Advances in Artificial Intelligence, Big Data and Algorithms
G. Grigoras and P. Lorenz (Eds.)

© 2023 The Authors.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA230817

A Blockchain-Based Decentralized IoT Sensing as a Service Framework for Smart Supply Chain

Chao FU[a], Shize ZHANG[a1], Chao PENG[a], Meng SUN[b], Xiaohong YU[c], Yu LI[d], Zhiqing HUANG[d]
[a] *China Electric Power Research Institute Co., Ltd., Beijing, China*
[b] *State Grid Materials Co., Ltd., Beijing, China*
[c] *State Grid Beijing Electric Power Company, Beijing, China*
[d] *Beijing University of Technology, Beijing, China*

Abstract. Smart supply chain services rely on the utilization of massive amount of data collected by sensor networks deployed in different enterprises. Sensing as a Service (S^2aaS) is a promising Internet of Things (IoT) business model pattern for data exchange. The current centralized IoT S^2aaS models are not suitable for IoT big data exchange due to the issues on privacy disclosure, single point of failure, data security, performance, etc. In this paper, we propose a blockchain-based decentralized framework for IoT S^2aaS for smart supply chain, which can ensure the IoT solution owners have full control of their data and securely exchange data with data consumers without intermediaries. We introduce the system model and the layered architecture of our proposed framework, based on which we give a concrete scheme, smart contract is used to perform the whole process of IoT S^2aaS. We implement a software prototype on Ethereum. Experiment results show the validity and effectiveness of our proposed solution.

Keywords. Smart supply chain, Internet of Things, decentralized sensing as a service model, blockchain, smart contract

1. Introduction

A supply chain is a sequence of steps that involve the procurement, planning, logistical planning, manufacturing, and distribution of a product[1]. The Internet of Things(IoT) plays an important role in the smart supply chain[2].Various types of IoT devices have been widely deployed in suppliers to obtain different types of data, such as production data, equipment data, etc. Smart supply chain applications process and analyze supply chain big data collected from suppliers to revolutionize many aspects of supply chain management[3].

Smart supply chain services rely on the use of a large amount of data obtained from different suppliers[4]. For example, the State Grid Corporation of China has created a "5E One Center" modern smart supply chain management platform, which plays a great role in ensuring the supply of materials for the State Grid Corporation. With a wide range

[1] Corresponding Author: Shize ZHANG, China Electric Power Research Institute Co., Ltd.; e-mail: zhangshize@epri.sgcc.com.cn

of IoT devices deployed in the suppliers, the "5E One Center" system can obtain real-time production status data, industrial control data, video monitoring data, and equipment testing data of the production line are obtained, achieving real-time monitoring of production and inspection processes, alerting key process issues, and providing optimization suggestions for production processes.

Sensing as a Service (S^2aaS) model is a vision and a business model that promotes data exchange between data owners (who own IoT solutions) and data consumers (i.e. government, regulatory authorities, companies, etc). Different S^2aaS solutions have been investigated [5]-[11]. The S^2aaS models proposed by these research work are cloud-based centralized models, which are not suitable for smart supply chain IoT data exchange due to the big challenges in data security and privacy, scalable data management, performance, etc. [5][6][7][19].

Blockchain is an emerging technology for decentralized and transactional data sharing across a large network of untrusted participants. It is becoming as publicly available common infrastructure for building decentralized systems and applications [20]. Smart contract depicts complex logics by program common process into code and represents the implementation of contract-based agreement [22]. Blockchain provides a secure environment for smart contract to execute. The blockchain and smart contract have been explored for IoT [12], especially in data secure exchange and access control [15]-[19]. The integration of Bitcoin in S^2aaS and the schema for the S^2aaS process of exchanging IoT data for cash using Bitcoin are discussed in [13][14]. And some applications are built on them, such as naming and storage system [20], health records management [21], crowdsourcing system [22], etc. Ethereum [23] is a widely used blockchain platform.

