Platform for Integrating Internet of Things Based Smart Healthcare System and Blockchain Network

Adhitya Bhawiyuga*, Arya Wardhana*, Kasyful Amron*, Annisa Puspa Kirana†
* Faculty of Computer Science
University of Brawijaya
Malang, Republic of Indonesia
Email: bhawiyuga@ub.ac.id, aryawardana1997@gmail.com, kasyful@ub.ac.id
† Department of Informatic
State Polytechnic of Malang
Malang, Republic of Indonesia
Email: puspakirana@polinema.ac.id

***Abstract*—In an internet of things (IoT) based smart healthcare system, personal health data can be produced by means of various wearable devices connected through a wireless body area network (WBAN). Since they contain sensitive information, whole collected data should be stored in a platform that guarantee both data confidentiality and integrity. In this sense, permissioned blockchain i.e. Hyperledger can be seen as an auspicious candidate for providing a secure yet scalable storage platform thanks to its data integrity and peers decentralization features. Considering that both IoT and blockchain network have different working mechanism, therefore, a software platform is required to provide an integration layer between the IoT and blockchain network. In this paper, we propose the design of IoT-to-Blockchain platform for integrating internet of things based smart healthcare system and blockchain network. The proposed system consists of five components including blcokchain-to-device interface, chain code execution interface, membership service provider (MSP), peers node and ordering node. The proposed system consists of five components including HTTP-based API gateway as device-to-blockchain interface, membership service provider (MSP), peers node and orderer. At first, the API gateway receives sensing data from IoT gateway device. Upon reception, the API gateway acting as client invokes chain code installed in each blockchain peer by emitting a transaction proposal to peers. Peers then simulate the execution and send back the endorsed data to the client i.e. API gateway. Collected endorsement from various peers are sent to orderer which then build a block containing an ordered transactions received from all clients and then broadcast that block to all peers. Each peer the performs a final verification before committing all ordered transactions on that block into its own local ledger. By using this mechanism, every peer in network can receive similar transaction orders which lead to identical ledger state changes.**

I. Introduction

The increasing number of physically interconnected devices have contributed to the development of internet of things (IoT) to improve the quality of human life. As one of the important aspect of human life, the health can take a benefit of IoT in form of smart and pervasive healthcare service leading to what we call as Internet of Health Things (IoHT) [1]. This kind of system typically consists of three main components including wearable sensing devices, an internet gateway device and a cloud based data center. At first, the personal health data of user are produced by means of various wearable sensing devices connected through a wireless body area network (WBAN). The collected data can then be either processed locally on the gateway device or delivered to the data center. While the former scenario offers a favorable real time data processing, the later one are still required provides a more complex data analysis [2]. Furthermore, In order to get more benefit, those data can be shared among various health stakeholders such as doctor, hospital, researchers and government agencies. In this case a secure and scalable data storage and access system is highly required to store such massive sensitive information.

One of the viable option to permanently store those huge amount of personal health data is by incorporating centralized cloud based data storage service [3], [4]. In this approach, the personal data are collected through the help of a gateway device and stored to a centralized data center for further data analysis. Additionally, in order to provide the data access to another stakeholders, the system employs an application programming interface (API) gateway in the form of webservice mechanism. While it offers a simple storage abstraction to the user, however, the centralized cloud health service may experiencing several issues especially in the context of data sharing. First, since personal health data contain sensitive information, therefore, there should be a mechanism to ensure the privacy of user's data. Since the centralized cloud service is operated by a certain entity i.e. cloud provider, therefore, the user may have a little control regarding to which entity his/her personal data will be shared with. In addition, since shared data are utilized for life critical decision support system, therefore, every entity should make sure that those health data are free from breach and tampering during data storage and transmission processes. Unfortunately, existing centralized cloud storage systems are lack of data confidentiality and integrity assurance.

