the world state.

Hyperledger Fabric Design

Smart contract /chaincode services Continued

- Transaction logic is written as chaincode (Go or JavaScript languages) and executes in secure Docker containers.
- Chaincode is installed on peers, which require access to the asset states to perform reads and writes.

Hyperledger Infrastructure



Hyperledger Nodes

The concept of node is common in all blockchain technologies. The “Node” becomes the communication end point in blockchain technology.

- Nodes connect to other nodes and that is how a blockchain is formed.
- Nodes use a type of peer-to-peer protocol for keeping the distributed ledger in sync across the network.

Hyperledger Nodes

- In Hyperledger, nodes need a valid certificate to be able to communicate to the network and the participants use applications that connect to the network by way of the nodes.
- Remember Hyperledger is a “Permissioned” blockchain.
- Participant’s identity is not the same as the nodes identity.
- When a participant executes or invokes a transaction, their certificate is used for signing that transaction.

Hyperledger Nodes

In Hyperledger, there is the concept of nodes. Basically, all Nodes are NOT equal. There are three distinct types of nodes:

- Client Node: That initiates the transaction
- Peer Nodes: Commits Transaction & keeps the data in sync across the ledger
- Ordered: They are the communication backbones and responsible for the distribution of the transactions

Hyperledger Nodes

Client Node

The client represents the entity that acts on behalf of an end-user. It must connect to a peer for communicating with the blockchain.

The client may connect to any peer of its choice. Clients create and thereby invoke transactions.

Hyperledger Nodes

Peer Node

- They are nodes that maintain the state and copy of a shared ledger.
- Peers are authenticated by certificates issued by MSP. In Hyperledger Fabric, there are three types of peer nodes depending upon the assigned roles:

Peer Node Type	Role
Peer	COMMSTX- Ledger/World State
Endorsing Peer	Endorse/Execute Chaincode
Ordering Peer	Include TX/Comms with Nodes

Hyperledger Nodes

Hyperledger supports two types of transactions.

- Code deploying transaction
- Code invoking transaction

Hyperledger Transactions

Hyperledger Transactions

Ordering Service Nodes (Orderers)

Responsible for consistent ledger state across the network

- Consensus Mechanism & Ensures order of Transactions

• Creates Blocks & Provides atomic delivery/broadcast

• Message Oriented Middleware options for orderer service in Hyperledger:

- SOLO: Single Node, supports multiple channels (Good for Development)

- Kafka: High throughput, scalable & Fault Tolerant

Hyperledger Transactions

Hyperledger Fabric allows users to define policies around the execution of chaincode.

These endorsement policies define which peers need to agree on the results of a transaction before it can be added to the ledger.

Fabric includes a small domain-specific language for specifying endorsement policies.

Transaction Policy

Node A, B, C must all endorse transactions of type XXX

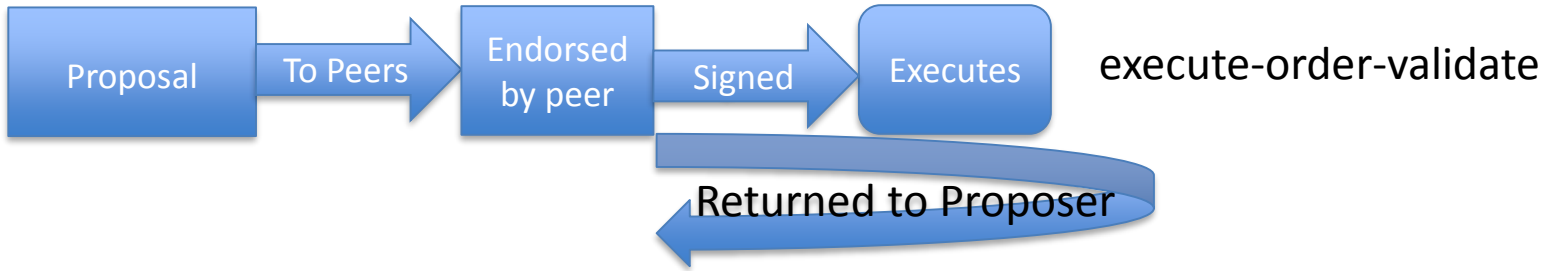
Hyperledger Transactions

- Fabric starts with a transaction proposal. (Triggers Chaincode)
- The transaction proposal is sent to some peers for endorsement.
- An endorsing peer executes the chaincode, which (if it succeeds) yields an actual transaction for the ledger.
- The endorsing peer then signs the transaction and returns it to the proposer.
- This is the Execute step in execute-order-validate.

Hyperledger Transactions

- When the creator of the proposal receives set signatures to satisfy the endorsement policy, it can submit the transaction (Sigs as well)to be added to the ledger.
- This is the Order step.