In this paper, we propose a decentralized IoT S^2aaS framework for smart supply chain based on blockchain. The main contributions are as follows:

- Propose a decentralized IoT S^2aaS framework for supply chain based on blockchain, which can ensure the IoT solution owners have full control of their data and securely exchange data with data consumers without intermediaries.
- Present a concrete scheme based on the framework. Smart contract is used to perform the whole process of IoT S^2aaS.
- Implement an S^2aaS software prototype based on Ethereum. Do experiments with real IoT data to validate our proposed solution.

The rest of the paper is organized as follows. Section II introduces the related work. Section III presents the decentralized IoT S^2aaS framework, which includes the system model and the layered system architecture. Section IV presents the system scheme, the S^2aaS smart contracts are introduced respectively. Section V introduces the prototype system. Section VI concludes the paper.

2. Related Work

2.1. S^2aaS model for IoT

In [5], the authors discussed the architecture design of a cloud-based Sensing as Service middleware. The middleware aims to provide API interfaces for Internet of Things applications to collect and analyze sensor data. The authors in [6] proposed Cloud of Things for sensing-as-a-service: a global architecture that enables remote sensing by

leveraging the global sensing resources of the Internet of Things (IoT), thereby expanding the scale of cloud computing. The author in [7] briefly discussed buying and selling IoT data via S^2aaS model. In [8], the author discussed the performance characteristics of the Sensing as a Service platform and designed for the platform's scalability. The author in [9] provided a systematic review of the Sensing as a Service platform, focusing on its architecture, current challenges, and future research directions. The authors in [10] proposed a more efficient and secure key protocol for the Sensing as a Service architecture in the Internet of Things. The results demonstrated that this solution can meet the security requirements effectively. The authors in [11] proposed a context-aware sensor search, selection, and ranking model under the S^2aaS architecture.

These research work focused on the S^2aaS model is cloud-based centralized model. Also the research work on the application of S^2aaS model for smart supply chain is very limited.

2.2. IoT Data Management Based on Blockchain

In [15], the authors described the capability of blockchain to maintain an immutable log of data exchanges as well as to perform access control. The authors in [16] proposed a framework for implementing blockchain to provide access control while maintaining the privacy of user data. The authors in [17] described a user-centric multi-level multiple granularity mechanism for IoT data sharing. The authors in [18] introduced a blockchain based distributed secure data storage system and the sharing of time-series sensor data. The authors in [19] proposed a network architecture for providing IoT data privacy via blockchains and IPFS.

These studies focused on how to use blockchain for IoT data secure exchange and performing access control.

3. Decentralized IoT S^2aaS Framework

In this section, we propose blockchain-based decentralized IoT S^2aaS framework. First, we introduce the overall system model. Then, we present the system architecture.

3.1. System Model

Our proposed blockchain-based decentralized IoT S^2aaS model is illustrated in Fig 1.

Data owners: They are suppliers who are in possession of data acquired by IoT systems. Data owners have full control of their data and have the choice of exchanging their sensor data with data consumers without intermediaries. They publish the sharing data's metadata on blockchain.

Data consumers: They are organizations or individuals that requests utilization of data owners' data. The data consumers post their data requests on blockchain. Data consumers can acquire IoT data from a larger number of data owners through certain data exchanging process, so they can use data far beyond their own collection and provide more accurate services.

S^2aaS control: The data exchanging process control logic is implemented as smart contracts in blockchain. The details will be described in section IV. And the service related key data will be stored on blockchain, such as access authorization data, data

exchange events, data owner and consumers' identity data, data registration, data requests, service credit score, etc.

