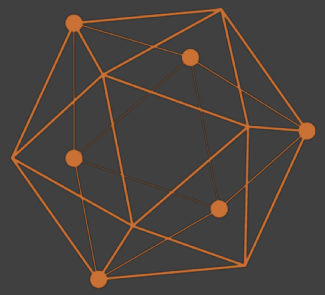


HYPERLEDGER PROJECT

Meetup #3 – Interfaces Use case & Architecture

dcentrum Community

Connect | Collaborate | Create

Community Friends : nagarro

DCentrum's 3 Month Blockchain Series: Hyperledger

dcentrum
Connect | Collaborate | Create

Join to experience perfect way to learn and solve real world problems using Hyperledger Fabric with the group of highly motivated Blockchain enthusiasts

Nov 2018 - Feb 2019

Hyderabad

Meeting Frequency
BiWeekly

Total Series Duration
3 Months

First and Second Meeting:

- ✓ Foundation
- ✓ Level Setting
- ✓ Concepts

During initial two meetings, we all will get to know each other and understand the basics of Blockchain to build solid foundation.

Third Meeting:

- ✓ Real World Use Case Selection
- ✓ Architecture Decisions

Will pick a real world use-case by looking at the domain expertise and interest of majority of participants. Group will also work to architect the solution to during this period.

Fourth and Fifth Meeting:

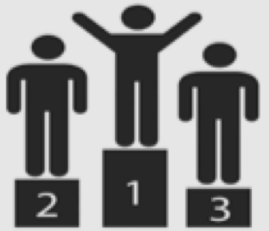
- ✓ Building Solution with collaboration

During this period of 45 days we all will build end-to-end robust enterprise level solution on chosen use-case

Sixth Meeting:

- ✓ Use Case Demo
- ✓ Hackathon

Will host and participate in an open Hackathon, which will be judged by esteemed jury of who's who of Blockchain World



Format for each meeting:

- *Welcome and General Discussion: 30 mins*
- *Presentation from members: 60 mins*
- *Shared Learning and Discussion: 60 mins*
- *News Briefs: 10 mins*
- *Open discussions on general topics: 20 mins*

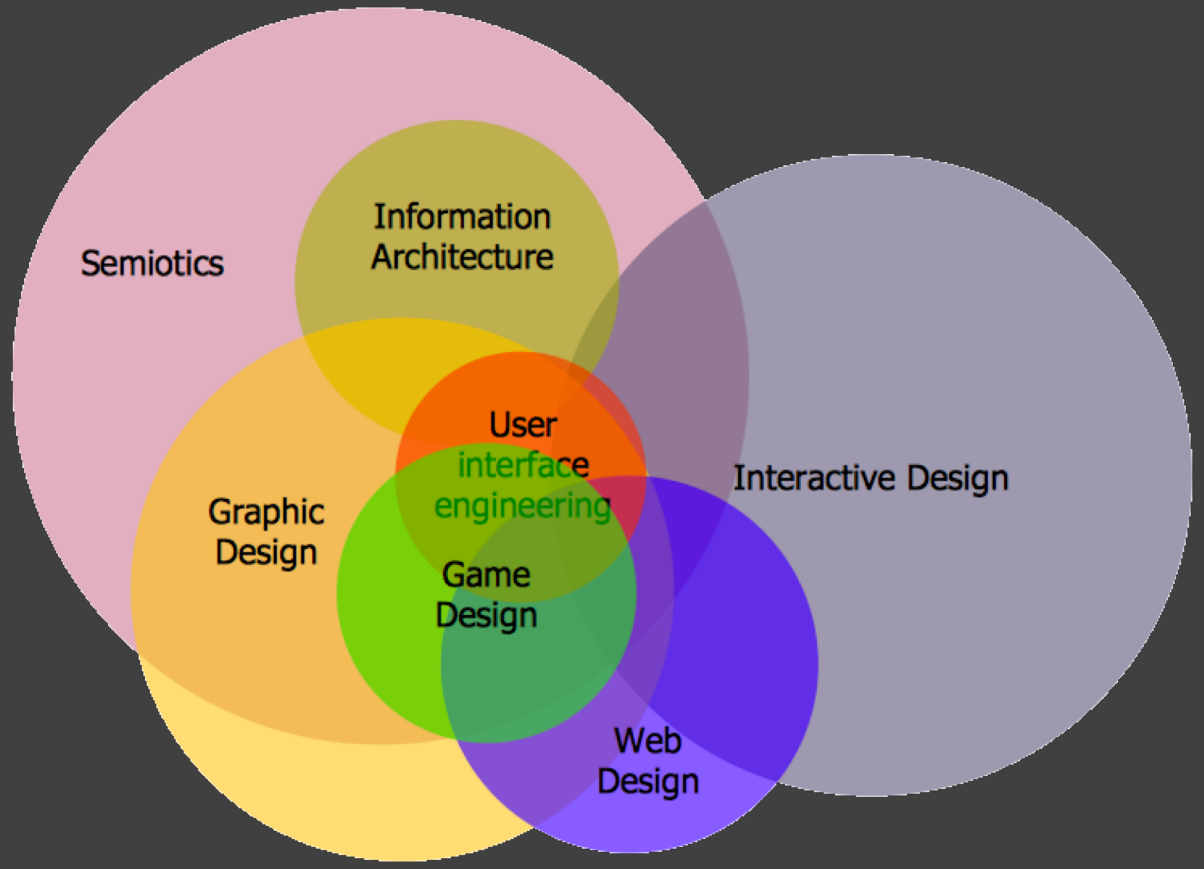
Agenda

The theme will be 20 mins presentations followed by 30 min discussions based on below 3 topics:

- ❑How do you connect using a UI/REST services Interface to Hyperledger Fabric
- ❑How do you pick use cases and parameters to finalize a use case? How to assess Business value.
- ❑Hyperledger Design Considerations
  - ❑Members of the network
  - ❑Planning for performance
  - ❑Development Best Practices
  - ❑Testing Best Practices
  - ❑Integration with CI and CD
  - ❑High level architecture configurations.
  - ❑Sample Architecture of applications involving HL Fabric

Meetup#3 – Session#1

Building REST Service Interfaces & User Interfaces for Fabcar Demo or any Blockchain network