Transaction Proposal



Hyperledger Transactions

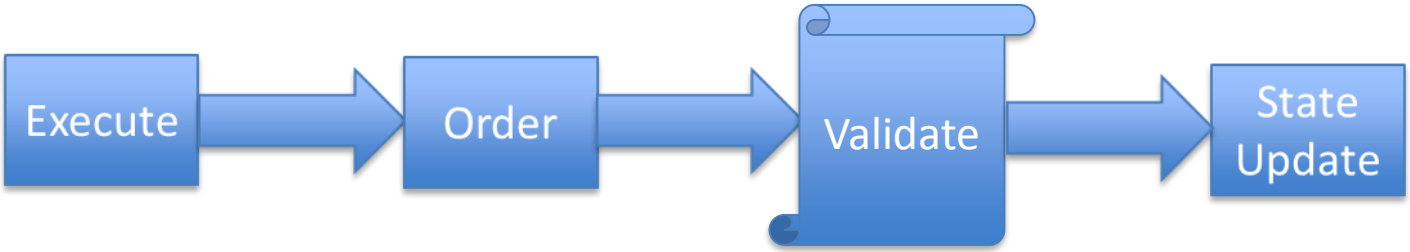
- Code deploying transaction submits, updates or terminates a chaincode.
- The validating nodes protects the authenticity and integrity of the code and its executing environment.
- Hyperledger Fabric uses a new execute-order-validate architecture, which lets transactions execute before the blockchain reaches consensus on their place in the chain
- Prior, blockchains generally used a “Sequential” approach

Hyperledger Transactions

Sequential execution style.

Non-Sequential execution style.

execute-order-validate



Hyperledger Transactions

Code invoking transaction

- Code invoking transaction is an API call to a chaincode function. It is similar to how a URI invokes a servlet in JEE. The displayed function is called upon the instantiation of the chaincode.
- Each chaincode maintains its own state and a function call is made to trigger chaincode state changes

Hyperledger Nodes

Transactions in Hyperledger – 10,000 FT Level

1. When a participant executes or invokes a transaction, their certificate is used for signing that transaction.
2. The network validates if the node's certificate should be trusted. (In the case the nodes certificate is revoked or has expired, the transaction that appeared to be signed by a valid certificate is broadcasted to the network.)
3. The transaction will be rejected because the certificate that the node used was expired or had been revoked.

Hyperledger Transactions

Transactions Per Second

- Blockchains are slow. Really.
- Blockchains are no match for example the Visa Network
- Lets Compare

Hyperledger Transactions

Transactions Per Second

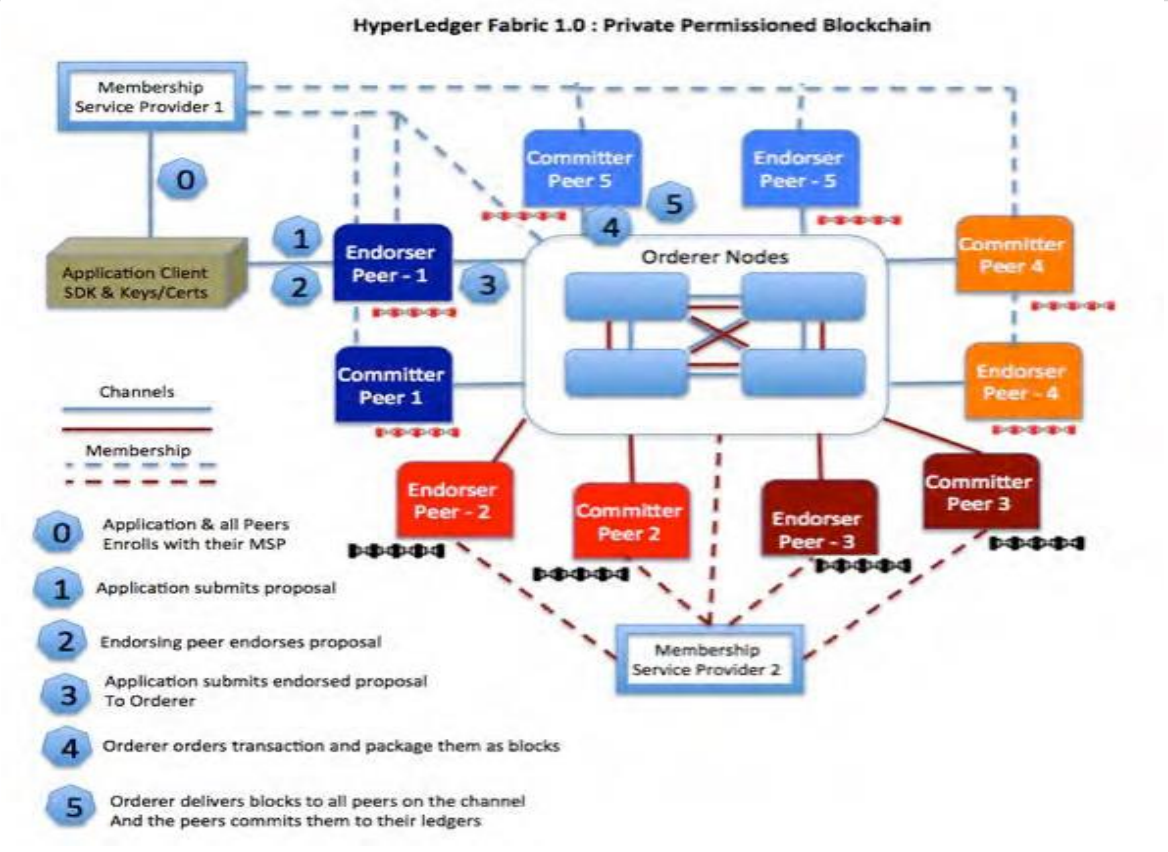
	VISA	BTC	Ripple	Paypal	Hyperledger
TPS	24,000	4	1500	193	3500**
Control	Centralized	Decentralized	Centralized	Centralized	Centralized
Notes	Push	P2P	Hybrid	Hybrid	P2P

**Hyperledger Fabric. As Chris Ferris noted "We haven't published performance figures for Fabric because there isn't a standard benchmark.".

