REDOCS 2017

Supervised by Aline Gouget, Gemalto

Data Confidentiality in Private Blockchain

Merve Sahin	Eurecom
Faiza Loukil	LIRIS Laboratory, University Lyon III
Mariem Ben Fadhl	MIS, Univrsité de Picardie
Ouassila Hoceini	LARI Laboratory(Algérie) and Télécom Sud Paris

Blockchain : An Overview

- Distributed ledger shared among a number of entities

  - Contains blocks of transactions

  - Blocks are linked & secured with cryptographic operations

    → Cannot be modified

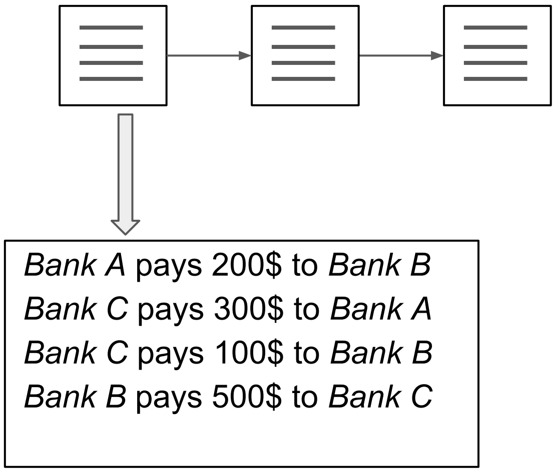
    → Validated with consensus

- Advantages:

  - No need for 3rd party

  - Multiple copies of the ledger

  → Reliable, tolerant to failures & takedowns



Private Blockchains

- **Private ledger** shared among a number of collaborating business partners

- Private, **confidential transactions** only visible to certain collaborators

- Example open source implementation:
  Hyperledger Fabric Project
  - IBM lead project under the umbrella of Hyperledger Project
  - ***Modular architecture*** for various business use cases

Objectives

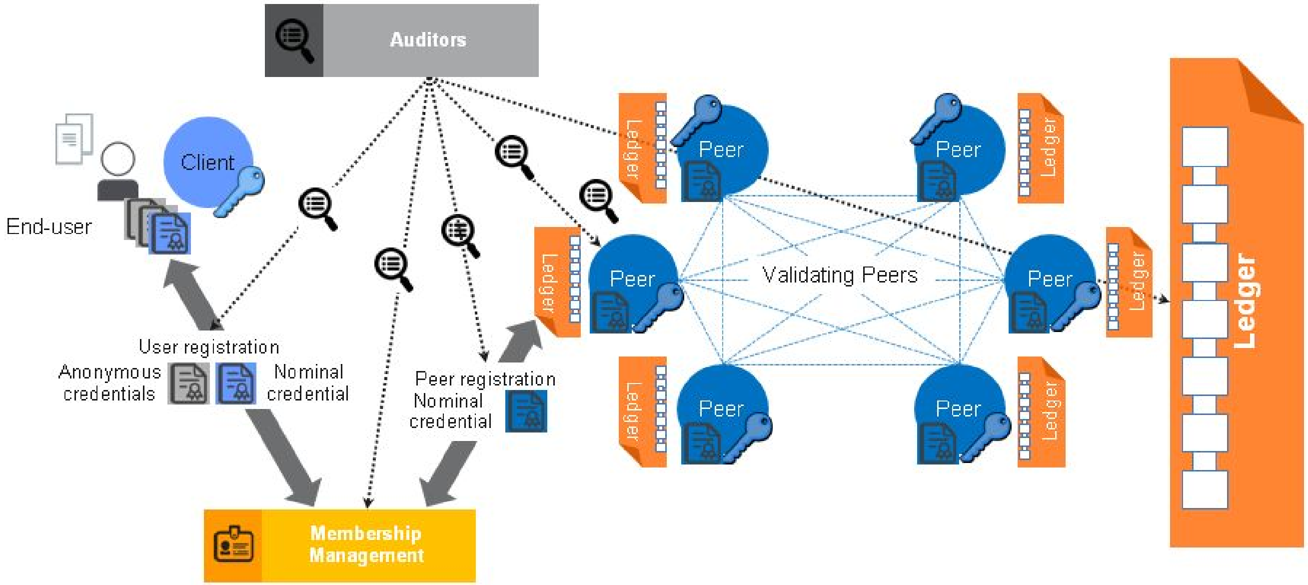
- Understanding the *Hyperledger Fabric* architecture

- Proposing use cases that utilize Hyperledger Fabric
    - Evaluate security requirements
    - Possible improvements on

