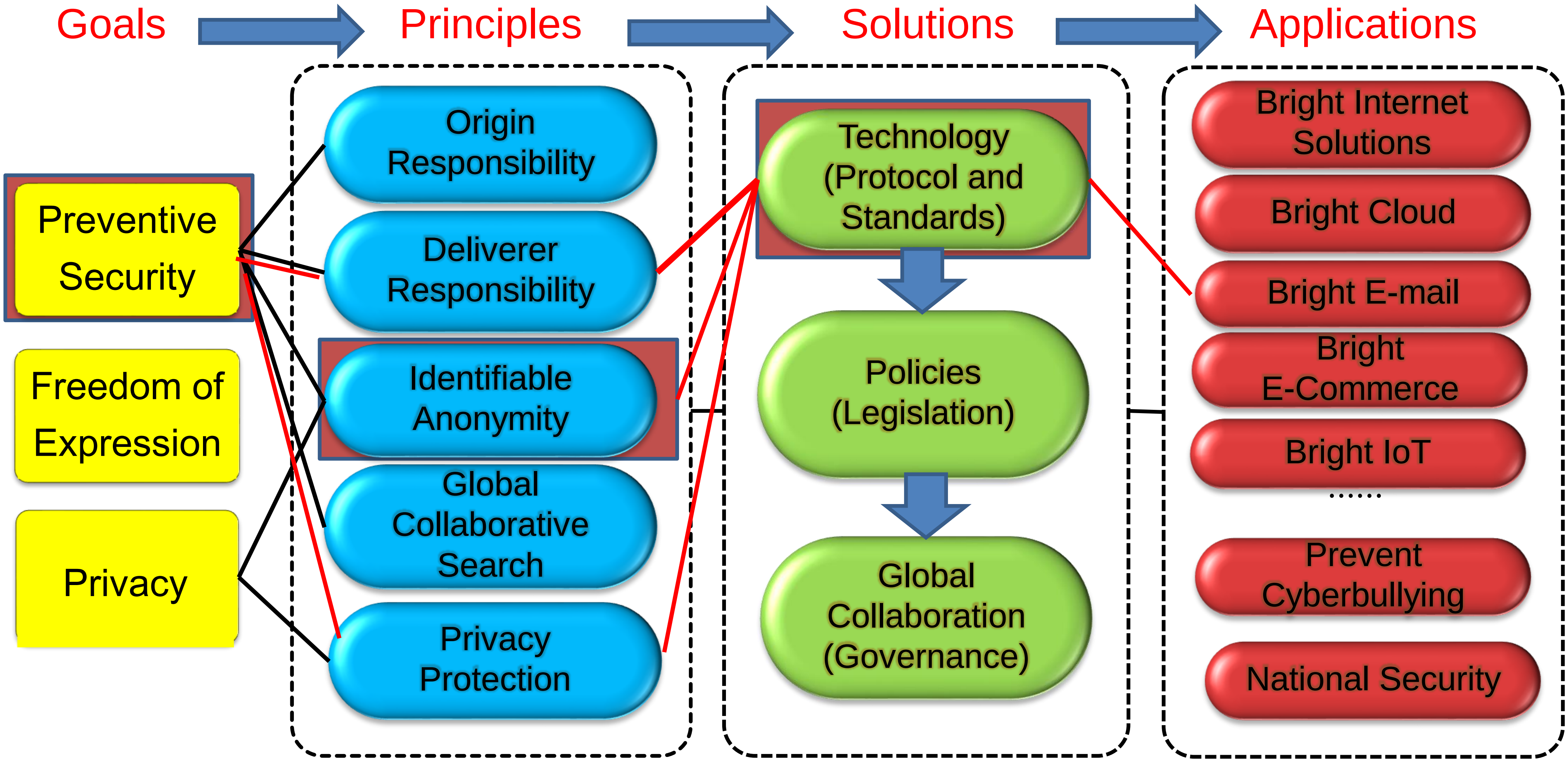
A Bright e-mail System based on Hyperledger Fabric Blockchain

John CHOI, Markany(juchoi@markany.com)
Jae Kyu LEE, KAIST(jklee@kaist.edu.kr)

Chang Won KIM, MarkAny (permedia@markany.com)
Hong Joon Ha, MarkAny (hongjuoon@markany.com)

Bright Internet: Achieving “Identifiable Anonymity”



Proposal: Bright Email

- Bright e-Mail system based on Hyperledger Fabric Architecture

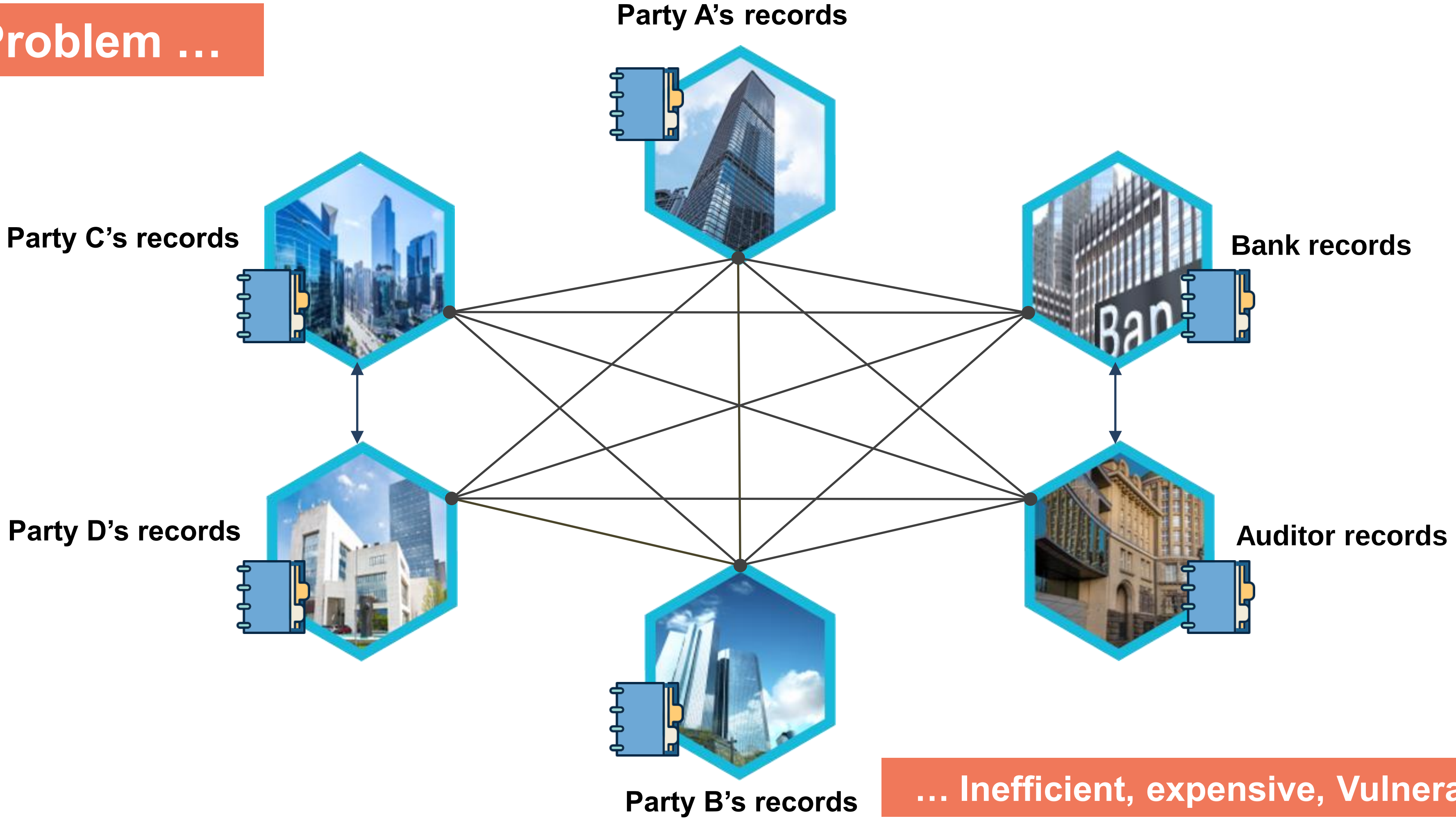
- Achieving ‘Deliverer Responsibility’ and ‘Identifiable Anonymity’ by Utilizing Membership Management Module (CA-Cert, E-Cert, T-Cert)
- Is it possible simply to apply Hyperledger Fabric Architecture to Bright Mail?
    - Benefits of Blockchain Applications: ensuring No Denial, but maintaining Redundant DB
    - Technical Issue: Structure of Blockchain Shard to Reduce ‘Scalability problem’

Contents

1. **Hyperledger Fabric Blockchain**
2. **Bright Mail System**
3. **Technical Issues**

Problems of BitCoin and Ethereum Blockchain

Problem …

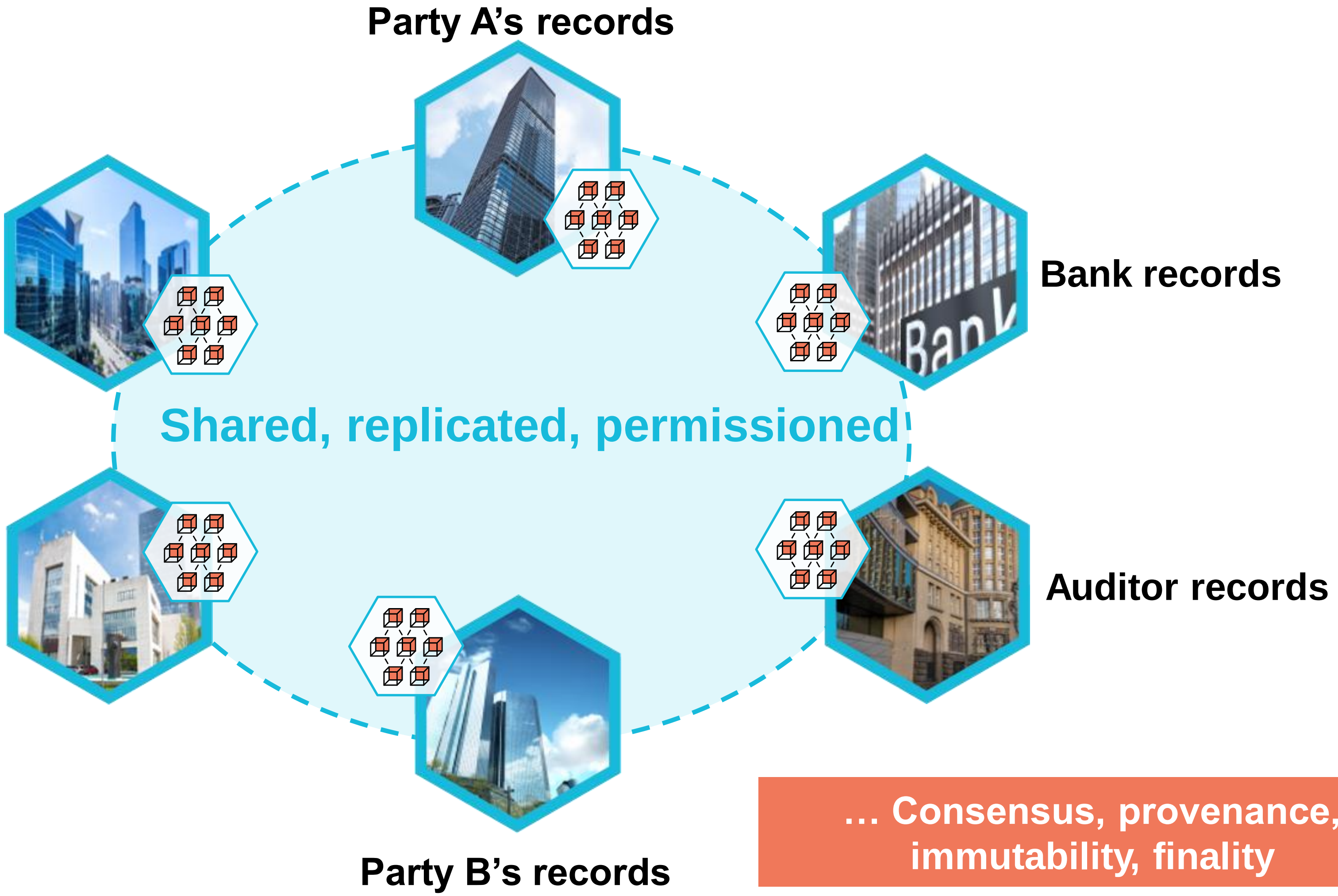


… Inefficient, expensive, Vulnerable

Too Large Blocksize, and Too Slow for processing transactions, because of too many connections and expensive consensus in a public chain, while privacy and confidentiality are at Risk