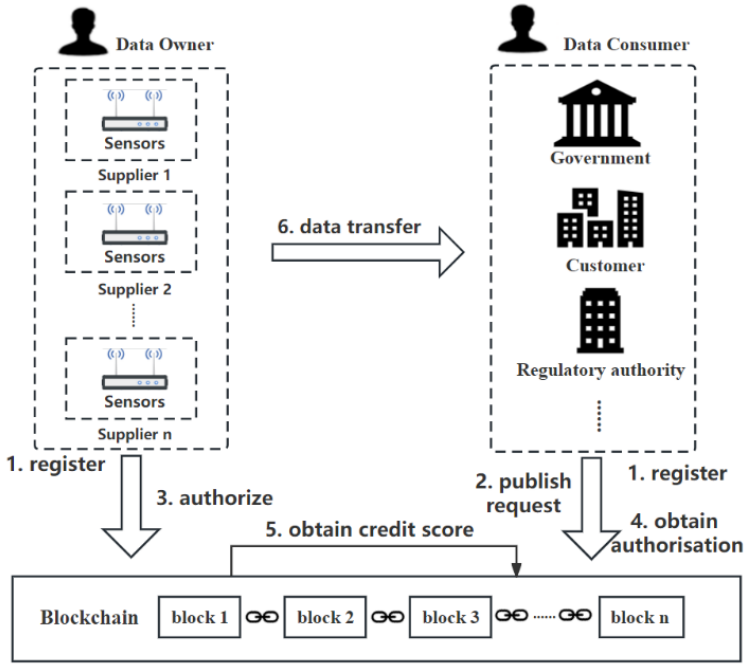

Figure 1. System model

The main data service process is described as the following steps. Note that the system is simply shown as blockchain in Fig.1, the whole system architecture is detailed in following part B.

Step1. Data owner registers sharing data's metadata in blockchain. Data information will be displayed in the data list by type for data consumers to select.

Step2. Data consumer initiates a data request to the data owner, and the request will be automatically added to the request list by system.

Step3. Data owner polls the request list, selects the request according to the description of the request. After that data owner provides an encrypted authority token based on the data consumer's public key.

Step4. Data consumer decrypts the token to obtain the access path of the data. At this point a complete data interaction process is completed and the data exchange details are recorded on the blockchain.

Step5. Data owner will receive a certain credit score as a reward. Credit score will be recorded on the blockchain.

Step6. Data consumer use the data access path in step 4 to get the data offchain.

3.2. System architecture

We designed the blockchain-based decentralized IoT S^2aaS system architecture, which contains five layers, namely application layer, service layer, contract layer, blockchain layer and storage layer as shown in Fig 2.

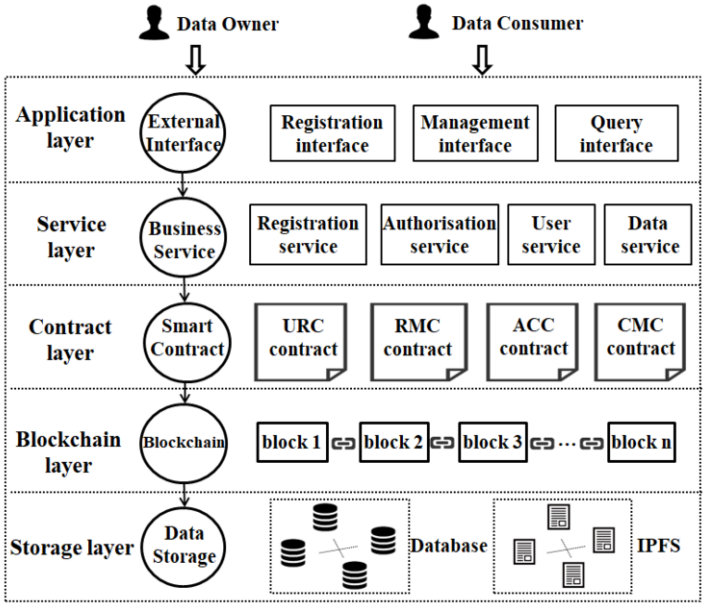

Figure 2. System architecture

Application layer: This layer provides the user interfaces for data consumers and data owners to perform S²aaS operations on different terminals (PC, mobile devices), also it connects to the blockchain network nodes.

Service layer: This layer is responsible for interacting with functions defined in the contract layer and providing corresponding service interfaces for the Application layer. It decouples the definition of S²aaS functions from the specific smart contract implementation.

Contract layer: This layer contains the smart contracts used to perform the whole process of IoT S²aaS. There are four kinds of smart contracts: the Registration Contract (URC), the Data Management Contract (DMC), the Access Control Contract (ACC), and the Credit Management Contract (CMC). Detailed information about these contracts will be expounded in section IV.

