CargoSmart.ai

打造航运联盟链

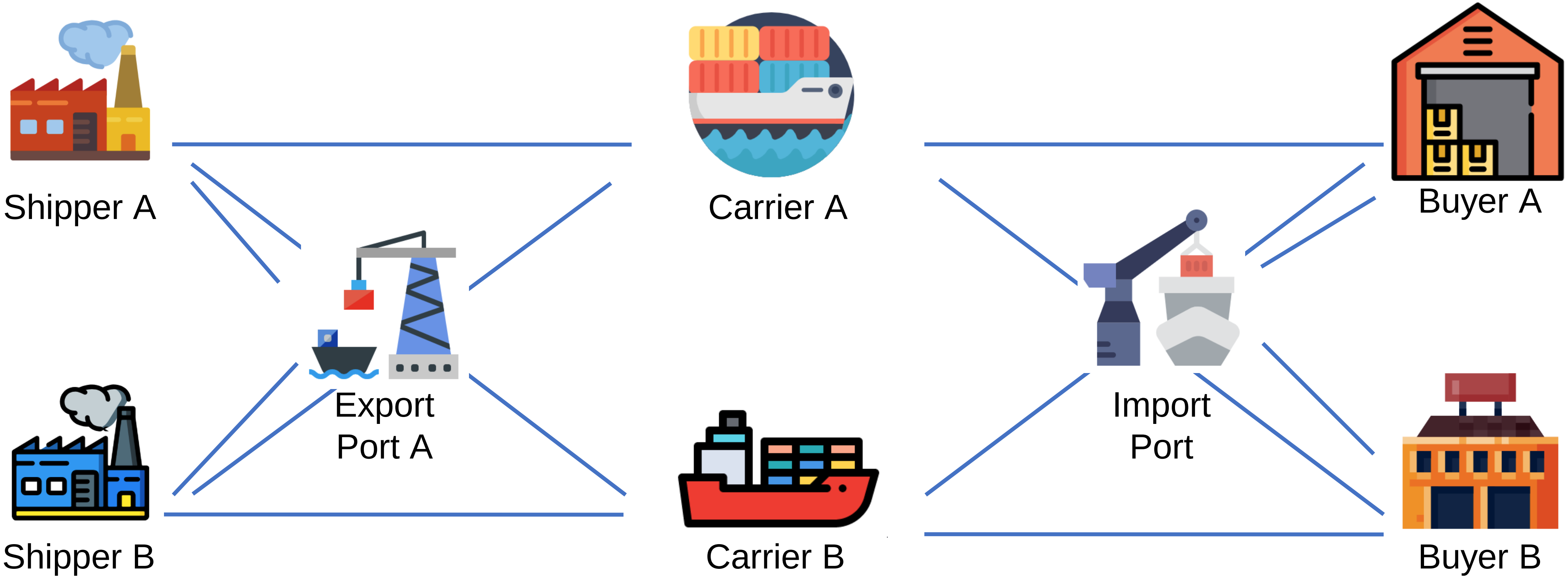
by 陈鹄 (Ken Chen)