Hyperledger Fabric Blockchain

Solution …



… Consensus, provenance, immutability, finality

permissioned, distributed, and shared ledger, while providing a secure, robust model for identity, auditability and privacy

Advantages of Hyperledger Fabric

- Practical Structure Suggested for existing Transactions
- Optimize Conflicting Goals: Consensus and Scalability

Privacy & Confidentiality

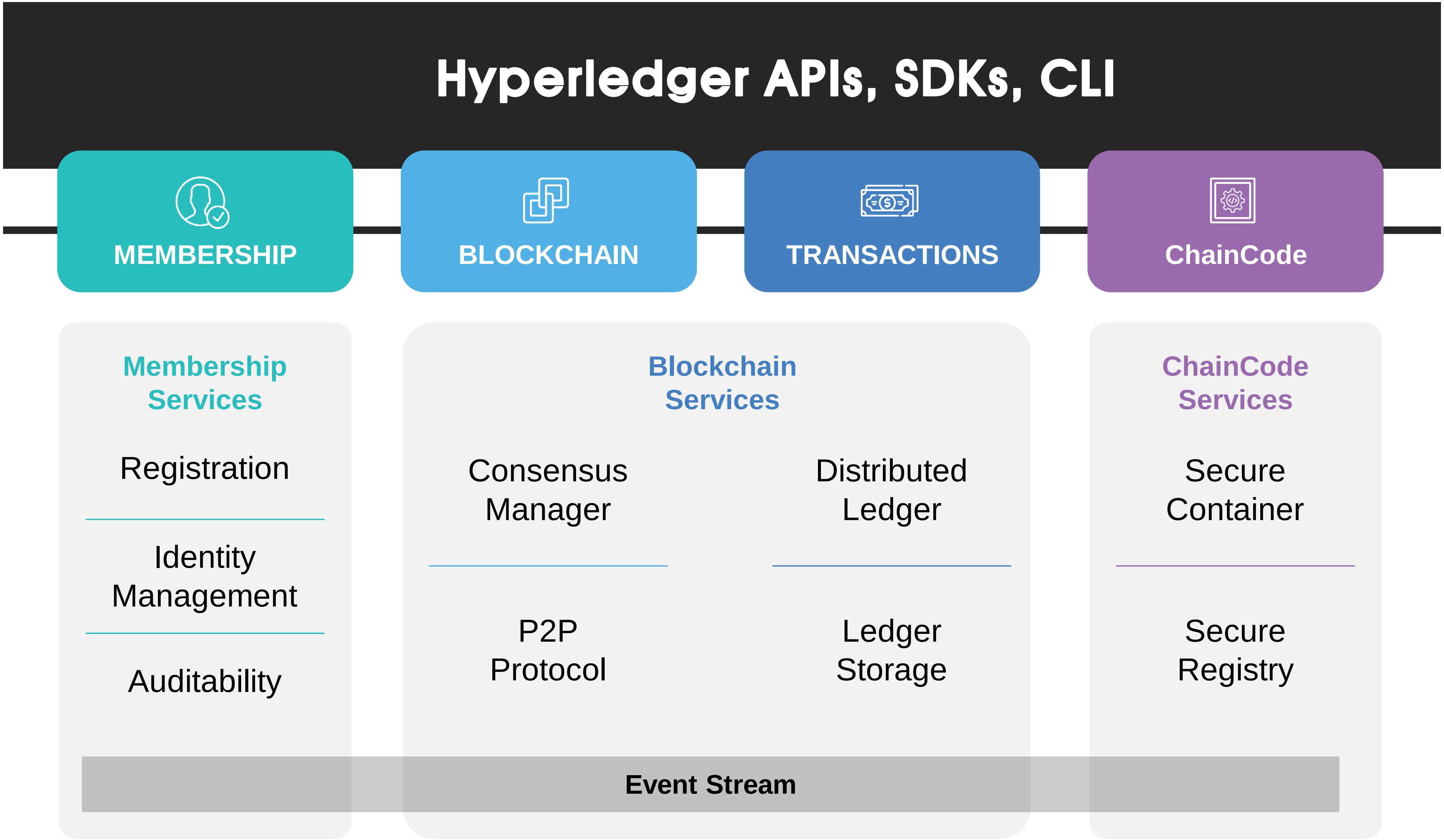
Auditability & 'Searchable'