      - **protecting transaction data in confidential transactions**

      - **reducing trust in centralized parties**

Hyperledger Fabric Architecture



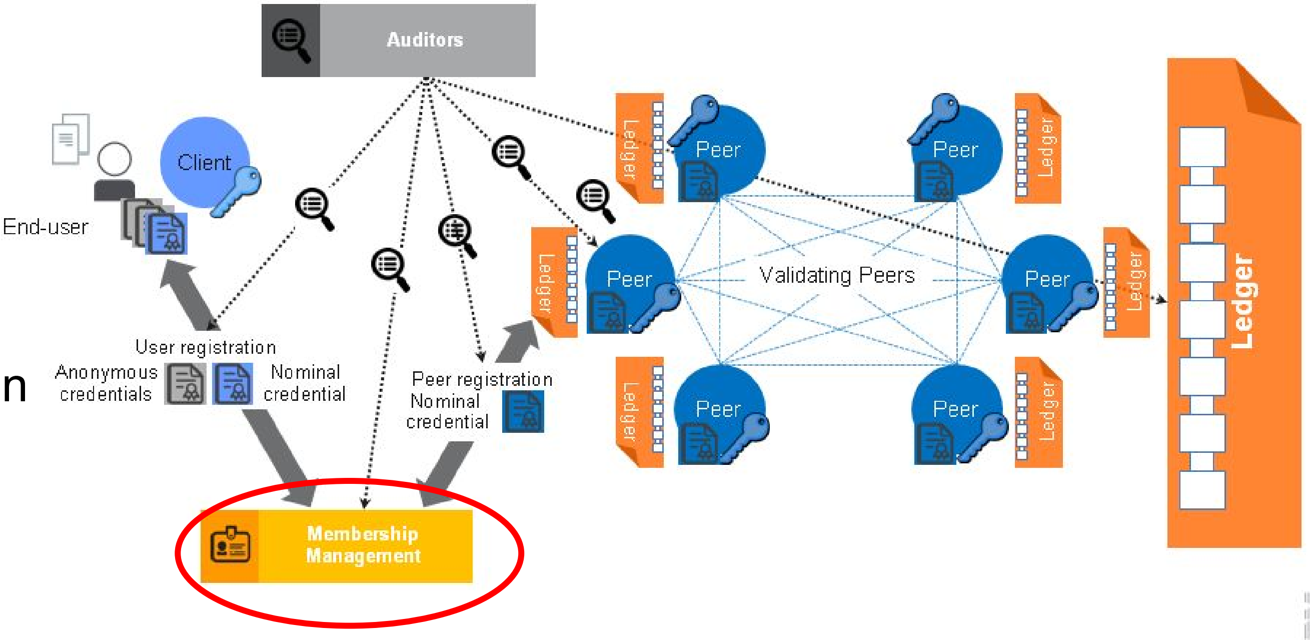
http://fabricrepo.readthedocs.io/en/master/protocol-spec/

Hyperledger Fabric Architecture

Membership Service Provider:

- Issues encryption and signature keys to clients and peers
- Centralized MSP or MSP for each organization

→ MSP can read all transactions!



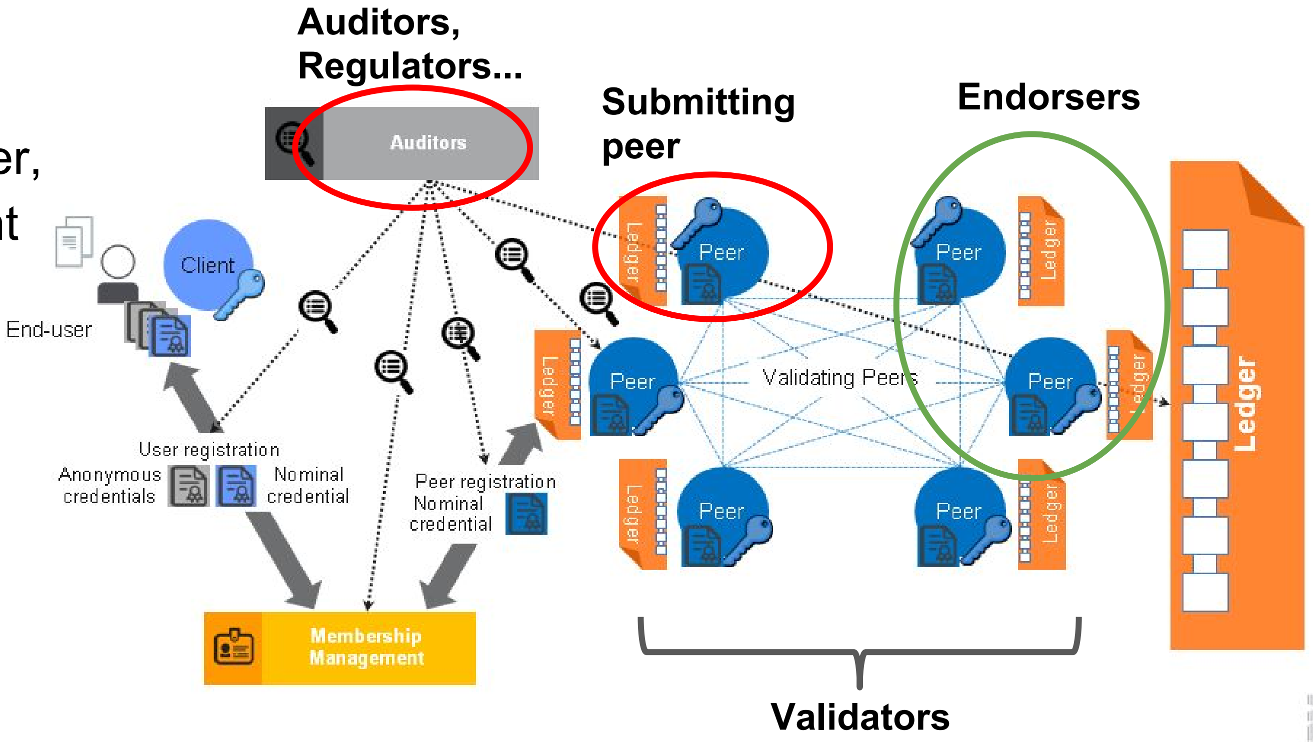
http://fabricrepo.readthedocs.io/en/master/protocol-spec/

Hyperledger Fabric Architecture

Peers:

- Keep a copy of the ledger, participate in management

- Different roles assigned by MSP

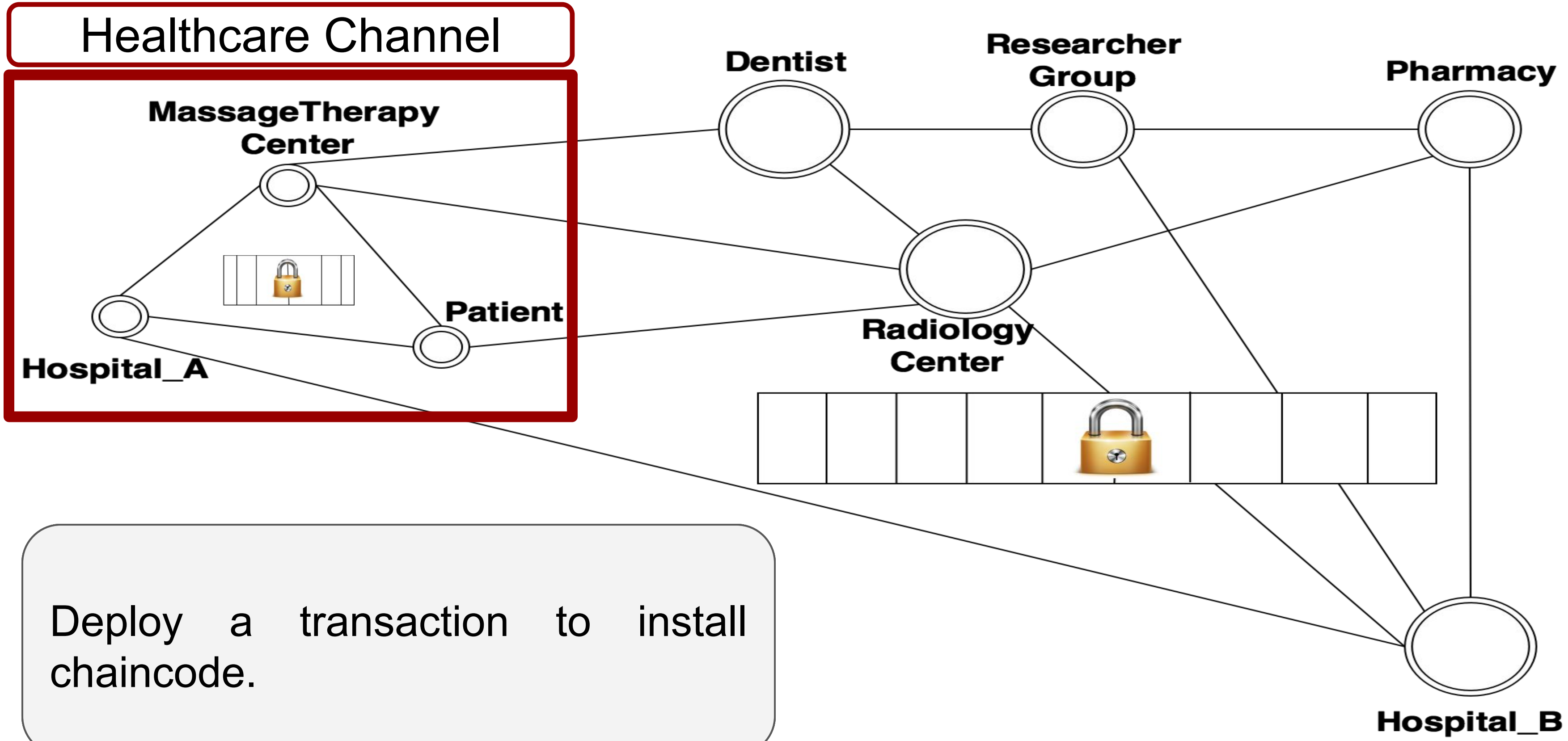


http://fabricrepo.readthedocs.io/en/master/protocol-spec/

Hyperledger Fabric: Concepts

- **Chaincode** (Smart contract)
  - *Reads* or *changes* the state of the ledger
  - Can be *deployed* or *invoked* by a transaction
- **Ordering (consensus) service**
  - Runs on a set of nodes (or single node)
  - Orders the transactions
  - Provides communication channel between peers
- **Channel** (Private subnet of communication)
  - Provides *confidential transactions* between 2+ peers
    - Invisible to the rest of the collaborators
  - One *leading peer* communicates with the ordering service

Healthcare Use Case



Deploy a transaction to install chaincode.

Towards an involved channel

Trust

Problems

- **Centralized Key Management**
  - MSP generates the couple public and private keys.

Solutions

- **Key Decentralization**
  - Create the public and private key and request a MSP certificate.

Towards an involved channel

Confidentiality
Integrity
Authenticity

Problems

- **Eavesdropping Attack**
  - Network traffic can be recuperate by an attacker if transactions are not encrypted.