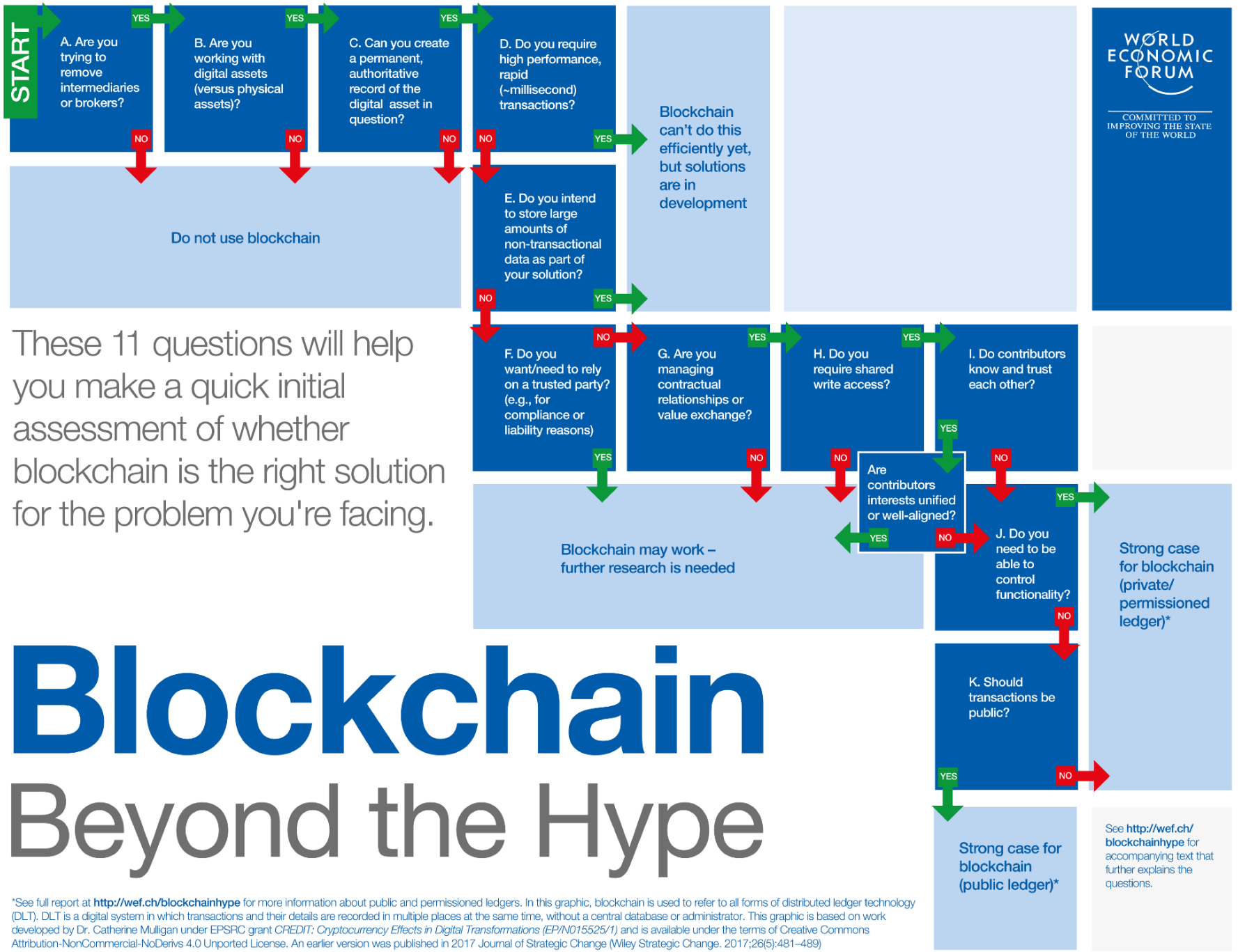
- By Sreenivas Chinni

Meetup#3 – Session#2

Building Use Cases and Framework to analyze & evaluate them

- By Raju Cherukuri (Rishi)

Use Cases from:
Sajin Padiyath
Suman Papanaboina
Pradeep Pentakota



These 11 questions will help you make a quick initial assessment of whether blockchain is the right solution for the problem you're facing.

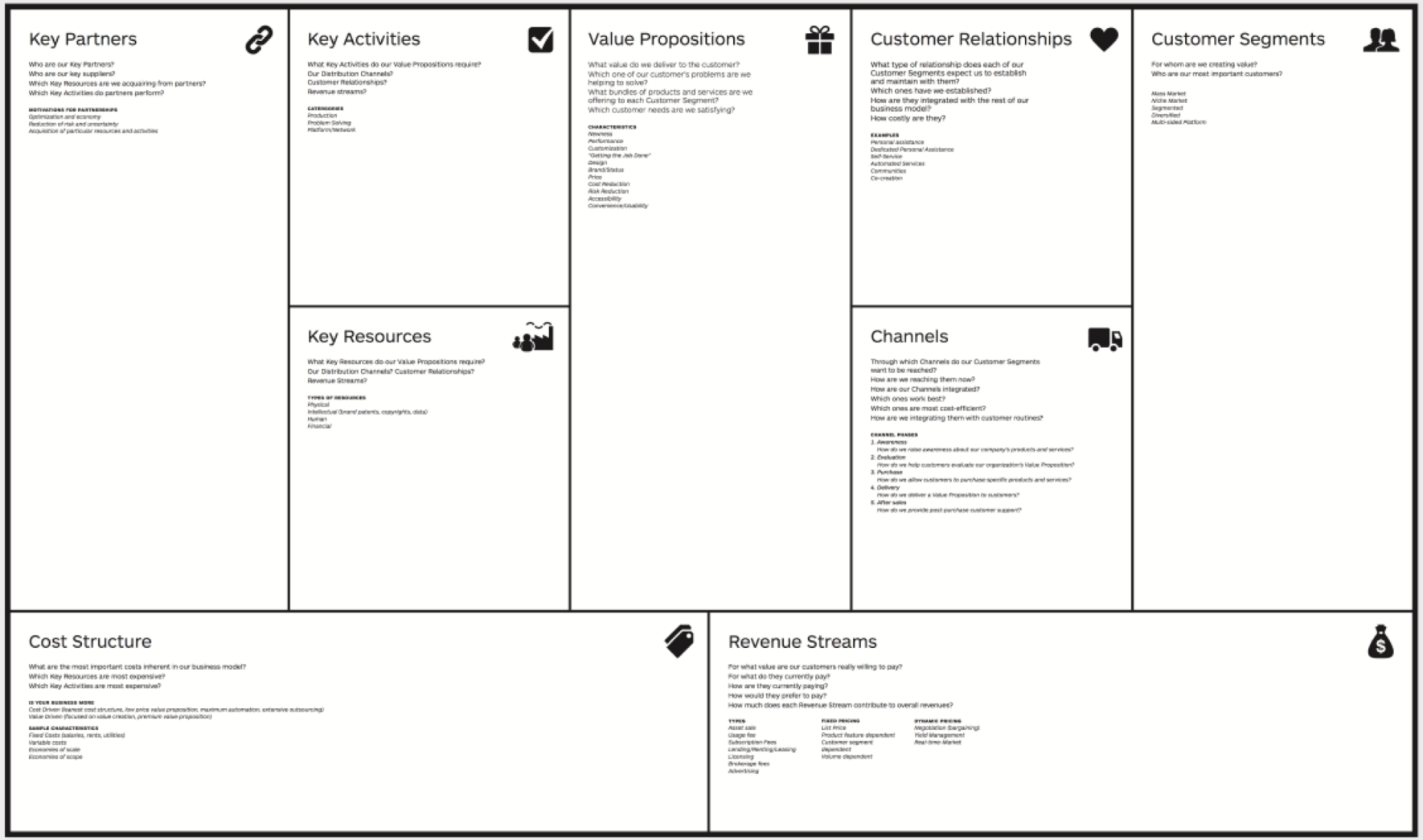
Blockchain
Beyond the Hype

See **http://wef.ch/blockchainhype** for accompanying text that further explains the questions.

*See full report at **http://wef.ch/blockchainhype** for more information about public and permissioned ledgers. In this graphic, blockchain is used to refer to all forms of distributed ledger technology (DLT). DLT is a digital system in which transactions and their details are recorded in multiple places at the same time, without a central database or administrator. This graphic is based on work developed by Dr. Catherine Mulligan under EPSRC grant *CREDIT: Cryptocurrency Effects in Digital Transformations (EP/N015525/1)* and is available under the terms of Creative Commons Attribution-NonCommercial-NoDerivs 4.0 Unported License. An earlier version was published in 2017 Journal of Strategic Change (Wiley Strategic Change. 2017;26(5):481–489)

The Business Model Canvas

Designed for:	Designed by:	Date:	Version:



Strategyzer
strategyzer.com

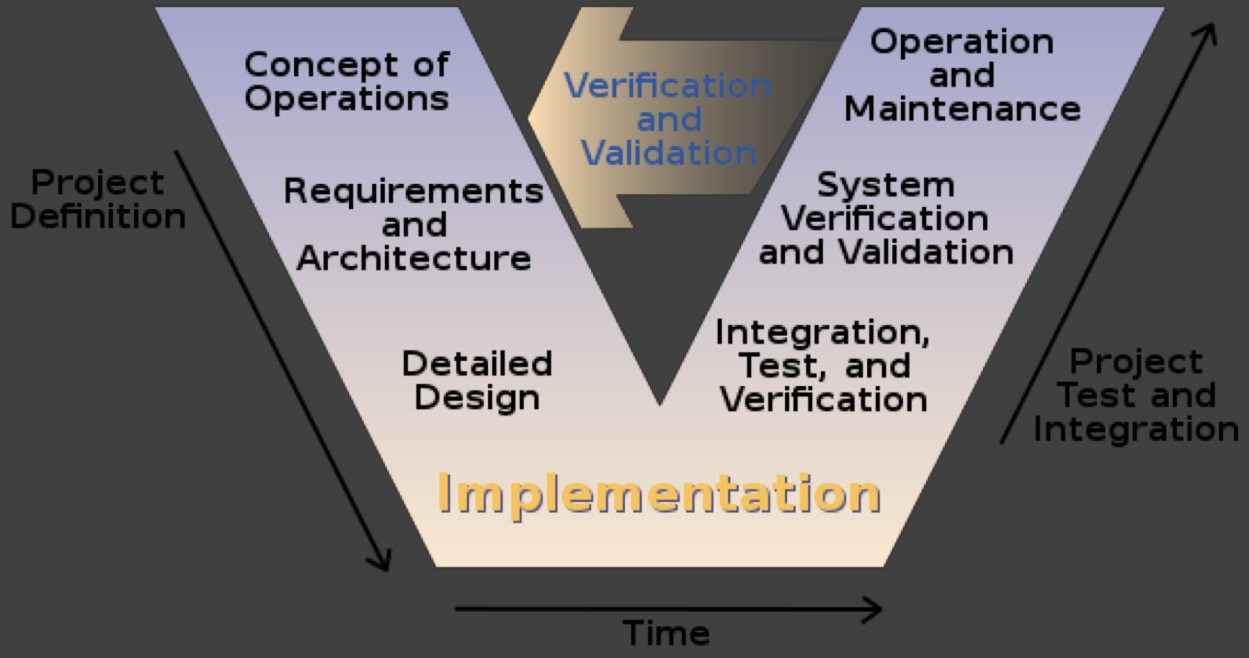
Osterwalder explaining the Business Model Canvas in 6 Minutes
https://www.youtube.com/watch?v=RpFiL-1TVLw