Transparency

scalability

Modularity

Hyperledger Fabric: Services



Hyperledger APIs, SDKs, CLI

MEMBERSHIP

BLOCKCHAIN

TRANSACTIONS

ChainCode

Membership Services

Registration

Identity Management

Auditability

Blockchain Services

Consensus Manager

Distributed Ledger

P2P Protocol

Ledger Storage

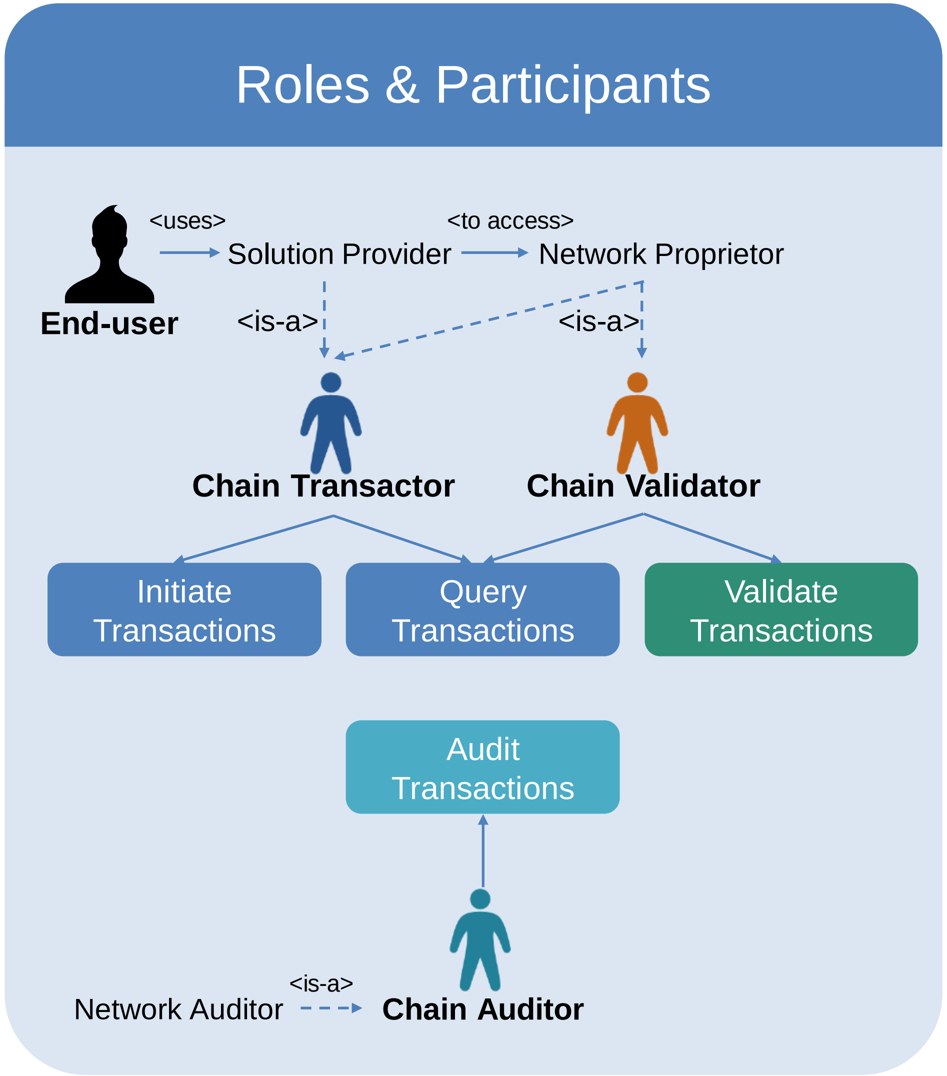
ChainCode Services

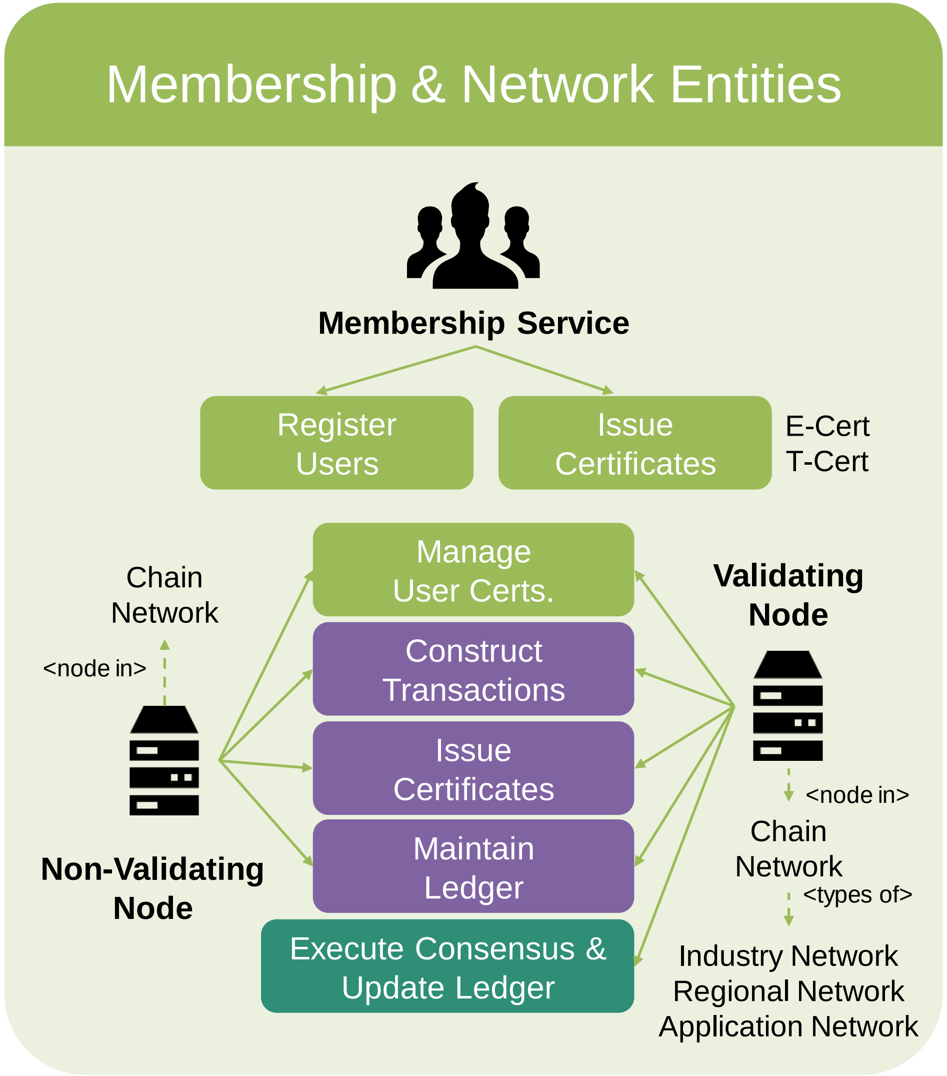
Secure Container

Secure Registry

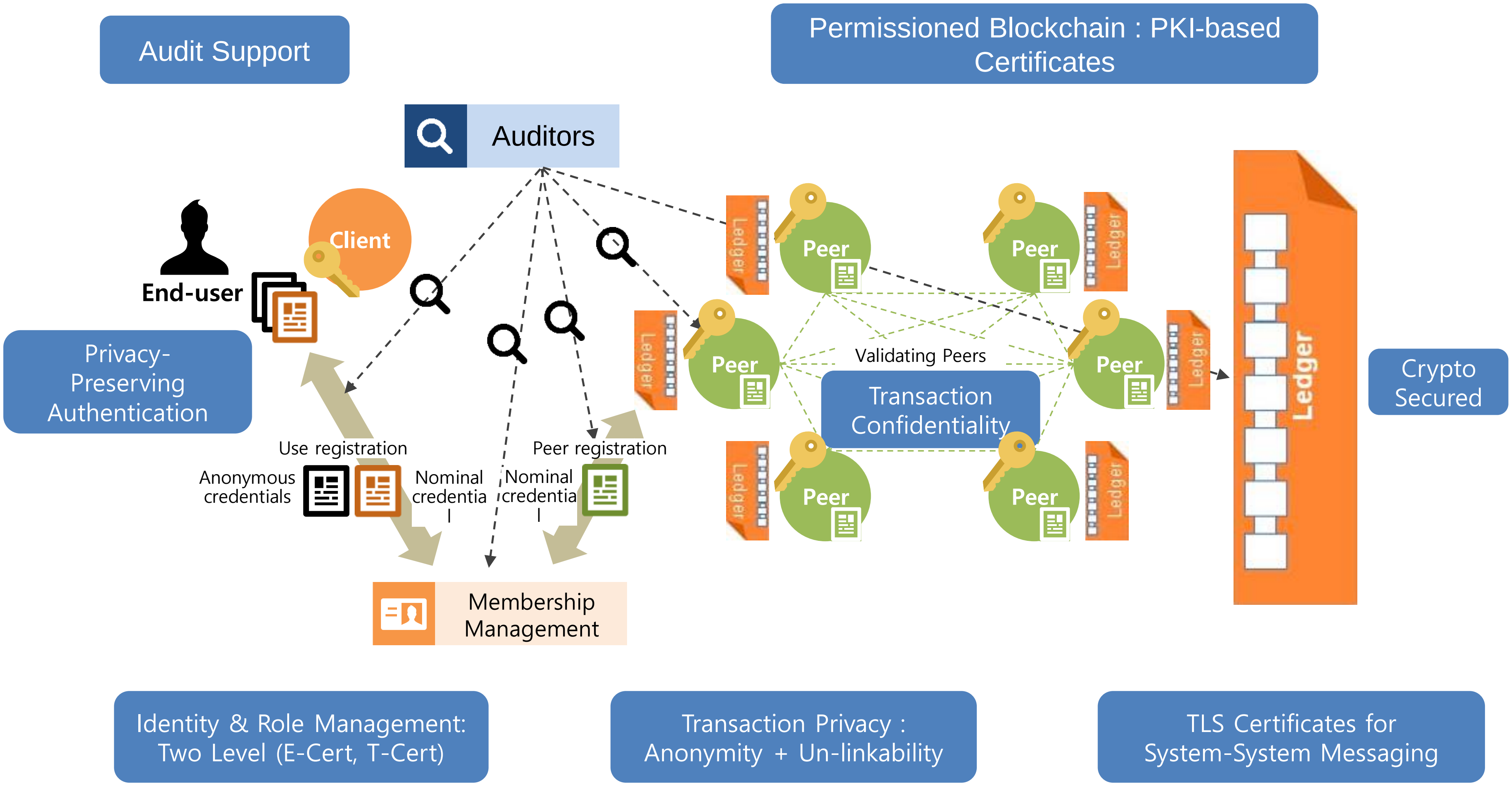
Event Stream

Hyperledger : System Context





Hyperledger : Security Review



Audit Support

Permissioned Blockchain : PKI-based Certificates

Privacy-Preserving Authentication

Crypto Secured

Identity & Role Management: Two Level (E-Cert, T-Cert)

Transaction Privacy : Anonymity + Un-linkability

TLS Certificates for System-System Messaging

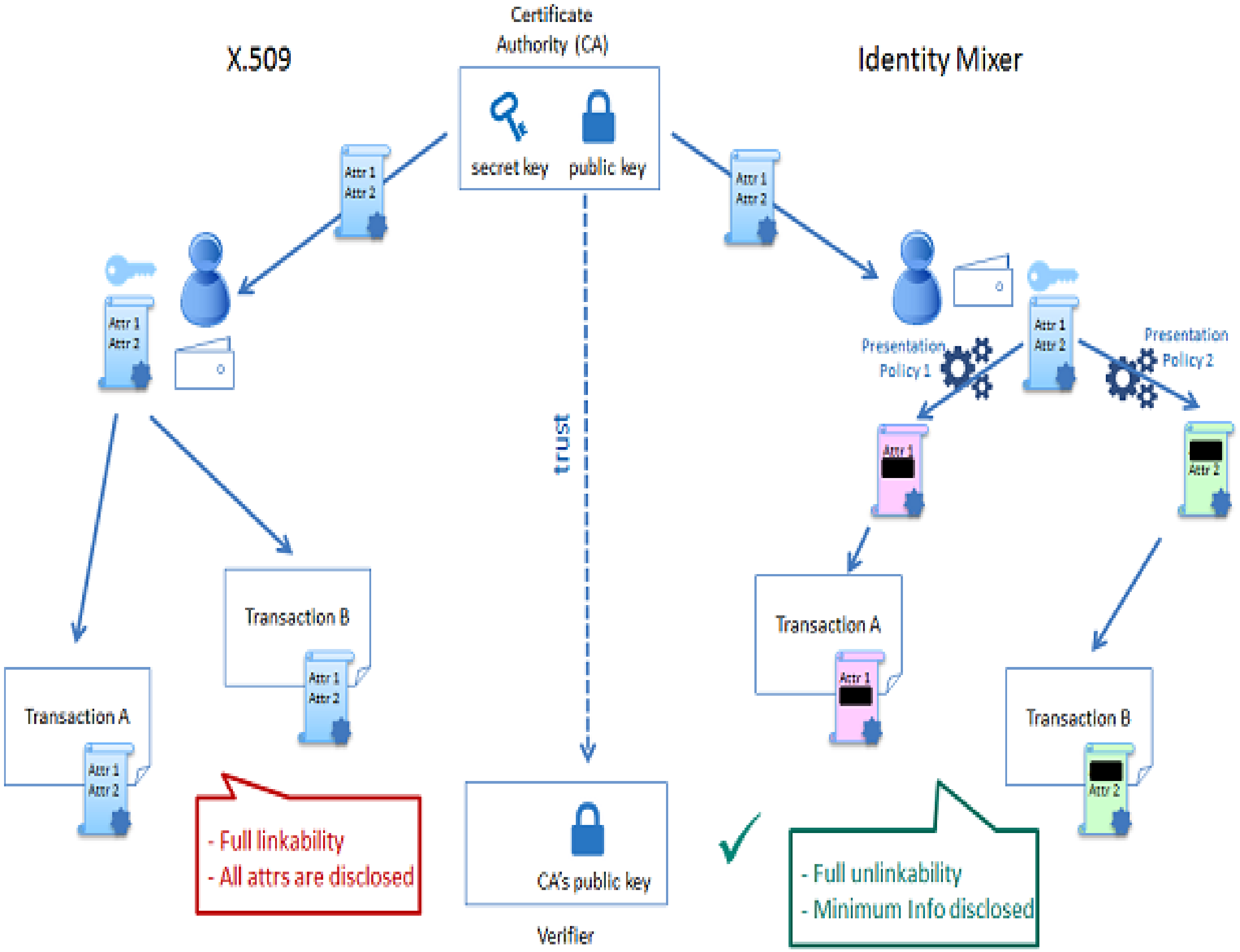
Hyperledger Fabric Blockchain

- Resolve Difficult Problems
  - Scalability problem: Introduce concept of ‘Private Blockchain’ and limit participation of network nodes, in the name of ‘Membership’
  - Consensus Algorithm: introduce Practical Byzantine Fault Tolerance (PBFT) Algorithm
- Enhance Privacy and Confidentiality
  - ID, behavior, transaction and conditions, and parameters of other nodes should not be disclosed to network participants except parties directly involved
  - Secret data in transaction should be decrypted and readable to only interested parties
- Searchable

- Confidentiality should be kept while contents of the ledgers should be searchable to the involved parties

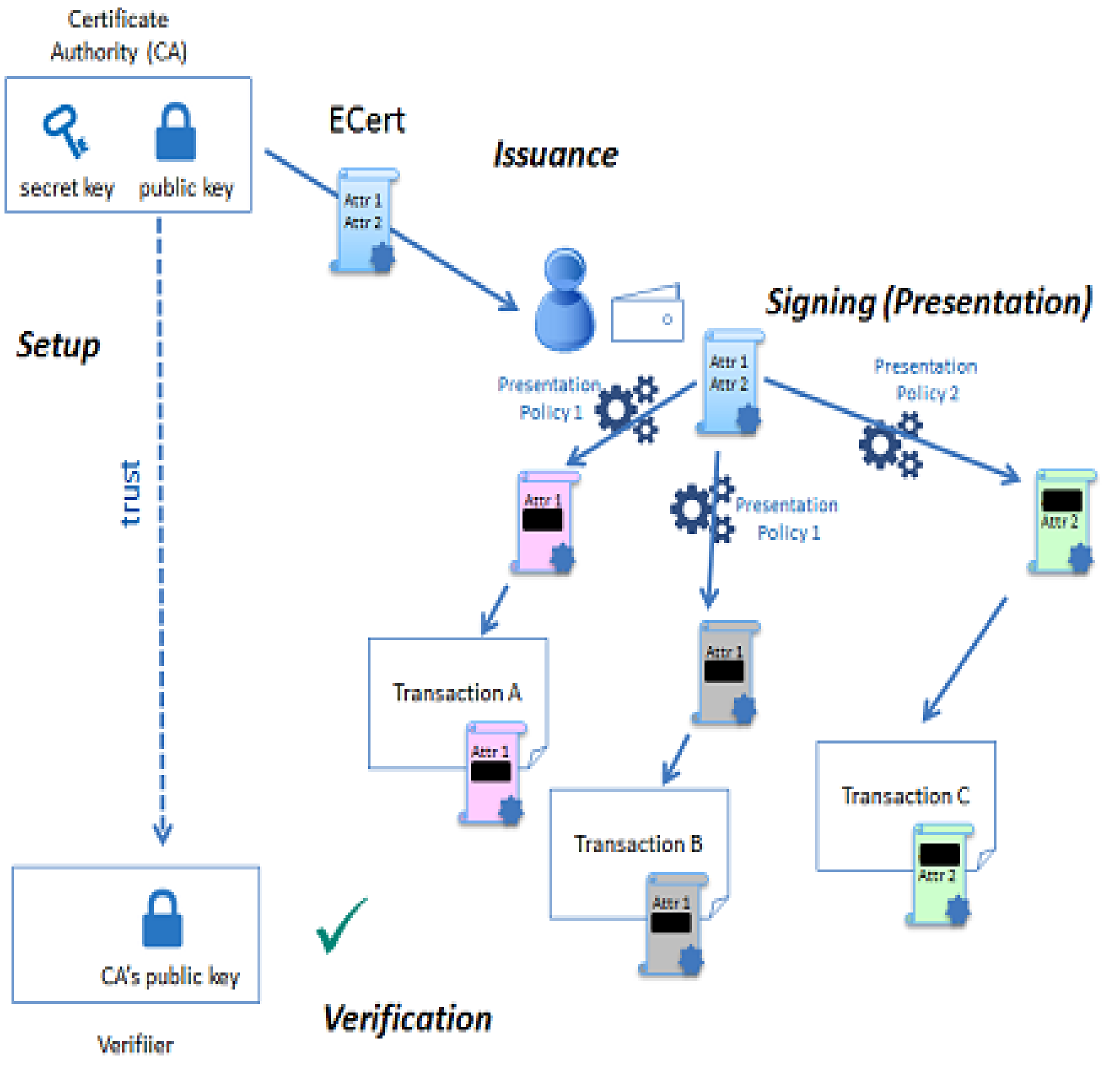
-ex) Sellers to join the bidding should reveal offers in ledgers to Buyers in the network