Blockchain layer: This layer mainly stores the sequence of S²aaS operations and it provides consensus on the order in which the operations were written. S²aaS operations are encoded in transactions on the underlying blockchain. It maintains an immutable log of data exchange activities.

Storage layer: This layer mainly refers to the distributed IoT data which is stored locally in separate data owners' databases. The traditional databases or IPFS may be used by data owners. Data consumers can accessed the IoT data after getting the authorization data from data owners.

4. A concrete implementation for S²aaS model

In this section, we present a concrete scheme of S²aaS model. Smart contracts are used t o perform the whole process of IoT S²aaS.

4.1. S^2aaS Smart Contracts

Four types of smart contracts are implemented, which are User Register Contract (UR C), Data Management Contract (RMC), Access Control Contract (ACC), and Credit M anag-ement Contract (CMC). Fig 3 depicts the structure and basic relationship of these smart contracts. UCC contracts and C-MC contracts are automatically created after the system serv-ice is started.

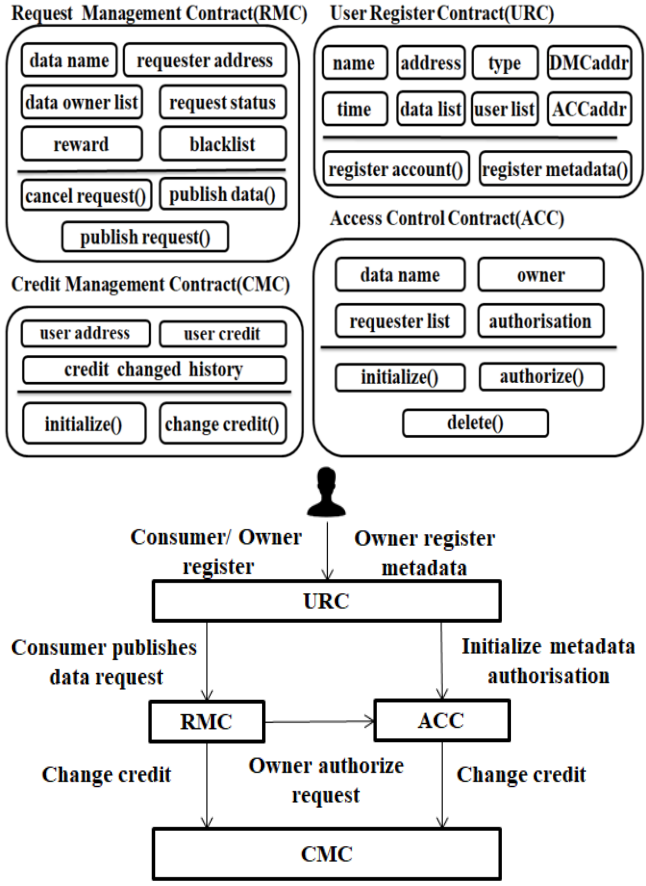

Figure 3. Smart contracts structure

User Register Contract. The URC contract is used to record information about user registration and data registration. When the user registers, the contract records the user's name, account address, and registration time, and also generates a user-specific RMC contract. When the data is registered, the contract records the name, category, and basic description of the data, and generates a data-specific ACC contract. In addition, the contract maintains two lists of user list and data list, respectively storing user information and data information

Request Management Contract. The RMC contract is responsible for recording and managing the operations related to the data request by the consumer. The contract records information about the data request, including the data name, the consumer

address, the request status, the owner list, and so on. When the consumer initiates a data request, the system first finds the RMC contract address belonging to it through the URC contract, and then changes the control request flow by modifying the field in the contract.

Access Control Contract. The ACC contract maintains access control rights for data. After the user successfully registers the data in the URC, the data permissions are initialized in the generated ACC contract, including the name of the data, the data owner, and the consumer list have their corresponding authorities. After the user obtains the access path of the data, the data can be downloaded by means of a URL or a P2P seed. The access interaction record for each data in the contract is written to the contract. on time, etc., the system will deduct the corresponding credit points. At present, the system's point contract only records the change process of the points when the account is related to the operation, and is not related to the actual incentive measures. In the future, we will gradually improve the credit score model to create more effective and diversified incentives.