Meetup#3 – Session#3

Hyperledger Fabric
&
Blockchain Design Considerations

- By Suman Papanaboina

Questions Moderated by:
Raju Cherukuri (Rishi)



Agenda

Let us discuss about Hyperledger design considerations

- ❑When do we need a Blockchain
- ❑When do we need to choose Hyperledger
- ❑What composes a Blockchain
- ❑Blockchain Application Layers
- ❑Roles in a Blockchain Project
- ❑Setting up a Blockchain Network
- ❑Example Use Case
- ❑Blockchain Design Considerations
- ❑Importance of Off Chain storage
- ❑Integrating with Enterprise Systems

When we need blockchain ?

We need blockchain When want to establish **Trust** between multiple **Third Parties** using fully decentralized network by eliminating middle man.

The blockchain’s core value is enabling its participants to provably and permanently agree on exactly what data was entered, when and by whom, without relying on a trusted intermediary

Tamper Evidence

Decentrilization

Transparency

Why Hyperledger?

- Enterprise Concerns in Blockchain adoption

  - Data Privacy and confidentiality
  - Efficiency and speed of transactions
  - Modular design approach
  - Hybrid approach
  - B2B focus
  - Mining and legal restrictions on location of nodes
  - No need of incentives/crypto currency

  ->Hyperledger solves all these concerns

Why Hyperledger

- Fabric/Sawtooth are permissioned ledgers and data stored on blockchain is accessible only to authenticated members
- Fabric provides concept of Channels and Private Data to further make data confidential
- Both Fabric and Sawtooth are modular and provides pluggable consensus mechanisms
- Both Fabric and Sawtooth are scalable
- No need of mining in Fabric or Sawtooth
- No need of crypto currency
- Both Fabric/Sawtooth can co exist with other apps in enterprise
- Both Fabric/Sawtooth allows secure data sharing in B2B scenarios

Blockchain

Blockchain (Immutable Ledger + world state)

