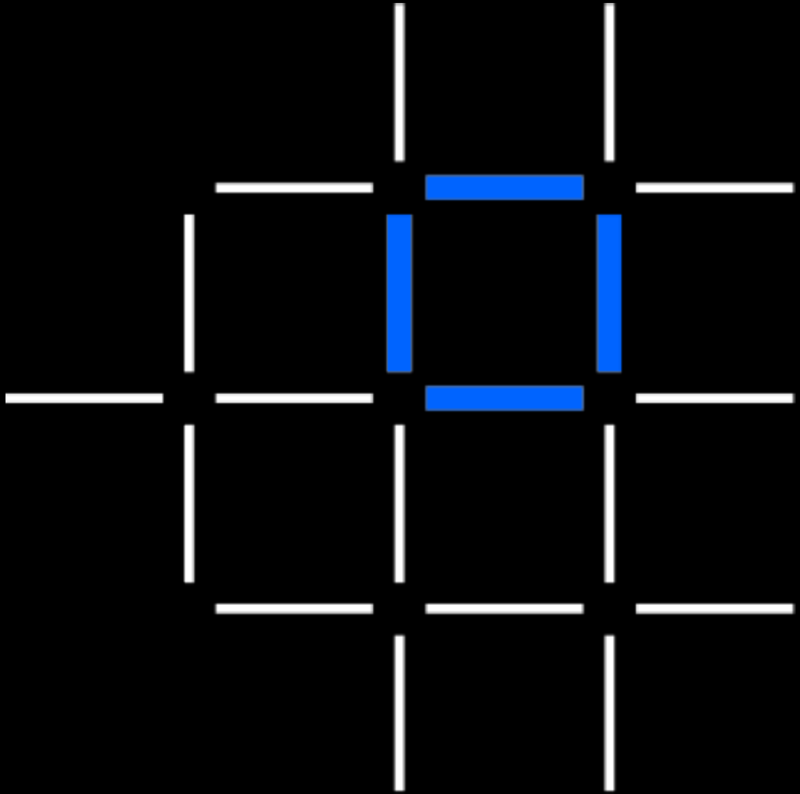
Understanding blockchain security

Unit 08



IBM Skills Academy

V1.0, July 2018

Learning objectives

Hyperledger Fabric security

Hyperledger Composer security

IBM Blockchain Platform security

Summary

What you should be able to do

Upon completion of this unit, you should be able to:

- Explain how to position blockchain security as part of a bigger enterprise context.
- Describe the key security features of the following items:
  - Hyperledger Fabric
  - Hyperledger Composer
  - IBM Blockchain Platform

Security refresher

Transport layer technology (TLS)

A protocol that provides privacy and data integrity between two communicating applications.

Lightweight directory access protocol (LDAP)

A protocol for managing information of organizations, individuals, and other resources, such as files and devices in a network, including their credentials.

Certificate authority (CA)

A trusted entity that issues electronic documents (certificates) that represents the digital identity of an entity.

Keystore

A repository of security certificates.

Certificate revocation list (CRL)

A list of digital certificates that have been revoked by the issuing CA before their scheduled expiration date. They should no longer be trusted.

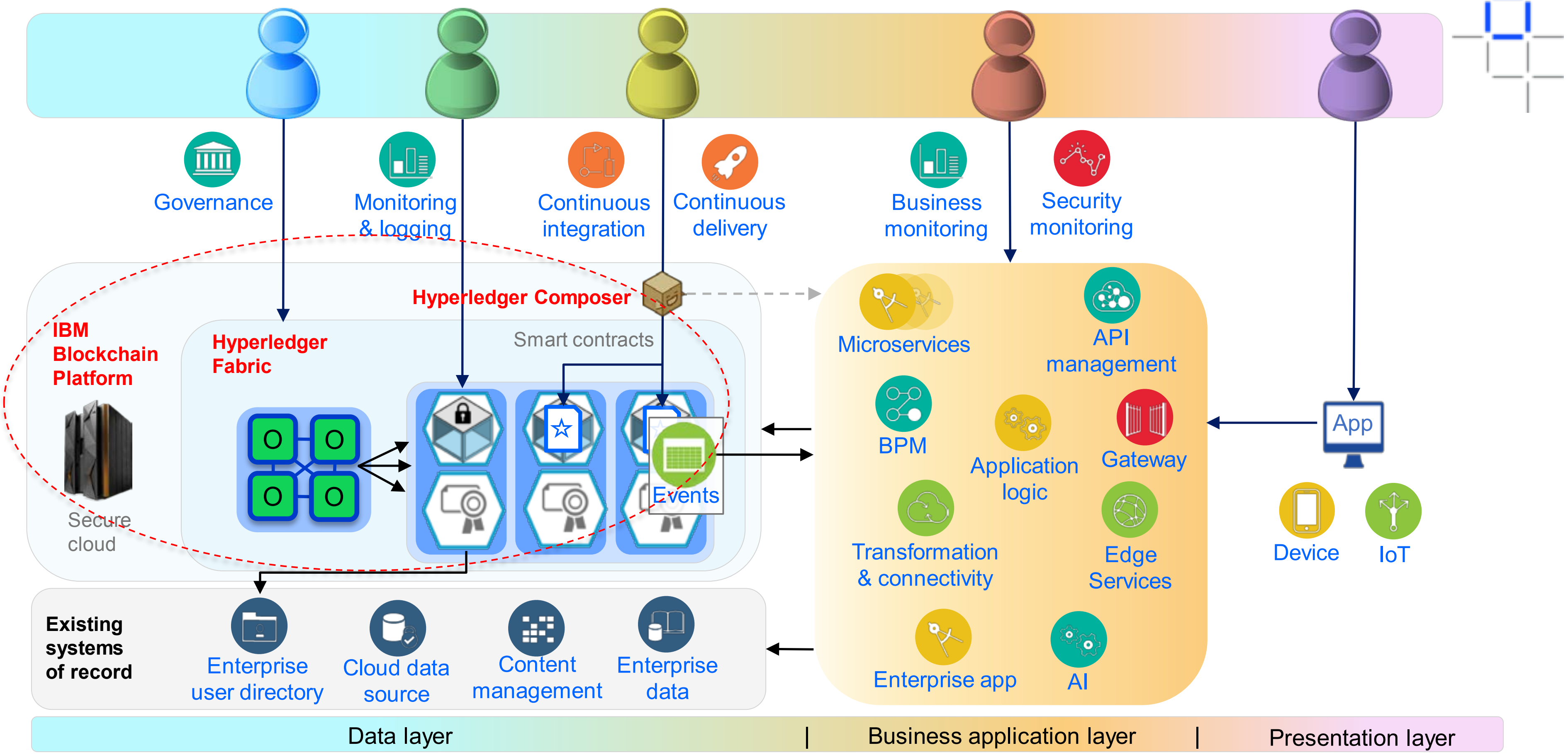
Hardware security module (HSM)

A physical computing device that manages digital keys for strong authentication and faster processing.

Open authorization (OAuth)

An open standard for token-based authentication and authorization to access URL addressable resources, which enables a user's account information to be used by third-party services without exposing the user's credentials.