Credit Management Contract. The CMC contract is automatically created when the system service is started, and the relevant credit points rules are established in the contract. When the user completes a request interaction normally, the system will automatically add credit score to the account according to the credit points rule. Conversely, if user violates relevant regulations, the corresponding credit score will be subtracted.

4.2. The Process of Proposed Decentralized S²aaS

This section is to present the process of decentralized S²aaS. There are four operations: user register, data register, request publishing and data authorization. Data owner and data comsumer interact with the blockchain by blockchain client.

As shown in Fig 4, we have designed a simple S2aaS operation sequence diagram. And also describes the corresponding smart contract updates.

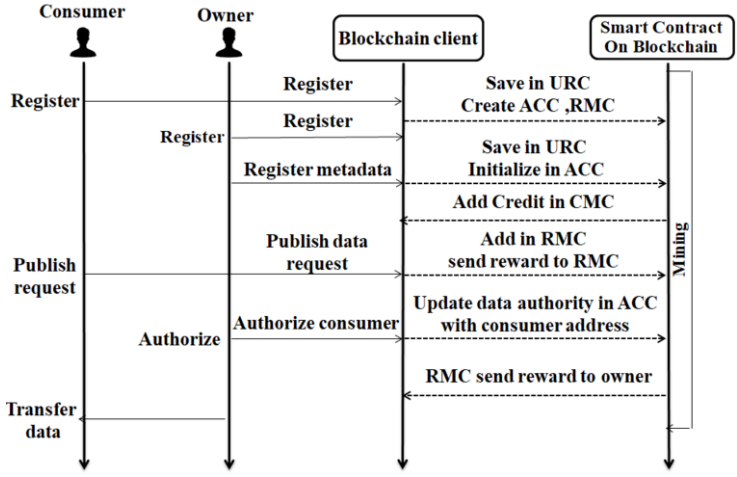

Figure 4. The process of S²aaS and smart contract updating

In the following sections we will elaborate on the core algorithms involved in the contracts.

(1) User registration algorithm

User provides the basic information and the public key U_p to the system, and writes the registration contract through the blockchain transaction. The exact flow of the algorithm is shown in Table 1.

Table 1. User registration algorithm

Algorithm 1: User Registration
Input: owner eth address U_a ,user nickname U_n ,user type U_t , user public key U_p ,user description U_d ,user email U_e ,user list U_l ,integration contract address IC_a
Output: DMC contract address DMC_a ,register result R_r ,error message Err_{msg}
1: R_r = false;
2: Err_{msg} = null;
3: **if** U_a exists in U_l **then**
4: $Err_{msg} \leftarrow U_a$ has been registered;
5: **return**
6: **end if**
7: U_{info} = {}
8: $DMC_a \leftarrow$ create DMC(U_a)
9: $U_{info} \leftarrow$ { U_a , U_n , U_t , U_d , U_p , DMC_a }
10: U_l **put** U_{info}
11: $IC_a(U_a) \leftarrow 0$
12: R_r = true
13: **return** DMC_a , R_r , Err_{msg}

(2) Metadata Registration Algorithm

The data owner Ua who has registered the user information can write the data source information into the URC contract. After the registration is successful, the system will automatically initialize the access authority for data. The exact flow of the algorithm is shown in Table 2.

Table 2. Metadata registration algorithm

Algorithm 2: Metadata Registration
Input: owner address U_a ,data name D_n ,data type D_t ,data description D_d , ACC contract address ACC_a ,Balance $,int-egration contract address IC_a
Output: Data list D_l ,register result R_r ,error message Err_{msg}
1: R_r = false;
2: Err_{msg} = null;
3: **if** U_a ($) < 1 **then**
4: $Err_{msg} \leftarrow U_a$ not enough to pay the transaction ;
5: **return**
6: **end if**
7: $Transfer_{status}$ = *transfer*(from $\leftarrow U_a$, to $\leftarrow TMA_a$, value $\leftarrow$ 1)
8: **if** $Transfer_{status}$ = true **then**
9: ACC_a (D_l) **put** { U_a, D_n, D_d,}
10: $TIC_a \leftarrow$ { U_a ,register data Integration }
11: **end if**
12: **return** ACC_a, R_r , Err_{msg}