Nov 18, 2018

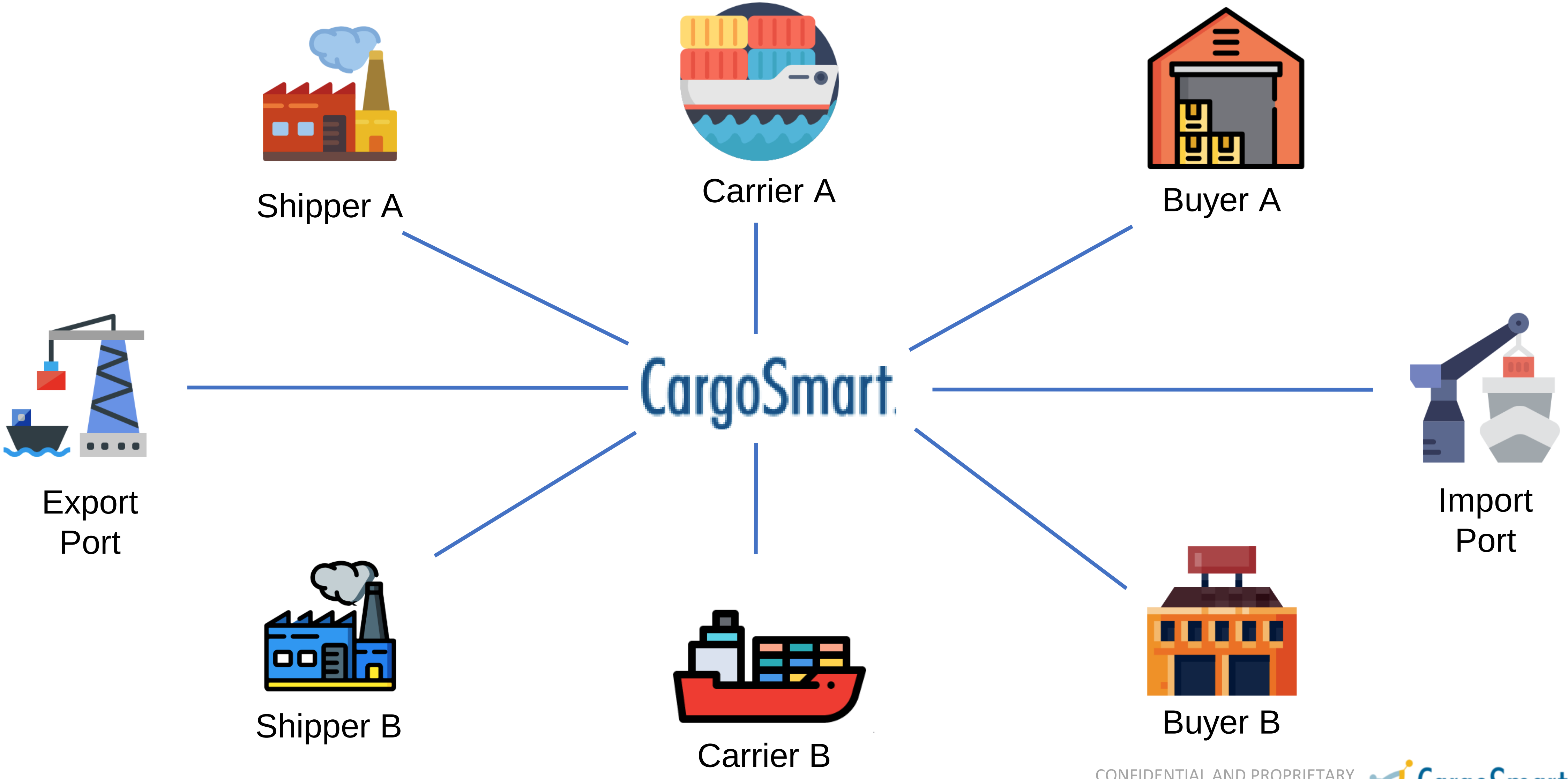
Agenda

- Shipping Reality & Vision of GSBN
- Blockchain Types and Frameworks
- Hyperledger Fabric & Data Segregation
- GSBN Design Consideration

Shipping Reality - EDI (Electronic Data Interchange)



