Decentralized Resource Sharing Platform for Mobile Edge Computing

Hongbo Zhang[1,2], Sizheng Fan[1,2], and Wei Cai[1,2(✉)]

[1] The Chinese University of Hong Kong, Shenzhen, Guangdong, China
[2] Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China
{hongbozhang,sizhengfan}@link.cuhk.edu.cn, caiwei@cuhk.edu.cn

Abstract. Recently, the Internet of Things (IoT) technology is booming in the industrial field. More and more industrial devices begin to connect to the internet. Compared with cloud computing, edge computing can well shorten the delay time on information transmission and improve the Quality of Service (QoS) of task computing, which promotes the development of the industrial Internet of things (IIoT) to some extent. The state-of-the-art edge computing service providers are specifically designed for customized applications. In our previous work, we proposed a blockchain-based toll collection system for edge resource sharing to improve the utility of these Edge Nodes (ENs). We provide a transparent, quick, and cost-efficient solution to encourage the participation of edge service providers. However, there exists a debatable issue since the system contains a centralized proxy. In this paper, we introduce the consortium blockchain to record the results of the service matching process in order to solve the issue. Besides, we propose a service matching algorithm for IIoT devices to select the optimal node and implement it using smart contract.

Keywords: Industrial Internet of Things · Mobile Edge Computing · Blockchain

1 Introduction

Internet of Things (IoT) can be regarded as a global network that consists of various connected devices that rely on sensing, communication, networking, and information processing technologies. It has made significant progress in recent decades [17]. IoT devices are widely used in industrial control, network equipment systems, public safety equipment, environmental monitoring, and many other fields. In order to satisfy the requirements of smart city, smart factory, and medical system, there are also a large-scale of IoT devices deployed to perform

This work was supported by Project 61902333 supported by National Natural Science Foundation of China, by the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS).

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2020
Published by Springer Nature Switzerland AG 2020. All Rights Reserved
X. Wang et al. (Eds.): 6GN 2020, LNICST 337, pp. 101–113, 2020.
https://doi.org/10.1007/978-3-030-63941-9_8

tasks such as monitoring, sensing, pre-processing, and real-time decision-making. More and more scientists hope to apply the IoT to the industry, which will help them achieve industry 4.0 [13].

Driven by the development of the 5G network, the industrial Internet of things (IIoT) is attracting growing attention all over the world [9]. In industry, IIoT devices are often used to monitor the regular operations on factory equipment. The future IoT system combined with 5G can monitor the vehicle data in real-time, and the data of the vehicle can be calculated at the edge to give the vehicle control instructions [14]. The IIoT paradigm in healthcare enables users to interact with various types of sensors via secure wireless medical sensor networks (WMSNs) [1]. In these application scenarios, IoT devices handle tasks with large amounts of data. However, those IoT devices are relatively weak in performance, and they are heterogeneous. So, they are not feasible to directly support the intensive computing load brought by the large-scale IoT data.

In order to handle the mentioned real-time data processing scenarios which require low latency and high Quality of Service (QoS), we introduce Mobile Edge Computing (MEC). MEC is a novel paradigm that extends the computing capabilities and storage resources from cloud computing to the edge of the mobile network [8]. It can reduce the significant delay in delivering the computing tasks to the cloud. Due to the dense geographical distribution, support for high mobility, and open platform [4], users can upload their computing tasks to Edge Nodes (ENs) no matter when and where. With the mentioned features, MEC can support applications and services with lower latency and higher QoS, which significantly promotes the development of IoT applications.

In our previous work, we designed and implemented EdgeToll, a blockchain-based toll collection system for heterogeneous edge resource sharing [15]. By leveraging the payment channel technique, EdgeToll provides a transparent, quick, and cost-efficient solution to encourage the participation of edge service providers. The payment channel is an efficient way to trade for multiple frequent transactions between two stakeholders. It requires stakeholders to deposit tokens and set up the receiver in this channel first. We set the payment channel as uni-directional, which means that only the receiver of the channel can withdraw coins. Instead of building a payment channel directly between users and edges, we introduce a proxy to handle payment delivery. In the payment stage, the proxy receives a signature on the agreement of splitting coins from users and then sign the same amount signature to edge computing services providers based on the address in the public blockchain network. Because the verification of signature and the delivery of payment signature are all operations with nearly no cost, the payment channel can reduce the cost of public transactions on the public blockchain.