=

Distributed Computing

+

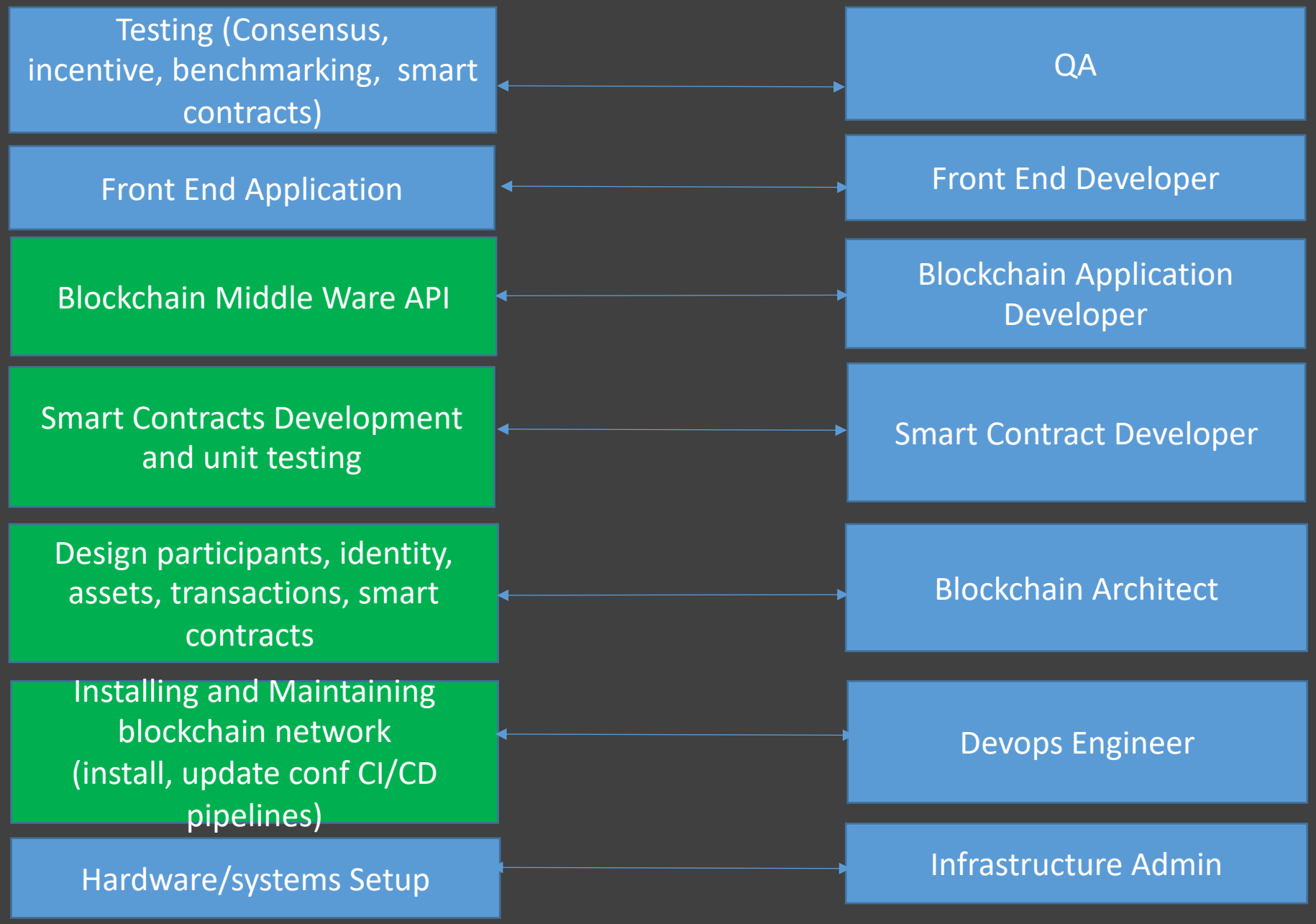
Cryptography

+

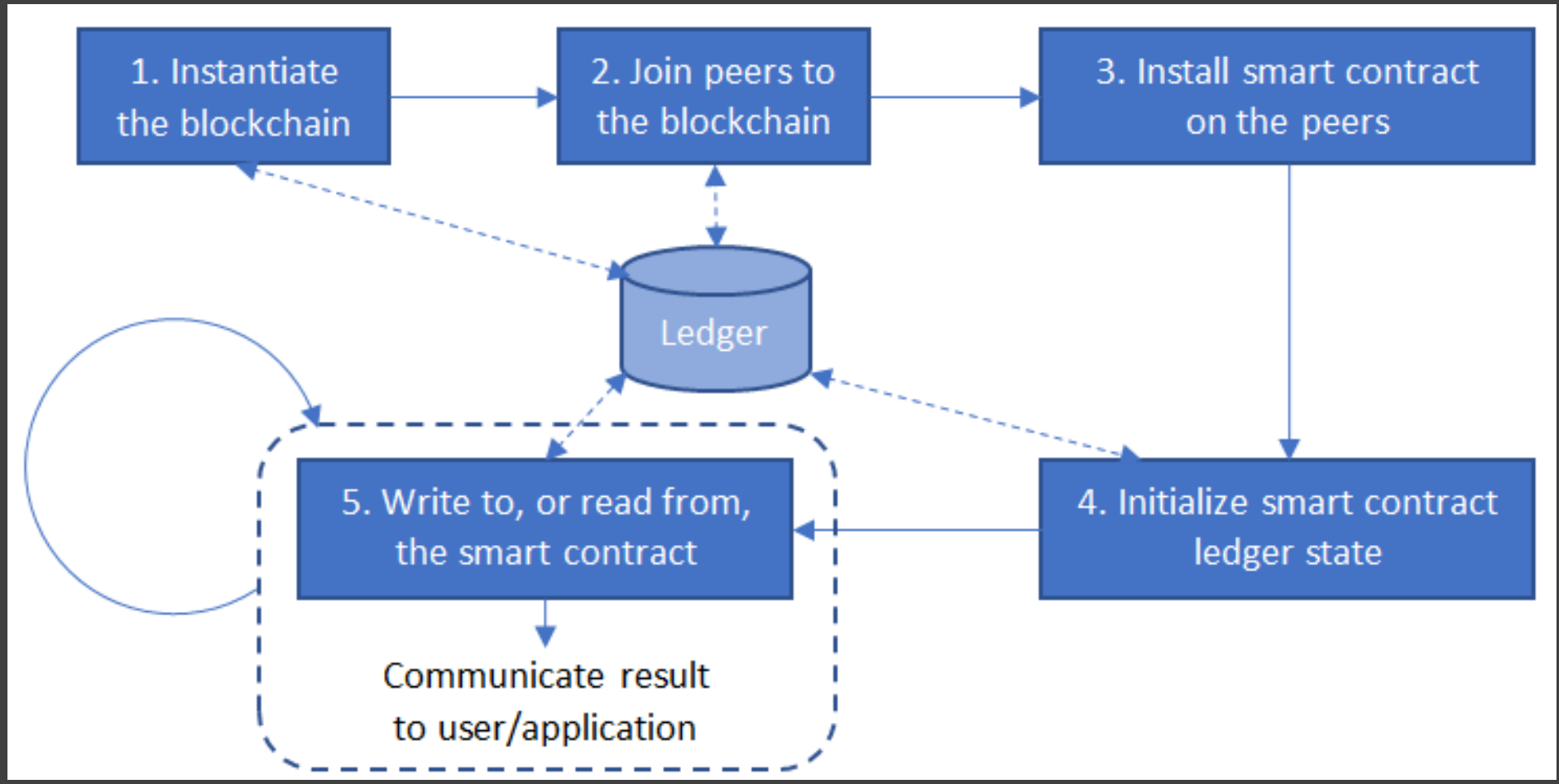
Mechanism/Incentive Design

NOTE: Incentive Design mostly needed for public blockchains. Permissioned blockchains does not need incentive design.