Solutions

- **Communication Security**
  - Create a common symmetric key and use the Message Authentication Code

Towards an involved channel

Peer Consent
Sensitive Data Control
Audit Functionality

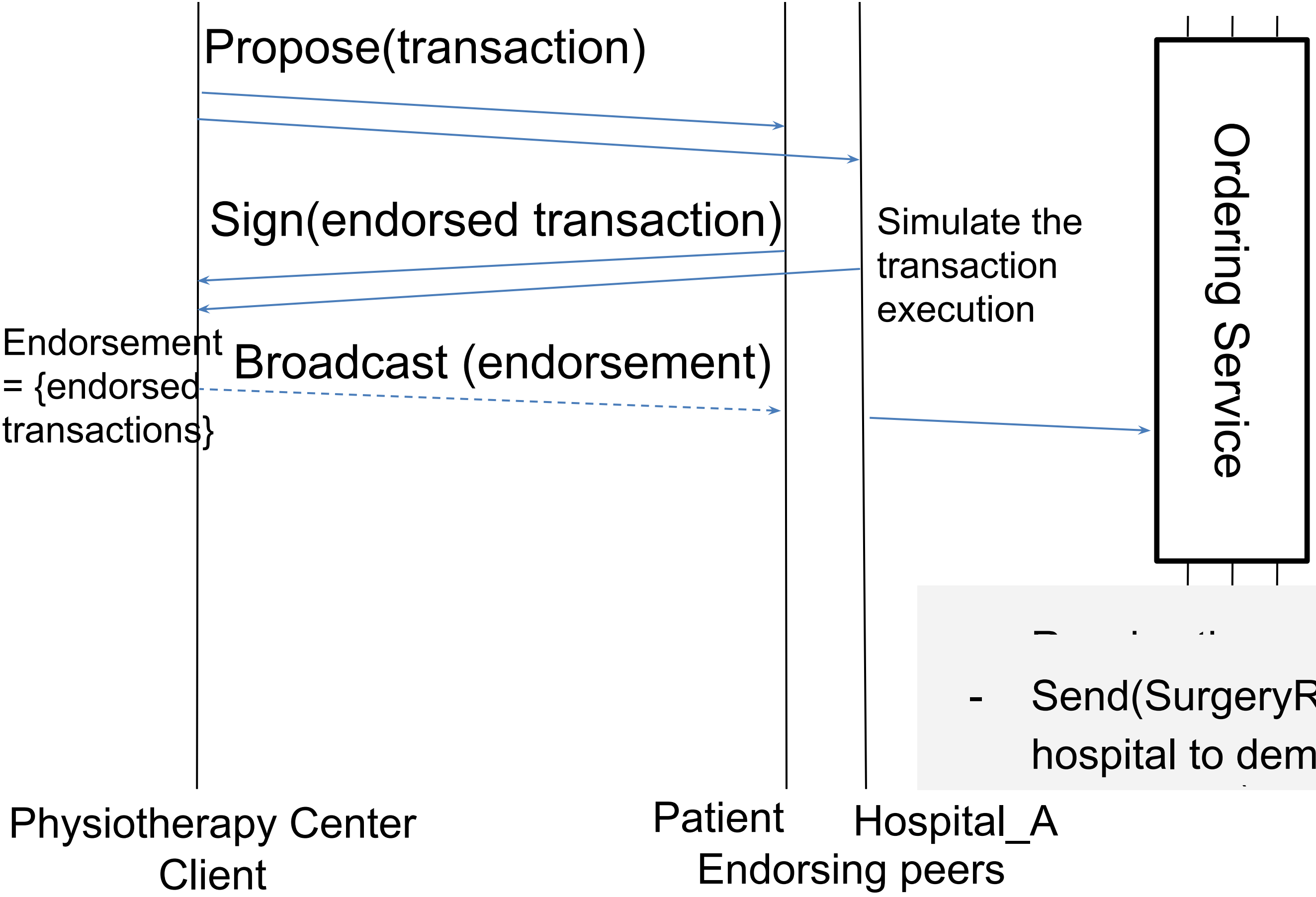
Problems

- **Unauthorized Access Attack**
  - Analysing the sensitive data can help to deduce individuals behaviours and preferences.

Solutions

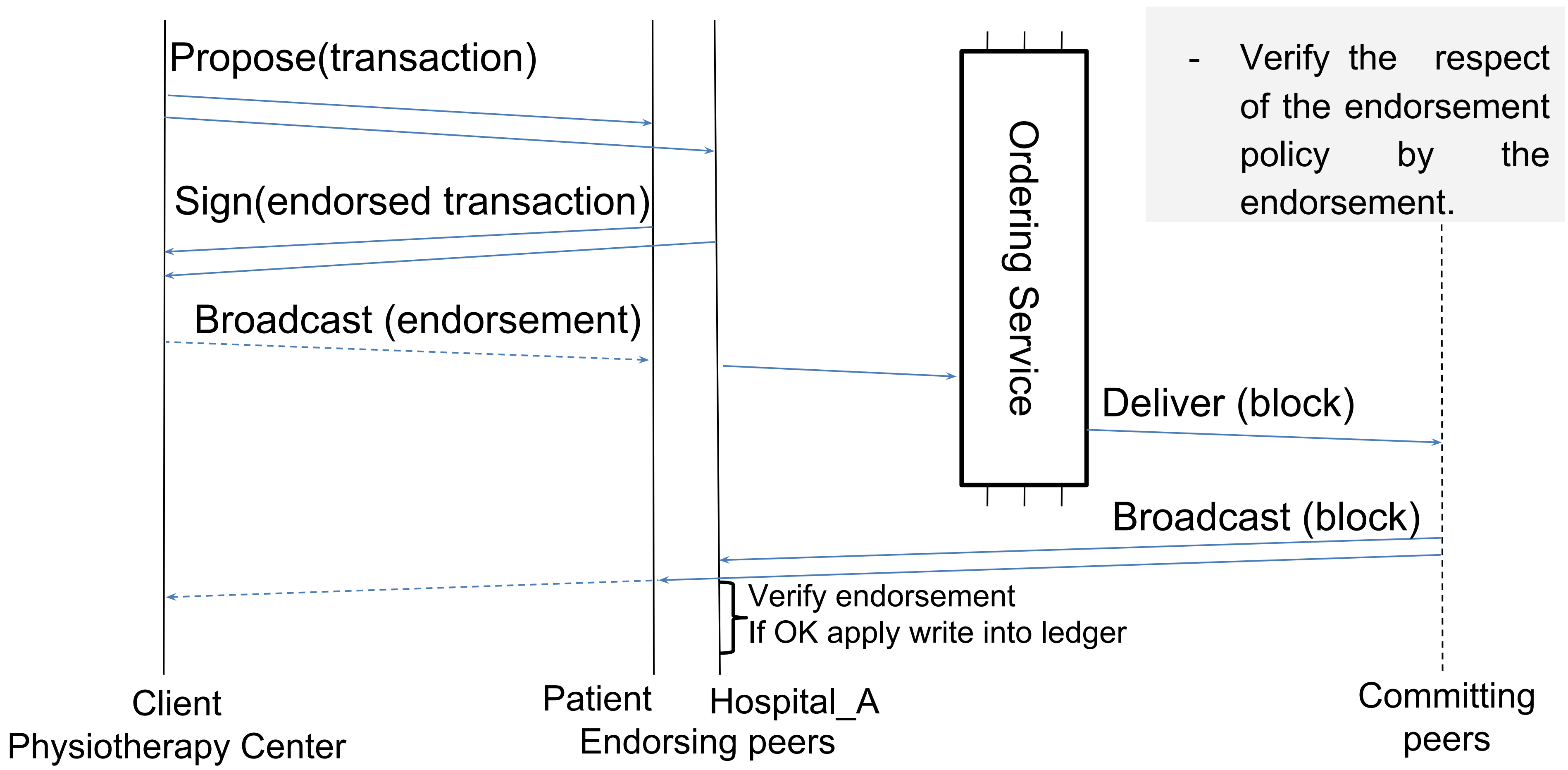
- **Privacy**
  - Claim the peer consent before disclosing sensitive data out of the channel.