However, it is quite controversial to introduce a centralized proxy in our system, which violates the decentralization spirit of the blockchain. As a third party, the proxy is responsible for the service matching process. The system might be vulnerable to have a centralized proxy. It might cause significant collusion if the proxy is unsupervised. It is possible that proxy colludes with one of the edge

service providers and prefers to recommend that provider's ENs to users. To solve the mentioned issue, we construct a consortium blockchain to record the service matching results on the consortium blockchain. We combine different edge service providers and proxy in this consortium.

Compared with the public blockchain, the consortium blockchain has many advantages, such as higher efficiency, higher scalability, and more transaction privacy. Rather than having all nodes joining the consensus process, the Practical Byzantine Fault Tolerance (PBFT) consensus process of the consortium blockchain is controlled by a set of selected nodes. At least 10 out of 15 nodes in the consortium need to sign and approve the block for it to be valid [2]. In our case, we select the proxy and edge service providers as the consensus nodes. In other words, every matching result needs to be signed and approved by at least two-thirds of the consortium.

To better attract IIoT devices and edge service providers to use our system, we deploy a service matching smart contract on the consortium blockchain, which is convenient for both IIoT devices and edge service providers. As buyers, the IIoT devices call the smart contract to search for the recommended ENs to handle computational tasks. As sellers, the edge service providers call the smart contract to record their ENs' location information. After receiving the information on the task, the proxy recommends the nearest EN according to the location of the IIoT device, which is an excellent way to reduce the time-consuming. Moreover, the smart contract can handle the above situations automatically, and the service matching results are recorded as transactions on the consortium blockchain.

The rest of the paper is organized as follows. We review related work in Sect. 2 and illustrate the system model in Sect. 3. Then, we present the technical design of the blockchain framework in Sect. 4. The test-bed implementation of the proposed system is shown in Sect. 5 to validate our system. In Sect. 6, we conclude our work and have a discussion about future work.

2 Related Work

2.1 Cloud and Edge Integration

Integrating edge to cloud platform involves a series of research topics in data and computational offloading. Traditional approach offloading schemes adopt virtualization techniques to host multiple copies of virtual machines in both clouds and edges [12]. At the same time, another group of researchers has investigated the possibility of dynamic code partitioning [3,6]. However, despite the form of offloading, the ENs intrinsically provide resource services for end-users through direct network connectivity. In this work, we assume the end-users are requesting micro-services installed in the ENs to simplify our model.

2.2 Blockchain in Edge Computing

Many existing works in MEC adopts blockchain for various purposes. For example, [11] presents an in-home therapy management framework, which leverages

blockchain to preserve the therapeutic data privacy, ownership, generation, storage, and sharing. [7] proposes a blockchain-based framework for video streaming with MEC, which uses blockchain to build decentralized peer-to-peer networks with flexible monetization. To incentivize users with no mutual trust and different interests, [16] uses the reward-penalty model to align incentives in the ecosystem on MEC. Meanwhile, they implement the model using the blockchain smart contract to solve the high centralization problem in the ecosystem.

2.3 Payment Channel

The payment channel [10] is designed for "off-chain" transactions to overcome the long response latency and the monetary costs introduced by frequent transactions. It allows users to exchange tokens for multiple times with a minimum number of smart contract invocations. The state-of-the-art payment channels can be classified into two types: uni-directional payment channel and bi-directional payment channel. A uni-directional payment channel only allows single directional transactions, while a bidirectional payment channel [5] allows both parties to send transactions. The duplex payment channel is composed of two unidirectional payment channels, which allows transactions to be sent from both directions.

3 System Model

3.1 System Overview

In this section, we start by basically introduce the previous system. The system contains three types of users, including proxy, users, and ENs from different companies. Before the previous system showed up, users need to register different companies' accounts to use their ENs' computing resources, which is quite inconvenient for users. Hence, the previous system builds up payment channels between proxy and users and between the proxy and ENs. In this way, users only need to register one public blockchain address and pay the bills to proxy through the payment channel. The proxy then sends the tokens to the selected ENs through the payment channel. Besides, the payment channel is implemented in a smart contract. In this way, the previous system can provide a convenient, low cost and transparent payment platform for edge computing.