Shipping Reality - CargoSmart B2B



WORLD SHIPPING SUMMIT 2018

6TH NOVEMBER
SHANGHAI NATIONAL EXHIBITION
AND CONVENTION CENTER

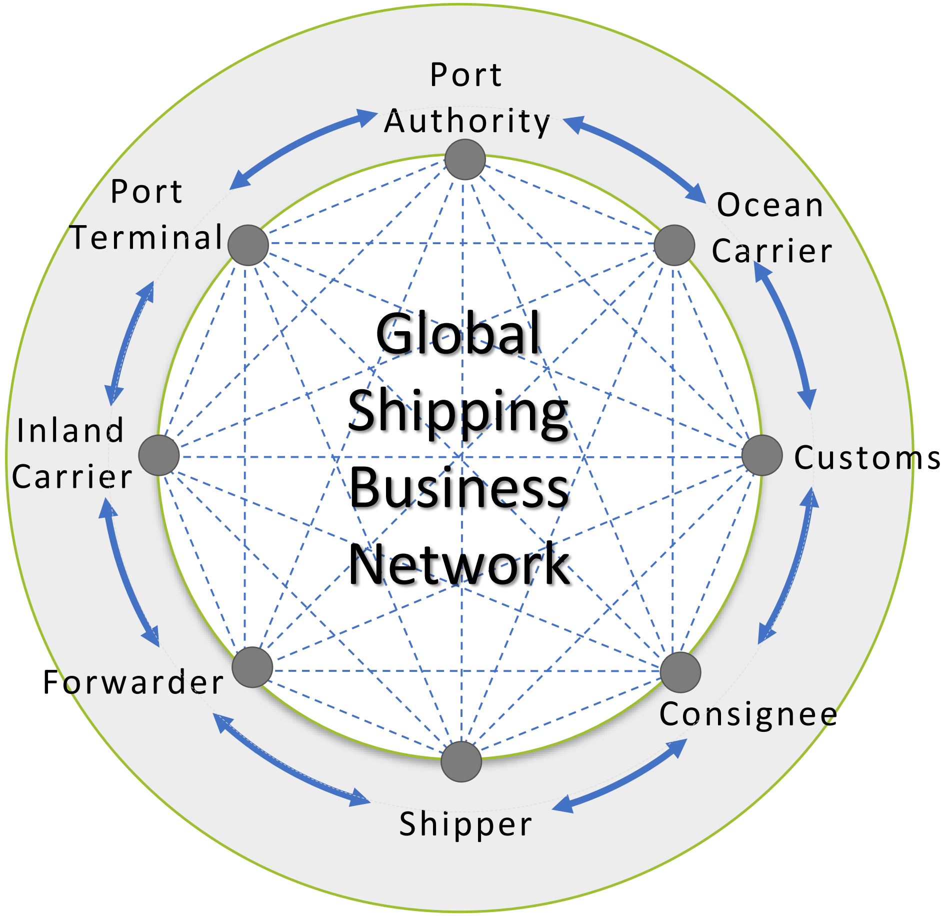
Image source: http://en.coscoshipping.com/art/2018/9/6/art_6923_80963.html

Global Shipping Business Network (GSBN)

ON-SITE SUPPORTING EVENTS
COSCO SHIPPING
中国远洋海运集团战略客户签约仪式
COSCO SHIPPING GROUP STRATEGIC CUSTOMERS AGRE
AND WORLD SHIPPING SUMMIT RECEPT
2018年11月6日
November 6th, 2018
中国 · 上海
Shanghai, Ch

Global Shipping Business Network - Vision

“Creating a digital baseline for the future of Shipping”