Taking into account aforementioned issues, the permissioned blockchain can be seen as promising candidate for providing a secure yet scalable storage platform thanks to its data integrity and peers decentralization features. In general, a blockchain network consists of several connecting independent

978-1-7281-5163-2/19/$31.00 ©2019 IEEE

nodes in which every node maintains a distributed ledger for recording any changes in asset as the result of a transaction [6]. Once a transaction is triggered by the client, a validation mechanism is employed to maintain consistency and integrity of the shared data across all contributing nodes. In contrast with public blockchain like Ethereum or Bitcoin in which every people can join its network [5], a permissioned blockchain i.e. Hyperledger limits members of a certain network to a set of known and identifiable node registered by authentication authorities. Furthermore, in Hyperledger, a node has ability to make a private channel to share the distributed ledger only to certain group of nodes. In this case, the permissioned blockchain can take the advantage of much simpler Byzantine-fault tolerant (BFT) consensus as compared to a complex proof-of-work (PoW) mechanism in public blockchain. The combination of both the simplicity of nodes consensus and the ability to limit the data sharing between nodes make the Hyperledger suitable for storing a sensitive data e.g. personal health data generated from IoT healthcare wearable devices. Considering that both IoT and blockchain network have different working mechanism, therefore, a software platform is required to provide an integration layer between the IoT and blockchain network.

In this paper, we propose the design of IoT-to-Blockchain platform for integrating internet of things based smart healthcare system and permissioned blockchain network. The proposed system consists of five components including HTTP-based API gateway as device-to-blockchain interface, membership service provider (MSP), peers node and orderer. At first, the API gateway receives sensing data from IoT gateway device. Upon reception, the API gateway acting as client invokes chain code installed in each blockchain peer by emitting a transaction proposal to peers. Peers then simulate the execution and send back the endorsed data to the client i.e. API gateway. Collected endorsement from various peers are sent to orderer which then build a block containing an ordered transactions received from all clients and then broadcast that block to all peers. Each peer the performs a final verification before committing all ordered transactions on that block into its own local ledger. By using this mechanism, every peer in network can receive similar transaction orders which lead to identical ledger state changes.

II. Proposed Design

Fig. 1 illustrates the general environment of proposed IoT-to-Blockchain gateway for IoT and blockchain network integration with three main actors : wearable sensing devices, gateway device and IoT-blockchain platform. The wearable sensing device has a role to perceive the user's physical condition through the usage of various sensors including heartbeat, body temperature or activity sensors. The collected sensor data are then transmitted to the gateway device through a wireless connection such as wifi [7] or bluetooth low energy[8]. Once the data are received, the gateway device relays that sensor data to IoT-blockchain platform through blockchain API gateway using a cellular or backbone network connection. Those data are then internally processed in blockchain network for further storing by its peers.

In detail, the transaction flow of each component is illustrated in Fig. 2. At first, IoT or gateway devices send their sensing information to blockchain network through an API gateway. Upon reception, the API gateway acting as client invokes chain code installed in each blockchain peer by emitting a transaction proposal to endorsing peers. Notice that the proposal may contains several fields including : chain code identifier, payload or parameter, timestamp and client signature. Endorsing peers then simulate the execution and send back the endorsed data to the client i.e. API gateway. Collected endorsement from various peers are sent to orderer which then build a block containing an ordered transactions received from all clients and then broadcast that block to all peers. Each peer the performs a final verification before committing all ordered transactions on that block into its own local ledger. By using this mechanism, every peer in network can receive similar transaction orders which lead to identical ledger state changes.

III. System Implementation

In this section, we present the implementation of proposed IoT-blockchain platform. Hyperledger is a big project containing several distinct sub projects, hence, in this paper, we only utilize two of them namely : Hyperledger Fabric which is permissioned distributed ledger framework designed for enterprise application and Hyperledger Composer which is additional tools for building business network application on top of Hyperledger Fabric. The implementation presented in this section will be divided into three main parts : chaincode model for defining ledger's data format and its corresponding transaction, chaincode permission for taking care of privacy and chaincode transaction processor for system logic part.

A. Chaincode Model

The chaincode model contains the information related to asset data structure, participant allowed to access this asset as well as the transaction for altering the value of asset as presented lin Listing 1. At first, we define the asset's data structure using special *asset* keyword. Here, we represent the data sensor identifier and its content as a string. Then, we define the allowed participant in the ledger by using *participant* keyword. In the end, we define transaction method called *UpdateSensorData* which represents a chaincode function to change a sensor value in blockchain. Notice that, the detail implementation of this function is presented in Section III-C.