Based on the existing system, we use consortium blockchain to solve the centralized proxy problem and introduce a new type of system user, which is the company. Companies take control of different ENs. One EN belongs to only one company. As illustrated in Fig. 1, ENs, proxy, companies, and users all have their specific address on the consortium blockchain. In our case, users are the IIoT devices, and companies are the edge service providers. The consortium blockchain has several functions.

First, we can implement the service matching through the smart contract and deploy it on the consortium blockchain. After being deployed on the blockchain,

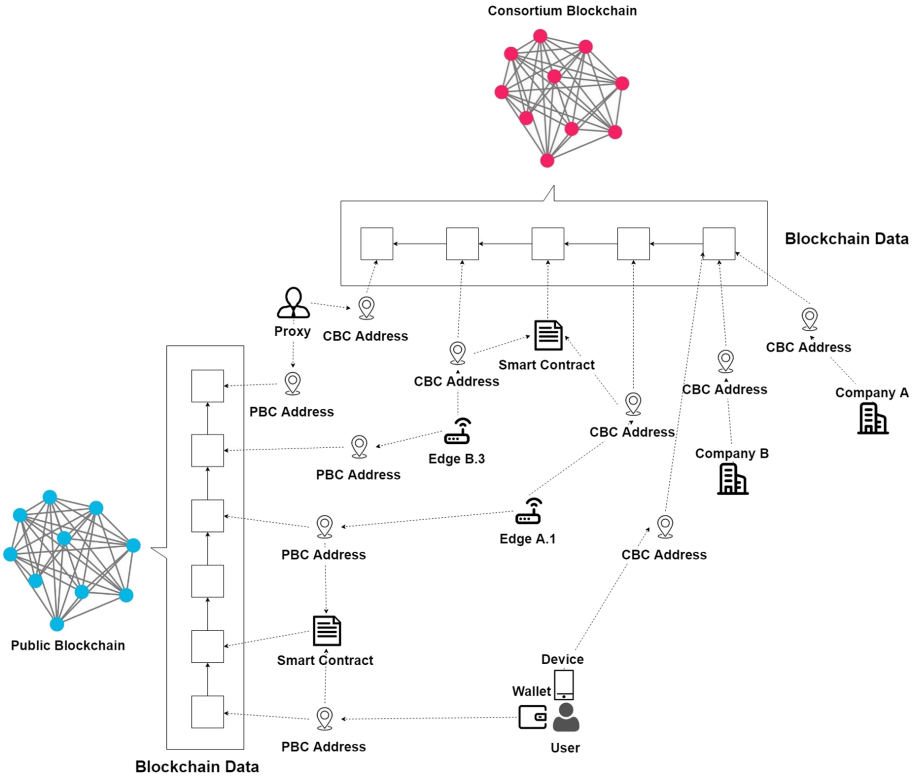

Fig. 1. System architecture

the smart contract has a specific address. Through the address, ENs can call the smart contract to record their locations to rent their surplus computing resources to devices that need computing resources. Devices can send their location to the smart contract to find the recommend EN and rent for the computing resources from the recommend node. Through the smart contract, the whole procedure of services matching can be done automatically. The result of an auction will be sent back to the ENs and devices which join the auction. To better demonstrate the procedure of using our system, we draw the sequence diagram in Fig. 2. Compare to the public blockchain, the performance of consortium blockchain is much better, and the cost of calling or deploying smart contract is zero, which save the costs of paying the gas fee. Meanwhile, the consortium blockchain reserves the characteristics of transparent, traceable, and unalterable. All the transactions, such as calling and deploying the smart contract, will be recorded on the consortium blockchain.