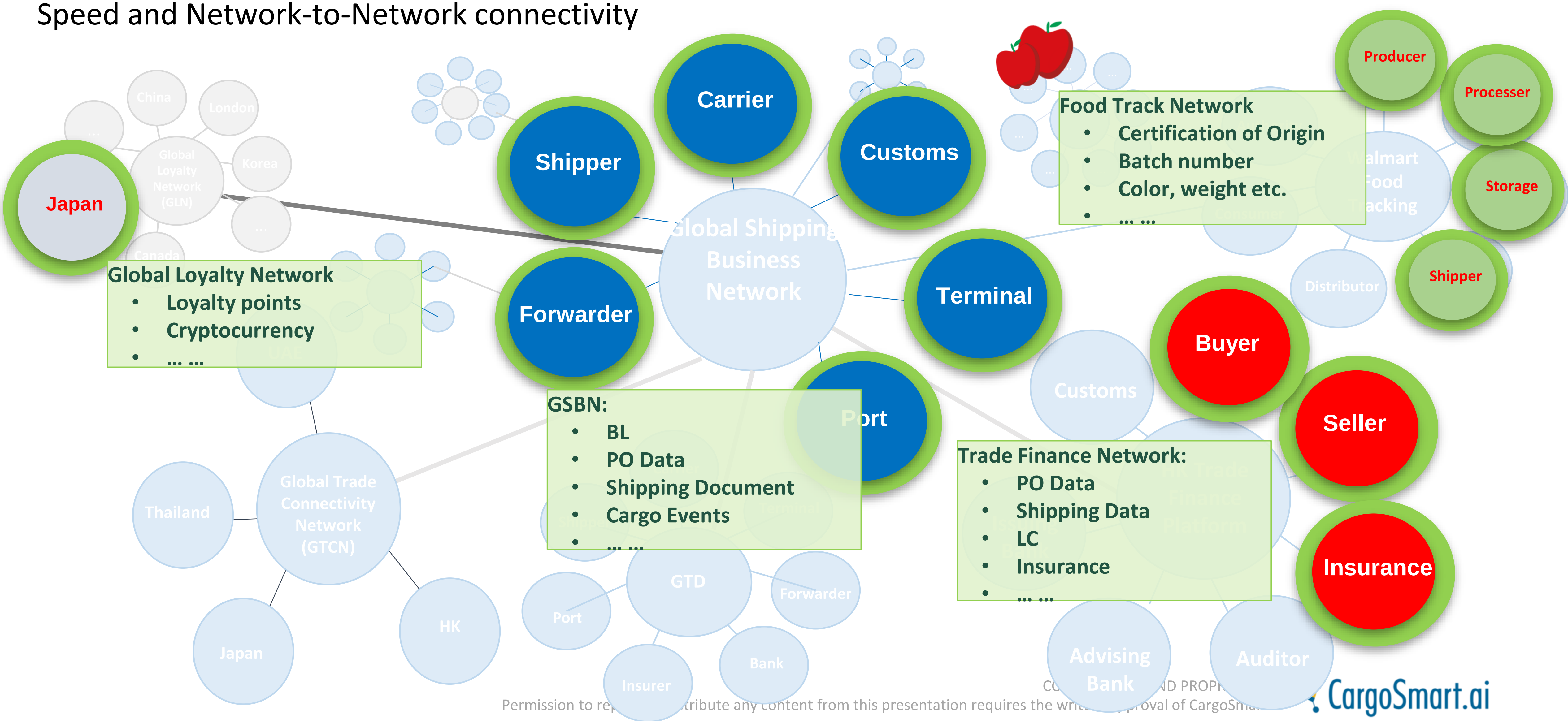


Key Features

- Industry standard schema
  - De-centralized trust
  - Enabler for digital transformation
- Peer-to-Peer networking
  - New way of connecting - *Speed*
  - Enabler for cross-industry value creation
- Open and Extensible
  - Consortium governance
  - Industry work group setup

Business Social Network Effect

Speed and Network-to-Network connectivity



Global Loyalty Network
- Loyalty points
- Cryptocurrency
- … …

Food Track Network
- Certification of Origin
- Batch number
- Color, weight etc.
- … …

GSBN:
- BL
- PO Data
- Shipping Document
- Cargo Events
- … …

Trade Finance Network:
- PO Data
- Shipping Data
- LC
- Insurance
- … …

CargoSmart.ai

Blockchain Types and Frameworks

Permissionless vs Permissioned Blockchain

- Permissionless (公链)
  - E.g. Bitcoin, Ethereum
  - Anyone can join; Publicly accessible data
  - Large-scale distributed ledger; Slower transaction confirmation
  - Incentive / Abuse Prevention Policy (e.g. Mining; Transaction Fee)

- Permissioned (私链，联盟链)
  - E.g. GSBN
  - Selected parties; Access control applied
  - Semi-public data; Segregation
  - Incentive policy is not necessary but can be helpful

Blockchain Types and Frameworks

Characteristics	Ethereum	R3 Corda	Hyperledger Fabric
Description of Platform	Generic blockchain platform	Specialized distributed ledger platform for financial industry	Modular blockchain platform
Governance	Ethereum developers	R3	Linux Foundation
Mode of operation	Permissionless	Permissioned	Permissioned
Consensus	POW	Validity and Uniqueness	full-circle verification of the correctness of a set of transactions comprising a block
Smart Contracts	Yes (Solidity)	Yes (Java and other JVM languages)	Yes (Go, Node.js, Java)
Currency	Ether	None	None