Security context of enterprise blockchain



IBM **Blockchain**

Learning objectives

[Hyperledger Fabric security]

Hyperledger Composer security

IBM Blockchain Platform security

Summary

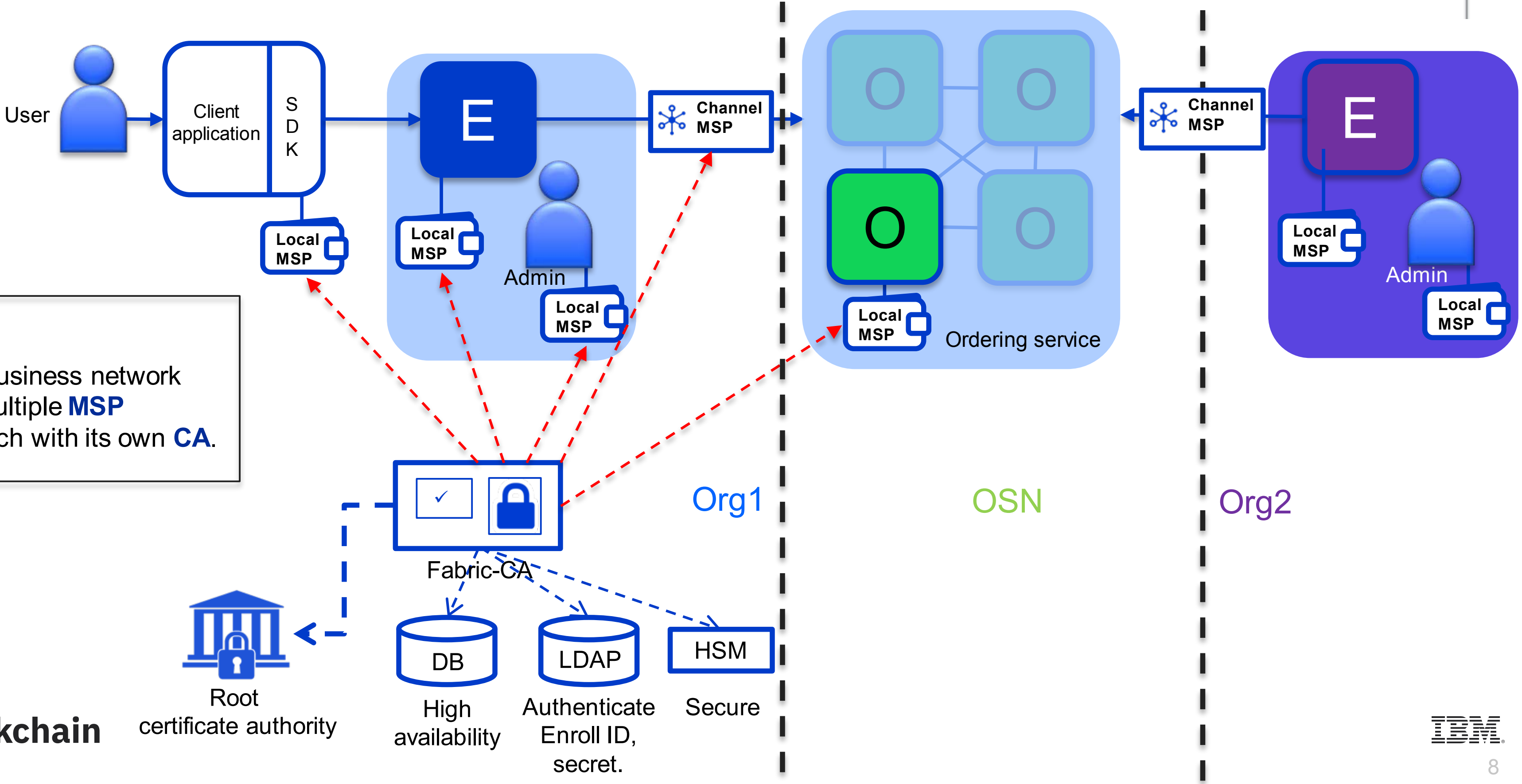
Hyperledger Fabric

Security

What to secure?	How to secure it?
Identities [users, admins, peers, and orderers]	Membership service provider, certificate authority, and Enrollment
Data at rest [transactions, ledgers, and PI/SPI]	Application-level encryption, channels, and private data collections
Data in transit [communication]	Application-level encryption
Consensus [validity]	Endorse/Order/Validate, and Transaction signing

Hyperledger Fabric

Membership service provider and certificate authority

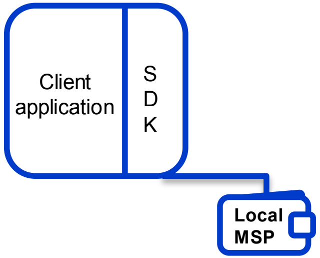


The same business network can have multiple **MSP** services, each with its own **CA**.

Hyperledger Fabric

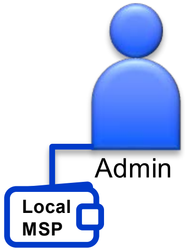
MSP - Identities

User identities



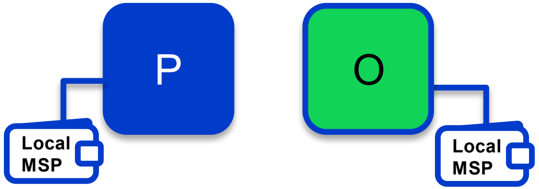
user@org1.example.com	
Keystore	<private key>
signcert	user@org1.example.com-cert.pem

Admin identities



admin@org1.example.com	
Keystore	<private key>
signcert	admin@org1.example.com-cert.pem

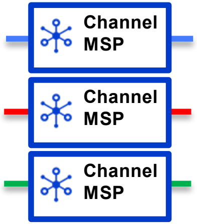
Peer and orderer identities



peer@org1.example.com	
admincerts	admin@org1.example.com-cert.pem
cacerts	ca.org1.example.com-cert.pem
Keystore	<private key>
signcert	peer@org1.example.com-cert.pem
CRLs	<List of revoked admin certificates>

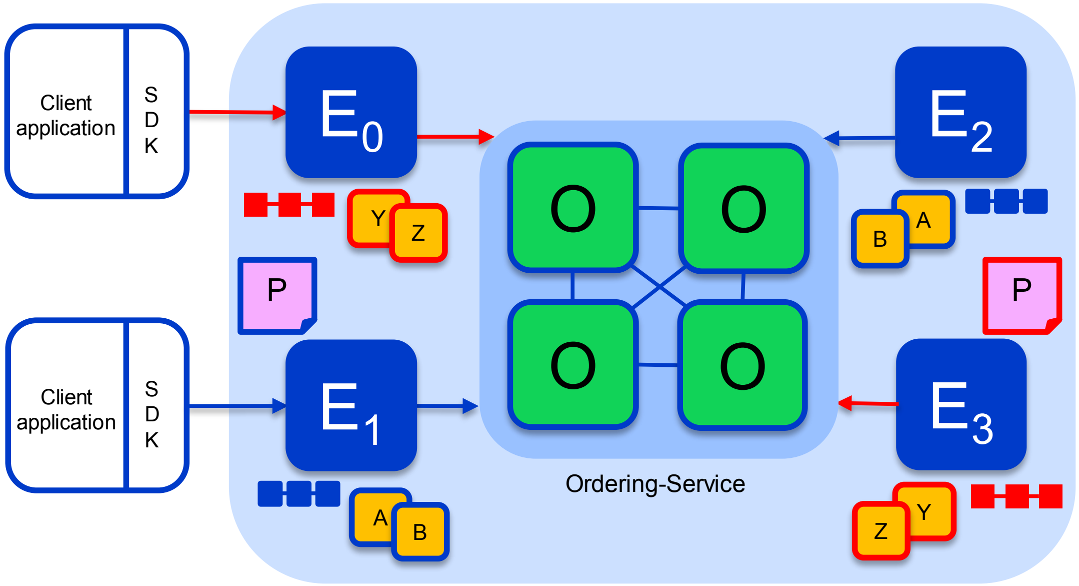
Hyperledger Fabric

MSP - Channel MSP information



Channels

ID = MSP1	
admincerts	admin.org1.example.com-cert.pem
cacerts	ca.org1.example.com-cert.pem
CRLs	<list of revoked admin certificates>