Second, the system can provide a more decentralized proxy with the consortium blockchain. In other words, the result of the service matching process needs to be supervised. As a third-party platform, proxy recommends ENs according

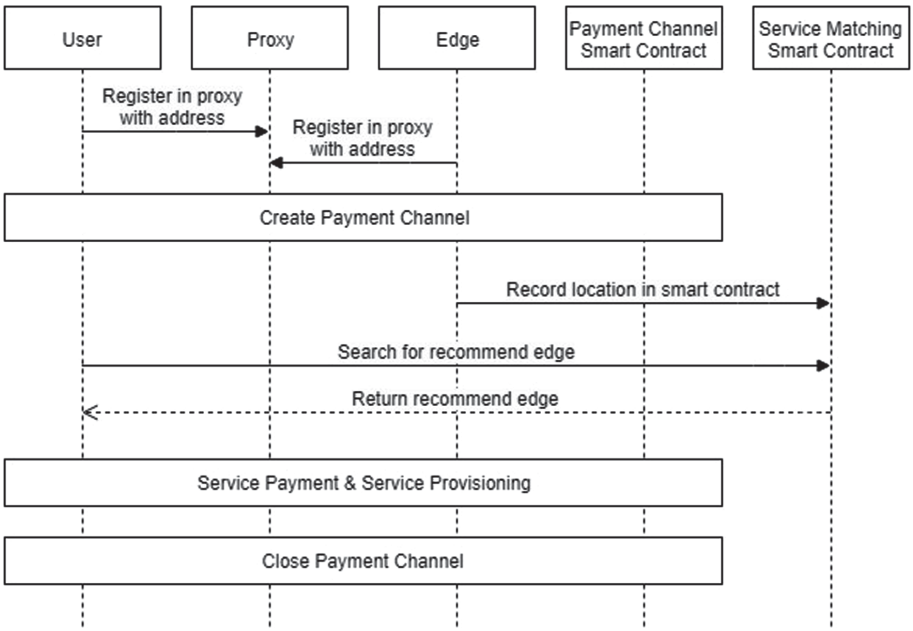

Fig. 2. Sequence diagram

to the list of ENs given by devices. Devices regularly follow the recommendation from proxy and bid for the computing resources from that ENs. However, the fairness of recommendation results can not be guaranteed since the whole procedure is entirely decided by proxy. If we do not deal with the situation, it will lead to a centralized proxy in our system, which violates the decentralization spirit of blockchain. In order to handle this centralized part of our system, we introduce consortium blockchain with the PBFT consensus algorithm.

4 The Blockchain Framework

We implement a decentralized edge computing resource sharing platform combining the advantages of two types of blockchain in our system design.

4.1 Software Architecture

Figure 3 illustrates the software architecture for the system.

4.2 Consortium Blockchain

Group Member Management. In order to better manage the system, the nodes on the blockchain are divided into two groups, which are sealers and