(3) Request publishing algorithm

Data consumers need to fill in data types, request deadline Rf, rewards, etc. when publishing data requests. In this process, they need to pay the same amount of virtual currency to the DMC. If the payment fails, the request fails. The consumer can set a blacklist Bl, and the address in the blacklist cannot participate in the data exchange process. The exact flow of the algorithm is shown in Table 3.

Table 3. Request publishing algorithm

Algorithm 3: Request Publishing
Input: consumer address R_a ,blocklist B_l ,DMC contract address DMC_a ,request type R_t ,request reward R_r ,request description R_d ,request finish deadline R_f ,URC contract address URC_a , TIC contract TIC_a
Output: publish result P_r ,request status R_s , error message Err_{msg}
1: P_r = false
2: Err_{msg} = null
3: **if** R_a **not exist** in URC_a **then**
4: $Err_{msg} \leftarrow R_a$ has not been registered
5: **return**
6: **end if**
7: **if** $R_f \leq$ **now then**
8: $Err_{msg} \leftarrow$ Request finish deadline invalid
9: **Return**
10: **end if**
11: $R_{owner} \leftarrow R_a$
12: $Transfer_{status} \leftarrow transfer$(from $\leftarrow R_a$, to $\leftarrow URC_a$, value $\leftarrow R_r$)
13: **if** $Transfer_{status}$ **equals** *true*
14: R_{list} **put** { R_a , R_t , R_r , R_d , R_f }
15: $P_r \leftarrow$ *success*
16: $R_s \leftarrow$ *opening*
17: $TIC_a \leftarrow$ { R_a , R_s }
18: **end if**
19: **return** P_r and R_s and Err_{msg}

(4) Access contract algorithm

The data owner will poll the request list in the system and select the data request. After the published data is confirmed by the consumer, the system automatically encrypts the access permission identifier using the public key of the consumer, and writes the identifier to the corresponding consumer of the ACC. After that, DMC will send the rewards to the owner's account. At the same time, CMC will add the corresponding credit score to the consumer and owner. The exact flow of the algorithm is shown in Table 4.

Table 4. Access contract algorithm

Algorithm 4: Authorize consumer
Input: ACC contract address ACC_a ,URC contract address URC_a ,user address U_a ,consumer address R_a ,data name D_n ,authority key A_k ,data list D_l
Output: authorize result A_r , error message Err_{msg}
1: **if** D_n **not exist** in D_l **then**
2**:** $Err_{msg} \leftarrow D_n$ has not been registered
3: **return**
4: **end if**
5: owner $\leftarrow D_l(D_n)$(owner)
6: **if** owner **not equals** U_a **then**
7: $Err_{msg} \leftarrow$You don't have permission for this data
8: **return**
9: **end if**
10: ($encrypt_{Ak}$, $encrypt_{status}$)$\leftarrow$encrypt(A_k) with $URC_a(R_a)$
11: **if** $encrypt_{status}$ = true **then**
12: $ACC_a(D_n)(R_a) \leftarrow encrypt_{Ak}$
13: R_r = true
14: **end if**
15: **return** R_r and T_s and Err_{msg}

5. A S²aaS Prototype System

We implemented a IoT S²aaS prototype system and did experiments with real IoT data to validate our proposed solution. Fig 5 is the prototype system diagram. The top-most layer provides the interface used by data owner and data consumer to perform data exchange.