Identity Mixer



- a trust model and security guarantees
- provide advanced privacy features such as “unlinkability” and minimal attribute disclosure.
- A user stores her credentials in a credential wallet application. User derives a fresh and unlinkable presentation token from her credentials according to an access control policy

Identity Mixer Verification



- E-Cert: A peer or a client generates a secret key and creates a request for an enrollment certificate, and e-cert is issued in the form of an Identity Mixer credential

- E-Cert is stored together with the corresponding credential secret key on the peer side or by the client SDK. Then, a client/a peer generates a fresh “unlinkable” presentation token and discloses the attributes required by the access control policy, and then sign a transaction

Bright e-Mail System

- **Security and Certificates**
  - **Utilization of Membership Management: CA-Certificate, E-Certificates, T-certificate**
  - All the transactions should follow regulations and thereby should be accessed and investigated by Regulators
  - All activities are initiated with cryptographic Certificates which can put into user’s confidential data
  - Register issue ID for network participation
  - Network members can participate into transactions with key issued by ID membership, while users joining transaction can hide ID to keep privacy

- **Maintaining Replicated Data in Distributed Ledgers**
  - Maintaining replicated data and possibly introduction of BI Index, ‘source and deliverer responsibility’ can be greatly enhanced

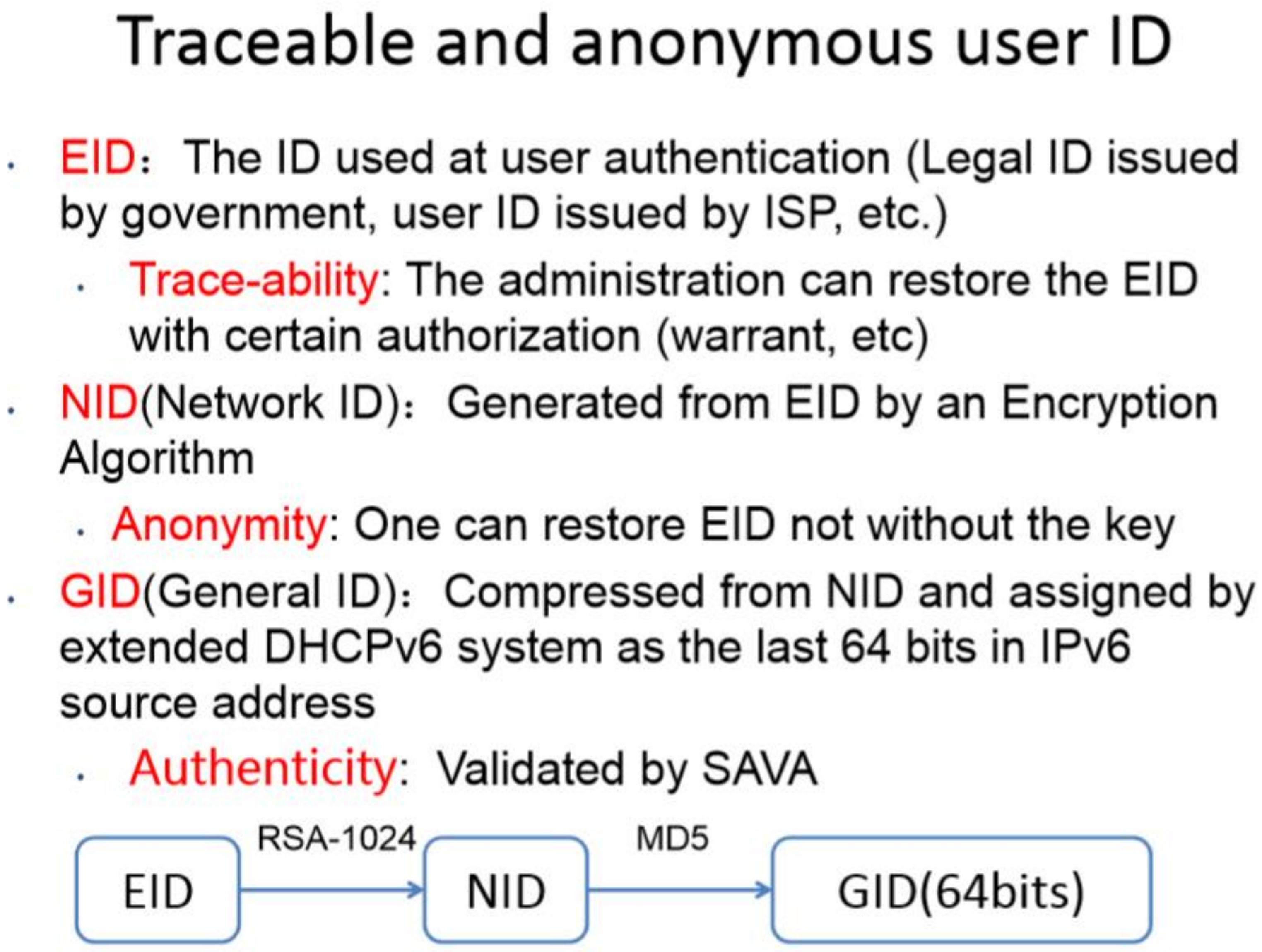
Contents

1. **Hyperledger Fabric Blockchain**

2. **Bright Mail System**

3. **Technical Issues**

Designing IPv6 for Traceable Anonymity (Jun Li, 2015)

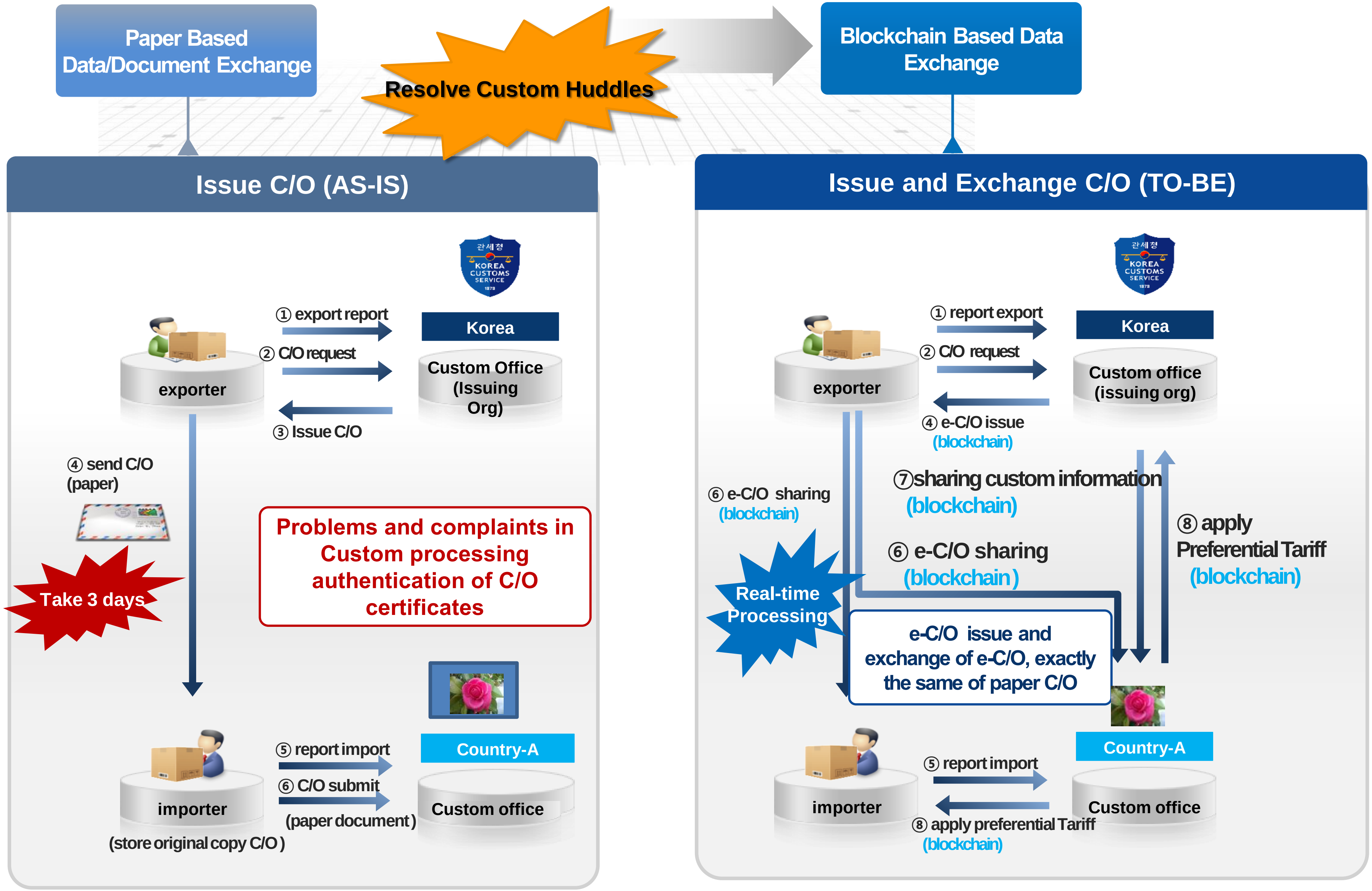


Traceable and anonymous user ID

- EID： The ID used at user authentication (Legal ID issued by government, user ID issued by ISP, etc.)
  - Trace-ability: The administration can restore the EID with certain authorization (warrant, etc)
- NID(Network ID)： Generated from EID by an Encryption Algorithm
  - Anonymity: One can restore EID not without the key
- GID(General ID)： Compressed from NID and assigned by extended DHCPv6 system as the last 64 bits in IPv6 source address
  - Authenticity: Validated by SAVA

Blockchain Based Certificate of Origin (C/O) System

MarkAny



Paper Based Data/Document Exchange

Resolve Custom Huddles

Blockchain Based Data Exchange

Issue C/O (AS-IS)

exporter

① export report

② C/O request

③ Issue C/O

Korea

Custom Office (Issuing Org)

④ send C/O (paper)

Take 3 days

Problems and complaints in Custom processing authentication of C/O certificates

importer

(store original copy C/O)

⑤ report import

⑥ C/O submit

(paper document)

Country-A

Custom office

Issue and Exchange C/O (TO-BE)

exporter

① report export

② C/O request

④ e-C/O issue (blockchain)

Korea

Custom office (issuing org)

⑥ e-C/O sharing (blockchain)

⑦sharing custom information (blockchain)

⑥ e-C/O sharing (blockchain)

⑧ apply Preferential Tariff (blockchain)

Real-time Processing

e-C/O issue and exchange of e-C/O, exactly the same of paper C/O

importer

⑤ report import

⑧ apply preferential Tariff (blockchain)