Hyperledger Transaction Flow

Transaction Flow

Hyperledger



Hyperledger Infrastructure

Hyperledger Ledger Basics

Hyperledger Ledger Basics

Ledger Basics

- The ledger is the sequenced, tamper-resistant record of all state transitions. State transitions are a result of chaincode invocations (“transactions”) submitted by participating parties.
- Each transaction results in a set of asset key-value pairs that are committed to the ledger as creates, updates, or deletes.

Hyperledger Ledger Basics

Fabric Ledger has two parts:

- State data: Representation of current state of the assets. Asset state data can be changed upon changes to the state of the data.
- Transaction Logs: Record of all the transactions (in the order they are received) which modified the state data, and once the data is written it is immutable and cannot be modified.

Hyperledger Ledger Basics

What DB is used?

The ledger system in Hyperledger fabric uses levelDB. By definition, LevelDB allows concurrent writers to safely insert data into the database by providing internal synchronization

State database options include LevelDB and CouchDB. LevelDB is the default key-value state database embedded in the peer process.

CouchDB is an optional alternative external state database. (Binary data)

Hyperledger Ledger Basics

What DB is used for Where and Why?

	Transaction Logs	State Date (World)
Type	Immutable	Mutable
Operations	Create, Retrieve	ALL-CRUD
DC	levelDB	levelDB/CouchDB
Attitude	Embedded in peers	Key-Value Paired(JSON, Binary)
Query	Simple	Couch DB for Complex

Hyperledger Infrastructure

Hyperledger Fabric Playground

- Hyperledger Composer Playground is a tool which provides an environment that quickly models and tests a blockchain network.
- The composer has a simple Graphical UI to edit and test the business blockchain network.

https://composer-playground.mybluemix.net/login

Hyperledger Fabric Playground

Hyperledger Composer(Fabric Composer):

- Fabric Composer is a newer open-source application development framework, which simplifies the creation of Hyperledger Fabric blockchain applications, thus reducing the time and complexity of development.
- The tool aims at helping users to create blockchain applications based on Hyperledger Fabric without needing to know the low-level (Go Programming) details involved in blockchain networks

Hyperledger Fabric Playground

- If you want to build your blockchain application directly on Hyperledger Fabric you have to write your Chaincode in ***GO or Java Programming Language*** which is comparatively different than JavaScript because its composer is quite easy to code smart contract using Model file (.cto) and angular JavaScript.
- Hyperledger Composer primarily uses JavaScript for ***chaincode*** development.

Hyperledger Composer

Hyperledger Composer has following main components

- Business Network Archive: Capturing the core data in a business network including the business model, transaction logic, and access controls, the Business Network Archive packages these elements up and deploys them to a runtime.
- Stored as ".bna" files.

Hyperledger Composer

Hyperledger Composer has following main components

- Composer Playground: This web-based tool allows developers to learn Hyperledger Composer, model out their business network (domain), test that network, and deploy that network to a live instance of a blockchain network
- CRUD Capacity
- Templates

Hyperledger Composer

- Composer uses what's called connection profiles to define the system to connect to.
- A connection profile is a JSON document that acts as part of a business network card.
- The connection profile is most often provided by creators of the system they refer to

Hyperledger Composer

- You can use queries to get data about the state of the blockchain. Queries are defined within a business network, and can include variable parameters. Queries are sent using the Composer API.
- Events in Composer are defined in the business network definition in the same way as participants or assets. Events are emitted by the transaction processor function once it has been defined. An event indicates to external systems that something important has occurred on the ledger. Applications subscribe to emitted events using the composer-client API

Hyperledger Composer

- Developers of the business network can create a set of access controls. Access controls are rules that determine which assets participants have access to in the business network and the conditions in which they can access them.
- A historian is a specialized type of registry that records successful transactions conducted on the business network

Hyperledger Composer

Hyperledger Composer consists of the following high-level components:

- Execution Runtimes
- JavaScript SDK
- Command Line Interface
- REST Server
- LoopBack Connector
- Playground Web User Interface
- Yeoman code generator
- VSCode and Atom editor plugins

Hyperledger Composer

Hyperledger Composer modeling language, an object-oriented modeling language that defines the domain model for a business network definition.

- The modeling language is saved as a .cto file.

The CTO file contains:

- A single namespace, in which all resource declarations are implicitly.
- A set of resource definitions that includes assets, transactions, participants, and events
- The option to import resources from other namespaces

Hyperledger Composer

Namespace

There is a system namespace which contains base definitions of asset, event, partcipant,and transactions. These base definitions are abstract types that are implicitly extended by all new assets, events, participants, and transactions

- Events and transactions in the system namespace are defined by an eventID or transactionID and a timestamp.
- The system namespace also includes definitions of

registries, historian records, identities, and system transactions

Hyperledger Composer

In Composer, resources are:

- Assets, participants, transactions, and events
- Enumerated types
- Concepts

Hyperledger Composer

Resource definitions all have the following inherent properties. A namespace defined by the namespace of its parent file

- A name and an identifying field
- An optional super-type that the resource definition extends
- An optional “Abstract” declaration to indicate that this type cannot be created.
- A set of named properties defined. Properties and data are owned by each resource
- A set of relationships to other Composer types that are not allowed by the resource but may be referenced from the resource.