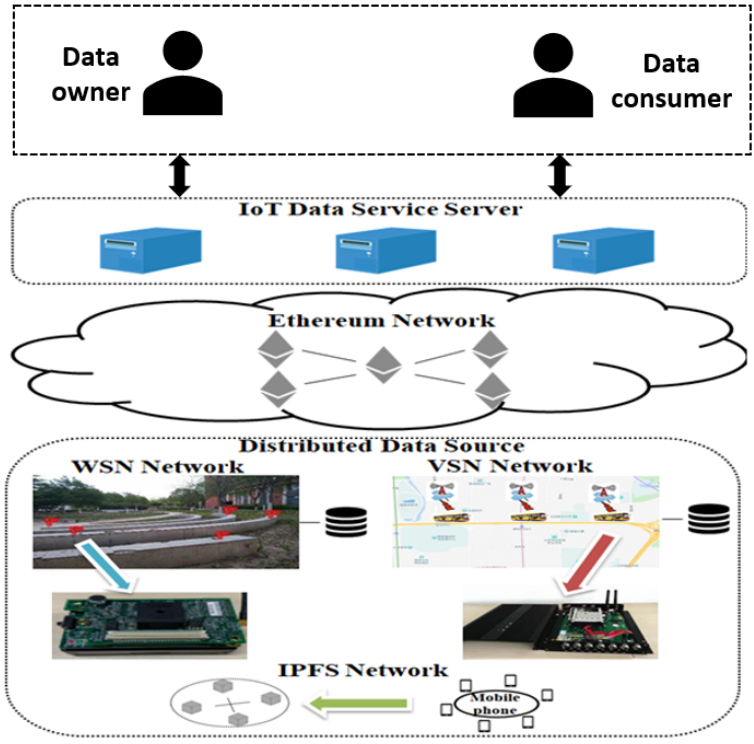

Figure 5. Prototype system diagram

IoT Data Service Server: System server was implemented with program language including nodejs, html and css. Physical devices used in the experiment were 10 Lenovo PC with 4g memory and 1G hard disk. The system platform interacts with Ethereum through the web3js library. Web3js provides an easy way to interact with smart contracts so that we don't need to implemented with additional code.

Ethereum Network: The blockchain network was composed of 10 Ethereum nodes on Alibaba Cloud host because Ethereum has a virtual machine EVM that is perfectly compatible with smart contracts. Our system interacts with the blockchain through the Ethereum client with geth v1.8.0. Smart contracts in the system are programed in Solidity and developed and tested on Remix.

Data Source: We had three distributed IoT networks to collect different IoT data.

(1) WSN Network: Around 50 Crossbow sensor nodes are deployed in Beijing University of Technology to collect temperature, humidity and illuminance data and upload them to the server maintained in Beijing IoT research center.

(2) VSN network: Beijing IoT research center developed a IoT terminal to collect vehicle status and city environmental data(temperature, humidity, etc). The buses

equipped with this terminal send the sensed data to Beijing IoT research center server through the 4G network.

(3) Mobile phone sensing network: We developed trial application running on smartphone to collect students' information, such as daily walking steps, trajectories, shopping hobbies, etc, the data is uploaded to IPFS Network.

Experiments: We used above three data sources as the data owners. And simulated several data consumers to request the IoT data collected by these IoT networks using the data exchange process listed in section III part A. The results show the successful data exchanges between data owners and simulated consumers.

6. Conclusion

In this paper, we propose a blockchain-based decentralized IoT S^2aaS framework. Smart contracts are used to perform the whole process of IoT S^2aaS. A series of algorithms based on smart contract were proposed to construct a concrete scheme under the framework. At last, a prototype system is built with real IoT networks to validate our proposed solution. So far we only implemented the basic process of S^2aaS,but there exists much more complex scenes to handle. Also we intend to test this framework in real supply chain scenarios and carry out user studies for an efficient evaluation of our system for future improvement.

References

[1] Abdul Zahra, Musaddak Maher, Ilhan GARIP, and Yazen S. Almashhadani. "Internet of Things-Based Smart and Connected Supply Chain: A Review." International Journal of Antennas and Propagation 2022 (2022).

[2] Abdel-Basset, Mohamed, Gunasekaran Manogaran, and Mai Mohamed. "Internet of Things (IoT) and its impact on supply chain: A framework for building smart, secure and efficient systems." Future Generation Computer Systems 86.9 (2018): 614-628.