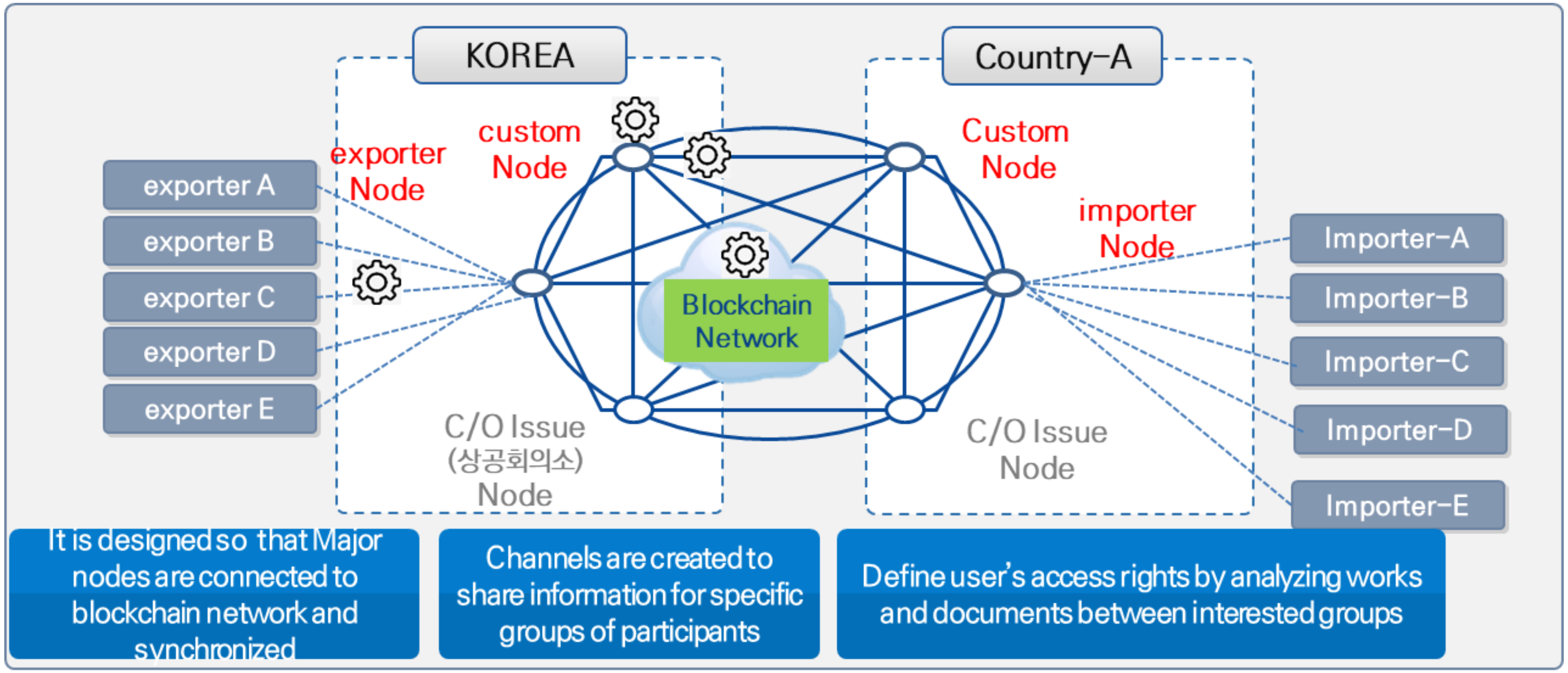
Country-A

Custom office

Overall Architecture

Of

E-C/O system

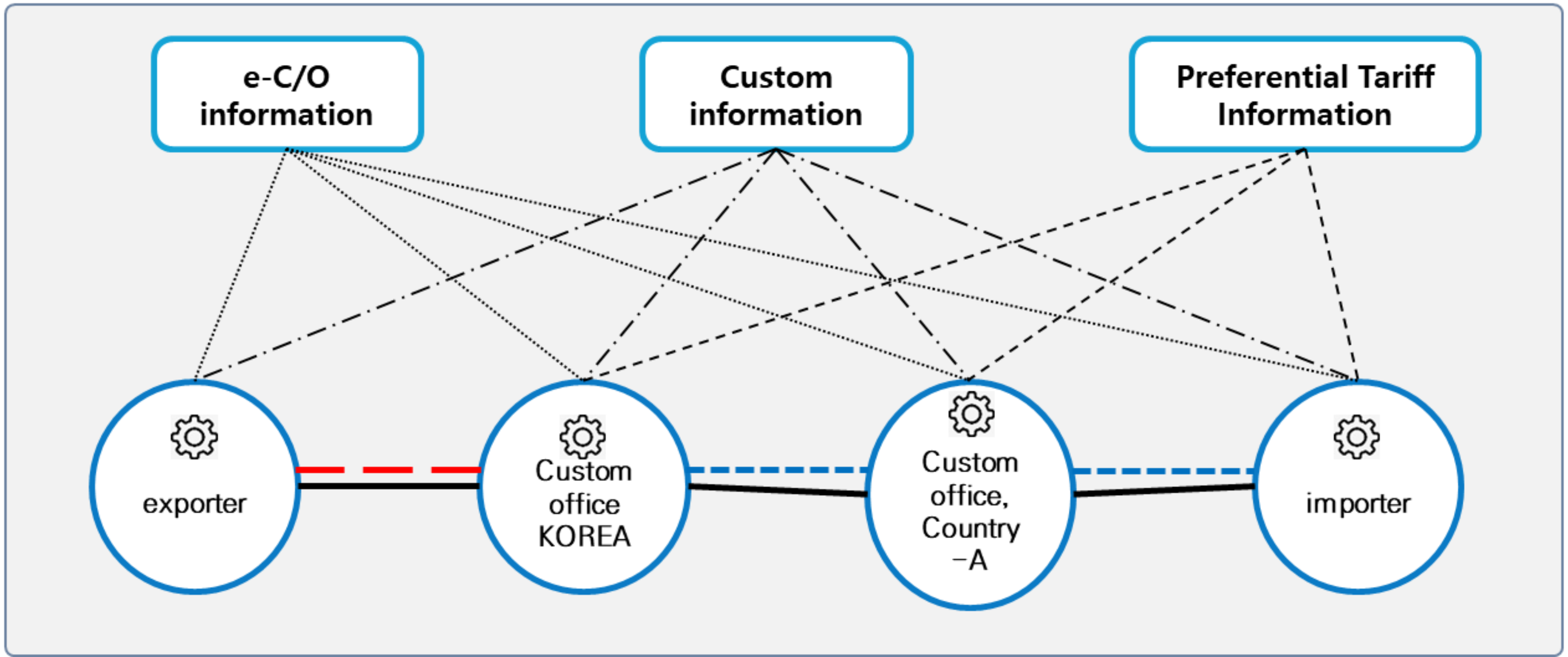


It is designed so that Major nodes are connected to blockchain network and synchronized

Channels are created to share information for specific groups of participants

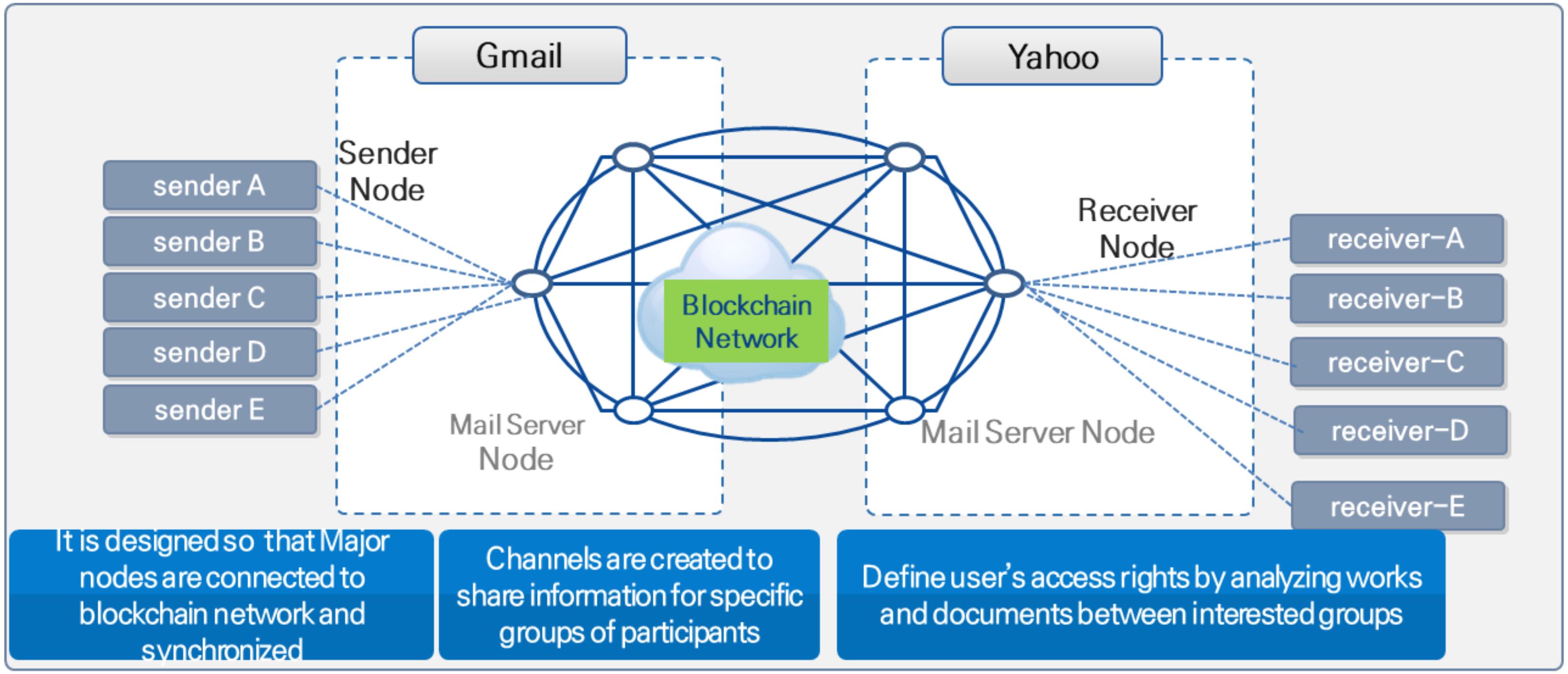
Define user's access rights by analyzing works and documents between interested groups

e-C/O Data exchange



🟥 Issue channel
⬛ Exchange channel
🟦 Use channel

Bright e-mail System

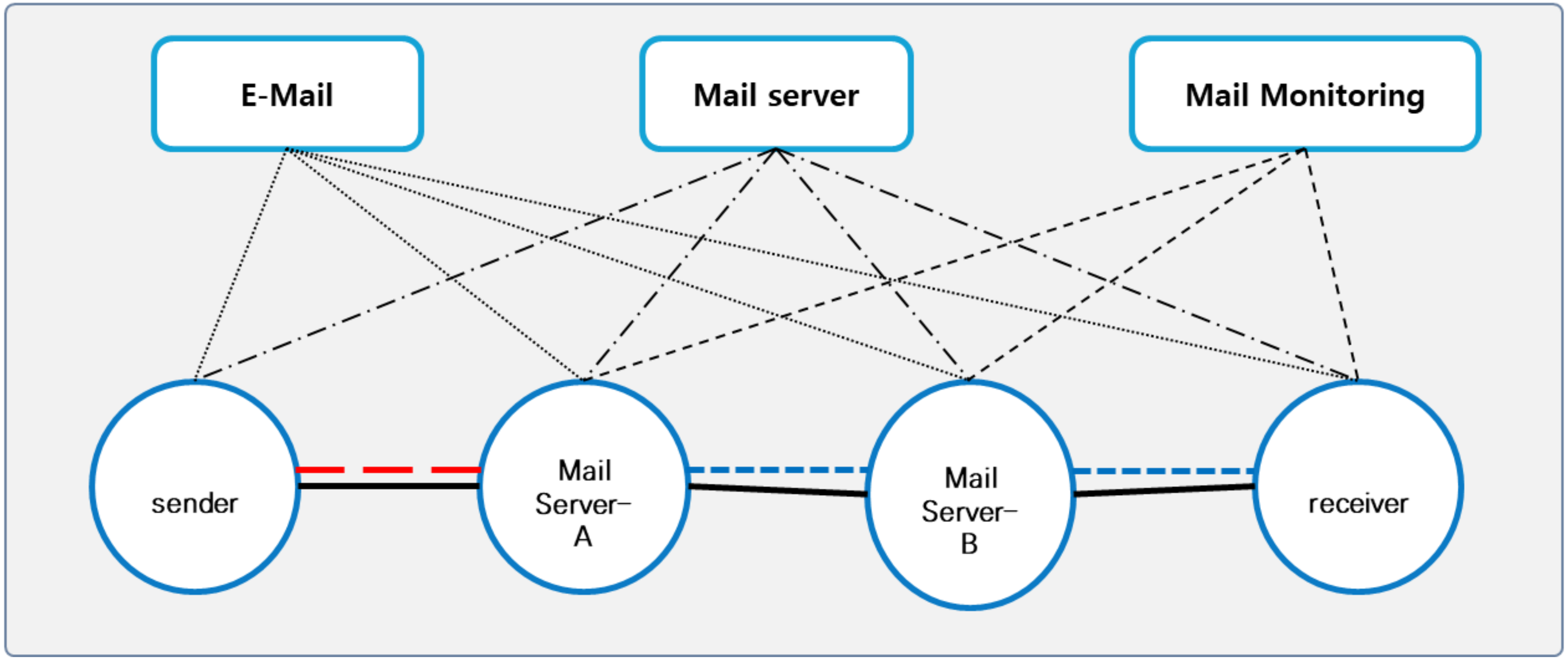


It is designed so that Major nodes are connected to blockchain network and synchronized

Channels are created to share information for specific groups of participants

Define user's access rights by analyzing works and documents between interested groups

e−mail exchange



🟥 Sender channel
▬ Exchange channel
┅ communication channel

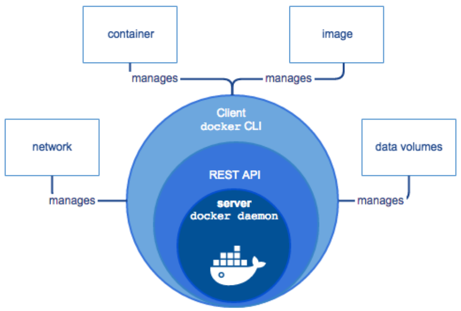
Docker Swarm

- Docker Swarm assigns containers for network, CPU, Memory, and storage in multi-hosts environment
- Docker Swarm is basically Client-Server application programs consisted of CLI (command line interface), REST API and Server.