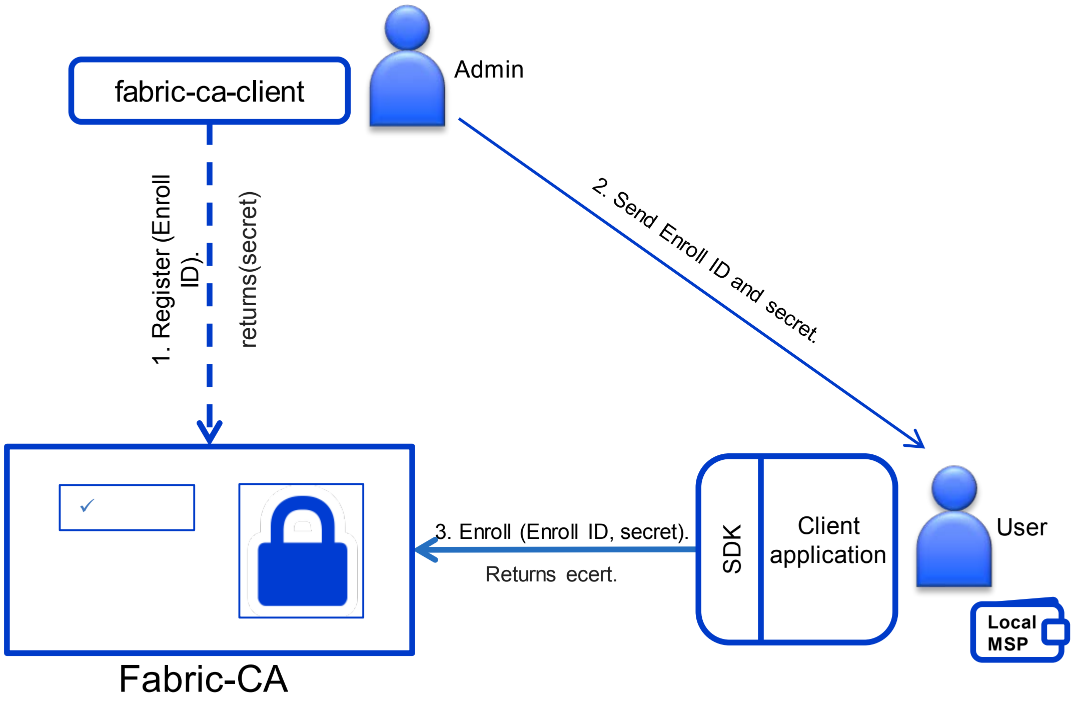


Hyperledger Fabric Network

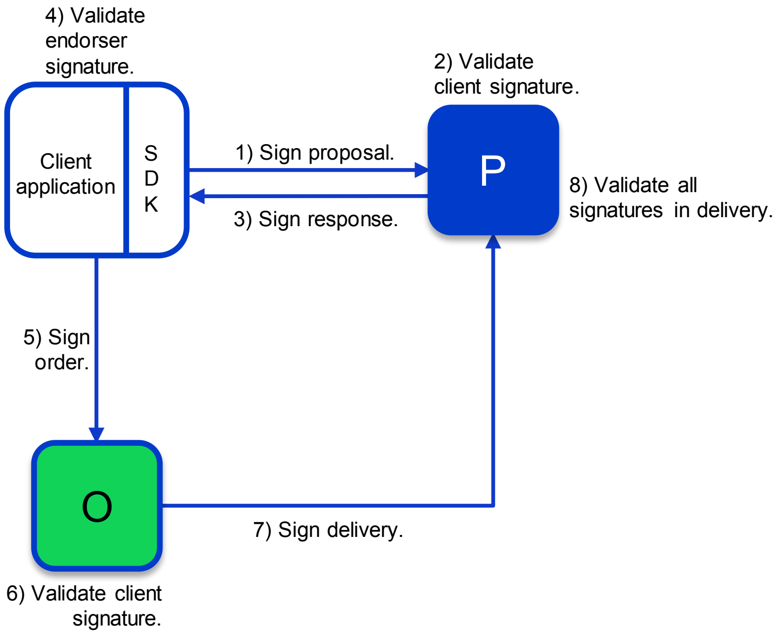
Hyperledger Fabric

MSP – Enrollment and transaction signing

New user registration and enrollment



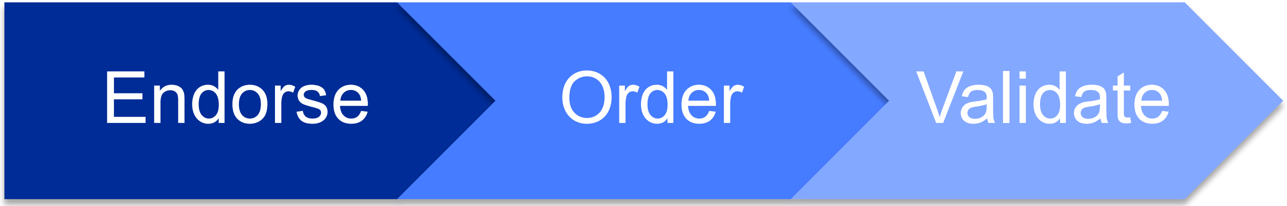
Transaction signing


4) Validate endorser signature.
Client application
SDK
1) Sign proposal.
3) Sign response.
P
2) Validate client signature.
8) Validate all signatures in delivery.
5) Sign order.
O
6) Validate client signature.
7) Sign delivery.

Hyperledger Fabric

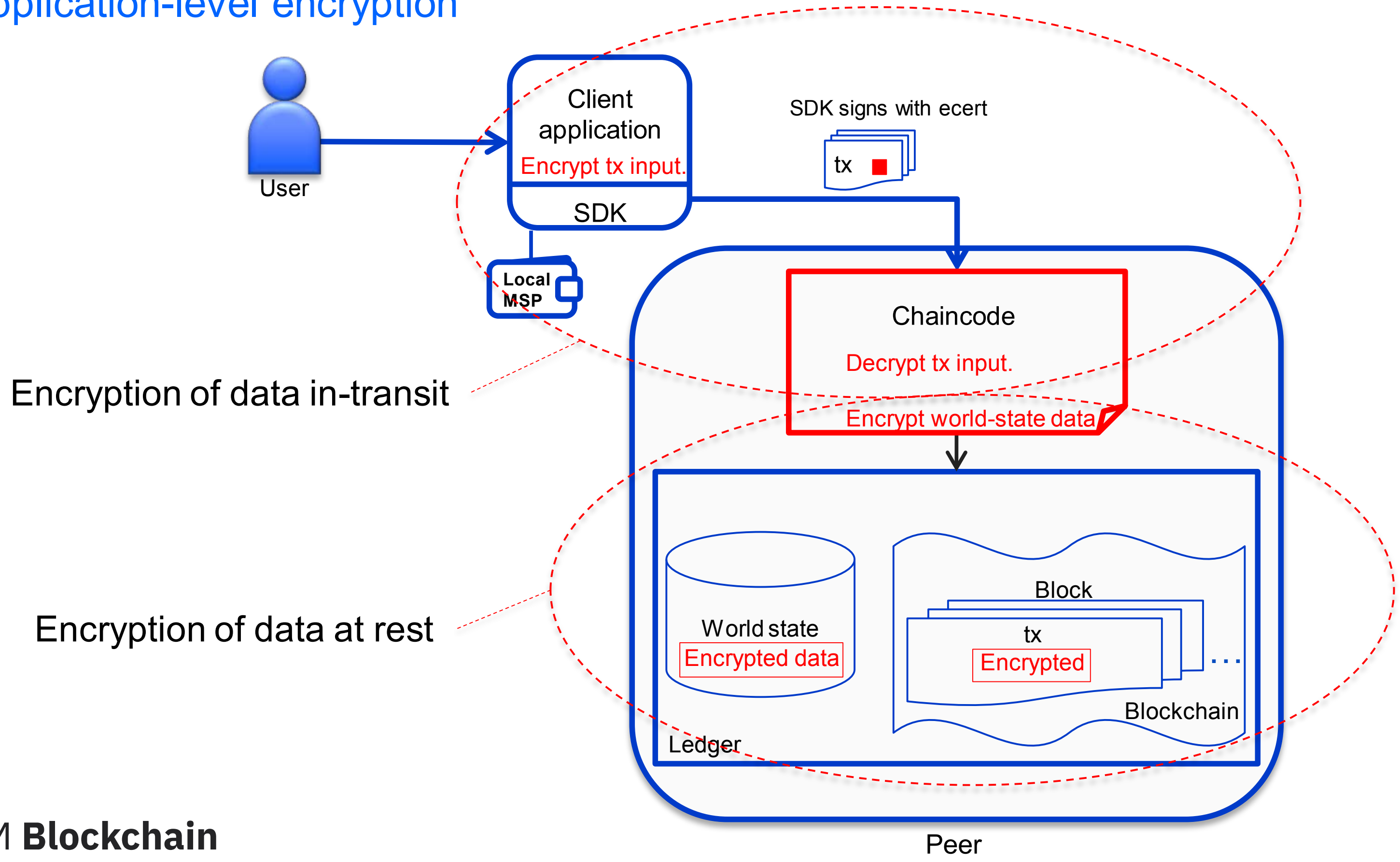
Consensus built-in security

Separation of duties provides inherent overall security before data is committed on the ledger.