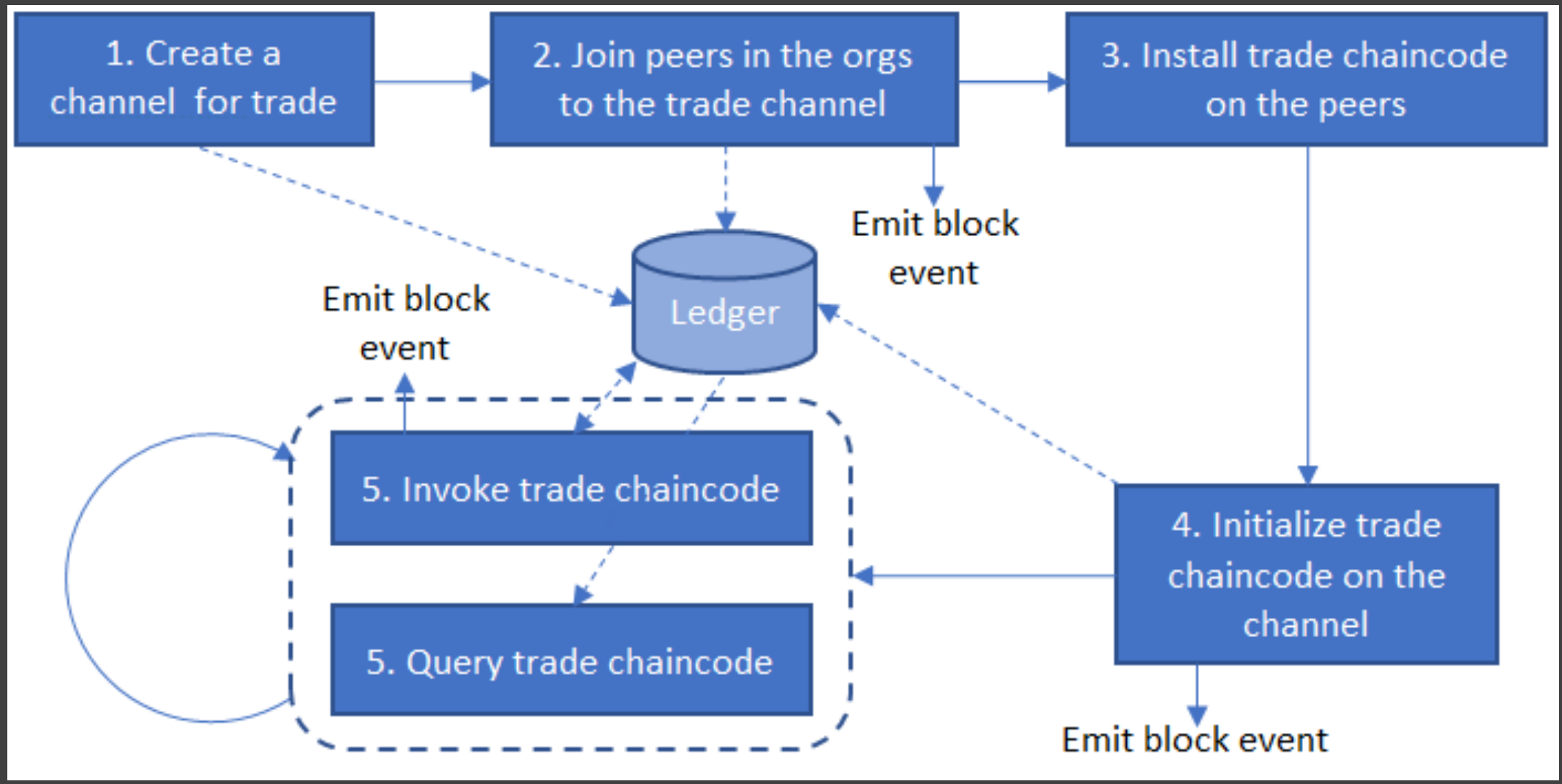
Blockchain Roles



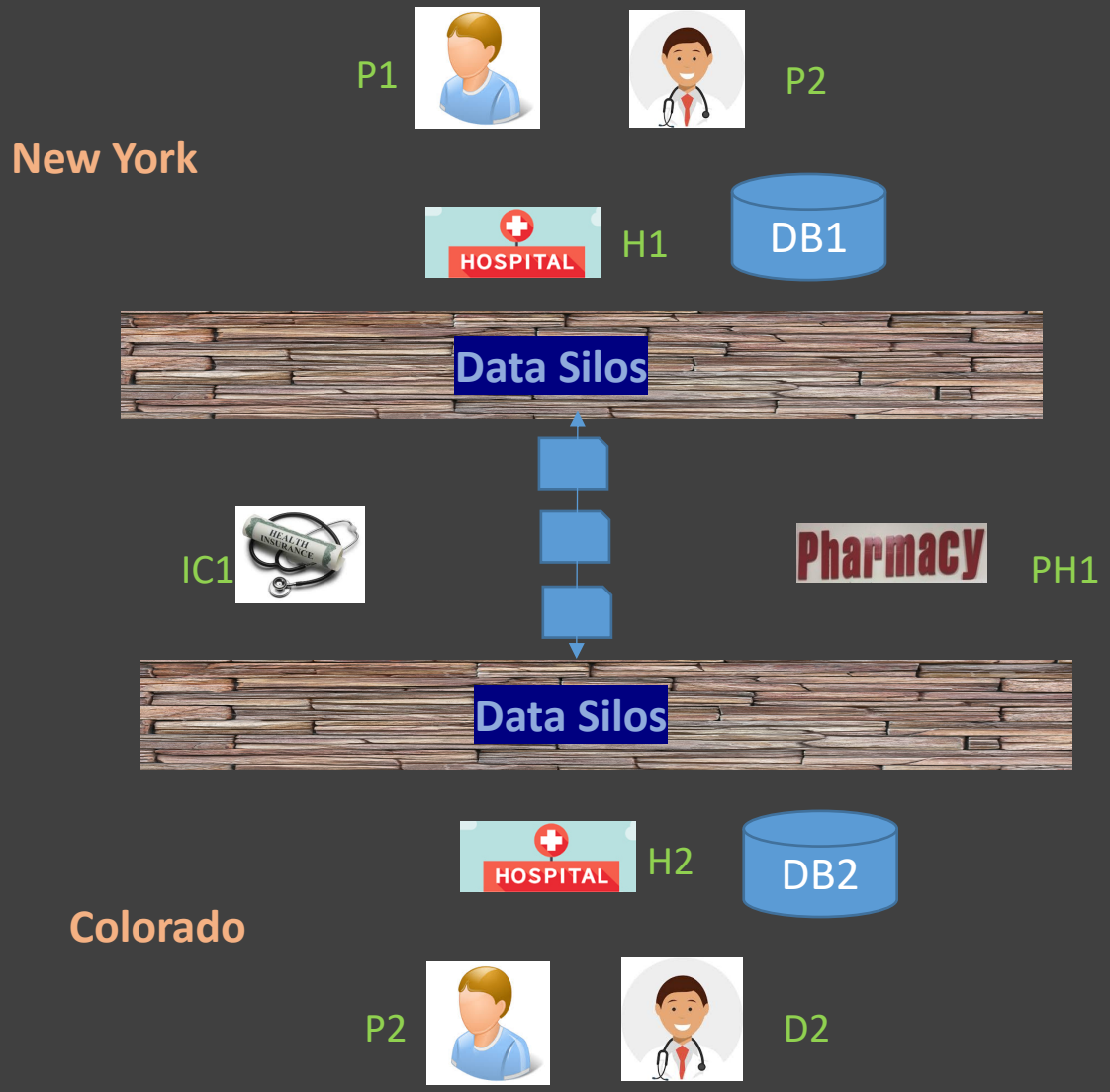
Instantiate Blockchain Network



Instantiating Channel



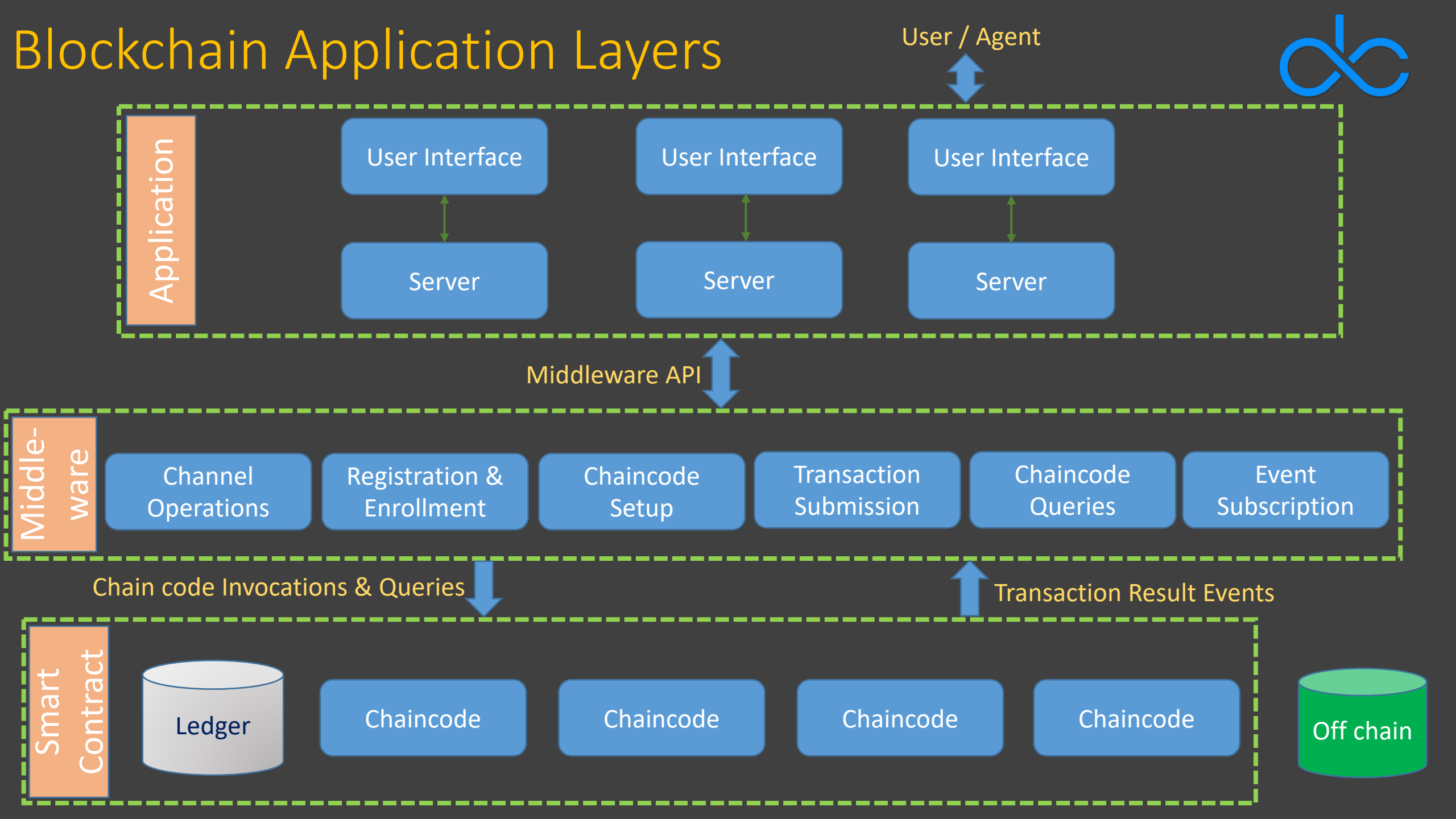
Current State of Healthcare



Sharing Data across providers is tedious

Patient Data is in provider control

Blockchain Application Layers



Designing Business Network

- Identify Participants
  - Individual
  - Orgs
  - systems/devices
- Design participants identity
- Design Number of Peers
- Design Orders
- Design Endorsers
- Design Channels
- Design Private Data (if needed)
- Design endorsement policy