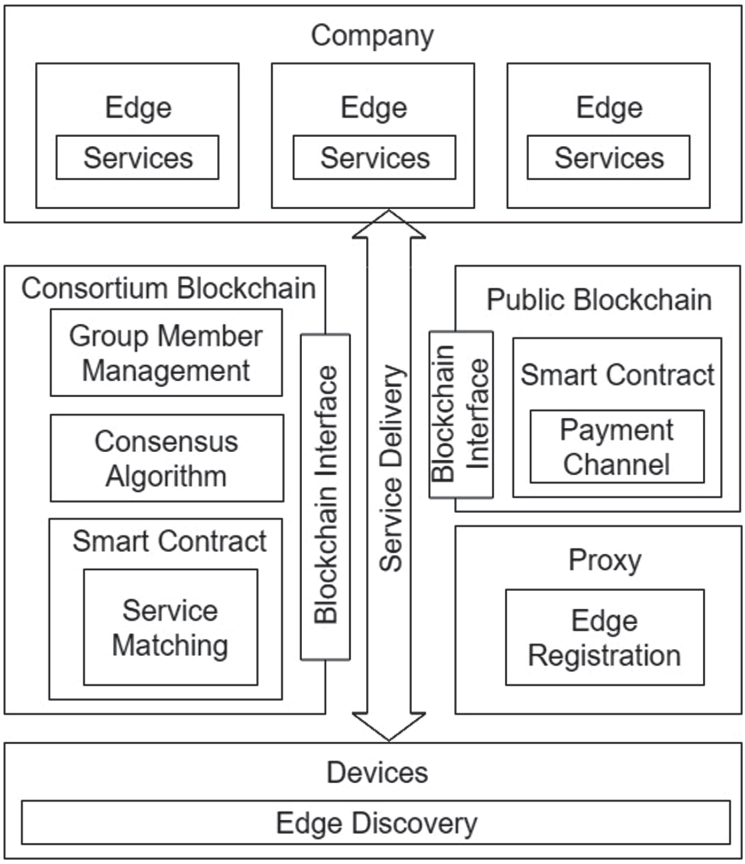

Fig. 3. Software architecture

observers. For both observer nodes and sealer nodes, they can send transactions in the consortium blockchain. The transactions are waiting in the transaction pool to be sealed into blocks. Compared to observer nodes, sealer nodes take part in the consensus process while sealing transactions into blocks. In our system, there are four types of system users, including proxy, devices, companies, and ENs. Different ENs belong to different companies. They are divided into two groups according to their types. For proxy and companies, they are the sealer nodes in the consortium blockchain. While for devices and ENs, they are the observer nodes in the consortium blockchain. Under this kind of classification, it is easier to manage the system when newcomers are accessing the system. For new devices and ENs, they can be directly added to the consortium blockchain after registration because they do not join the consensus process. They need to provide identifying information during registration to prevent DDoS attacks. However, the registration of the sealer node is stricter. Because sealer nodes

take control of the consortium blockchain. It requires agreements from more than two-thirds of the companies in the consortium.

Consensus Process. In the consortium blockchain, we use the PBFT consensus algorithm for the consensus process. Compared to other consensus algorithms, the PBFT consensus algorithm has benefits such as low latency, high efficiency, and high scalability. With low latency and high efficiency, the consortium blockchain can satisfy the demands of high-frequency transactions during auctions. The consensus process used in our system mainly includes three phases, including pre-prepared, prepare, and commit. Before the pre-prepare phase, one of the sealer nodes is selected to obtain the latest block and populate an empty block right after the latest block. Then, load transactions from the transaction pool and seal transactions into the block. The selected sealer node is selected in turn to guarantee fairness. After that, generate a prepared packet and broadcast it to other sealer nodes. In the pre-prepare phase, sealer nodes first need to check several requirements to judge whether the received prepare packet is valid. For example, they need to check whether the parent hash of a block is the hash of the highest block recently to void forking. If the prepared packet is valid, cache it locally to filter the replicated prepare packet. Then, generate and broadcast the signature package to state that this node has finished block execution and verification. In the prepare phase, sealer nodes need to check the validity of the received signature package. After receiving a valid signature package send from more than two-thirds of sealer nodes, the node starts to broadcast the commit package.

Similarly, in the commit phase, sealer nodes receive and check the validity of the commit package. The new block is finally confirmed after receiving a committed package sent from more than two-thirds of sealer nodes. By using the PBFT consensus algorithm, the system can remain stable as long as there are more than two-thirds of non-malicious nodes.

Service Matching. Similarly, we implement the service matching algorithm by using the smart contract and deploy it on the consortium blockchain. The smart contract also has a unique address and can be called by any other nodes in the consortium blockchain. Devices can call the smart contract according to the address and receive a recommended ENs according to the service matching algorithm. Since proxy and companies are responsible for the consensus process, the results of service matching are supervised by them. It can prevent the proxy from colluding with any other company. Companies will not allow the situation happened because it is related to their benefits. As a result, companies and proxy will supervise each other, and the fairness of service matching can be guaranteed.

5 Test-Bed Implementation

In order to better demonstrate our system, we implement a prototype and conduct several experiments on it. In this section, we introduce the enabling

technologies, the system deployment of the prototype, and demonstrate it with several shortcuts.