Hyperledger Composer

Here is an example of Vehicle as a super-type, and a Car being considered an asset with a set of parts:

asset Car extends Vehicle {

o String model

--> Part[] Parts

Hyperledger Composer

In composer, concepts are abstract classes that are not considered an asset, participant, or transaction

- Concepts do not have an identified by field because concepts cannot directly abstract concept Address {

o String street

o String city default ="New York"

o String country default = "US"

o Integer[] counts optional

}

concept CanadaAddress extends Address {

o String zipcode

}be stored in registries or referenced in relationships

Hyperledger Composer

Other progamming areas

- Arrays
- Primative
- Field Validators
- Relationships
- Imports
- Decorators

Review Questions

The primary purpose of Hyperledger Composer is:

a) Allowing blockchain applications to run on computers with slow processing power

b) Accelerate the time to develop a blockchain application

c) Make it easy to integrate blockchain technology into legacy systems

d) Both B and C

Review Questions

The connection profile:

a) Defines the participants that can connect to each other

b) Defines the system to connect to

c) Defines the systems to avoid

d) Defines connections between assets in a network

Review Questions

Composer modeling language files are saved with which extension?

a) .EXE

b) .CTO

c) .CML

d) .CMP

Review Questions

A historian is:

a) A specialized business network that only allows certain participants

b) A specialized type of registry that records errors on the business network

c) A specialized type of registry that records all successful transactions

d) A specialized type of registry that records all participants on the network

Review Questions

Hyperledger fabric business network is divided into three categories.

a. Sawtooth, Fabric, and Indy

b. Composer, Fabric, and Chaincode

c. Blockchain, Chaincode, and Membership

d. Blockchain, Registration, Identity

Review Questions

What application is used by Hyperledger Fabric to communicate with the network?

a. JSON

b. Binary

c. SDK

d. RPC API

Review Questions

What application is used by Hyperledger Fabric to communicate with the network?

a. JSON

b. Binary

c. SDK

d. RPC API

Hyperledger Infrastructure

Hyperledger Fabric Playground

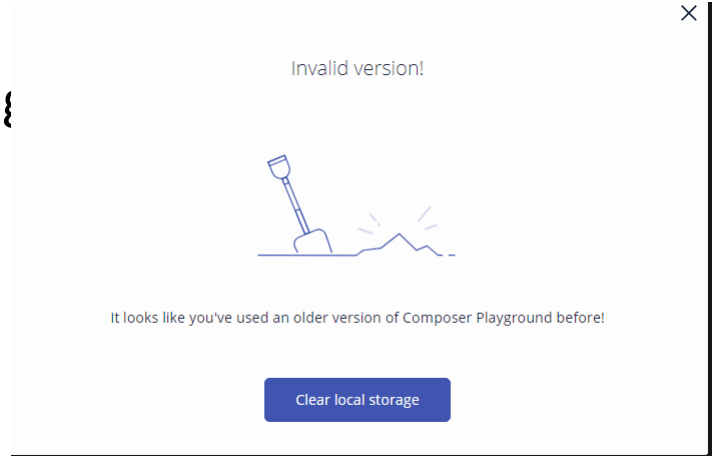
Hyperledger Fabric Playground

- Playground makes the highly complex blockchain network easy for running blockchain testing.
- There is an online and offline version of Playground
- Online playground runs the business network in browser memory
- Local playground is deployed in Hyperledger Fabric instances.

HYPERLEDGER
COMPOSER

Hyperledger Fabric Playground

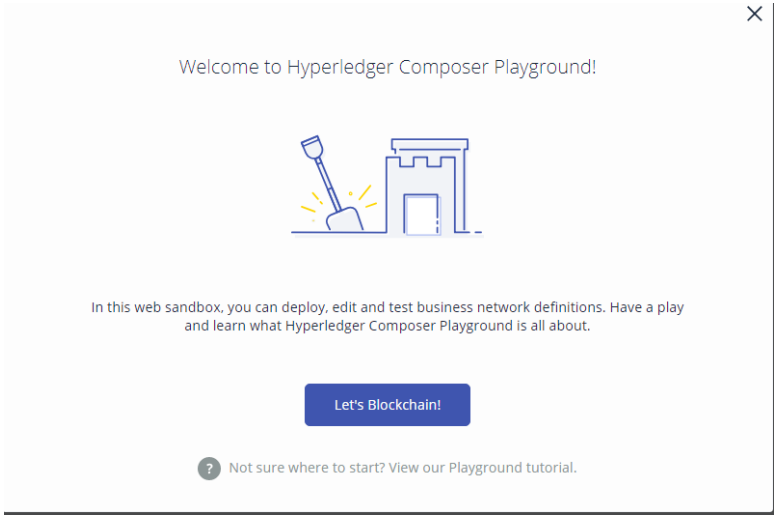
- Playground is located here
- It will let you know if your running a supported version



Hyperledger Fabric Playground

- Playground is a “sandbox”

- Use it to develop, test, validate, etc.
- Deploy a new Business Network
- Not a live blockchain



Hyperledger Fabric Playground

- Deploy a “Model”

```
/**
 * My Pearson Student Training network
 */
namespace org.example.mynetwork
asset Commodity identified by tradingSymbol {
   o String tradingSymbol
   o String description
   o String mainExchange
   o Double quantity
   --> Trader owner
}
participant Pearson identified by tradeId {
   o String tradeId
   o String firstName
   o String lastName
}
transaction Trade {
   --> Commodity commodity
   --> Trader newOwner
}
```

Hyperledger Infrastructure

REST

- Deploying a “Model” A REST (Representational State Transfer) Server is often used to proxy requests to Hyperledger chaincode.
- This provides a well defined process for accessing blockchain services.