Listing 1: Chaincode Model Implementation

```
1  namespace org.aryastorage.arya
2  asset Datasensor identified by datasensorId {
3   --> Device device
4   o String datasensorId
5   o String datasensor
6  }
7  participant Deviceowner identified by
       deviceownerId {
8   --> Datasensor datasensor
9   --> Device device
10  o String deviceownerId
11  o String namaowner
12 }
```

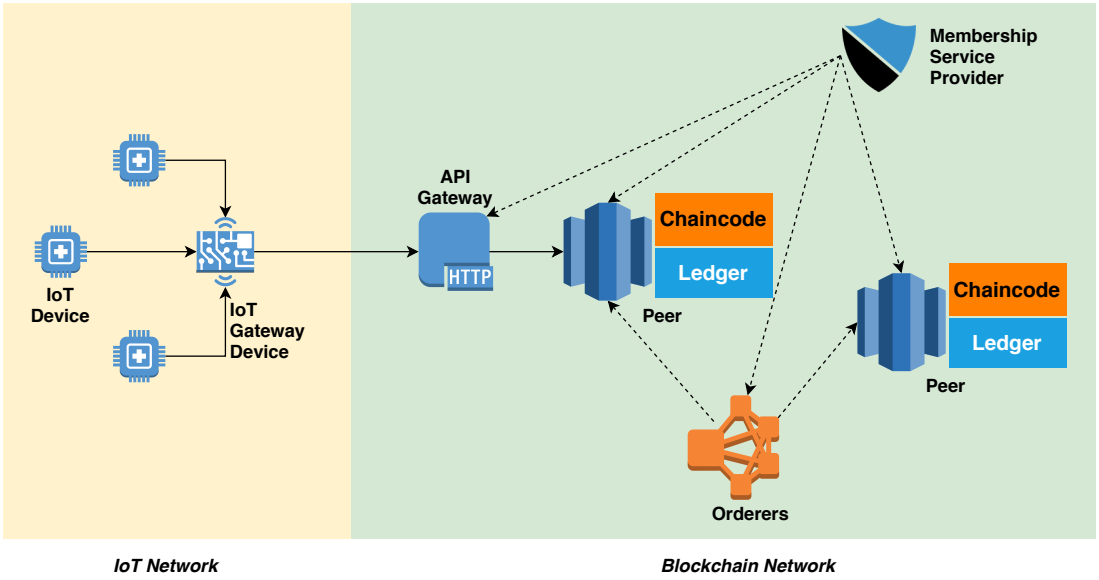

Fig. 1: General Environment.