Hyperledger Fabric

Application-level encryption



Hyperledger Fabric

Private data collections (SideDB) – Vehicle Manufacture Lifecycle example

Privacy requirements:

- No vehicle data should go through the ordering service as part of a transaction.
- All peers have access to general vehicle information, such as make, year, and color.
- Only a subset of peers has access to vehicle pricing and owner information.

Transaction

- Primary read/write set (if it exists).
- Hashed private read/write set (hashed keys/values)

Transaction

- Public channel data.
- Goes to all orderers/peers.

Collection: Vehicles

- Private Write Set
- Make, Year, Color.

Policy: Org1, Org2

"requiredPeerCount": 1,
"maxPeerCount":2,
"blockToLive":1000000

Collection: Vehicles

- Private data for channel peers.
- Goes to all peers, but not orderers.

Collection: Vehicle Private Details

- Private Write Set
- Price, Owner

Policy: Org1

"requiredPeerCount": 1,
"maxPeerCount": 1,
"blockToLive":3

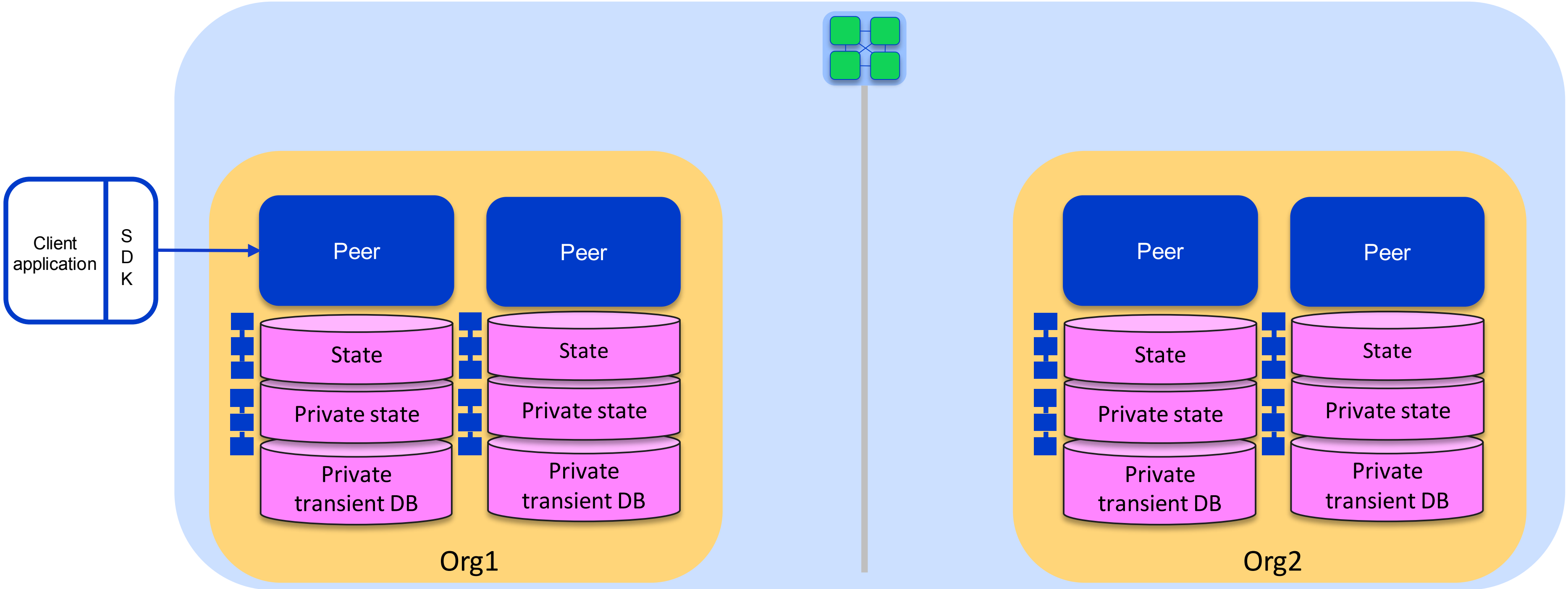
Collection: Vehicle Private Details

- Private data for a subset of channel peers.
- Goes to a subset of peers only.

Hyperledger Fabric

PDC – Vehicle Manufacture Lifecycle - Step 1: Propose the transaction.

The client sends a proposal to the endorsing peer.

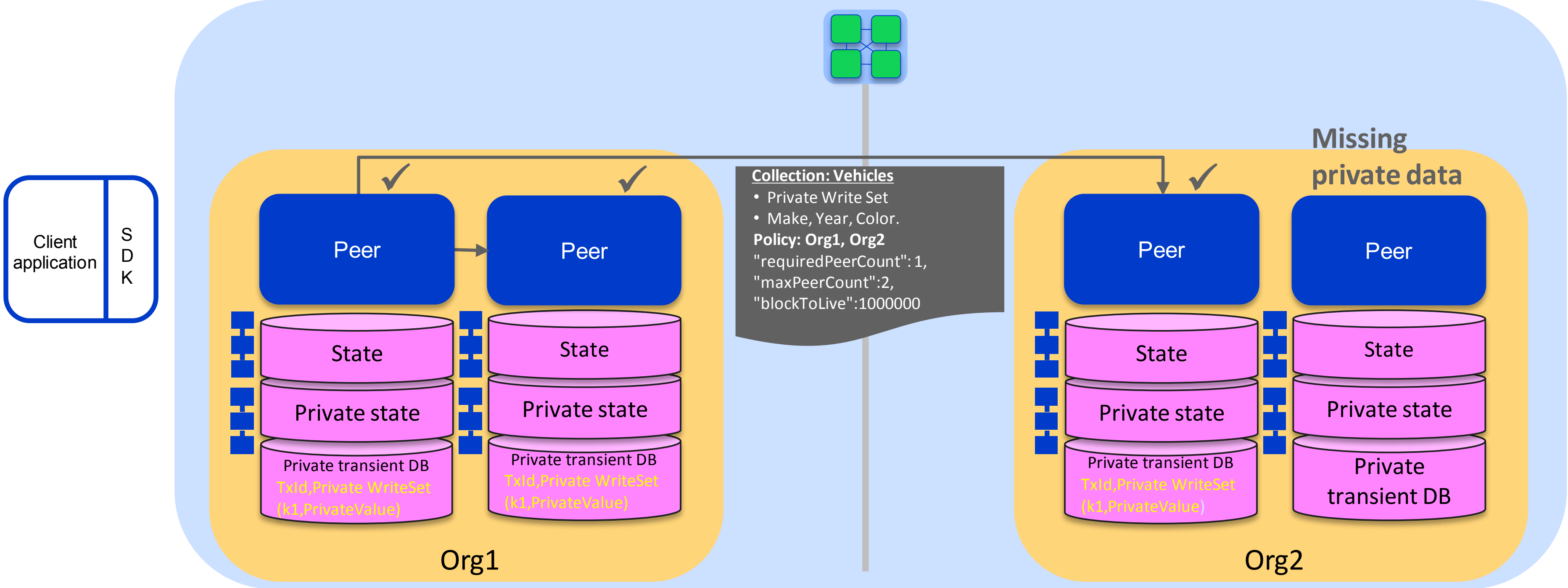


IBM **Blockchain**

Hyperledger Fabric

PDC – Vehicle Manufacture Lifecycle - Step 2a: Run the proposal and distribute the first collection.

Endorsing the peer simulates the transaction and distributes the **vehicles collection** data based on policy.



