Blockchain in IoT with HyperLedger Fabric Framework

Final Presentation by Group White

Outline

1. Project Scenario Recap
2. Why Hyperledger Fabric?
3. Security & Membership Services
4. Problem analysis
5. Smart Contract
6. Restful API
7. How we use Blockchain
8. Implementation
9. Future improvements
10. Live Demo

1. Project Scenario Recap

- Decentralized solution for quality assessment.
- Semiconductor Manufacturing Process
  - Check temperature and humidity.
  - Continuous sensor data capture.
- Store sensor data to hyperledger network for audit and visualization.

2. Why Hyperledger Fabric

- Distributed Ledger Technology (DLT)
- Enterprise-ready network **security**, **scalability**, **confidentiality** and performance.
- Modular blockchain architecture
- Endorsement policy and smart contract.

3. Security & Membership Services

- All participants have known identities (**Identity management**)
- Public Key Infrastructure is used to generate cryptographic certificates
- It is tied to organizations, network components, and end users or client applications
- Access control lists can be used to provide additional layers of permission (**Permissioned** blockchain)
- The members of a Hyperledger Fabric network enroll through a trusted **Membership Service Provider (MSP)**.
- **Private channels** are restricted messaging paths that can be used to provide transaction **Privacy and Confidentiality**

4. Problem analysis

- Project topic is about
  - Sensor data
  - Blockchain
  - Smart Contracts
- How to combine all three in a meaningful matter?

Ideas

- Decentralized storage of sensor data is beneficial to increase trust between business partners.
- A smart contract could analyze data in in a way to prevent forgery.
- What data can be trusted?
- Can sensor data be secured otherwise?

Trust in sensor data

- Anomaly detection (thresholds, maximum distance from the mean, different models)
- Is abnormal data invalid data?
- Do we ever want to reject sensor data? Yes!

Possible situations

- Strong and rapid fluctuation of values
- Fall of temperature below -60°C
- Rise of temperature above 1000°C
- Stagnant values when change is expected
- Date is in the past or exists already
- Decryption of data fails

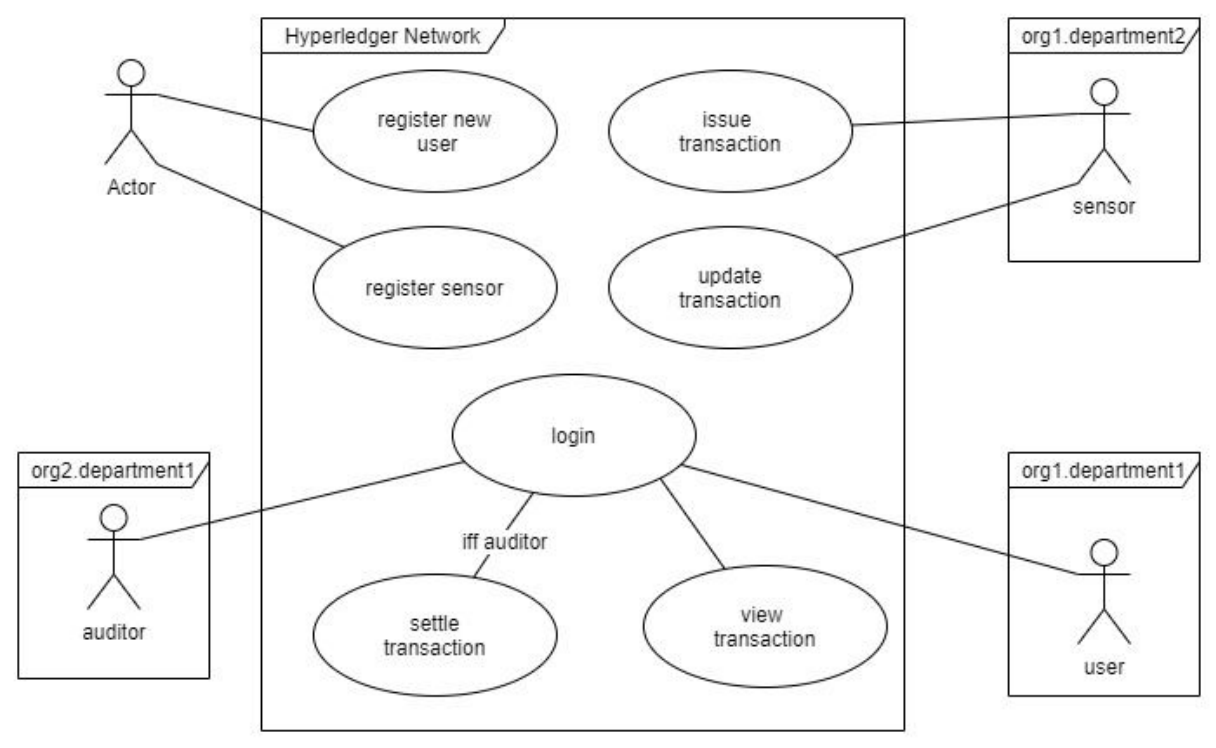
Rejecting data

- Business partners want to know the values, even when there are outliers or anomalies.
- It’s important not to reject these values
- Data can be rejected when timestamps aren’t matching up
- Data can be rejected when incorrectly encrypted (possible forgery)