REST

- The REST server uses a business network card specified during startup to connect to and discover the assets, participants, and transactions within a deployed business network.
- This information visibility is required in order to generate the REST API. This business network card is known as the discovery business network card.
- By default, the discovery business network card is also used to handle all requests to the REST API

REST

- Chaincode Services uses Docker to host (deploy) the chaincode without relying on any virtual machine or computer language.
- Docker provides a secured, lightweight method to sandbox chaincode execution.
- The environment is a "locked down" and secured container, along with a set of signed base images containing secure OS and chaincode language, runtime and SDK images for Golang
- Additional programming languages can be enabled, if required

REST

- Secure Registry Services enables Secured Docker Registry of base Hyperledger images and custom images containing chaincodes.
- Since assets in Hyperledger Fabric are represented in JSON or Binary, hyperledger includes the REST and JSON RPC APIs, events, and an SDK for applications to communicate with the network.
- This makes it easy for your developers to interact with the services

REST

Which of the following is NOT a high-level component of Composer?

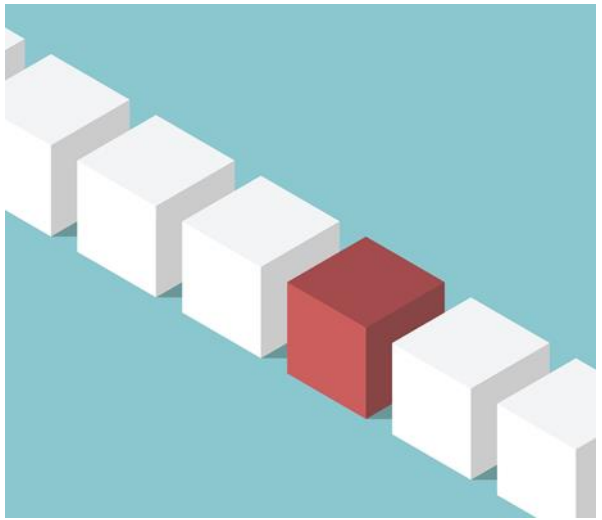
- a) Command Line Interface
- b) REST Server
- c) Playground Web User Interface
- d) C++ SDK

Hyperledger Infrastructure

Hyperledger Terminology

Hyperledger Terminology

- **Block** An ordered set of transactions that is cryptographically linked to the preceding block(s) on a channel.
- **Chain** The ledger's chain is a transaction log structured as hash-linked blocks of transactions.



Hyperledger Terminology

- **Chaincode** - A smart contract is code – invoked by a client application external to the blockchain network – that manages access and modifications to a set of key-value pairs in the World State
- Chaincode services are secured and lightweight.
- The environment is a “locked down” and is a secured container with a set of signed base images which contains secure OS and Chaincode language, runtime and SDK images for Golang, Java, and Node.js.

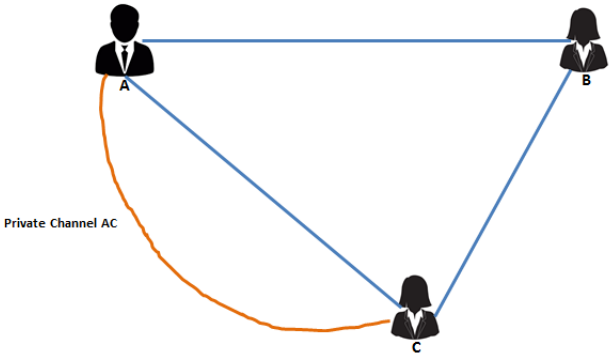
Hyperledger Terminology

As with every chaincode, it implements the Chaincode interface in particular, Init and Invoke functions.

- Init - is called during Instantiate transaction after the chaincode container has been established for the first time, allowing the chaincode to initialize its internal data.
- Invoke - is called to update or query the ledger in a proposal transaction. Updated state variables are not committed to the ledger until the transaction is committed

Hyperledger Terminology

- A channel is a private blockchain overlay which allows for data isolation and confidentiality.
- A channel-specific ledger is shared across the peers in the channel, and transacting parties must be properly authenticated to a channel in order to interact with it.



Hyperledger Terminology

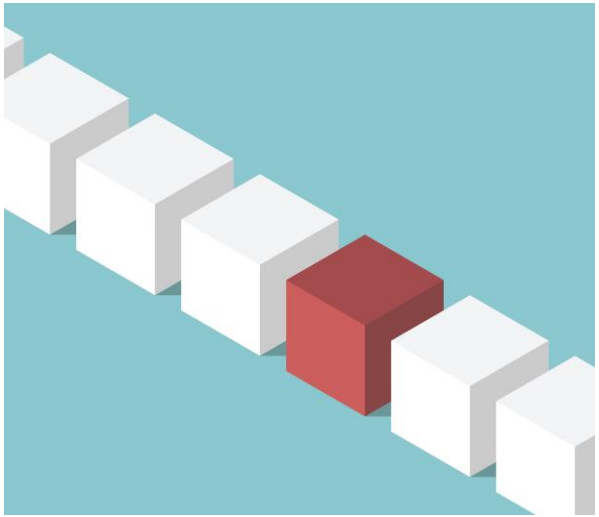
- Consensus is a broader term overarching the entire transactional flow, which serves to generate an agreement on the order and to confirm the correctness of the set of transactions constituting a block.