Collection: Vehicles

- Private Write Set
- Make, Year, Color.

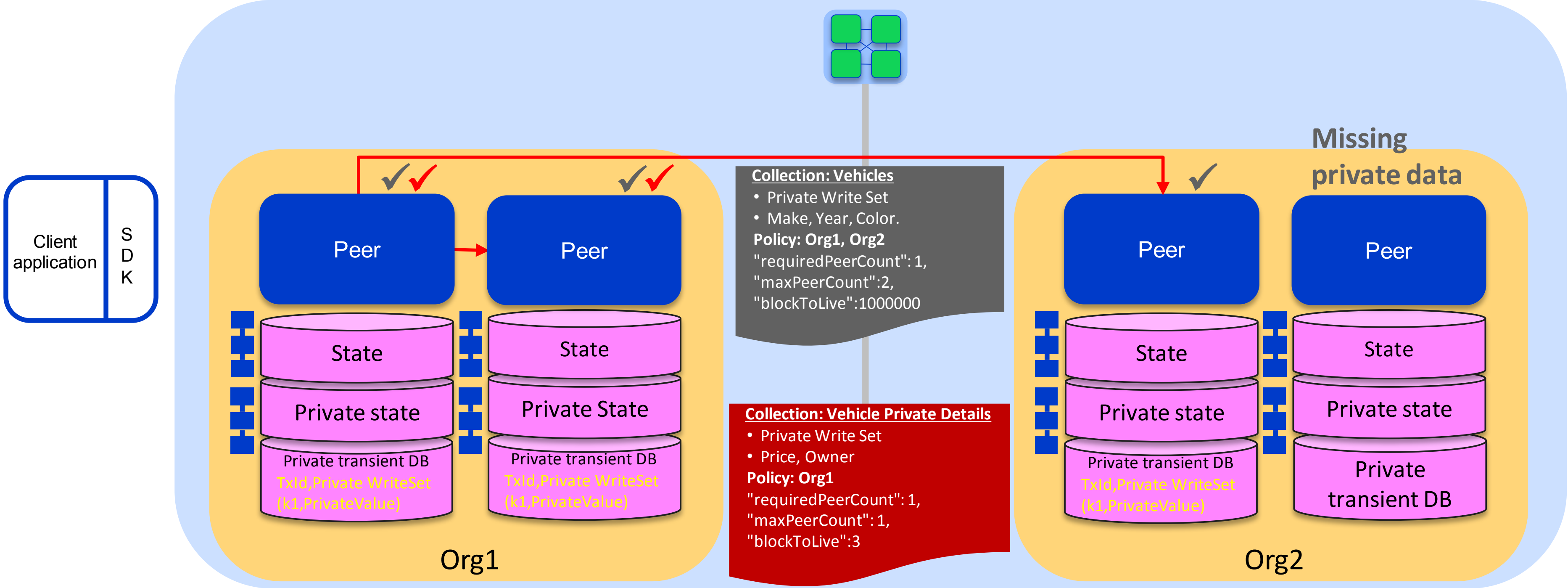
Policy: Org1, Org2

"requiredPeerCount":1,
"maxPeerCount":2,
"blockToLive":1000000

Hyperledger Fabric

PDC – Vehicle Manufacture Lifecycle - Step 2b: Distribute the second collection.

The endorsing peer distributes the **vehicle private details collection** data based on policy.



Collection: Vehicles

- Private Write Set
- Make, Year, Color.

Policy: Org1, Org2

"requiredPeerCount": 1,
"maxPeerCount":2,
"blockToLive":1000000

Collection: Vehicle Private Details

- Private Write Set
- Price, Owner

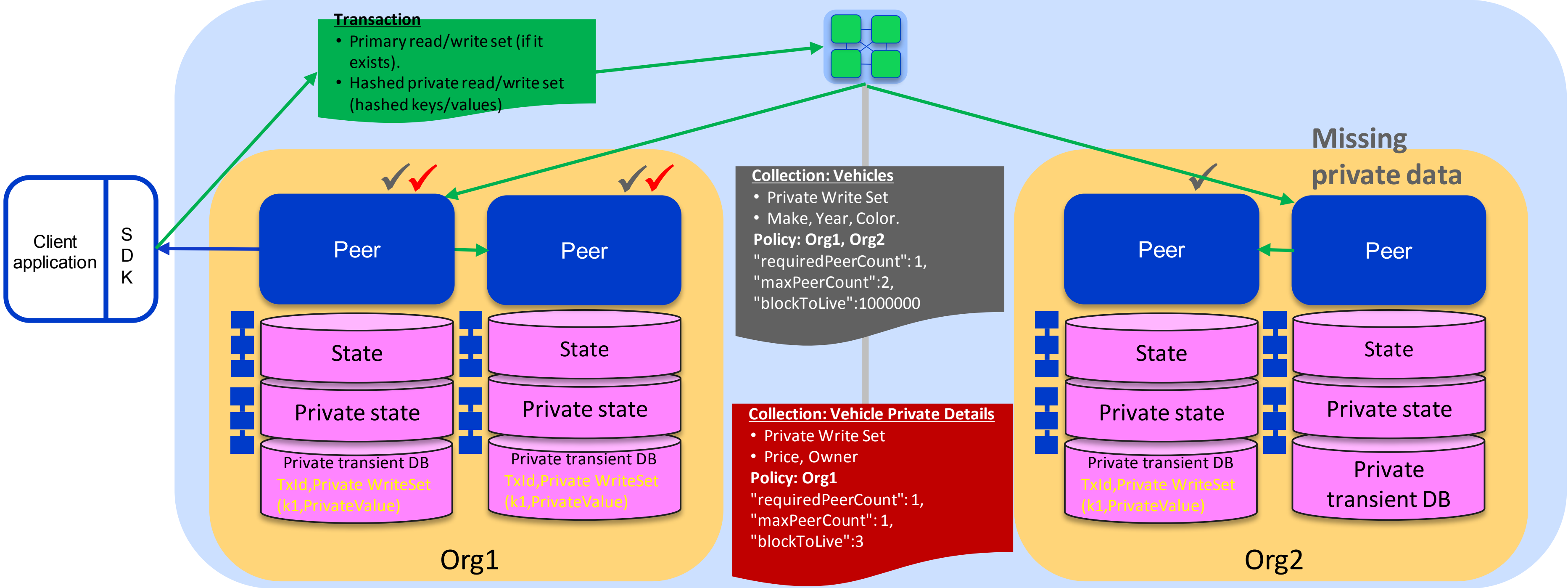
Policy: Org1

"requiredPeerCount": 1,
"maxPeerCount": 1,
"blockToLive":3

Hyperledger Fabric

PDC – Vehicle Manufacture Lifecycle - Step 3: Proposal Response / Order / Deliver.

The proposal response is sent back to the client, which then sends the proposal to the ordering service for delivery to all the peers.



Transaction

- Primary read/write set (if it exists).
- Hashed private read/write set (hashed keys/values)

Collection: Vehicles

- Private Write Set
- Make, Year, Color.