Transaction flow



- Send(SurgeryReport) to the patient and hospital to demand the peer consents.

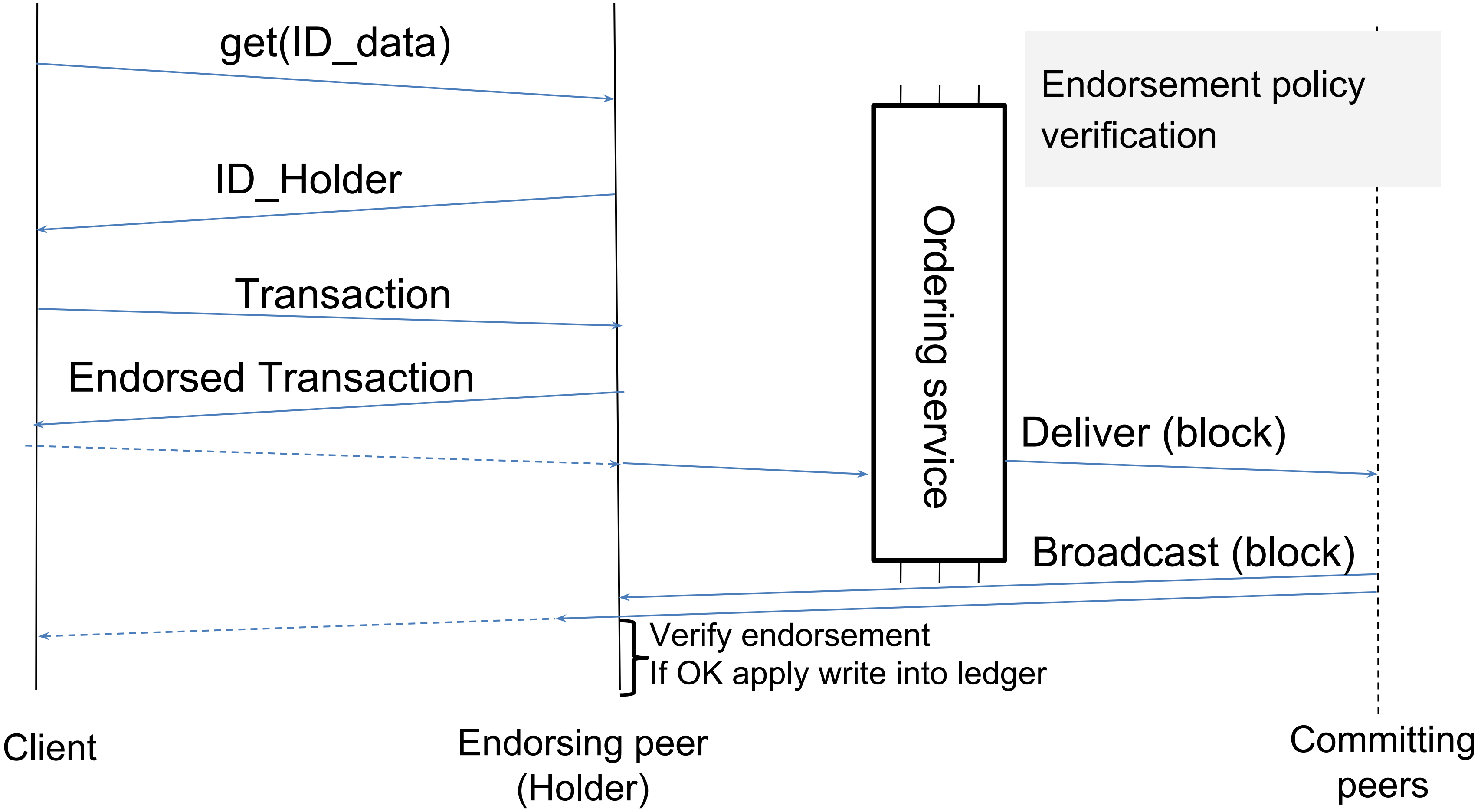
Transaction Flow



Storage System Use Case

- storage system that allows you to outsource data:
    - Put (data): to externalise the data
    - Get (ID_data): to recover the data
- A ledger to control the space available in each machine