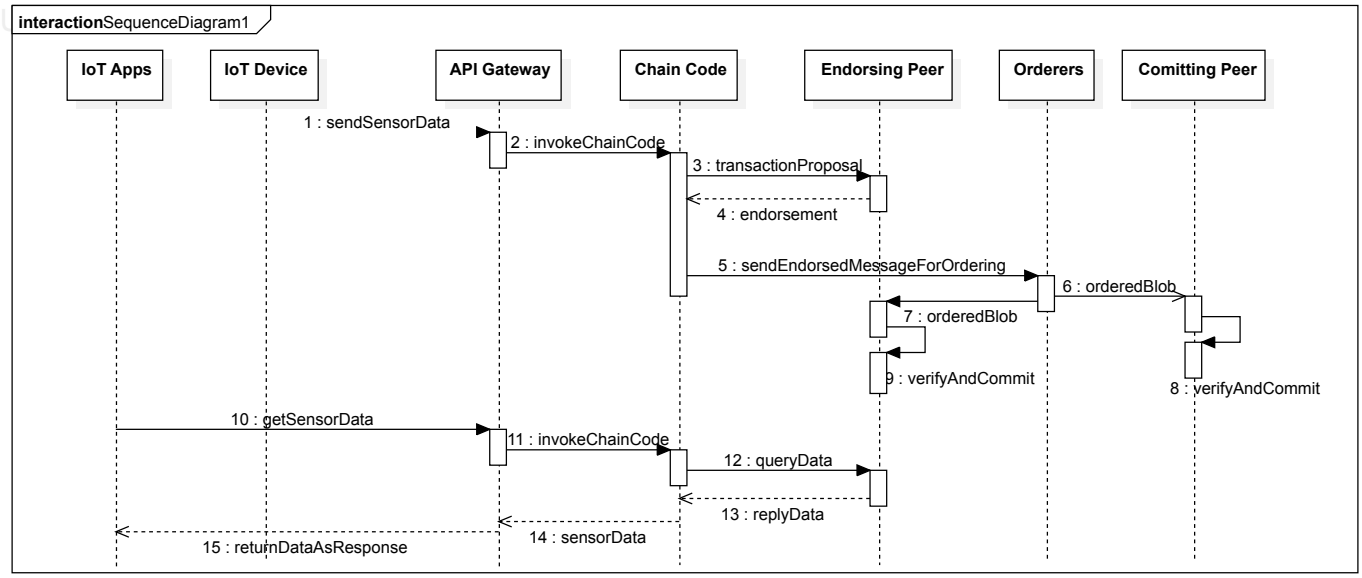

Fig. 2: Sequence Diagram of Blockchain Transaction Flow.

```
13  participant Device identified by deviceId {
14   o String deviceId
15   o String namasensor
16  }
17  transaction UpdateSensorData {
18   --> Datasensor sampDevice
19   o String newdatasensor
20  }
```

B. Chaincode Permission

The chaincode permission contains the information related to authentication mechanism that is who is the actor allowed to access the ledger value as well as authorization mechanism which dictates what actions can be performed by each of actor. Proposed chaincode permission of this paper is presented in Listing 2. In here, we define several rules indicating the action allowed for each user in formof access control list (ACL). For instance, in rule *AdmintoSystem*, we allow the admin actor to perform all operation for all Hyperledger related system resources. However, the admin actor is not allowed to perform any data changes on the IoT stored data. In the other hand, in rule *DeviceToDatasensor* ,the device actor is authorized to create and update data sensor without being able to delete it.

Listing 2: Chaincode Permission

```
1  /*======= Admin ========*/
2  rule AdmintoSystem {
3    description: "Allow Admin to access system
         resources"
4    participant:
5   "org.hyperledger.composer.system.NetworkAdmin"
6    operation: ALL
7    resource: "org.hyperledger.composer.system.**"
```

```
8     action: ALLOW
9   }
10
11  rule DeviceToDatasensor {
12    description: "Allow Device to Create and Update
13   datasensor"
14    participant: "org.aryastorage.arya.Device"
15    operation: CREATE, UPDATE
16    resource: "org.aryastorage.arya.Datasensor"
17    action: ALLOW
18  }
19
20  rule DeviceToUpdateSensorData {
21    description: "Allow Device to UpdateSensorData"
22    participant: "org.aryastorage.arya.Device"
23    operation: CREATE
24    resource: "org.aryastorage.arya.
          UpdateSensorData"
25    action: ALLOW
26  }
```

C. Chaincode Transaction Processor

The chaincode transaction processor provides a mechanism to change the value of data sensor's asset on distributed ledger as presented in Listing 3. At first, the chaincode function receives transaction parameter. Once the registry data is retrieved from distributed ledger, the chaincode performs a data update on data sensor value. Notice that, in this case, the value is only changed in world state of ledger while previous values will still be maintained in transaction log database.

Listing 3: Chaincode Transaction Processor

```
1  function UpdateSensorData(tx) {
2     var olddatasensor = tx.sampDevice.datasensor;
3     tx.sampDevice.datasensor = tx.newdatasensor;
4     return
5  getAssetRegistry('org.aryastorage.arya.Datasensor
       ').then(function (assetRegistry) {
6       return
7  assetRegistry.update(tx.sampDevice);
8   })
9   .then(function () {
10        var event =
11 getFactory().newEvent('org.aryastorage.arya',
12 'SampleEvent');
13        event.sampDevice = tx.sampDevice;
14        event.olddatasensor = olddatasensor;
15        event.newdatasensor = tx.newdatasensor;
16        emit(event);
17  });
18 }
```

IV. Result and Analysis

In this section we present the result and analysis of the testing performed to the proposed IoT blockchain platform in term of privacy, data synchronization and write latency testing. As the testing environment, we deploy peers into three different virtual machines as presented in Fig. 3. The first virtual machine is dedicated to deploy the orderer peer while the other two are provided for deploying two different organizations, each of which has single certificate authority as well as two distinct endorsing and committing peers.