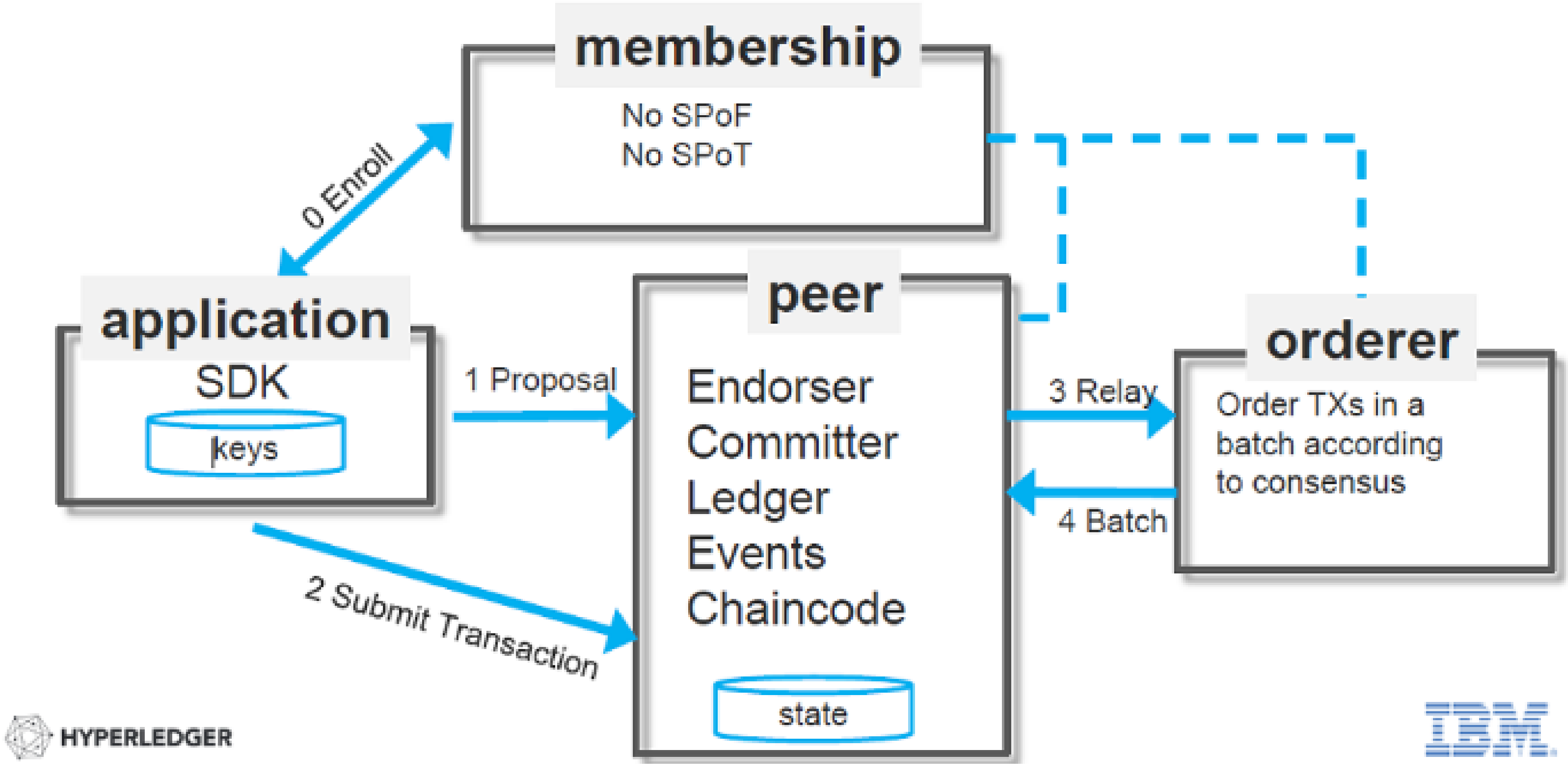
What We Choose and Why?

- Ethereum
  - Mining as consensus approach and gas concept is not suitable
  - No build-in data segregation
  - Account management cannot support org chart & ACL well
  - Solidity language has some restriction and not enough developer knows
- R3 Corda
  - Specialized in finance
  - Seems supporting group is a private company. Not large enough community.
- Hyperledger Fabric
  - Modular design, CA
  - Smart contract in Go, Node.js, JAVA
  - Stronger background supporting by IBM and Linux Foundation
  - Bigger community
  - Have cloud vendor support, such as Oracle

CargoSmart.ai

Hyperledger Fabric

Hyperledger Fabric Transaction Flow



Data Segregation (数据隔离)