1) Identify Types of Participants

Patient

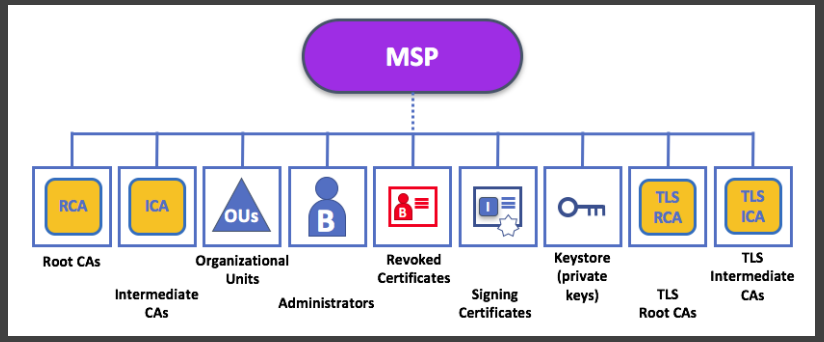
Doctor

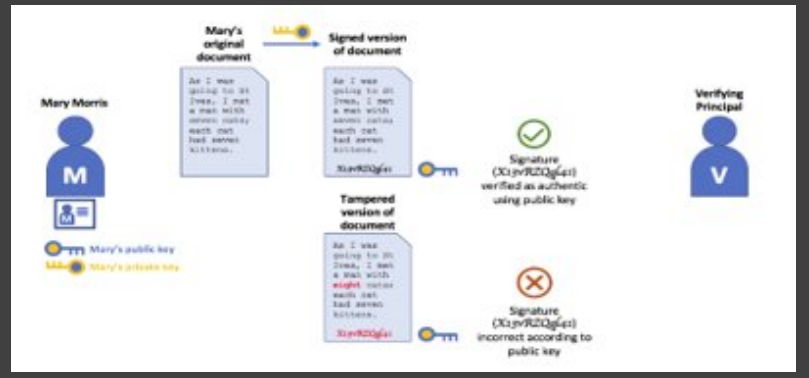
Insurance

Hospitals

Pharmacy

Design participant Identity





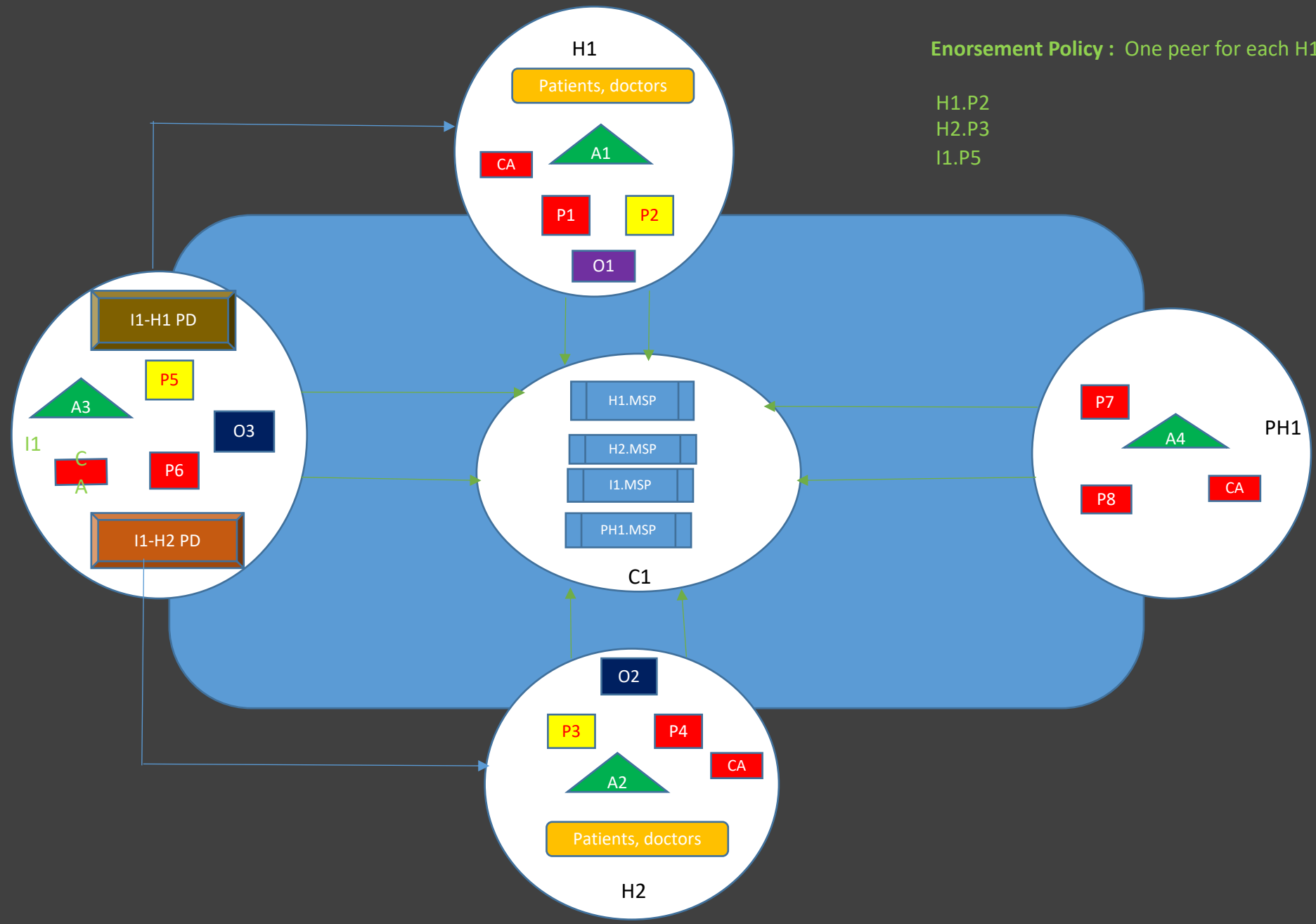
PD: Private Data collection

Endorsing Peer

Peer

Orderer

Application



Enrosement Policy : One peer for each H1, H2, I1

H1.P2
H2.P3
I1.P5

Designing Assets, Transactions & Chaincode

- Designing Assets and Transactions
  - Identify what Assets to store
  - Design Asset Lifecycle
  - Design Transactions & Asset flow between participants
- Designing Chain code
  - Design what Chain codes and what functions to write
  - Design Access control mechanisms for Chaincode
  - Design Transactions & Asset flow between participants

Designing Assets, Transactions & Chaincode

- Assets
  - Prescriptions
  - LabReports
  - Insurance Claims
- Transactions
  - Create/update Prescription/Labreport
  - Request Permission for Prescription/Labreport
  - Approve/Reject/Revoke permission for Prescription/Labreport
  - Claim Insurance
  - Settle Insurance Claim
- Smart Contracts/Chaincodes
  - Prescription SmartContract
    - Create/update/request/aprpove/reject/revoke prescription
  - Labreport SmartContract
    - Create/update/request/aprpove/reject/revoke prescription
  - Insurance Claim Contract
    - Claim/Settle insurance

Chaincode Attribute Based Access Control

- Only doctors should create/update prescriptions
- Only Lab Technicians can upload test reports
- Only Patients should Approve/Reject/Revoke access permissions to his/her medical records

- https://github.com/hyperledger/fabric/blob/release-1.1/core/chaincode/lib/cid/README.md

Blockchain Design considerations

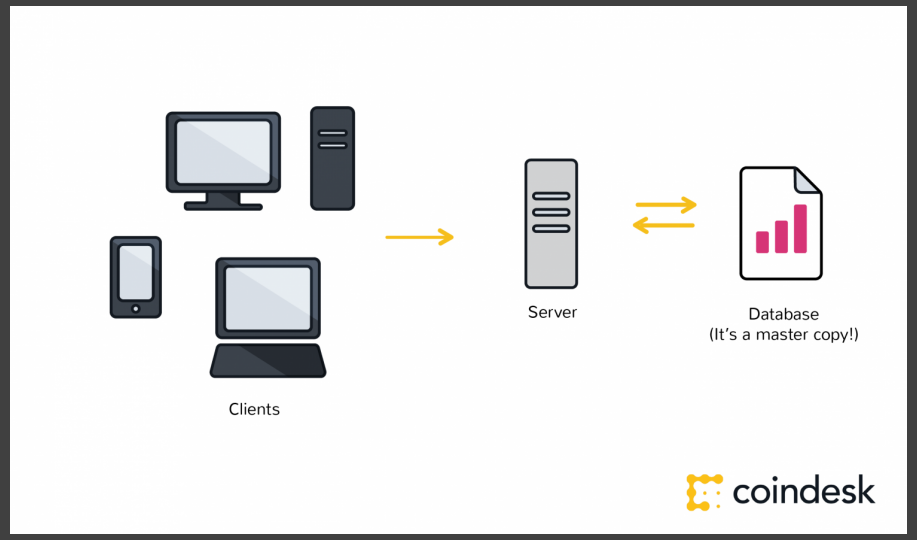
- Designing Data Storage

Blockchain ≠ Database

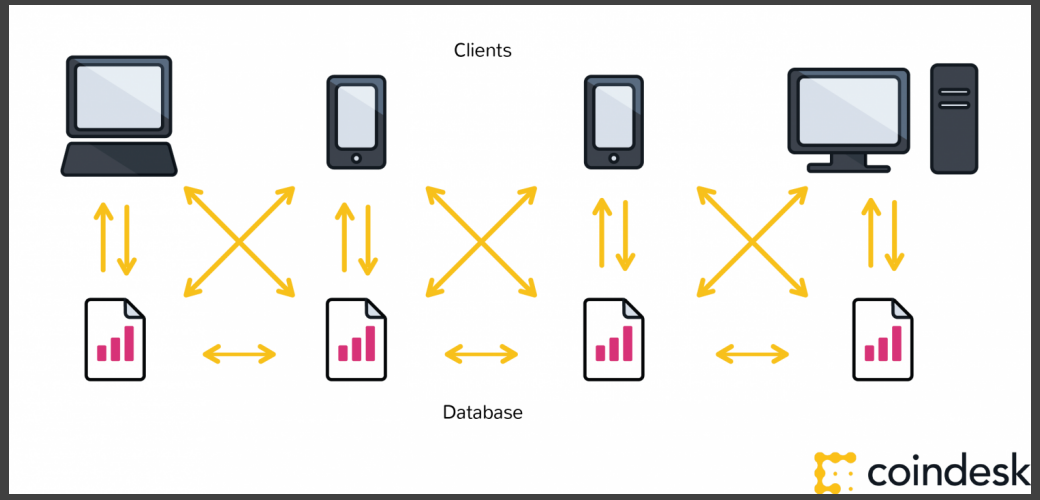
- On Chain
  - Store only the data that requires consensus among participants
  - Never store business sensitive data on-chain
- Off Chain
  - Store bulk data off chain and just store pointers to data on-chain

Blockchain ≠ Database

Client/Server Centralized systems



Decentralized systems



Blockchain ≠ Database

1. Blockchain applications storing many large pieces of data, straightforward on-chain storage is not a practical choice.