Policy: Org1, Org2
"requiredPeerCount": 1,
"maxPeerCount":2,
"blockToLive":1000000

Collection: Vehicle Private Details

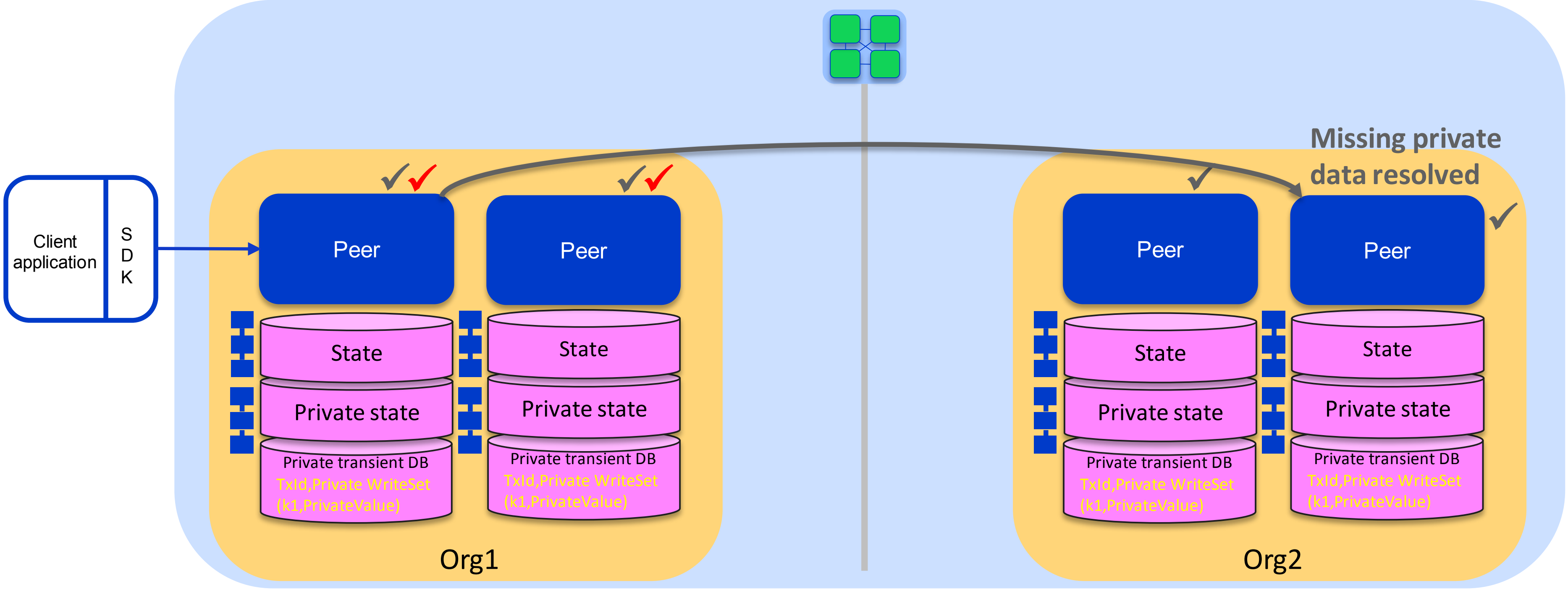
- Private Write Set
- Price, Owner

Policy: Org1
"requiredPeerCount": 1,
"maxPeerCount": 1,
"blockToLive":3

Hyperledger Fabric

PDC – Vehicle Manufacture Lifecycle - Step 4: Validate the transaction.

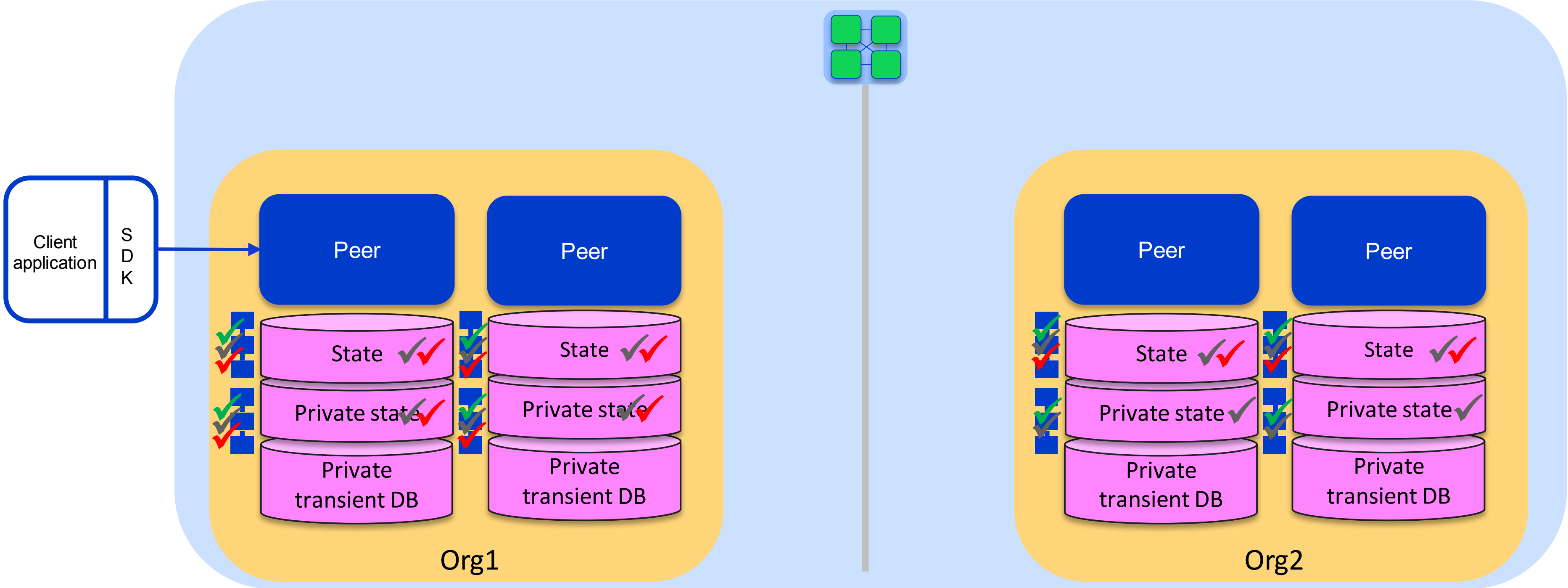
The peers validate the transactions. The private data is validated against the hashes. The missing private data is resolved by pull requests from other peers.



Hyperledger Fabric

PDC – Vehicle Manufacture Lifecycle - Step 5: Commit.

1) Commit private data to the private state DB.
2) Commit hashes to the public state DB.
3) Commit the public block and private write set storage.
4) Delete the transient data.



IBM **Blockchain**

Learning objectives

Hyperledger Fabric security

[Hyperledger Composer security]

IBM Blockchain Platform security

Summary

Hyperledger Composer

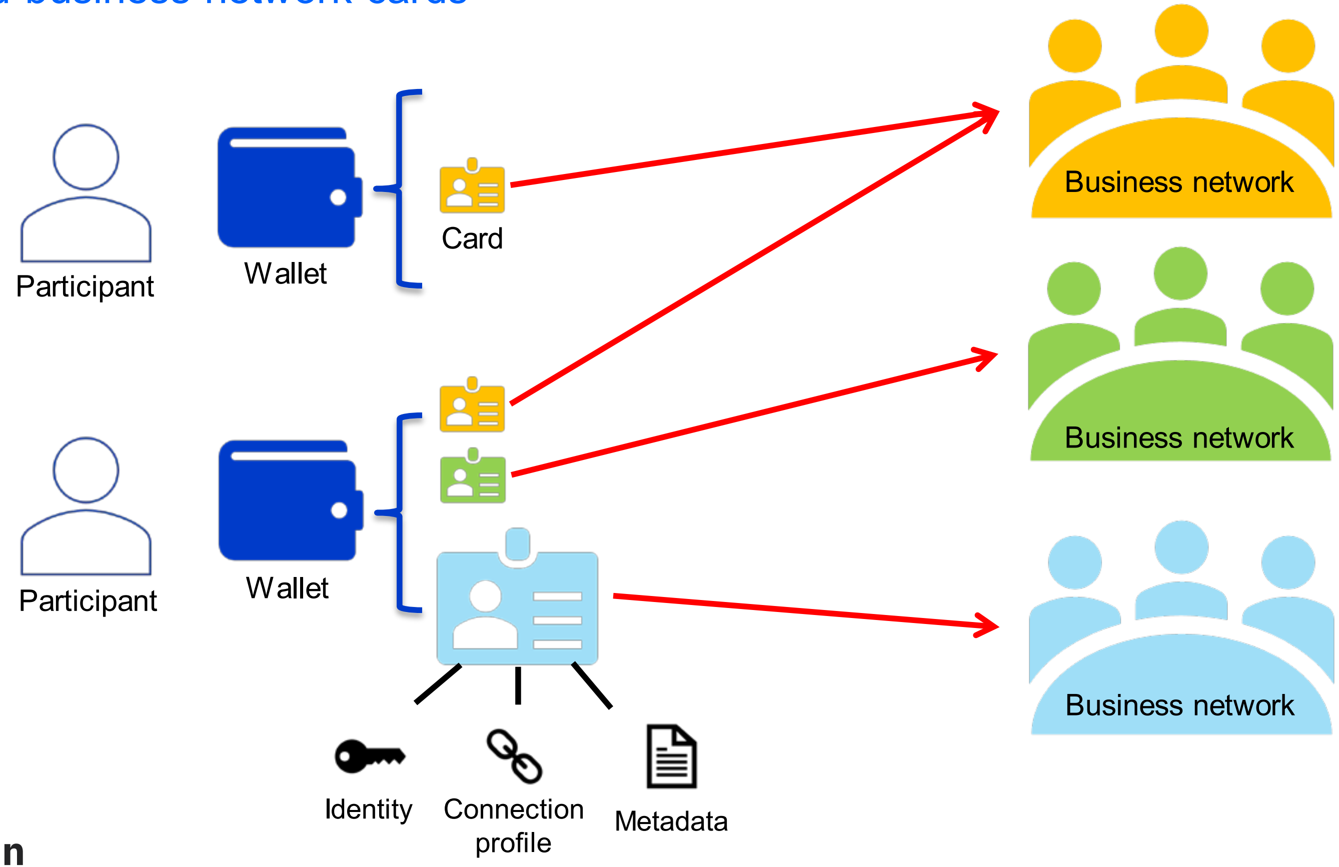
Security (*)