Transaction Flow



Transaction Definition

- Transaction:
  - Client ID
  - ChaincodeID
  - Flag (put or get)
  - Storage capacity (free disk space)
  - File size
  - Client signature

- Endorsed Transaction:
  - Transaction
  - Endorsers signatures

Weakness

- Centralized CA (mandatory trust point)
- Orders are fixed peers (unavailability issue)
- The number of validation peers is not defined (possible scalability problem)

Security Requirements

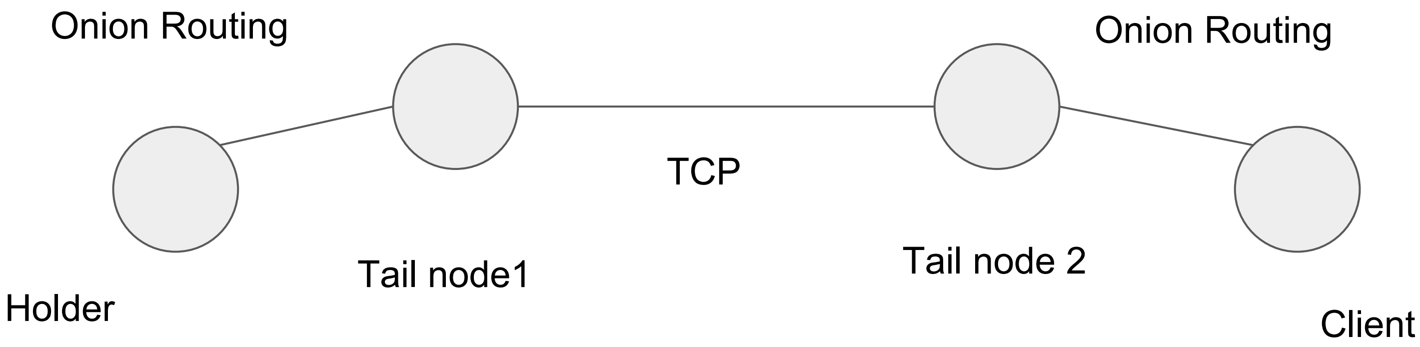
Distribute the architecture while maintaining a reasonable level of security:

- Mutual Authentication

- Check the available space in each machine

- Peers Anonymity

Anonymous Peer-to-Peer File Sharing : APFS [1]



Distributed Authentication

- The network is equipped with a fragmented private key and a public key [2]
- For the first connection:
  - Test Resources [3]
  - Obtaining certificate
  - Getting ID_holders (by APFS protocol) and sending certificate
  - Creating the transaction and signing by one fragmented private key chosen by the holders and the client

Distributed Authentication

- The transaction contains:
  - IDs of Tail nodes
  - Flag
  - Storage capacity
  - File size
  - Signing the transaction by the common key
  - Signing the transaction by tail nodes
- Sending the endorsed transactions to the orders
- Checking tail nodes signatures by validation peers

Distributed Authentication

- For a second connection:
  - Getting ID_holders (by APFS protocol) and sending certificate
  - Creating the transaction :
    - IDs of Tail nodes
    - Flag
    - Storage capacity
    - File size
    - Last transaction ID
    - Client signature
  - Sending the transaction to holders (endorsers) by APFS protocol

Distributed Authentication

- Verification of the transaction by holders:
  - Verification of tail signatures
  - Verification of common signature
- Signing the new transaction (if the verification succeeds) by holders
- Sending the endorsed transactions to the orders
- Checking tail nodes signatures by validation peers

Possible improvements

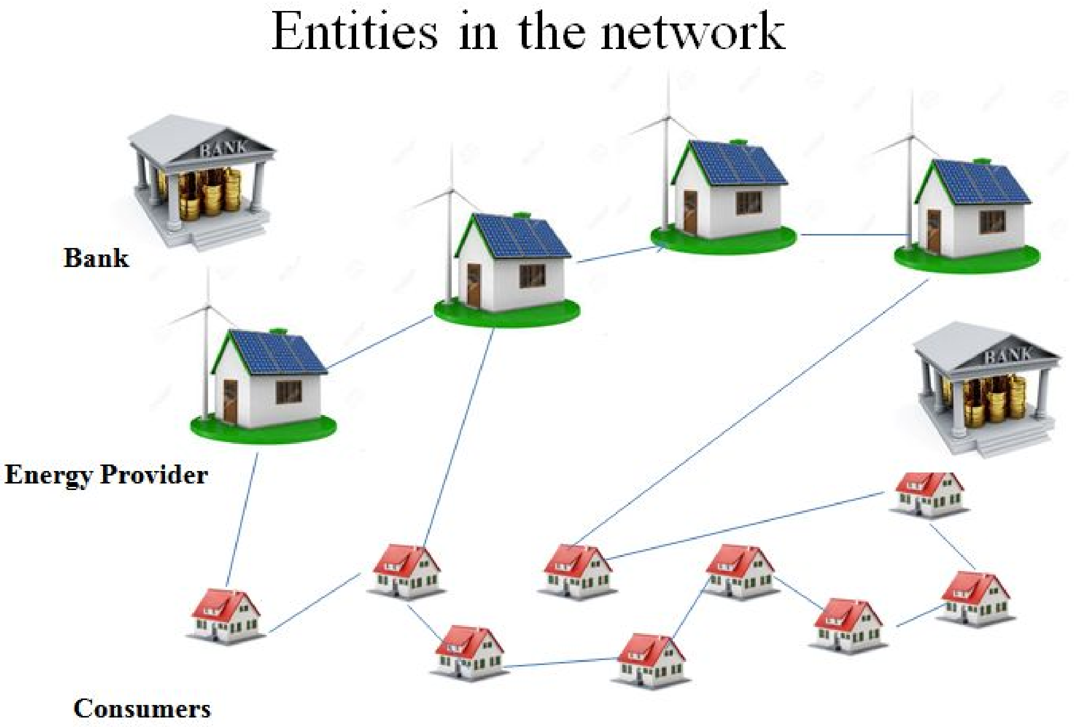
- The orders are chosen in a dynamic way (are not predefined as the case of Hyperledger)