2. Data stored on blockchain is replicated to every node on blockchain

3. Storing more data cause replication/storage overhead and scalability bottlenecks in private blockchain

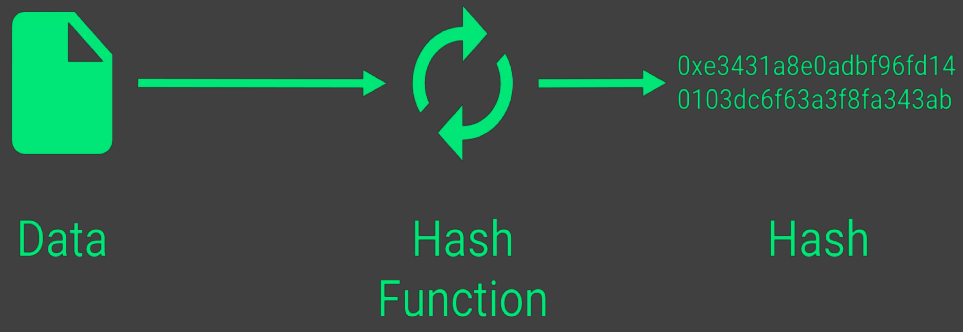
4. Storing more data cause more fees on public chains

5. Only data that need immutability and tamper proof evidence need to store on blockchain

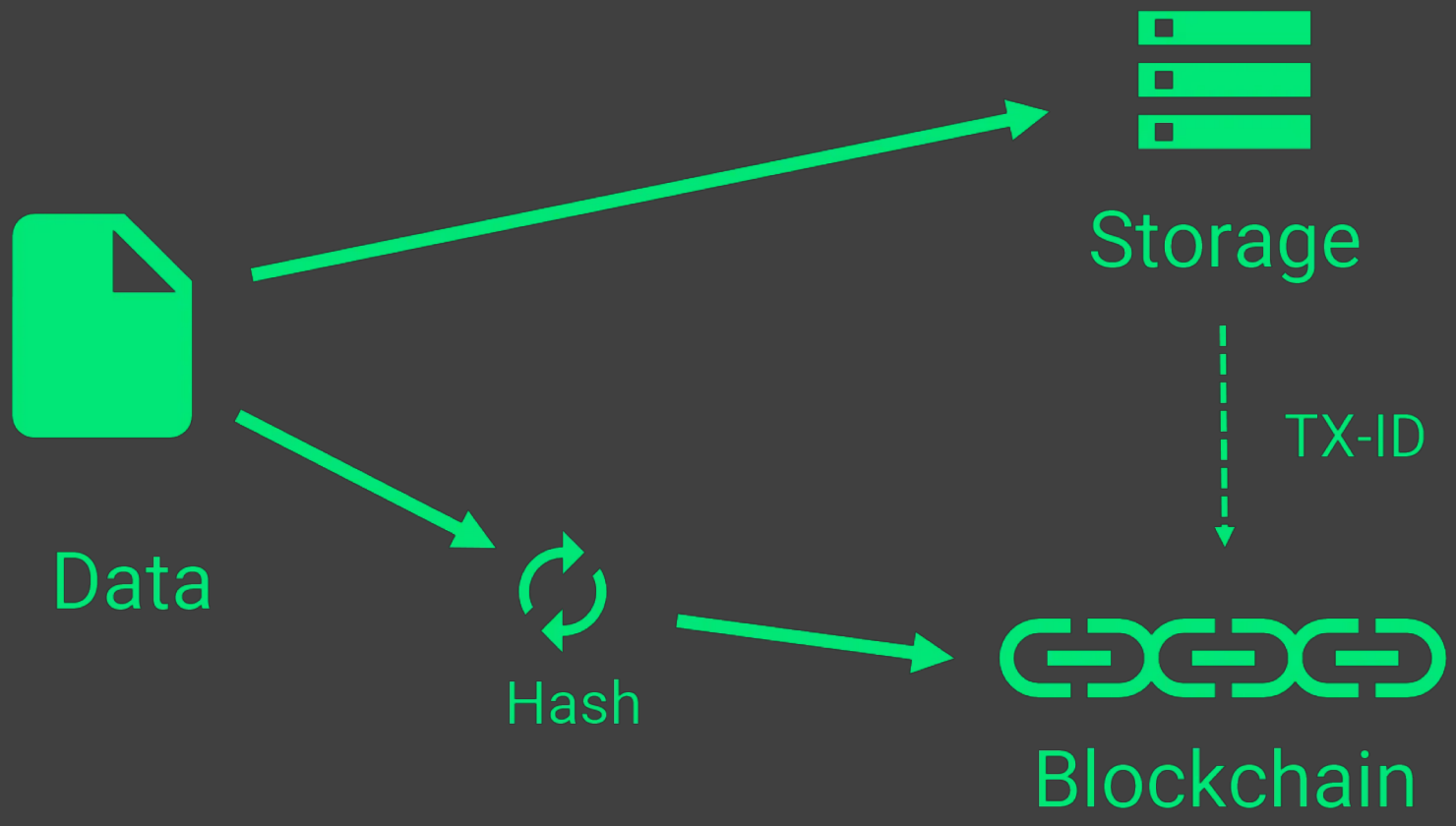
6. Blockchain doesn't have flexible query language

Off chain Storage

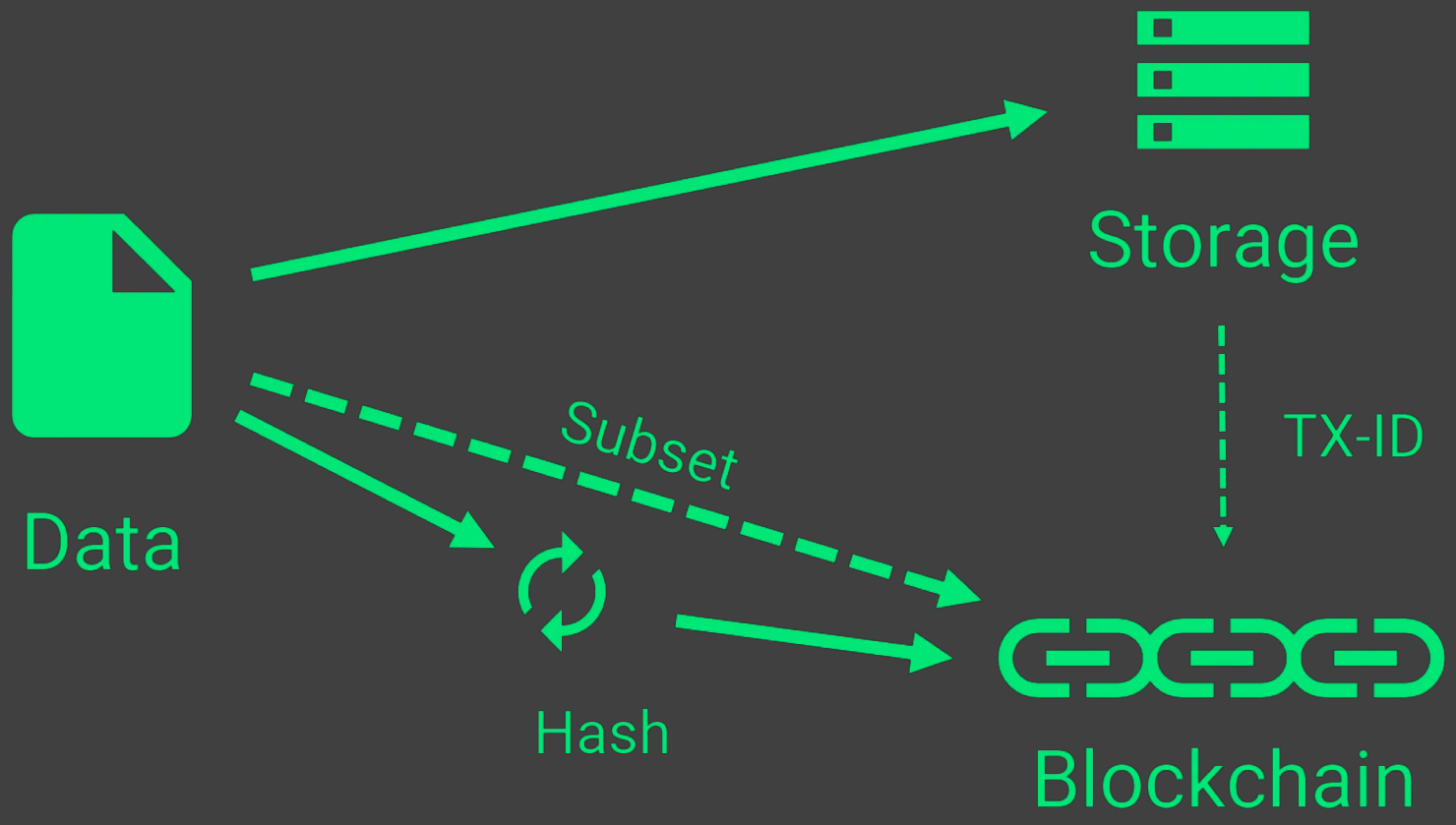
- Store only hashes in blockchain and actual content in other storage
- Incase of sensitive contents encrypt the contents and store encrypted content hashes on blockchain.
- Hash protects the content from tampering