-Server is Demon process, receiving docker API request, managing docker resources such as image, network, container, and volume. Server can communicate with other Demon to manage Docker service.

-REST API is provided by Docker engine, while Client communicates with and control Demon. It is accessible from all HTTP clients.

Docker Technology: Construction of Independent Implementation Environment and Resolution of Host System Reliance

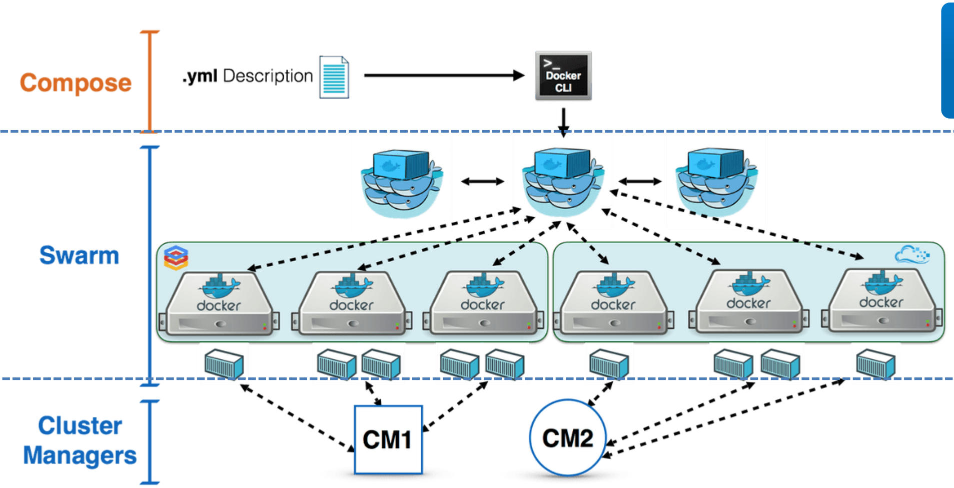


Efficient in Resource Utlization(CPU, Memory, Storage)

Convenient in operating multi-networks

Simple Re-packaging and redistribution

Relatively easy replacement



File based Installation and CLI implementation environment

Docker Multi-implementation Network

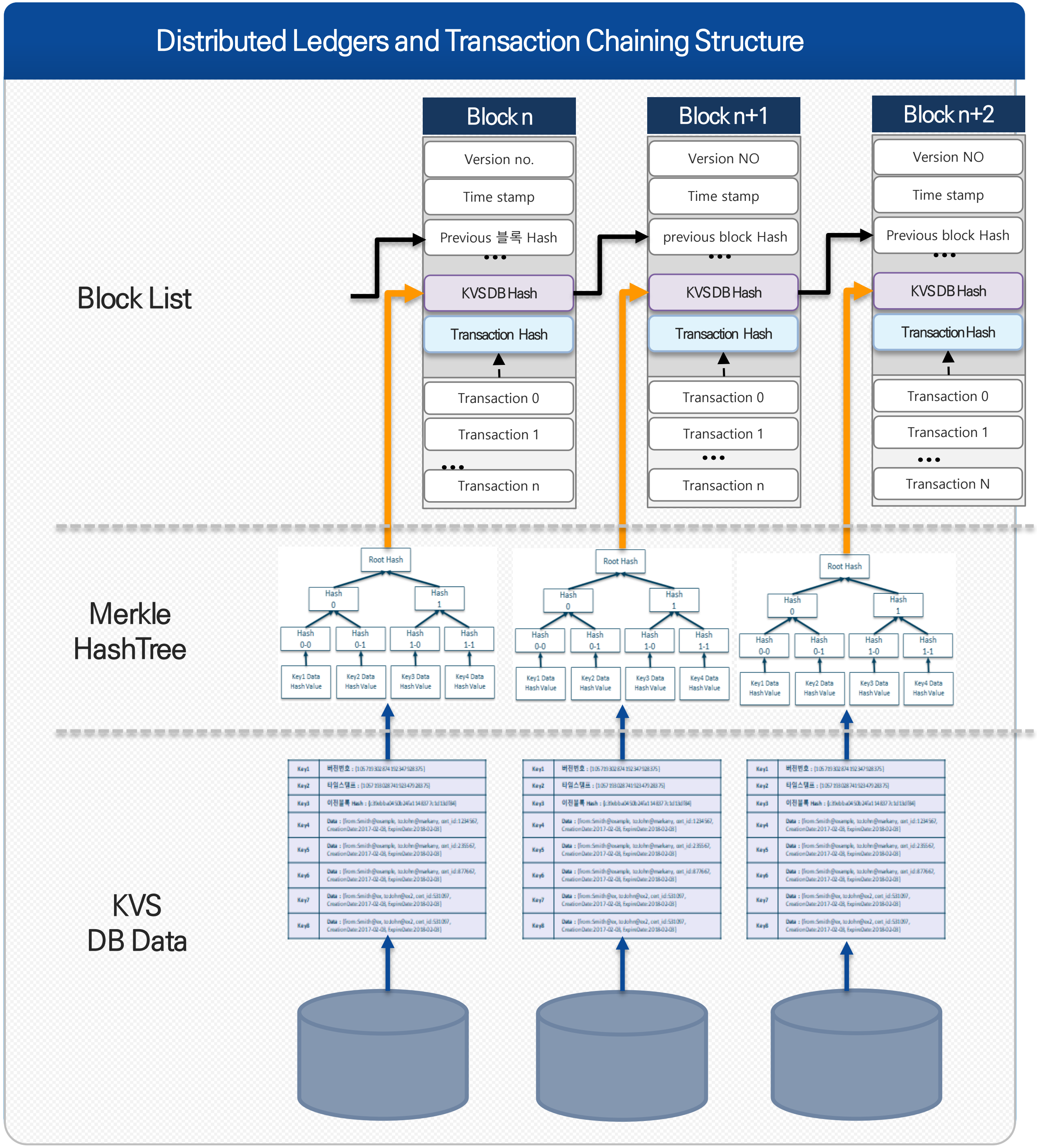
Docker Cluster MGT

Components

Node | Channel | Smart Contract | DL MGT | Consensus | Transfer MGT | Component

Distributed Ledger

- Blocklist contains basic information of hash value of previous block, time stamp, version number.
- Hash values of each transaction included in each block comprises Merkle Hash Tree and record value of Root Hash.
- Each transaction data is stored in NoSQL Database Key-Value Storage DB for fast processing



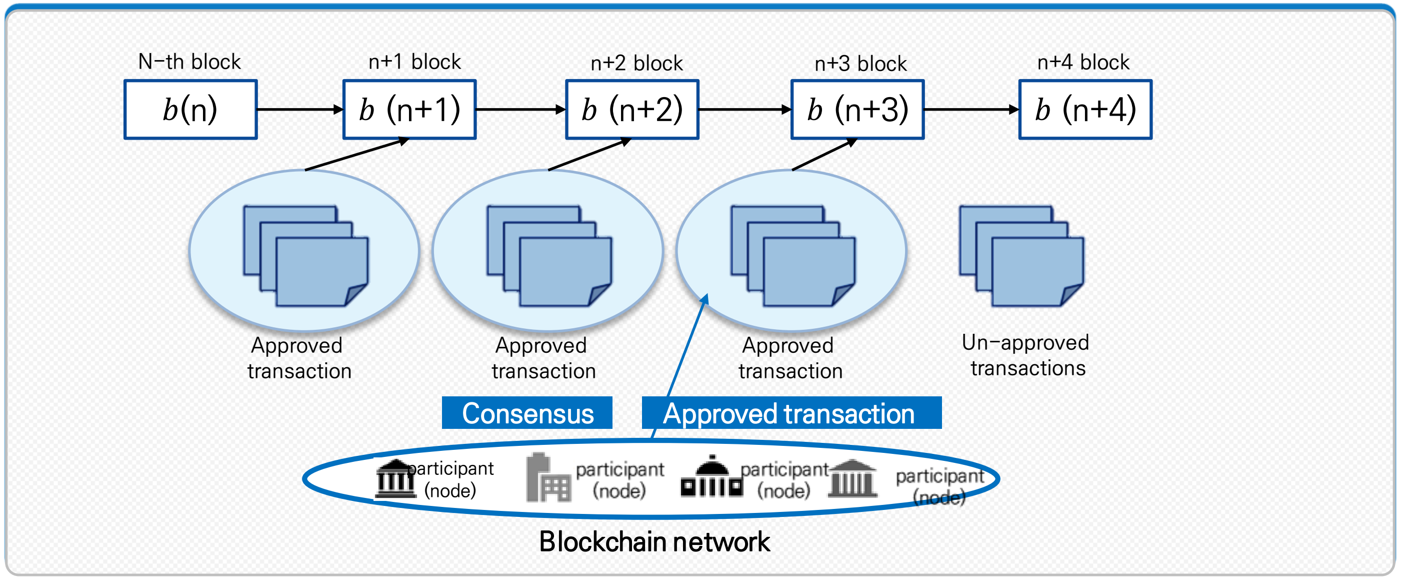
Distributed Ledgers and Transaction Chaining Structure

Key1	버전번호 : [105 719 302 874 192 347 928 375]
Key2	타임스탬프 : [1057 193 028 741 923 479 283 75]
Key3	이전블록 Hash : [c30ebba0450b24fa114837 7c1d13df84]
Key4	Data : [from:Smith@example, to:John@markany, cert_id:1234567, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key5	Data : [from:Smith@example, to:John@markany, cert_id:235567, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key6	Data : [from:Smith@example, to:John@markany, cert_id:877667, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key7	Data : [from:Smith@xx, to:John@xx2, cert_id:531097, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key8	Data : [from:Smith@xx, to:John@xx2, cert_id:531097, CreationDate:2017-02-08, ExpireDate:2018-02-08]