5.1 Test-Bed Specification

The edge computing server in our test-bed is Dell Precision 3630 Tower Workstation equipped with 16 GB RAM, Intel i7-9700 CPU, and NVIDIA GeForce GTX 1660. The edge computing server is also equipped with three wireless access points. The first one is TP-LINK WR886N, which adopts IEEE 802.11b/g/n standard with up to 450 Mbps data rate and 2.4 GHz radiofrequency. The second one is NanoPi R1, which adopts IEEE 802.11b/g/n standard with up to 450 Mbps data rate and 2.4 GHz radiofrequency. The third one is Phicomm K2P, which adopts IEEE 802.11b/g/n/ac standard with up to 1267 Mbps data rate and 2.4/5 GHz radiofrequency.

Figure 4 illustrates the test-bed implementation of our system. We use the workstation to work as an edge computing server. We equipped the server with three wireless access points to work as the ENs in MEC. The IIoT devices can submit their computational tasks through the wireless network. Then, edge computing server runs the computational tasks and return the results to IIoT devices.

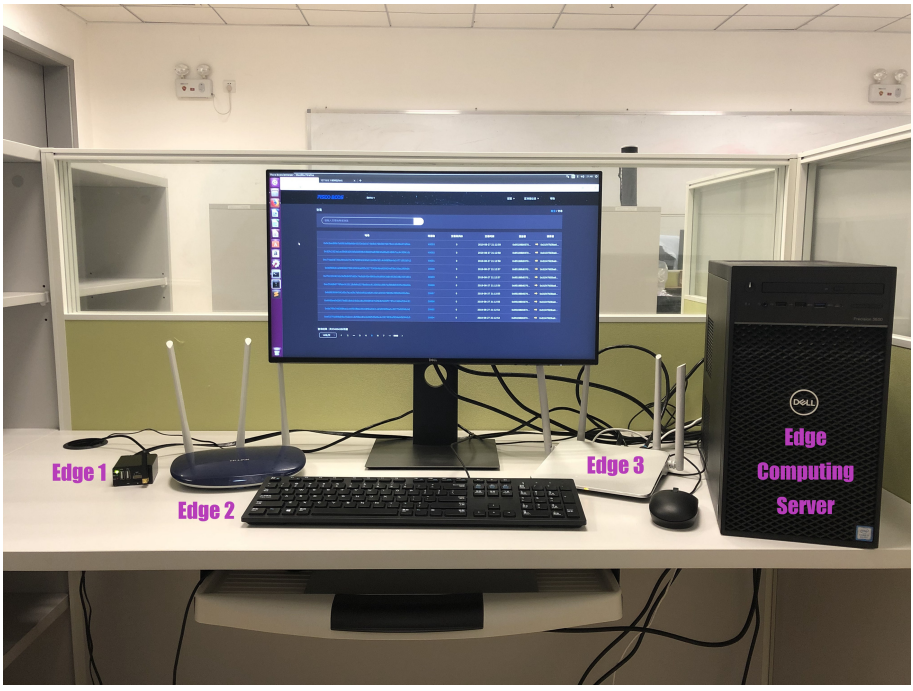

Fig. 4. Test-bed implementation

5.2 Enabling Technologies

To implement the prototype, we select various platforms and technologies. For the consortium blockchain platform, we choose FISCO-BCOS[1], which is an open-source consortium blockchain platform. For the programming language, we use Solidity[2] to write the service matching smart contract. Besides, FISCO-BCOS also provides an information port. With the information port, all system users can check the information of transactions and the smart contract. We choose information port as our client-side to demonstrate the information of the consortium blockchain. As for the interaction with consortium blockchain, we use the python-SDK[3] provided by the FISCO-BCOS.

5.3 Blockchain Deployment

We deploy the consortium blockchain in our local server. Initially, proxy deploys the service matching smart contract on the consortium blockchain. The smart contract has a unique address on the consortium blockchain. Both edge service providers and users can access the smart contract through the unique address. Edge service providers can record the information of their ENs by calling the addNote() function. As for users, they can get the recommended EN by calling the edgeMatch() function. To make our system more user-friendly and more convenient for users to use, we choose the information port as our client-side. After connecting to the local server through the wireless network, users can access the information port. Through the information port, users can easily obtain information on the consortium blockchain.