Hyperledger Terminology

- Endorsement refers to the process where specific peer nodes execute a chaincode transaction and return a proposal response to the client application.

Hyperledger Terminology

- Genesis Block is The configuration block that initializes the ordering service, or serves as the first block on a chain.
- The initializer or start of the blockchain



Hyperledger Terminology

- Gossip Protocol The gossip data dissemination protocol performs three functions
- Manages peer discovery and channel membership
- Disseminates ledger data across all peers on the channel
- Syncs ledger state across all peers on the channel.

Hyperledger Terminology

- State Database data is stored in a state database for efficient reads and queries from chaincode.
- Supported databases include levelDB and CouchDB.

Hyperledger Terminology

- Membership services provide identity, privacy, and confidentiality to the network.
- Reputation Manager enables auditors to view transactions pertaining to a participant, providing that each auditor has been granted proper access authority, based on the role of the participants.
- Blockchain services manages the distributed ledger through a peer to peer protocol that is built on HTTP/2.

Hyperledger Terminology

- A **Participant** is an actor in a business network. A participant might represent an individual or an organization.
- A participant can create assets and share assets with other participants.
- A participant can interact with assets by submitting transactions
- A participant has an identity set that can be validated to prove the identity of that participant.

Hyperledger Terminology

- World State is also known as the “current state”, the world state is a component of the HyperLedger Fabric Ledger
- The world state represents the latest values for all keys included in the chain transaction log.

Hyperledger Infrastructure

Membership

Membership Service Provider

- “Abstract component of the system that provides credentials to the clients, and the peers to participate in the Hyperledger Fabric network”
- MSP implementation is based on the PKI (Public Key Infrastructure).
- Service it Provides: Authorization Service
- Role based

Membership

- As a platform for permissioned blockchain networks, Hyperledger Fabric includes a modular Certificate Authority (CA) component for managing the network identities of all member organizations and their users.
- The requirement for a permissioned identity for every user enables ACL-based control over network activity and guarantees that every transaction is ultimately traceable to a registered user.

Membership

- As The CA (Fabric CA by default) issues a root certificate (rootCert) to each member (organization or individual) that is authorized to join the network.
- The CA also issues an enrollment certificate (eCert) to each member component, server-side applications and occasionally end users.
- Each enrolled user is granted an allocation of transaction certificates (tCerts).
- Each Cert authorizes one network transaction.

X509

- An X.509 certificate is any certificate under the X.509 specification standard for public key infrastructure and Privilege Management Infrastructure (PMI).

The X.509 provides standardized formats for:

- Attribute certificates
- Public key certificates
- Certificate revocation lists
- Certification validation algorithms

X509

- These certificates are used for identity validation and for transmission of encrypted data that only the owner (person, organization or software) of a specific certificate is able to decrypt and read.
- X.509 certificates act as secure identifiers, digital passports which contain information about the owner.
- The certificate is tied to a public key value which is associated with the identity contained in the certificate.

Hashing

Block

- A block contains data of the transaction, hash of the block and hash of the previous block.
- The structure of the block is:

Block 1
Data
Previous Hash
Hash Hash

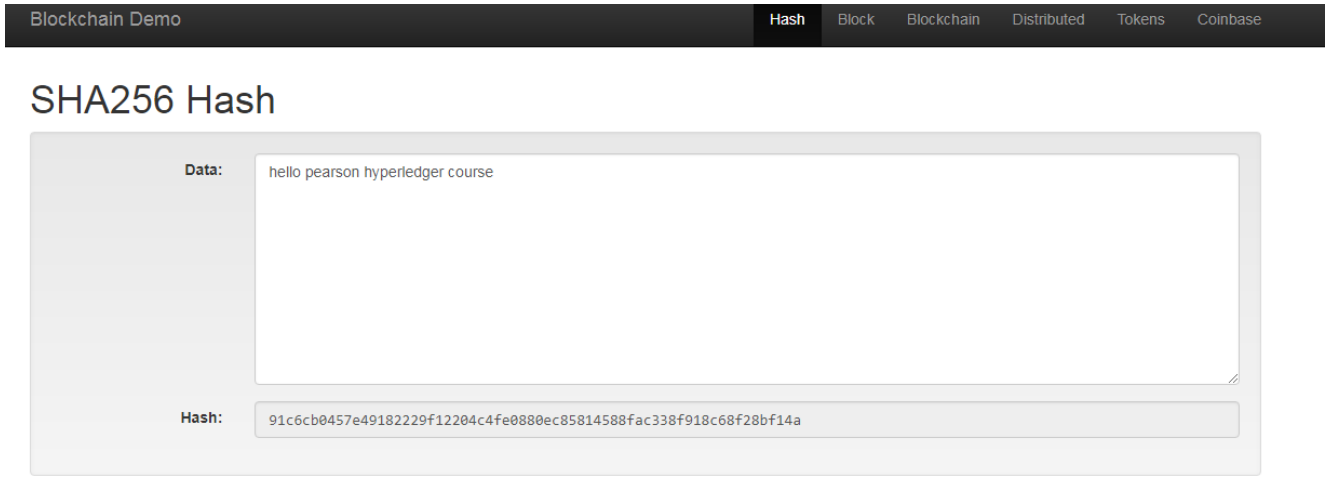
Hashing

Hashing means taking an input string of any length and giving out an output of a fixed length.