- A more reasonable choice of the number of validation peers

Smart grid use case

- Smart meter data can show what's going in a home, because smart appliances have identifiable load signatures.
- The Energy consumed in home can reveal sensitive information. Moreover, measures taken periodically could lead to an inference on the number of occupants, the moments when the occupants leave the house and the times when they return.

Smart grid use case



Smart grid use case

User Peers Policy: A set of permissions that a user grants to some information contained in a transaction. For example, if one peer (Provider or bank) requires access to one information like(Energy consumption of user1 between 8:00 and 12:00) for legitime(or not legitime) purpose, user1 checks this request before giving access to this peer. We define this in a register UPP(User Peer Policy) of permissions.

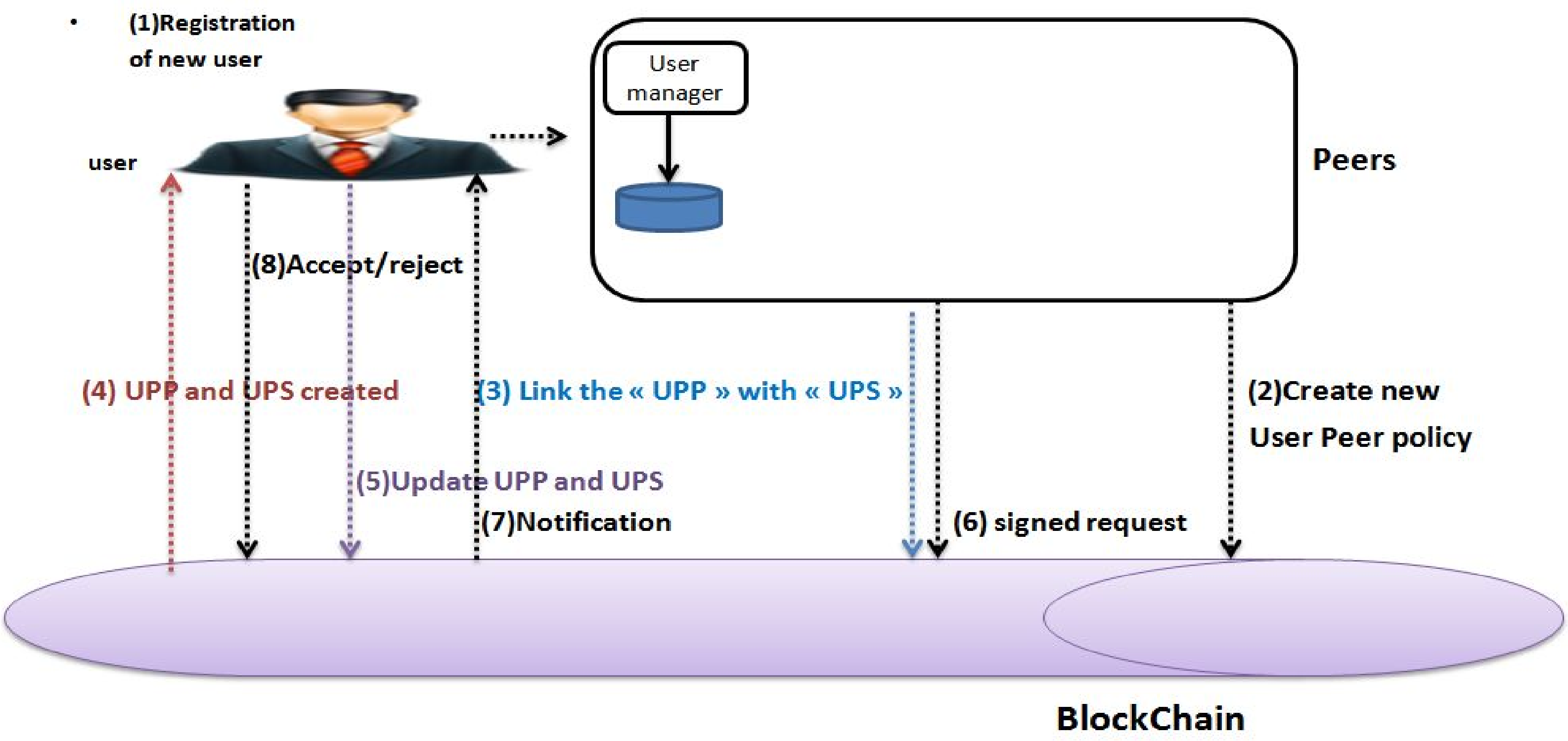
Smart grid use case

•We define for each user, a **user peer status (UPS)** register that contains the status of access rights to user information. In each entry of the register there is a pointer to the UPP register and the access rights for each peer.