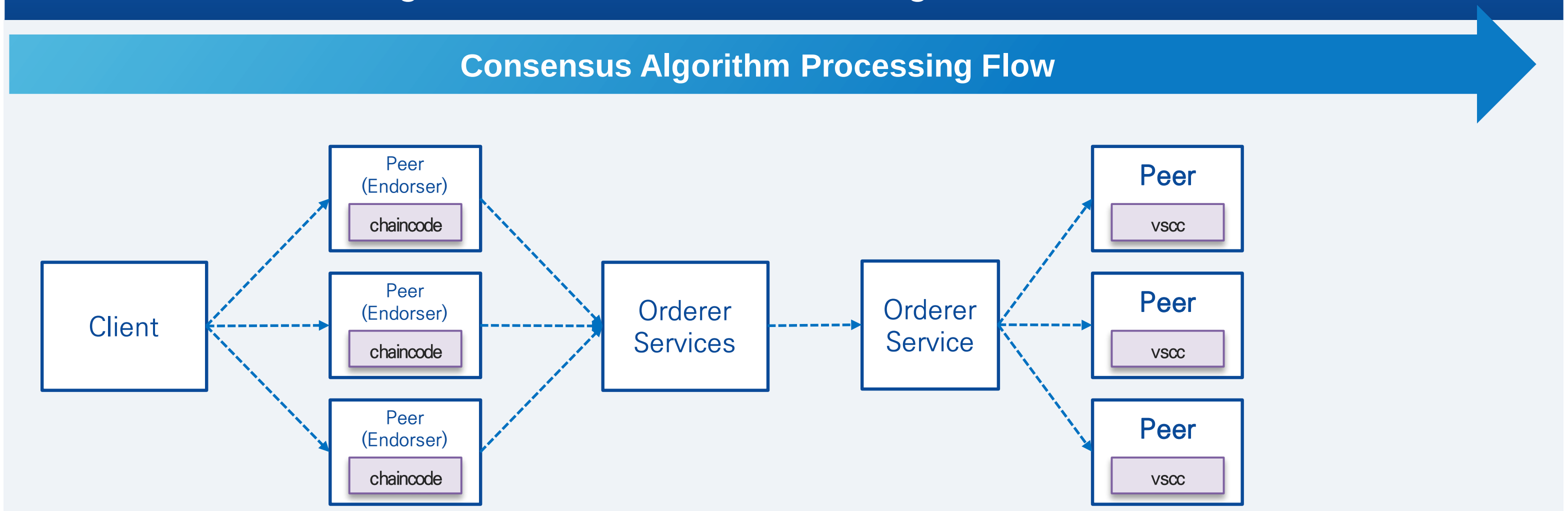
Key1	버전번호 : [105 719 302 874 192 347 928 375]
Key2	타임스탬프 : [1057 193 028 741 923 479 283 75]
Key3	이전블록 Hash : [c30ebba0450b24fa114837 7c1d13df84]
Key4	Data : [from:Smith@example, to:John@markany, cert_id:1234567, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key5	Data : [from:Smith@example, to:John@markany, cert_id:235567, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key6	Data : [from:Smith@example, to:John@markany, cert_id:877667, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key7	Data : [from:Smith@xx, to:John@xx2, cert_id:531097, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key8	Data : [from:Smith@xx, to:John@xx2, cert_id:531097, CreationDate:2017-02-08, ExpireDate:2018-02-08]

Key1	버전번호 : [105 719 302 874 192 347 928 375]
Key2	타임스탬프 : [1057 193 028 741 923 479 283 75]
Key3	이전블록 Hash : [c30ebba0450b24fa114837 7c1d13df84]
Key4	Data : [from:Smith@example, to:John@markany, cert_id:1234567, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key5	Data : [from:Smith@example, to:John@markany, cert_id:235567, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key6	Data : [from:Smith@example, to:John@markany, cert_id:877667, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key7	Data : [from:Smith@xx, to:John@xx2, cert_id:531097, CreationDate:2017-02-08, ExpireDate:2018-02-08]
Key8	Data : [from:Smith@xx, to:John@xx2, cert_id:531097, CreationDate:2017-02-08, ExpireDate:2018-02-08]

Consensus Algorithm



Consensus Algorithm = Endorsement + Ordering + Validation



1. client's transaction offer

2. Endorse PEER implement Chaincode, and Sign the result at RW set RWSet

3. According to endorsement policy, CLIENT collect endorsement results and submit transaction

4. Orderer service decides transaction sequence and creates one list

5. Confirm before each PEER confirms transaction
   - ✓ Whether condition of endorsement policy is satisfied
   - ✓ Whether there are conflicts between transactions MVCC)

- In Bright e-mail system, consensus might be minimized, requiring membership verification and content hash checking.

- Endorsement, Ordering, and Verification can be simplified into Request, Delivery, and Confirmation.

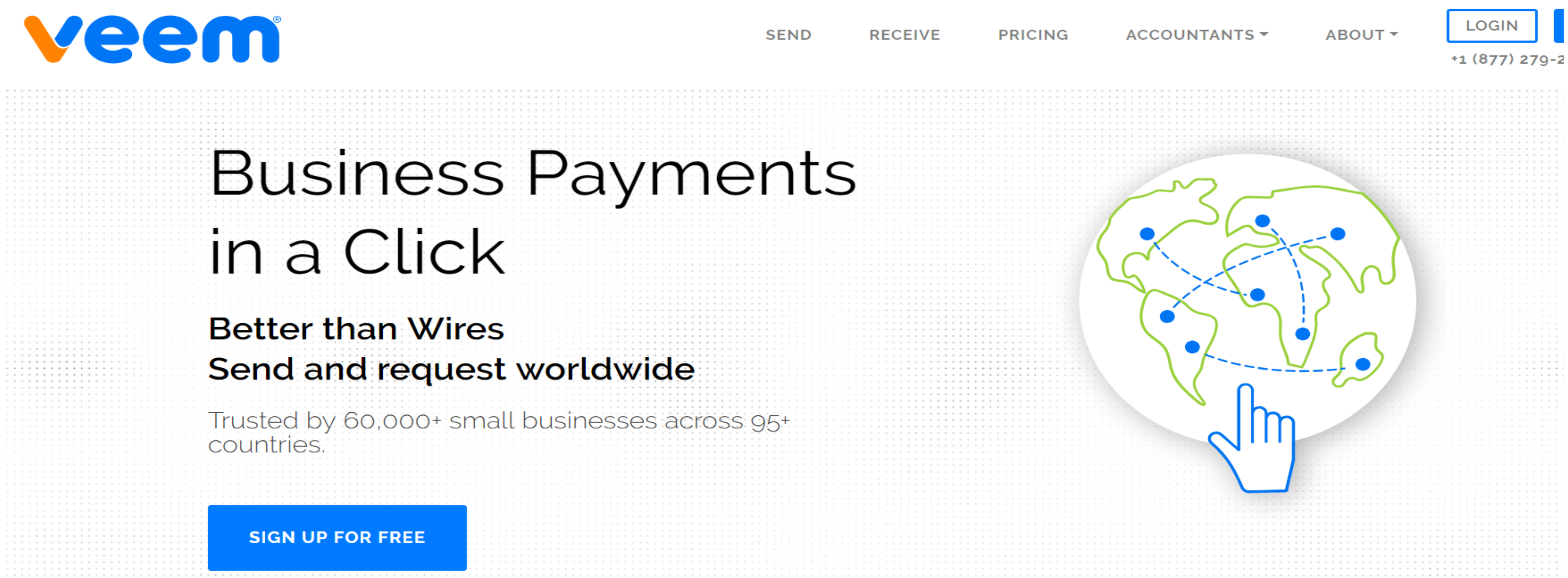
Contents

1. **Hyperledger Fabric Blockchain**

2. **Bright Mail System**

3. **Technical Issues**

AlignCommerce: Sending money to Foreigners

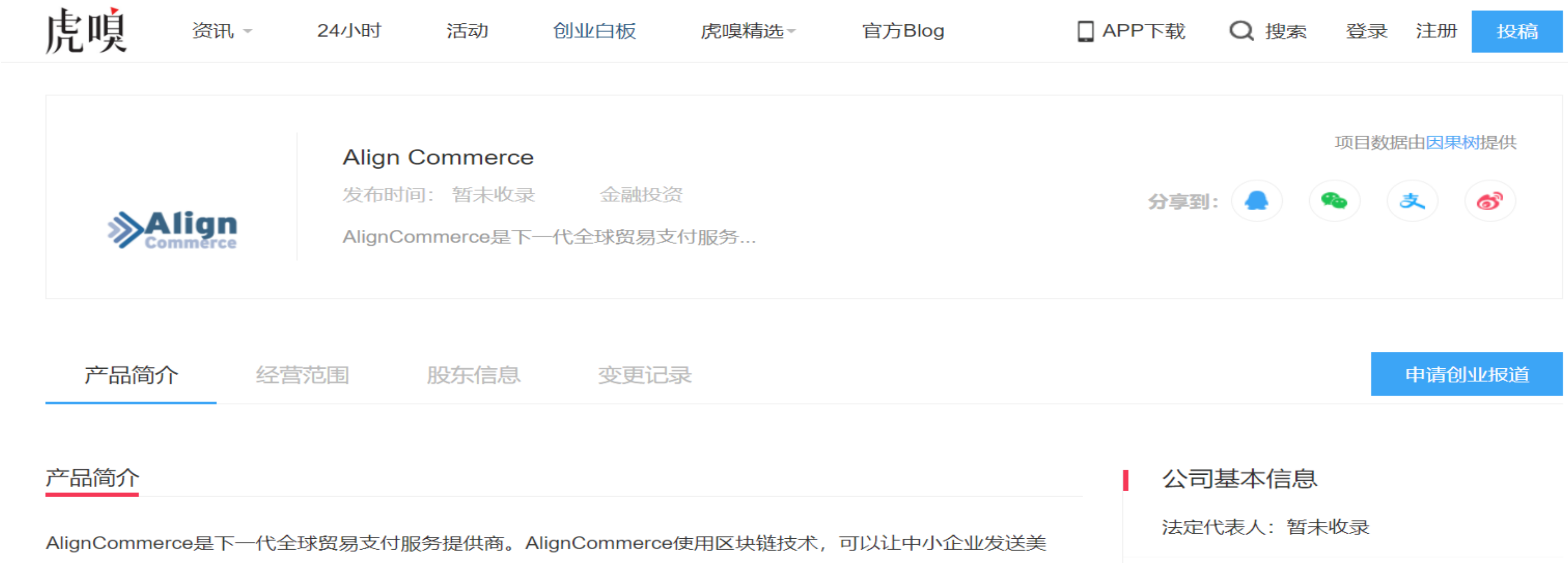


Business Payments in a Click

Better than Wires
Send and request worldwide

Trusted by 60,000+ small businesses across 95+ countries.

SIGN UP FOR FREE



Align Commerce

发布时间：暂未收录　　金融投资

AlignCommerce是下一代全球贸易支付服务...

项目数据由因果树提供

产品简介

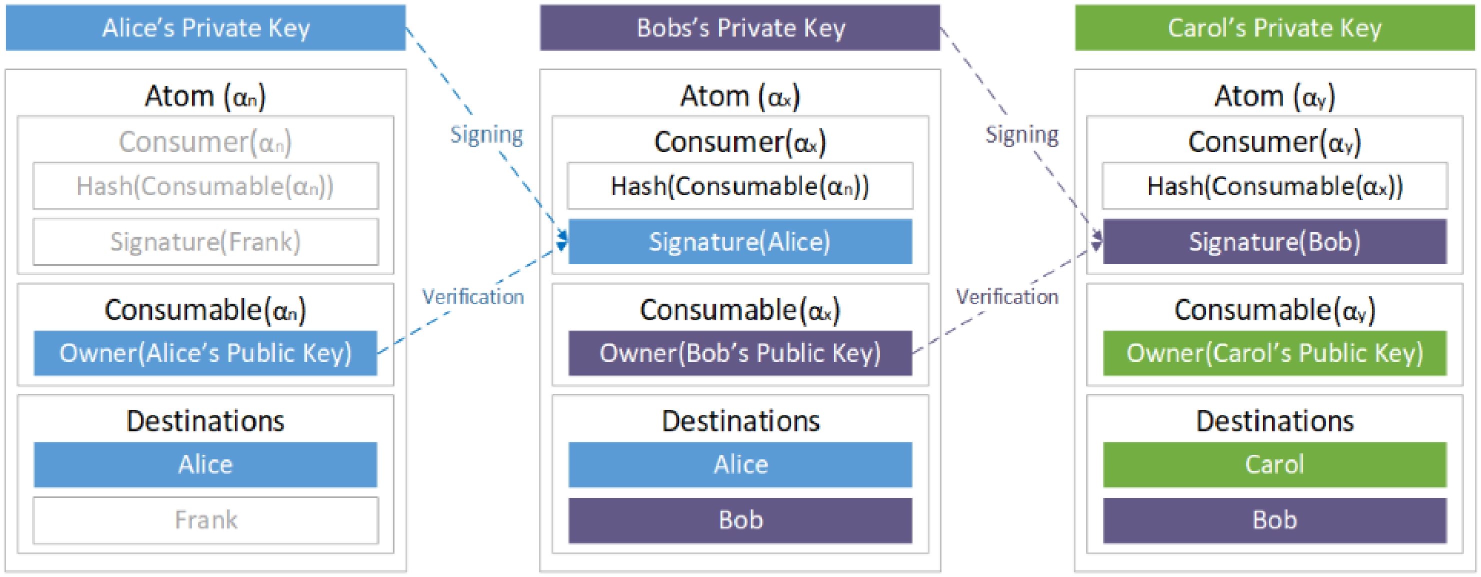
AlignCommerce是下一代全球贸易支付服务提供商。AlignCommerce使用区块链技术，可以让中小企业发送美

公司基本信息

法定代表人：暂未收录

- **Radix Tempo**: a peer-to-peer network of nodes with logical clocks to generate a temporal proof of the chronological order of events

- 3 Components: (1) A networked cluster of nodes (2)A global ledger database distributed across the nodes (3)An algorithm for generating a cryptographically secure record of temporally ordered events.