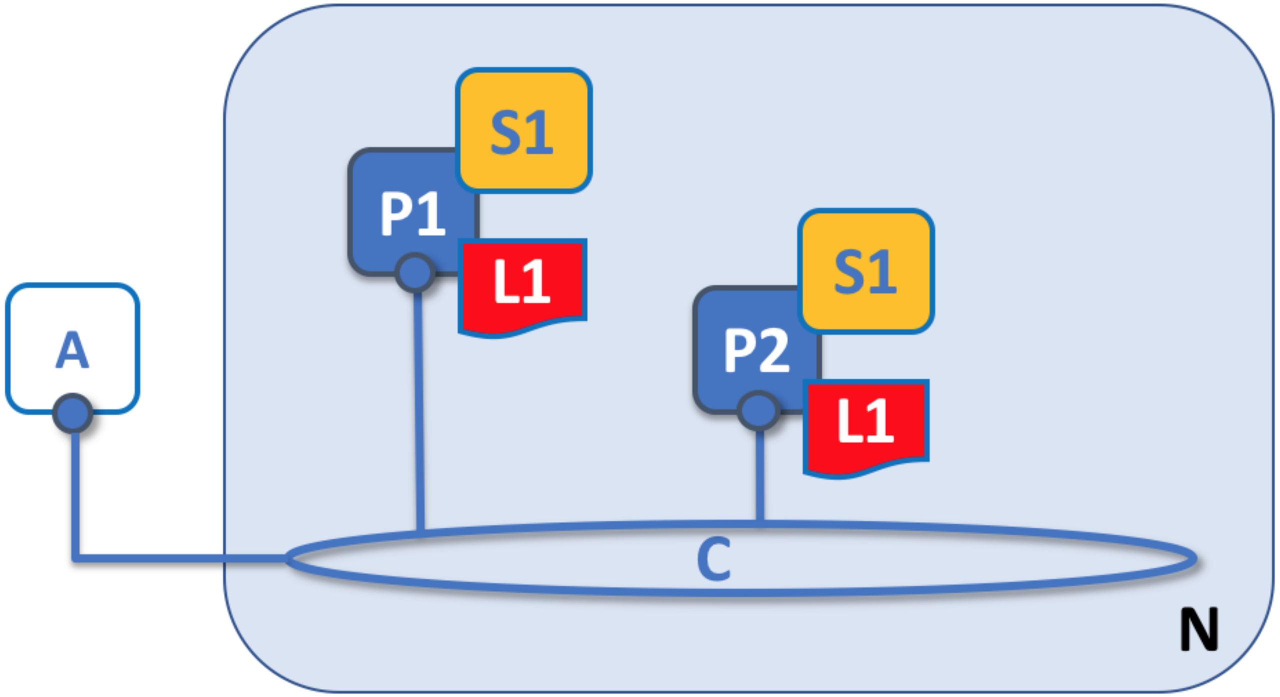
- Why?
  - Violate blockchain nature?

- Must-have item for consortium blockchain
  - Business Nature
  - Policy, e.g. General Data Protection Regulation (GDPR)
  - SaaS / PaaS

- How to do it in Hyperledger Fabric?
  - Channel
  - Private Data (Since v1.2)

Hyperledger Fabric - Channel

- a **private** “**subnet**” of communication between two or more specific network members, for the purpose of conducting private and confidential transactions.



N	Blockchain Network	L	Ledger
C	Channel	A	Application
P	Peer	PA C	Principal PA (e.g. A, P1) communicates via channel C.
S	Chaincode		

Image source: https://hyperledger-fabric.readthedocs.io/en/release-1.3/peers/peers.html

CargoSmart.ai

Hyperledger Fabric – Private Data

- where a group of organizations on a channel need to keep data **private from other organizations on that channel**
- Disseminated p2p; Kept confidential from orderer