[3] Phase, Avani, and Nalini Mhetre. "Using IoT in supply chain management." International Journal of Engineering and Techniques 4.2 (2018): 973-979.

[4] Azizi, Neda, et al. "IoT–blockchain: harnessing the power of internet of thing and blockchain for smart supply chain." Sensors 21.18 (2021): 6048.

[5] Alarbi, Muhamed, and Hanan Lutfiyya. "Sensing as a service middleware architecture." 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE, 2018.

[6] Abdelwahab, Sherif, et al. "Cloud of things for sensing-as-a-service: Architecture, algorithms, and use case." IEEE Internet of Things Journal 3.6 (2016): 1099-1112.

[7] Charith Perera. "Sensing as a Service (S^2aaS): Buying and Selling IoT Data," IEEE Internet of Things eNewsletters, November Issue, 2016, arXiv:1702.02380v1.

[8] Mukherjee, Tridib, et al. "Performance characterization and scalable design of sensing-as-a-service platform." Proceedings of the 30th Annual ACM Symposium on Applied Computing. 2015.

[9] YR, Sampath Kumar, and H. Champa. "An extensive review on sensing as a service paradigm in IoT: Architecture, research challenges, lessons learned and future directions." Int. J. Appl. Eng. Res 14.6 (2019): 1220-1243.

[10] Bentahar, Atef, et al. "Sensing as a service in Internet of Things: efficient authentication and key agreement scheme." Journal of King Saud University-Computer and Information Sciences 34.8 (2022): 5493-5509.

[11] Sherif Abdelwahab ; Bechir Hamdaoui ; Mohsen Guizani ; Taieb Znati, "Cloud of Things for Sensing as a Service: Sensing Resource Discovery and Virtualization" , 2015 IEEE Global Communications Conference (GLOBECOM).

[12] Konstantinos Christidis, and Michael Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things” IEEE Access, Vol:4, pp. 2292 – 2303, May 2016.

[13] Yu, Yong, et al. "LRCoin: Leakage-resilient cryptocurrency based on bitcoin for data trading in IoT." IEEE Internet of Things Journal 6.3 (2018): 4702-4710.

[14] Ensor, Alice, Sigrid Schefer-Wenzl, and Igor Miladinovic. "Blockchains for iot payments: A survey." 2018 IEEE Globecom Workshops (GC Wkshps). IEEE, 2018.

[15] Zyskind, Guy, Oz Nathan, and Alex Pentland. "Enigma: Decentralized computation platform with guaranteed privacy." arXiv:1506.03471 (2015).

[16] Guy Zyskind, Oz Nathan and Alex Pentland. Decentralizing Privacy: Using Blockchain to Protect Personal Data. In Proceeding of 2015 IEEE Security and Privacy Workshops (SPW), DOI: 10.1109/SPW.2015.27.

[17] Sayed Hadi Hashemi ; Faraz Faghri ; Paul Rausch ; Roy H Campbel “World of Empowered IoT Users” 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI).

[18] Shafagh Hossein, Anwar Hithnawi, and Simon Duquennoy. 2017. Towards Blockchain-based Auditable Storage and Sharing of IoT Data. arXiv preprint arXiv:1705.08230 (2017).

[19] Muhammad Salek Ali, Koustabh Dolui, Fabio Antonelli , “IoT Data Privacy via Blockchains and IPFS” IoT 2017,pp 22–25, Oct 2017.

[20] M. Ali,J. Nelson,R. Shea and MJ Freedman,Blockstack: A Global Naming and Storage System Secured by Blockchains,In USENIX ATC, pp.181-194, 2016.

[21] A. Azaria , A. Ekblaw, T Vieira and A. Lippman, MedRec: Using Blockchain for Medical Data Access and Permission Management, international Conference on Open and Big Data, IEEE, pp. 25-30, 2016.

[22] M. Li, J. Weng, A. Yang, W. Lu,Y. Zhang and L. Hou ,CrowdBC: A Blockchain-based Decentralized Framework for Crowdsourcing, 2018.

[23] Buterin, Vitalik. "Ethereum: platform review." Opportunities and challenges for private and consortium blockchains 45 (2016).