Alice's Private Key

Atom (α_n)

Consumer(α_n)

Hash(Consumable(α_n))

Signature(Frank)

Consumable(α_n)

Owner(Alice's Public Key)

Destinations

Alice

Frank

Signing

Verification

Bobs's Private Key

Atom (α_x)

Consumer(α_x)

Hash(Consumable(α_n))

Signature(Alice)

Consumable(α_x)

Owner(Bob's Public Key)

Destinations

Alice

Bob

Signing

Verification

Carol's Private Key

Atom (α_y)

Consumer(α_y)

Hash(Consumable(α_x))

Signature(Bob)

Consumable(α_y)

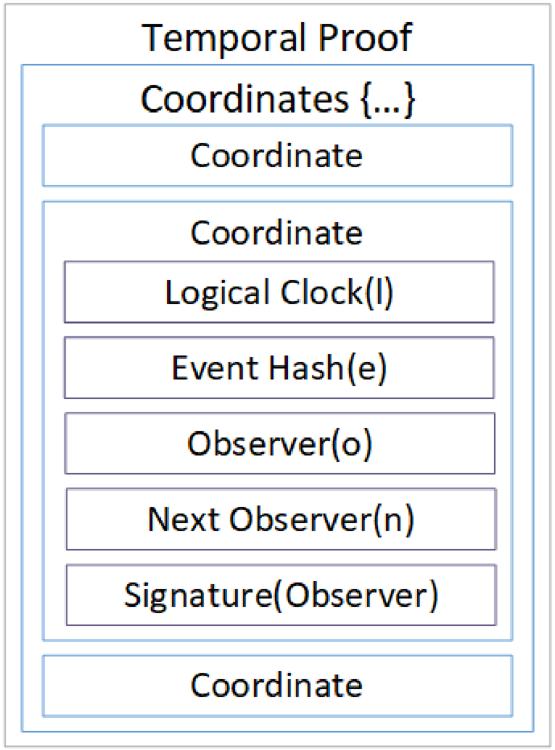
Owner(Carol's Public Key)

Destinations

Carol

Bob

: Ownership Transfer



Temporal Proof

Coordinates {…}

Coordinate

Coordinate

Logical Clock(l)

Event Hash(e)

Observer(o)

Next Observer(n)

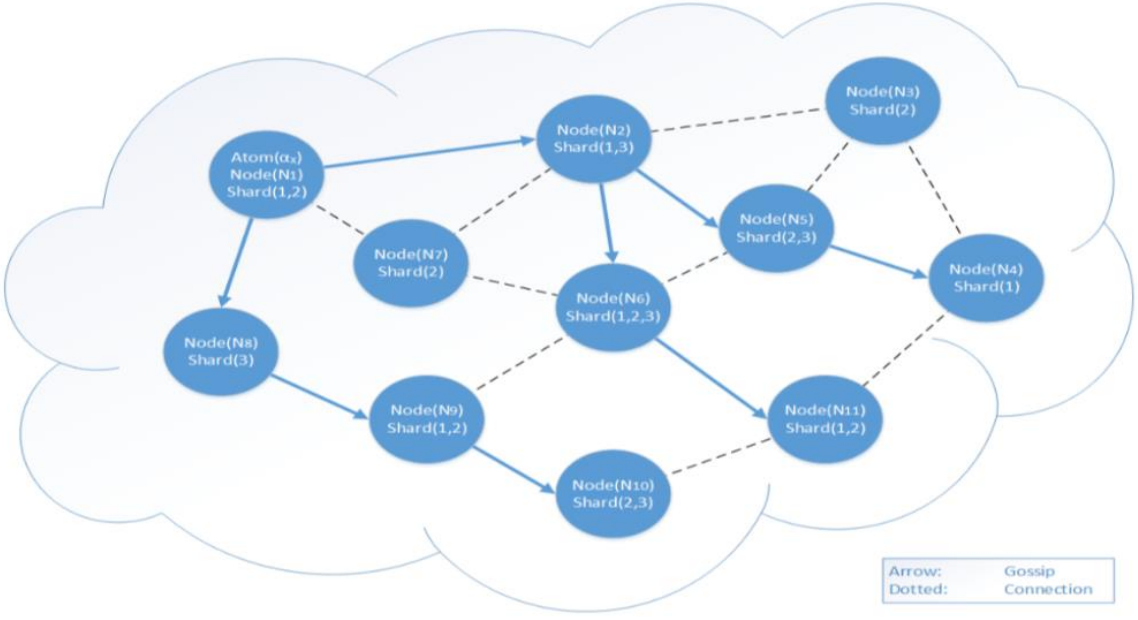
Signature(Observer)

Coordinate

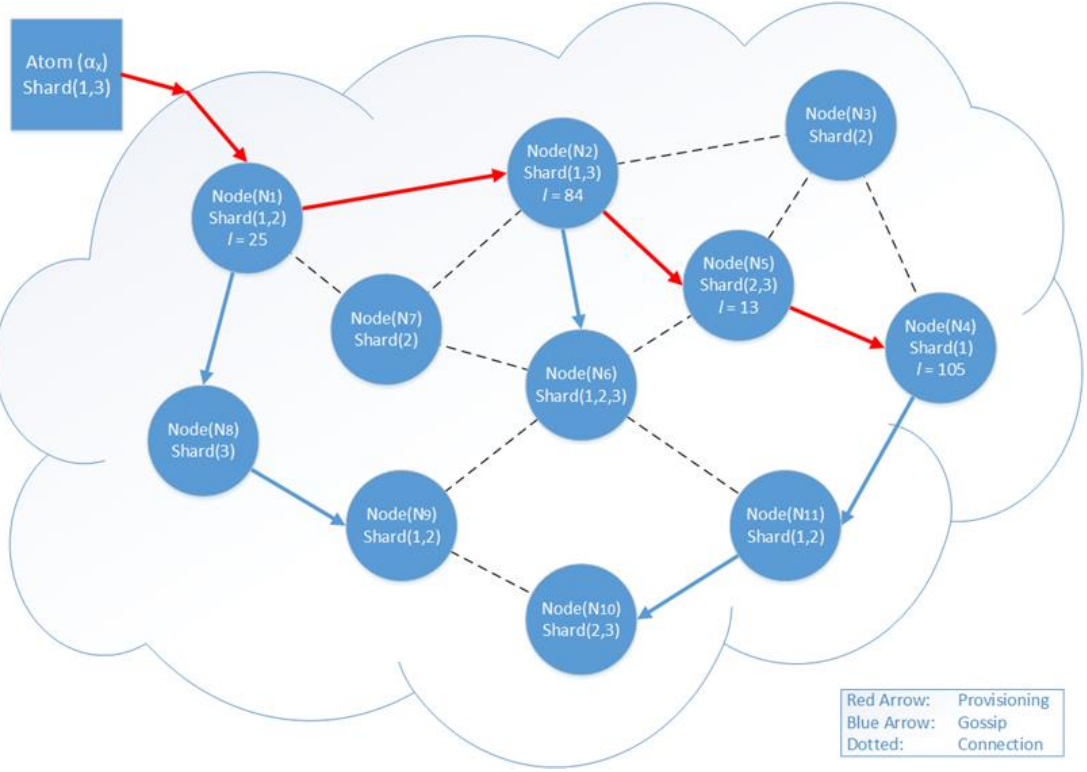
Temporal proof provisioning:

(1)append its (l,e,o,n) coordinate and signature to the Temporal Proof and transmit Atom(αX) and the Proof to the next node.

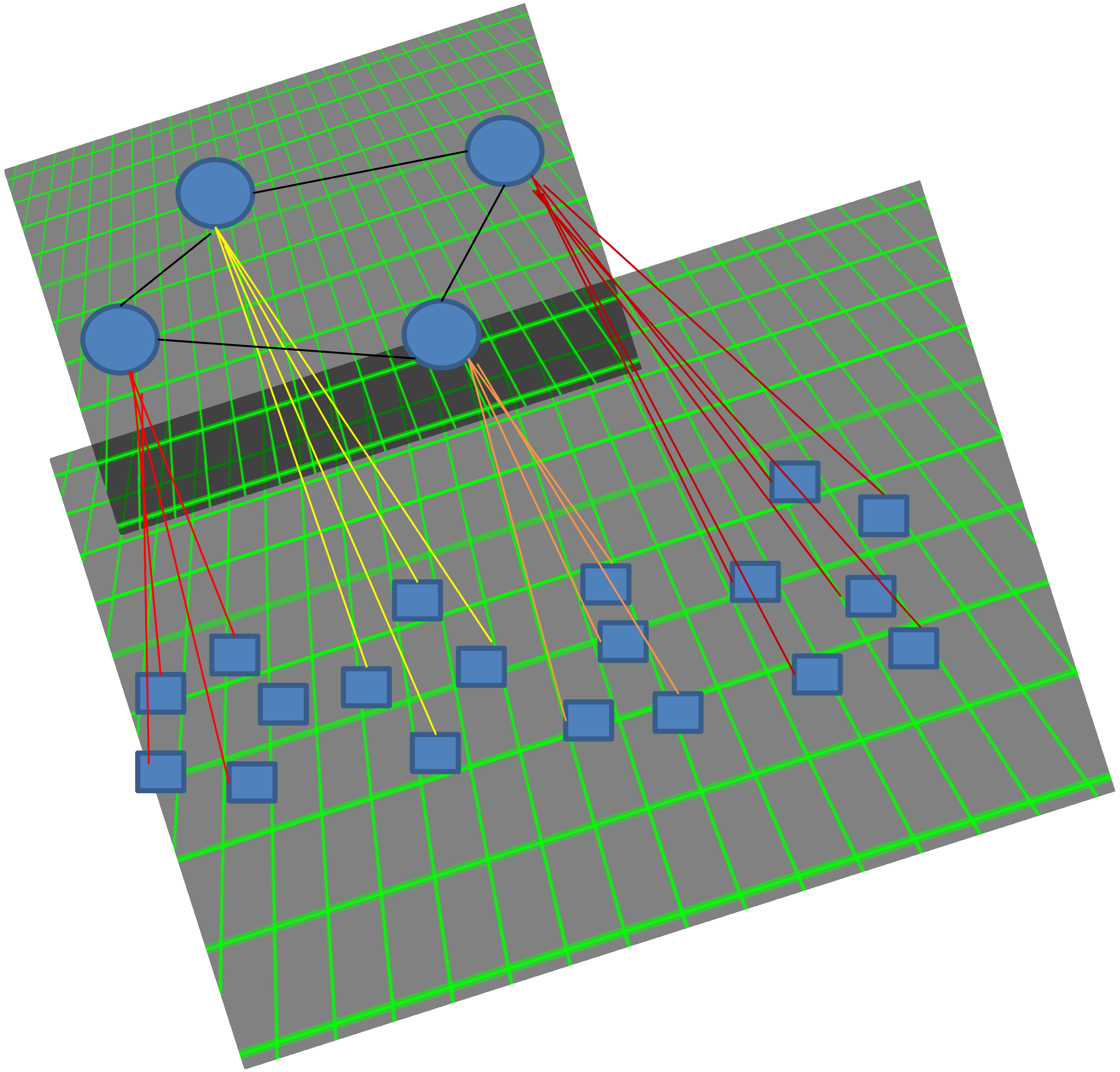
(2)a provable discrepancy is discovered by any node involved in the process.



Gossip of Atom α_X targeting Shards$(1,3)$



Sharding in Stratified Networks



Thank you