Storing Only Hash on blockchain



Storing Hash and Subset of data on blockchain



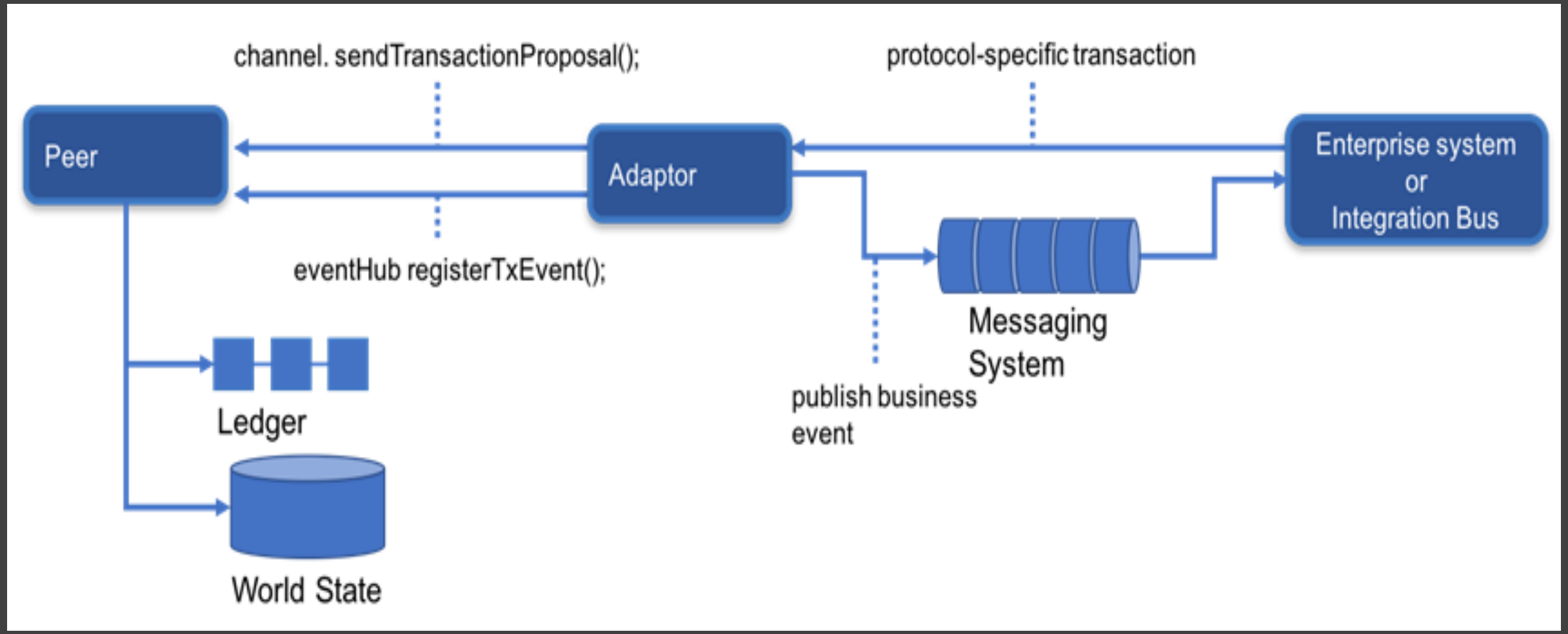
Off Chain storage

1. IPFS
2. IPDB
3. Swarm
4. BigChainDB
5. Storj
6. Sia
7. RDBMS
8. Distributed Databases

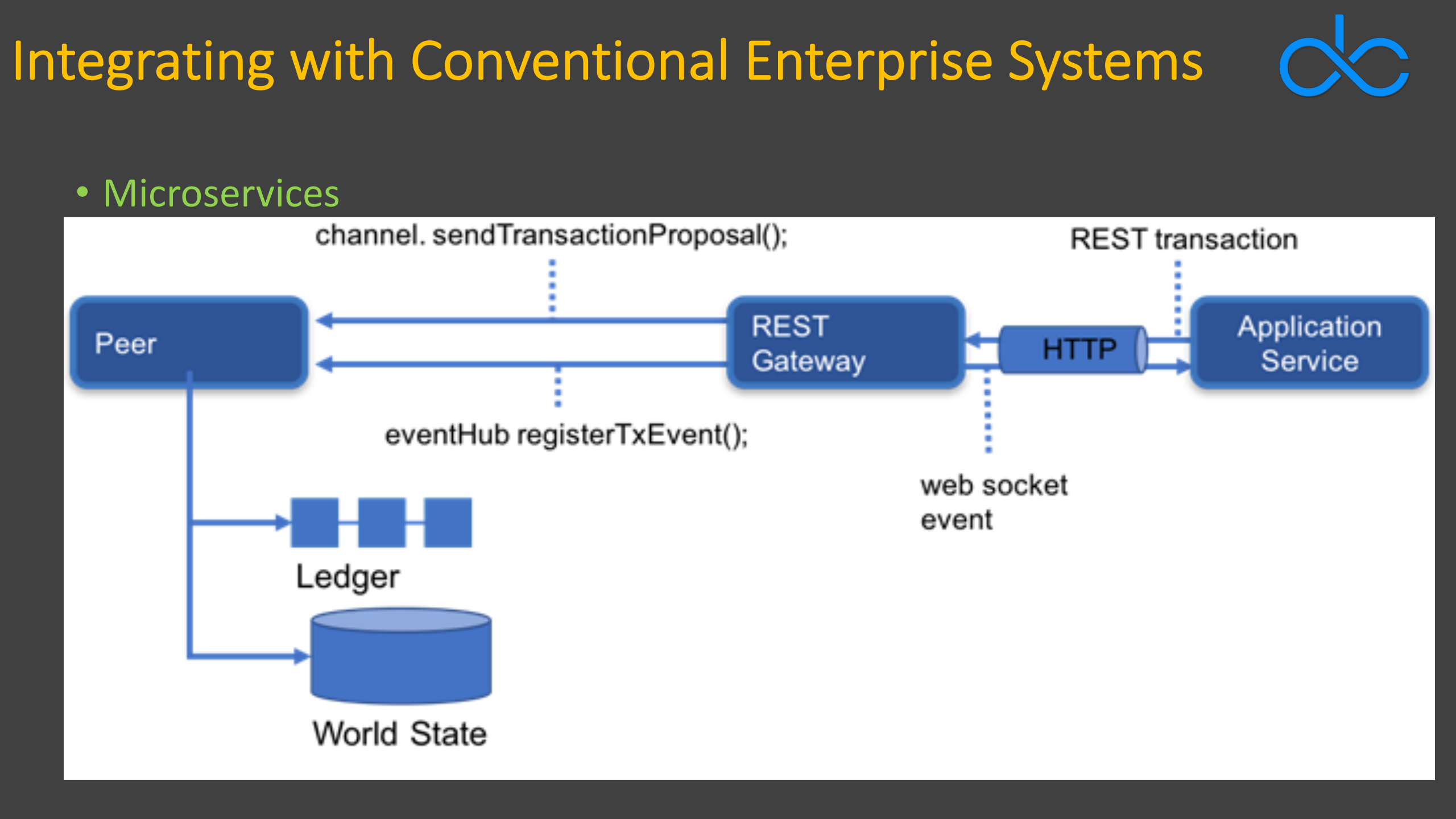
NOTE : RDBMS and Distributed databases controlled by central authorities and they are single pont of failure

Integrating with Conventional Enterprise Systems

- Event Driven



Integrating with Conventional Enterprise Systems



- Microservices

What next?

- Engage Better as a Community
- Use Slack & Trello Effectively
- Share Use Cases & Ideate
- Propose Online Sessions
- Finalize Use Cases & Start Building
- Meet again on January 5, 2019


WHO?
when?
which?
WHERE?
why?
WHEN?
How?
WHAT?