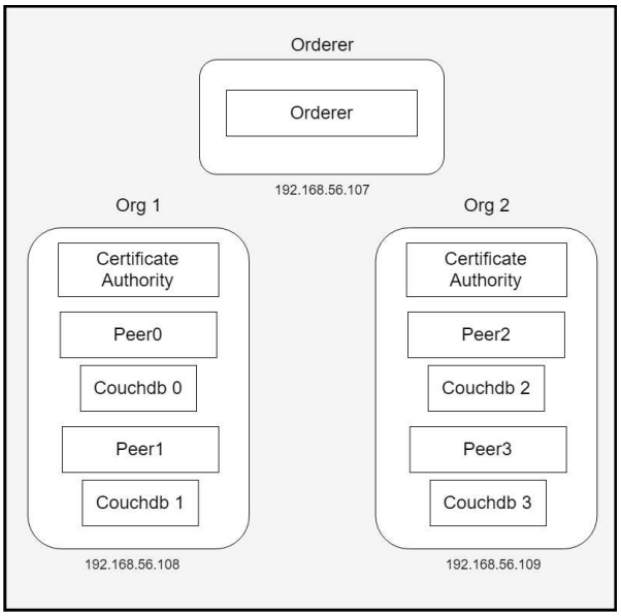

Fig. 3: Testing Environment.

A. Privacy Data Testing

This testing is performed to evaluate whether the proposed system can guarantee the privacy of stored sensing data as implemented in Section 2. At first, by using an administrator credential, we invoke the API gateway corresponding to data sensor value change using NodeRed API Tester. Fig. 4 shows that the data value change action is prohibited even for the administrator account. As we design the ACL in Section 2, the administrator account can only perform the access to the system related data including the device owner and its corresponding device while its stored data are not permitted. In this case, the value of data sensor can only be changed by device actor.

B. Data Synchronization Testing

This testing is performed to evaluate whether the proposed system can correctly synchronize the stored sensor data between peers in different organization. At first, we send a sensor data through API gateway deployed in virtual machine owned by organization 1 with IP address 192.168.56.108 as presented in Fig. 5. We then turn off that virtual machine and try to get the data from another machine owned by organization 2 with IP address 192.168.56.109 as shown in Fig. 6. As result, the stored data can still be accessed which proves that the data synchronization mechanism iw working properly.

C. Write Latency Testing

This testing is performed to measure the latency required to store sensing data with various amount of concurrent data sending i.e. 10, 20, 30, 40 and 50 data. In order to simulate that concurrent data sending, we utilize the JMeter application acting as device communicating with blockchain network through API gateway. Fig. 7 presents the result of latency measurement in seconds for different concurrent data sending.

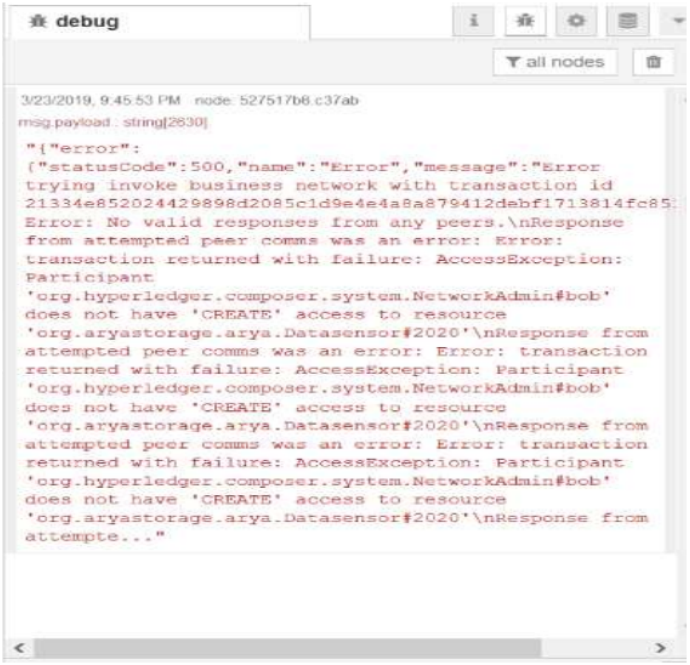

Fig. 4: The administrator is prohibited to change the sensor data value.