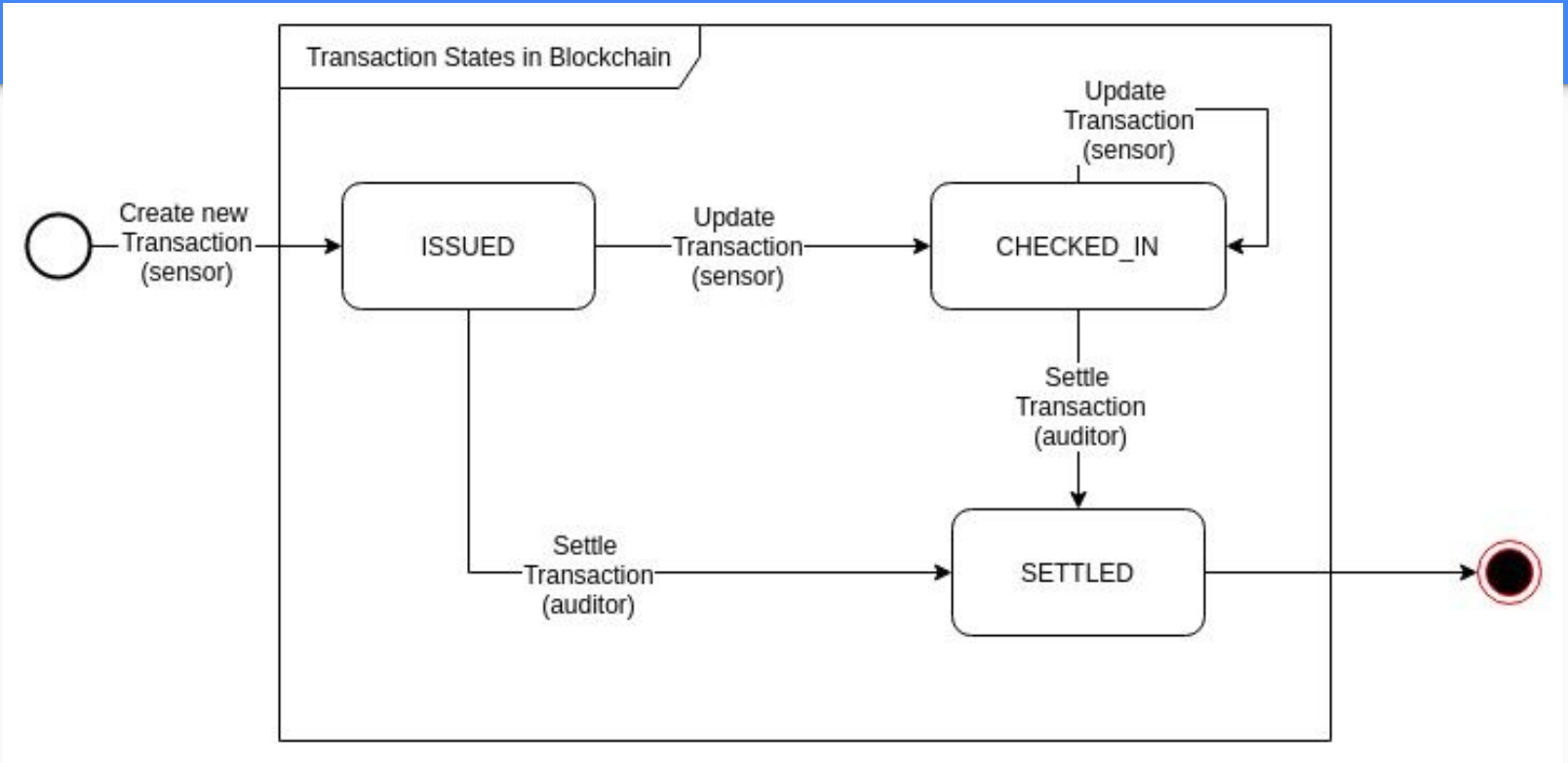
5. Smart Contract

- User Actions Management
  - 4 types of users(admin, user, auditor, sensor)
  - Admin user not involved in any transaction actions.
- Data parsing and State Management.
  - Transaction states(ISSUED, CHECK_IN, SETTLED)

Organizational Structure



Smart Contract Modeling for the scenario



6. RESTful API used for interaction

The HTTP-based RESTful APIs enables the communication between the web client, sensor data simulator and the blockchain network.

URI	Transaction	Action	Media Type
/api/user/register	Register new user.	POST	Application/json
/api/sensor/register	Register new sensor	POST	Application/json
/api/txn/issue	Issue a new transaction	POST	Application/json
/api/txn/update	Updating an existing transaction	POST	Application/json
/api/txn/settle	Settlement of existing transaction	POST	Application/json
/api/txn/list	Get all transaction	GET	Application/json
/api/txn/{batchNumber}	Get transaction detail	GET	Application/json
/api/txn/{batchNumber}/history	Get transaction history	GET	Application/json

7. How we use Blockchain

- Save data in non repudiable manner.
- Transaction authentication and authorization.
- Manufacturing process validation.

8. Implementation

This prototype platform includes

1. Blockchain network: Hyperledger Fabric
2. Smart contract : Java Implementation
3. Sensor simulator: standalone program in python.
4. WebUI: Flask, HTML and Bootstrap

9. Future Improvements

- Hyperledger Indy (Identity Management)
- Role Based Authentication(contd.)
- Automate auditing process in the smart contract.
- Implement Https in the apiserver.

10. Demo