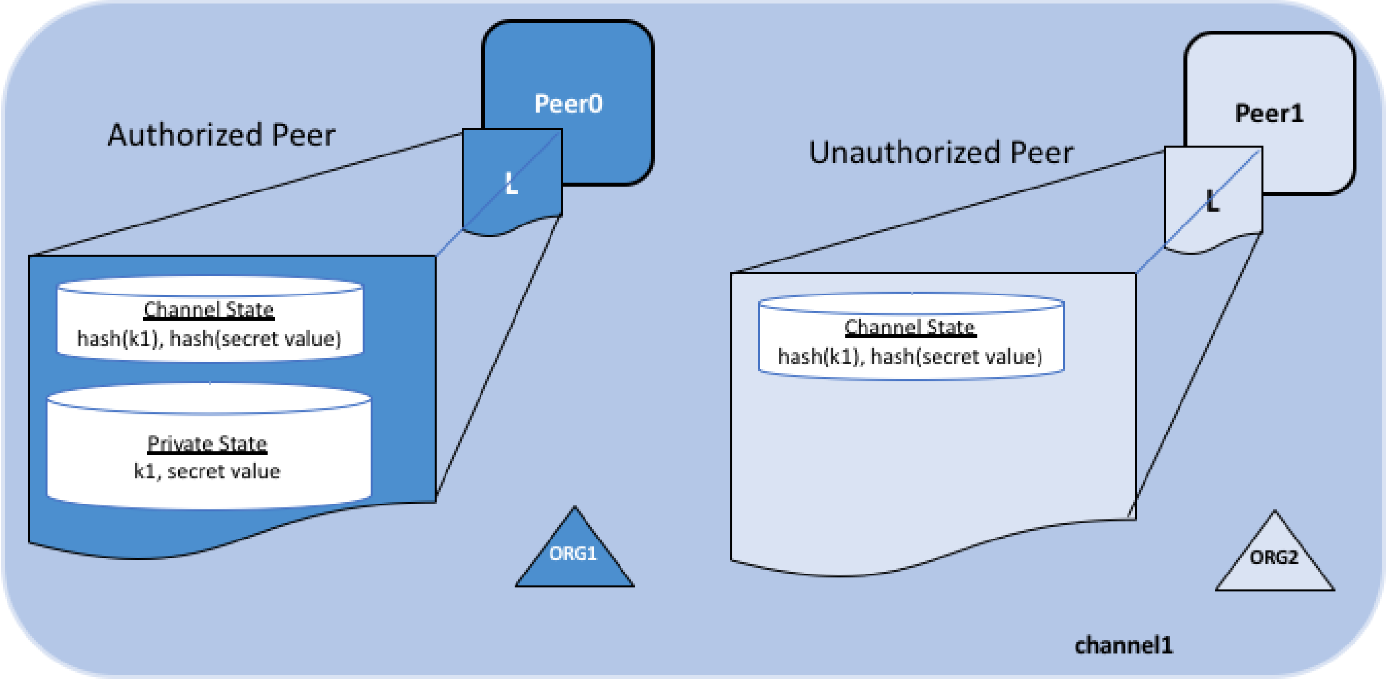


Image source: https://hyperledger-fabric.readthedocs.io/en/release-1.3/private-data/private-data.html

Channel vs Private Data

	Characteristics	Limitation / Drawback
Channel	• Entire transactions (and ledgers) are confidential • Top-level hard-segregation	• No Write Action can be done across channels • Boom if organizations are a lot, hard to maintain
Private Data	• Confidential to a subset of org in same channel • Config defined at the time chaincode instantiate or upgrade	• Privacy control is still on org, peer or client level

CargoSmart.ai

GSBN Design Consideration

GSBN Design Consideration

- Data Segregation
  - Operation Reality - Small companies can only share common node/peer
  - Business Requirement – Dynamically share asset as business process goes

- On-chain / Off-chain
  - Data characteristics
  - Policy restriction, e.g. GDPR
  - Transaction control

- PaaS / SaaS
  - SaaS - for normal member, can build app based on provided Biz API
  - PaaS - for founding member who owns network node, can provide Biz API
  - Well-established layers for different parties

Encryption

- DEK (Data Encryption Key) Key Type
  - Symmetric / Asymmetric (e.g. PKI)

- How should the DEK be shared?
  - Predetermined / Dynamically generated

- How many DEK are required?
  - Data level / Recipient level?

Sample Case and Solution

- Sample case
  - One copy of data
  - A shares to B and C

- Two major approaches
  - Multiple copies of encrypted data
  - One copy of encrypted data

- Solution
  - Multiple symmetric DEK or Asymmetric DEK
    - Multiple data
  - One DEK, multiple Asymmetric KEK (Key Encryption Key)
    - One data; Multiple encrypted key

Give and Take after using Fabric

- Only listen to the event from the org peer where user registered in:
    - *UNKNOWN: event message validation failed: [failed deserializing event creator: [expected MSP ID **xxxx**, received **yyyy**]]*

- Self-implemented MongoDB KeyValueStore
    - To store authenticated user's private keys, certificates, etc.
    - https://github.com/kenspirit/fabric-sdk-node-mongodb-kvs

Credits

- Icons in slide 3 and 4 are from Flaticon
  - Port icon are designed by geotatah
  - Crane icon, Factory icon A and Factory icon B is designed by Smashicons
  - Warehouse icon A is designed by Adib Sulthon
  - Ship icon A, Ship icon B, Warehouse icon B is designed by Freepik

Author

- 鹄思乱想
  - http://www.thinkingincrowd.me/about

CargoSmart.ai

Thank You!