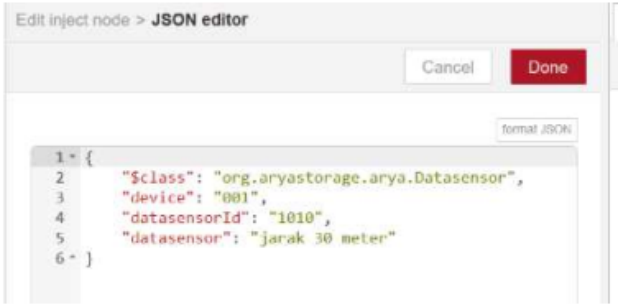

Fig. 5: The sensor data sent to peer in organization 1.

Fig. 6: The sensor data retrieved from peer in organization 2.

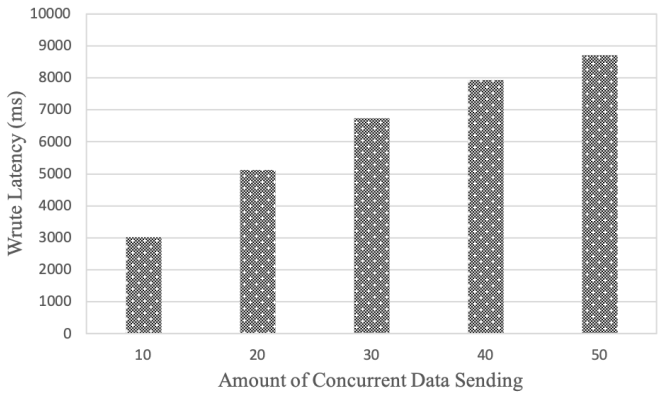

Fig. 7: Result of Write Latency Testing.

From the testing we can conclude that the latency increases linearly following the amount of concurrent data.

V. Conclusion

We proposed the design of IoT-to-Blockchain platform for integrating internet of things based smart healthcare system and blockchain network. The proposed system consists of five components including blcokchain-to-device interface, chain code execution interface, membership service provider (MSP), peers node and ordering node. The proposed system consists of five components including HTTP-based API gateway as device-to-blockchain interface, membership service provider (MSP), peers node and orderer. At first, the API gateway receives sensing data from IoT gateway device. Upon reception, the API gateway acting as client invokes chain code installed in each blockchain peer by emitting a transaction proposal to peers. Peers then simulate the execution and send back the endorsed data to the client i.e. API gateway. Collected endorsement from various peers are sent to orderer which then build a block containing an ordered transactions received from all clients and then broadcast that block to all peers. Each peer the performs a final verification before committing all ordered transactions on that block into its own local ledger. By using this mechanism, every peer in network can receive similar transaction orders which lead to identical ledger state changes. From the experiment, we conclude that the proposed system can perform a data synchronization between peers in different organization while still maintaining its privacy and reasonable write latency.

References

[1] Islam, SM Riazul, et al. "The internet of things for health care: a comprehensive survey." IEEE Access 3 (2015): 678-708.

[2] Baker, Stephanie B., Wei Xiang, and Ian Atkinson. "Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities." IEEE Access 5 (2017): 26521-26544.

[3] Ahmed, Sheeraz, et al. "Integration of cloud computing with Internet of Things and wireless body area network for effective healthcare." Wireless Systems and Networks (ISWSN), 2017 International Symposium on. IEEE, 2017.

[4] Plathong, Kuntinan, and Boonprasert Surakratanasakul. "A study of integration Internet of Things with health level 7 protocol for real-time healthcare monitoring by using cloud computing." Biomedical Engineering International Conference (BMEiCON), 2017 10th. IEEE, 2017.

[5] Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." Ethereum project yellow paper 151 (2014): 1-32.
[6] Androulaki, Elli, et al. "Hyperledger fabric: a distributed operating system for permissioned blockchains." Proceedings of the Thirteenth EuroSys Conference. ACM, 2018.
[7] Ahmed, N., H. Rahman, and Md I. Hussain. "A comparison of 802.11 ah and 802.15. 4 for IoT." *ICT Express* 2.3 (2016): 100-102.
[8] Ryu, Dae-Hyun. "Development of BLE sensor module based on open source for IoT applications." *The Journal of the Korea institute of electronic communication sciences* 10.3 (2015): 419-424.