What to secure?	How to secure it?
Identities [Users]	ID and business network cards
Access [Transactions and assets]	Access control lists and REST Server OAuth 2.0 support
Data in transit [Communication]	REST Server HTTPS/TLS support

(*) These features are used in addition to the Hyperledger Fabric and Hyperledger Composer security features.

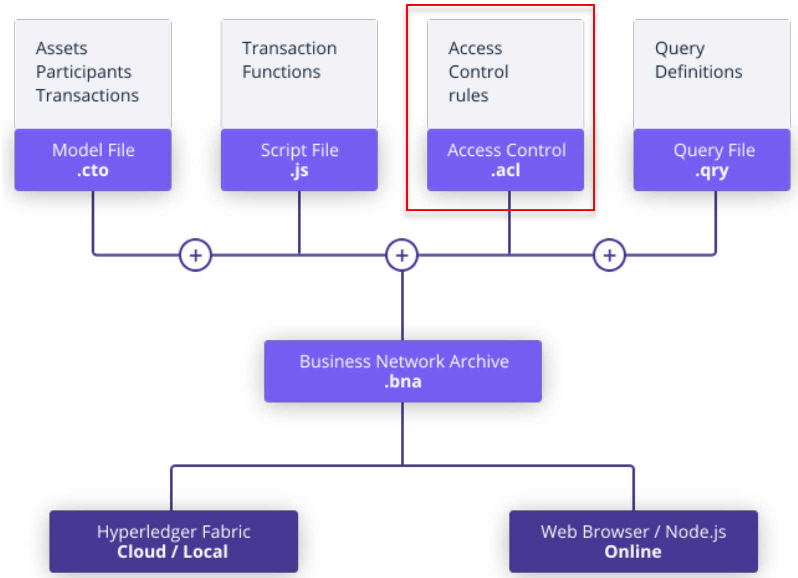
Hyperledger Composer

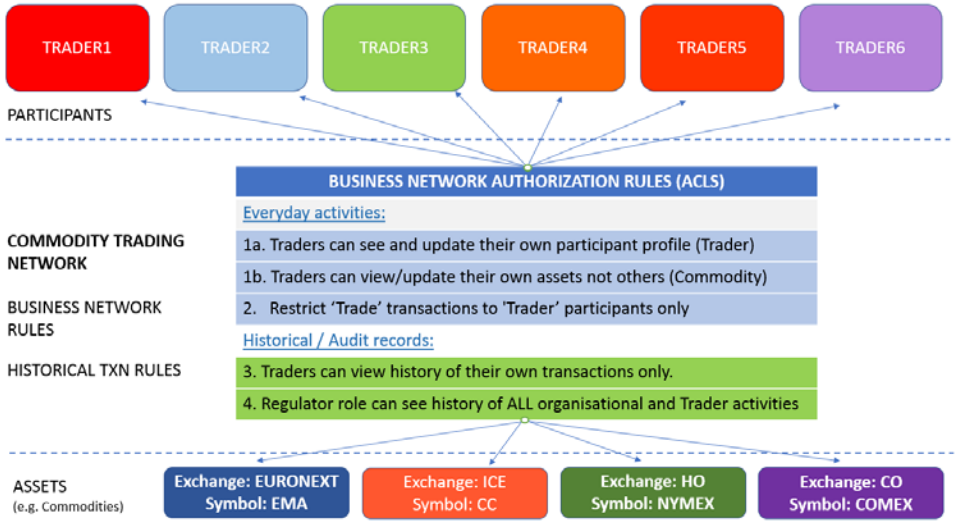
Identities and business network cards



Hyperledger Composer

Access control lists





BUSINESS NETWORK AUTHORIZATION RULES (ACLS)

Everyday activities:

1a. Traders can see and update their own participant profile (Trader)

1b. Traders can view/update their own assets not others (Commodity)

2. Restrict 'Trade' transactions to 'Trader' participants only

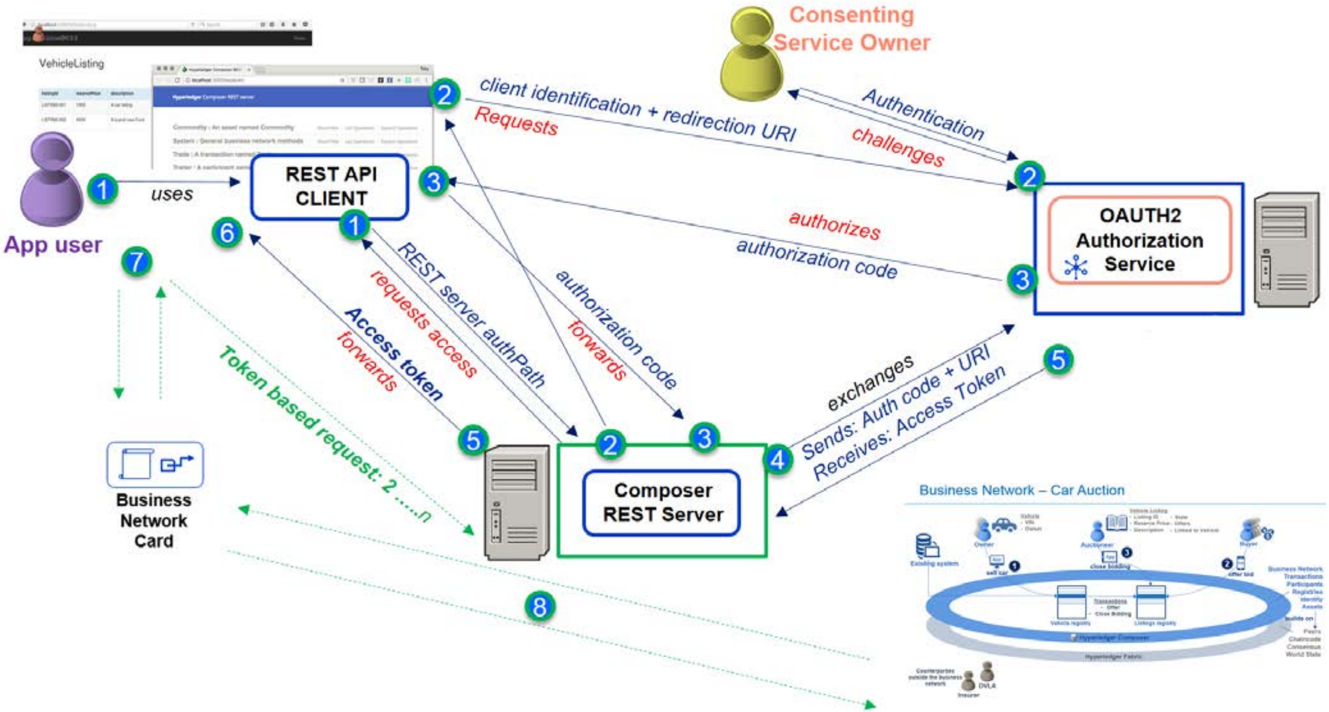
Historical / Audit records:

3. Traders can view history of their own transactions only.

4. Regulator role can see history of ALL organisational and Trader activities

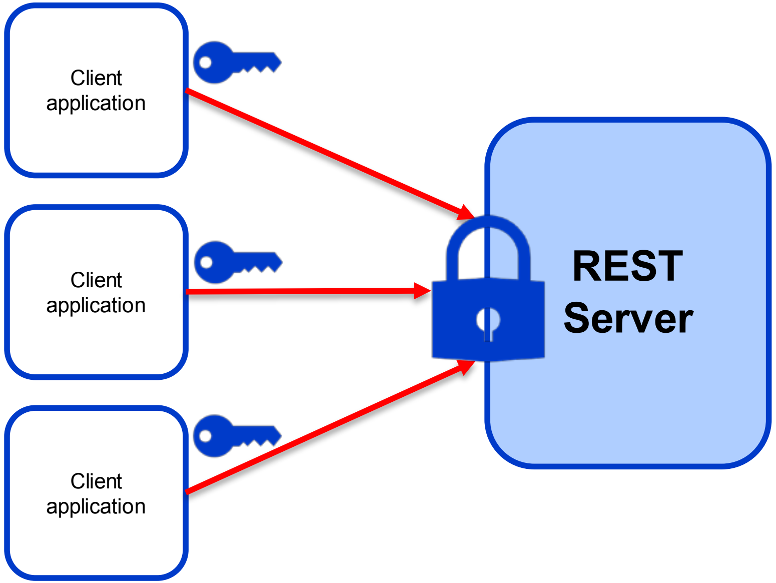
Hyperledger Composer

REST Server: OAuth2.0 authentication



Hyperledger Composer

REST Server: HTTPS and TLS



Learning objectives

Hyperledger Fabric security

Hyperledger Composer security

IBM Blockchain Platform security

Summary

IBM Blockchain Platform

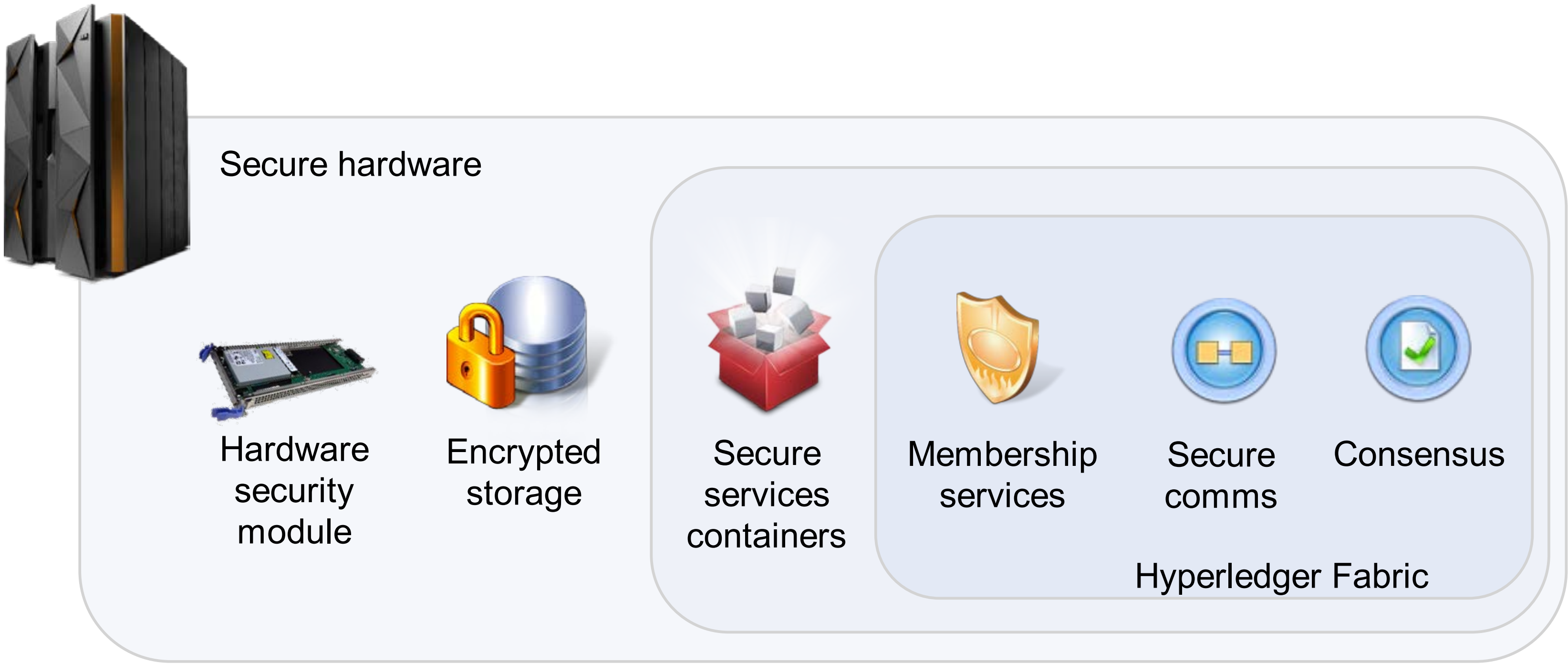
Security (*)

What to secure?	How to secure it?
Identities [All Hyperledger Fabric and Hyperledger Composer identities]	Hardware security module
Infrastructure [Containers and communication]	Secure services containers
Data at rest [Ledger]	Encrypted storage

(*) These features are used in addition to the Hyperledger Fabric and Hyperledger Composer security features.

IBM Blockchain Platform

Security capabilities: Part 1



Secure hardware

Hardware security module

Encrypted storage

Secure services containers

Membership services

Secure comms

Consensus

Hyperledger Fabric

IBM Blockchain Platform

Security capabilities: Part 2

Hardware security module

- **Keys are stored in HSM:** Certified to FIPS 140-2 level 4.
- **Fastest cryptographic acceleration**: Used by block hashing and digital signatures.

Encrypted storage

- **Data privacy**: Encryption of data in flight and at rest on the ledger.

Secure services containers

- **Secure appliance framework:** Provides infrastructure services that encapsulate the Hyperledger Fabric.
- **No root access**: You can access the system and software only through API, including trusted administrators.
- **Impervious to the injection of malware**: Installed from an encrypted, signed boot image.

Learning objectives

Hyperledger Fabric security

Hyperledger Composer security

IBM Blockchain Platform security

Summary

Unit summary

This unit covered blockchain security features of the following items:

- Hyperledger Fabric
- Hyperledger Composer
- IBM Blockchain Platform

Exercise objectives

The exercise covers the development of chaincode to create and retrieve encrypted data from the world state database. During the exercise, you will also learn how pass encryption keys to the chaincodes to be able to read and write encrypted data in a secure way.

References

For more information about the topics that are covered in this unit, see the following resources:

- https://hyperledger.github.io/composer/latest/tutorials/acl-trading
- https://hyperledger.github.io/composer/latest/introduction/introduction
- https://hyperledger.github.io/composer/latest/reference/acl_language
- https://hyperledger.github.io/composer/latest/tutorials/google_oauth2_rest
- https://hyperledger.github.io/composer/latest/integrating/enabling-rest-authentication.html
- https://hyperledger.github.io/composer/latest/integrating/enabling-multiuser.html
- https://hyperledger.github.io/composer/latest/integrating/securing-the-rest-server.html
- https://www.ibm.com/developerworks/community/blogs/df13d392-68c4-49ed-8378-88a091ea50b8?lang=en

Thank you.

IBM Skills Academy

IBM **Blockchain**

www.ibm.com/blockchain

developer.ibm.com/blockchain

www.hyperledger.org