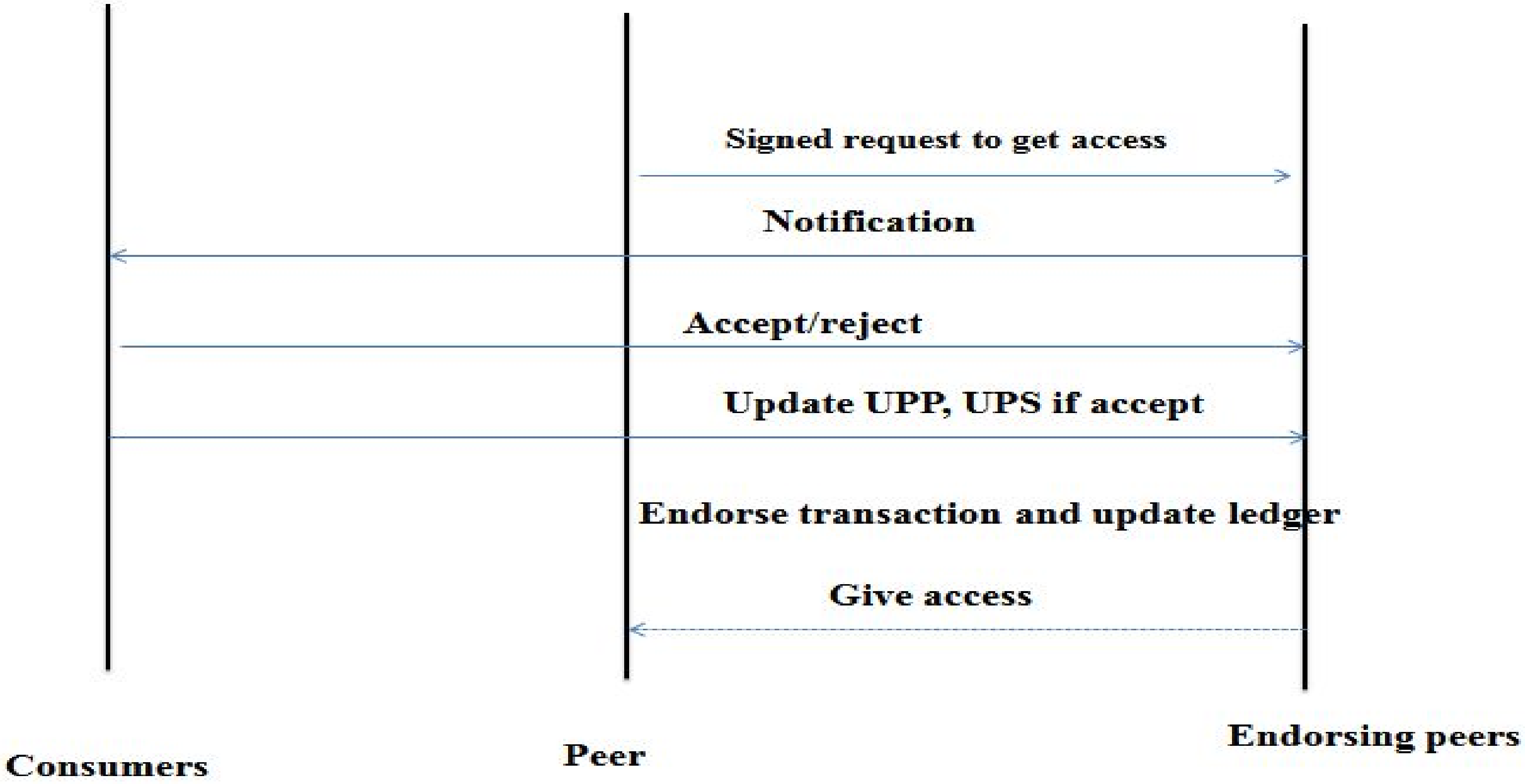
UPP
Access infos
Permissions

User 1	
UPP @	Status
UPP @	Status

Smart grid use case



Smart grid use case



Conclusions

- Hyperledger Fabric has
  - very flexible architecture
  - many use cases with different security requirements
- We analyzed 3 use cases on:
  - Healthcare

    - Share sensitive data only with certain parties
  - Data storage

    - Store data anonymously in a distributed manner
  - Smart grid

    - Protect personal data with access control adapted by user

Challenges in working with Hyperledger Fabric

- Poor documentation
  - References to ‘Section XX’
  - Broken links
- Vague terminology (multiple versions)
- Lack of clear, concrete use cases
  to show the functionality of every module
- Associating abstract concepts with technical implementation
- Curse of ‘flexibility’ ?
  - Supporting every use case is not easy !

References

[1] F Lesueur, L Mé, VVT Tong, Peer-to-Peer Computing, 2009. P2P'09, An Efficient Distributed PKI for Structured P2P Networks

[2] S. Park et al.: "SpaceMint: A Cryptocurrency Based on Proofs of Space", Cryptology ePrint Archive report 2015/528

[3] V. Scarlata, B. N. Levine, and C. Shields, “ Responder Anonymity and Anonymous Peer-to-Peer File Sharing,” in Proc. of the 9th International Conference of Network Protocol(ICNP), 2001. PMCid:1301537.

Thanks for your attention!

Questions?