- In the context of cryptocurrencies like Bitcoin, the transactions are taken as an input and run through a hashing algorithm which gives an output of a fixed length
- No matter how big or small your input is, the output will always have a fixed 256-bits length.
- Even if slight changes are made to the input the changes get reflected in the hash.
- Blockchain hash functions makes it immutable.

Hashing

https://anders.com/blockchain/hash.html



Review Questions

An X.509 certificate is used for _____

a. certification of transaction consensus

b. validating node performance

c. the issuing of private keys

d. identity validation

Review Questions

Blockchain's use of cryptographic hashing provides for _____

a. the maintaining of data integrity

b. making data blocks tamper proof

c. network security to work in unison

d. All of the above

Review Questions

The hashing function requires that _____

a. all recorded data in each block be the same

b. the network administrator agrees with the hash value

c. hash of the previous block is the same as in the present block

d. none of the above

Hyperledger Infrastructure

Blockchain As a Service (BAaS)

Blockchain as a Service

Blockchain as a Service (BaaS)

- Is generally deployed on a Cloud Computing solution.
- Develop, test, and deploy secure blockchain apps
- Nodes are setup by provider
- Proof of Concepts (POC)
- No need for CAPEX or OPEX for infrastructure

Blockchain as a Service

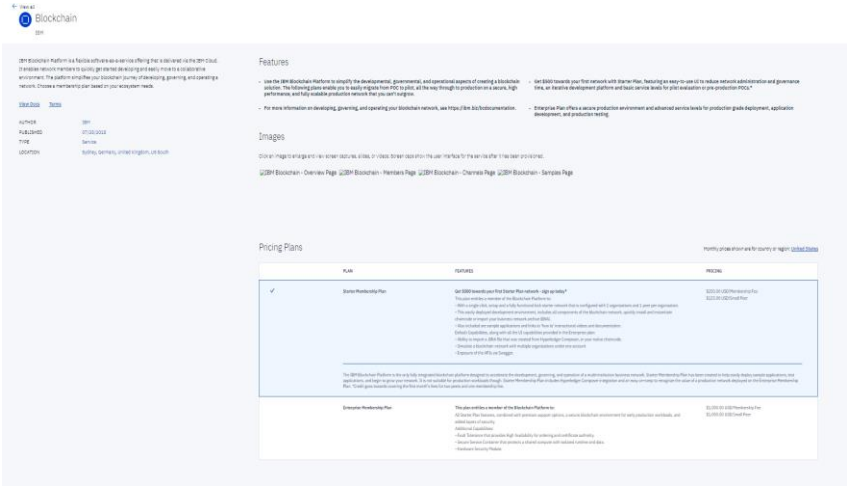
BaaS Providers:

- MS Azure
- AWS
- IBM
- Stratis
- HPE
- Oracle

Blockchain as a Service

IBM Bluemix (IBM Cloud)

- Two service models for Hyperledger Fabric
- Basic and Enterprise
- Basic is an affordable great service for POCs, Betas and training



Hyperledger Infrastructure

Hyperledger Development

Client App

- Model file, Transaction functions (chaincode), access control file and the static query file make the Business Network Archive (Package) for the Hyperledger Fabric
- The native query language can filter results returned using criteria and can be invoked in transactions to perform operations, such as updating or removing assets on result sets.
- Queries are defined in a query file (.qry) in the parent directory of the business network definition.

Hyperledger Development

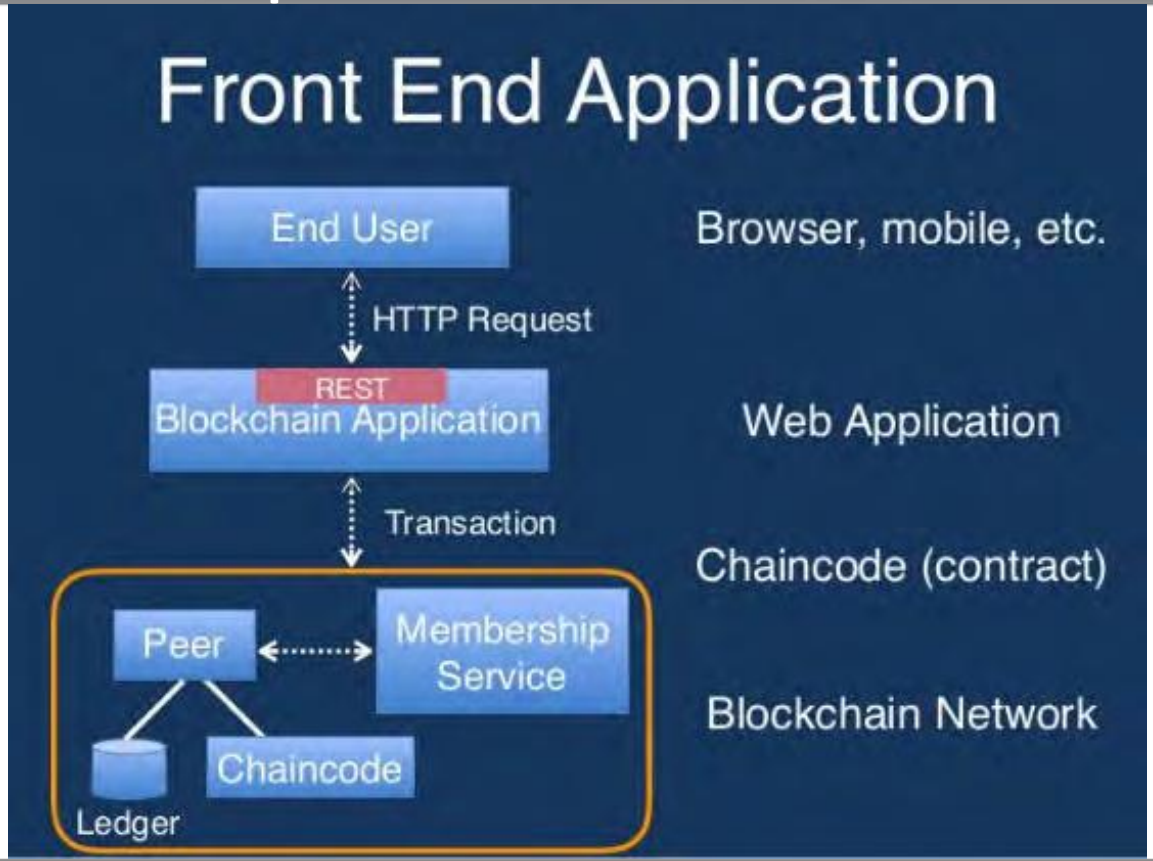
Client App

- Event creates notifications of significant operations on the Blockchain (e.g. a new block), as well as notifications related to a milestone achieved while processing a smart contract/chaincode.
- The client app can subscribe to this event and take appropriate business actions.

Hyperledger Development

Front End Considerations:

- **Rest server must be secured – HTTPS**
- **Should use authentication – [Passport]**
- **Should use multi-user mode for rest api**



Hyperledger Development

- When writing chaincode, you will want to make sure that you have the Go programming language installed and setup with the correct configuration.
- You will want to make sure that a directory is created for your chaincode application as a child

```
// Extract the function and args from the
transaction proposal
fn, args := stub.GetFunctionAndParameters()
var result string
var err error
if fn == "set" {
result, err = set(stub, args)
} else {
result, err = get(stub, args)
}
if err != nil {
return shim.Error(err.Error())
}
// Return the result as success payload
return shim.Success([]byte(result))
}
```

Hyperledger Development

- We can then implement the init function. Init is called during chaincode instantiation, and it will initialize any data.

Chaincode:

```
// Init is called during chaincode instantiation to initialize any data.
func (t *SimpleAsset) Init(stub shim.ChaincodeStubInterface) peer.Response
{
}
```

Hyperledger Development

- Now our example chaincode application implements two functions that can be invoked via the invoke function.

```
// Set stores the asset (both key and value) on the ledger. If the key exists,
// it will override the value with the new one
func set(stub shim.ChaincodeStubInterface, args []string) (string, error) {
if len(args) != 2 {
return "", fmt.Errorf("Incorrect arguments. Expecting a key and a value")
}
err := stub.PutState(args[0], []byte(args[1]))
if err != nil {
return "", fmt.Errorf("Failed to set asset: %s", args[0])
}
return args[1], nil
}
```

Hyperledger Development

- For more development resources check out the attachments

```
// Set stores the asset (both key and value) on the ledger. If the key exists,
// it will override the value with the new one
func set(stub shim.ChaincodeStubInterface, args []string) (string, error) {
if len(args) != 2 {
return "", fmt.Errorf("Incorrect arguments. Expecting a key and a value")
}
err := stub.PutState(args[0], []byte(args[1]))
if err != nil {
return "", fmt.Errorf("Failed to set asset: %s", args[0])
}
return args[1], nil
}
```

Review Questions

- The chaincode's interface inplements the following functions _____
- a. open and close
- b. query and update
- c. init. and run
- d. invoke and init

Review Questions

The “init” method is called when:

- a) there is an error in the code
- b) a chaincode receives an invoke function
- c) a chaincode receives an “instantiate” or “upgrade” transaction
- d) an asset is created

Review Questions

The “invoke” method is called in response to:

a) receiving an transaction to process transaction proposals

b) receiving an asset

c) sending a transaction

d) receiving an instantiate transaction

Review Questions

Every chaincode program must implement the:

a) Chaincode panel

b) Chaincode interface

c) Chaincode policy

d) Chaincode parameters

Hyperledger Infrastructure

Blockchain Certification -Hyperledger

CBSA Overview

- The Certified Blockchain Developer Hyperledger (CBDH) exam is an elite way to demonstrate your knowledge and skills in the Blockchain arena.
- You will become a member of a community of Blockchain leaders.

CBSA Overview

- The Certified Blockchain Developer Hyperledger (CBDH) exam is a professionally delivered exam which is proctored thru Pearson.
- Passing this certification will distinguish you as one that is knowledgeable from developer perspective

https://blockchaintrainingalliance.com/products/cbdh