5.4 Demonstration

Figure 5 shows the information port. The upper-left part shows the current block number, total transactions, dealing transactions, and current PBFT view. As we can see from the figure, there are already 40012 blocks and 40012 transactions in the consortium blockchain. Next to it is the curve showing the transaction amount in the last 15 days. Since we do not have any transactions during the last 15 days, the curve stays flat. Below is the information of some nodes on the consortium blockchain. The information includes node ID, current block number, PBFT view, and node status. In the bottom-left part, it demonstrates several block information, including created time and sealer node of the block. Users can click on it and see detailed information on another page. Transaction information is shown in the bottom-right part. Similarly, users can click on it to achieve more information.

[1] http://fisco-bcos.org/.
[2] https://github.com/ethereum/solidity.
[3] https://github.com/FISCO-BCOS/FISCO-BCOS-DOC/tree/release-2/docs/sdk/python_sdk.

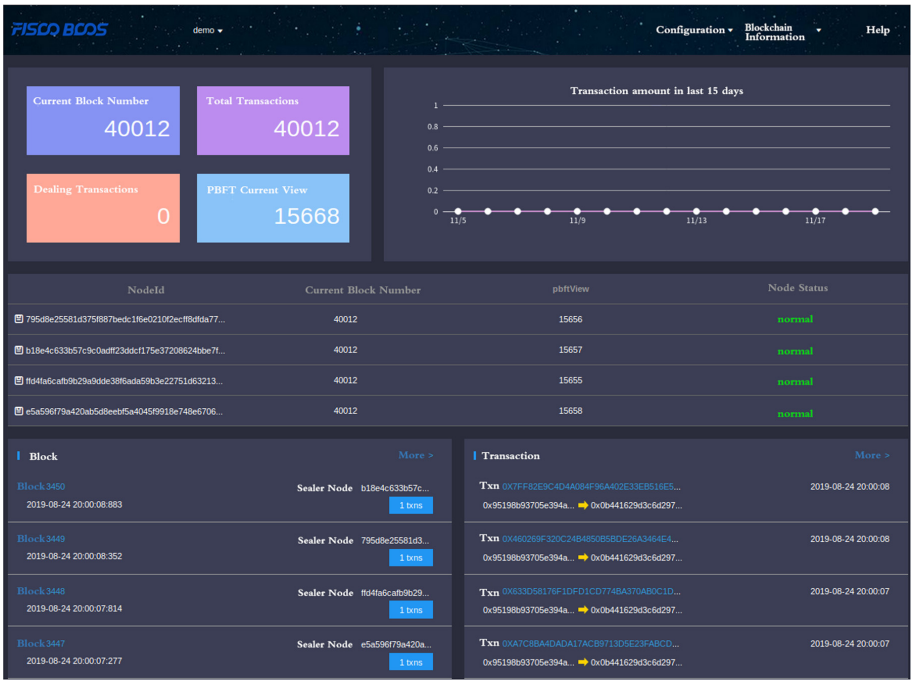

Fig. 5. Shortcut

6 Conclusion and Future Work

In our previous work, we provide a low-latency and cost-efficient solution for a decentralized, transparent, and auditable toll collection system by leveraging the payment channel technique. In this work, we demonstrate a decentralized toll collection system which solves the centralized proxy problem in our previous work. By adding consortium blockchain, we introduce a low-cost solution to solve out the centralized proxy problem. The decisions used to made by a centralized proxy can be implemented in the smart contract, and the smart contract will be deployed on the consortium blockchain. In this way, the results will be supervised by edge computing companies and proxy.

In the future, a more efficient and rational service matching model will be considered, and the proxy will consider the type of both ENs and tasks. In particular, the task allocation process will focus on high efficiency and low cost. In order to attract IIoT devices and edge service providers, a new dynamic pricing strategy will be proposed, which not only focuses on the incentive mechanism but also aims to improve the utility of ENs.

References

1. Al-Turjman, F., Alturjman, S.: Context-sensitive access in industrial internet of things (IIoT) healthcare applications. IEEE Trans. Ind. Inform. **14**(6), 2736–2744 (2018). https://doi.org/10.1109/TII.2018.2808190
2. Cai, W., Wang, Z., Ernst, J.B., Hong, Z., Feng, C., Leung, V.C.: Decentralized applications: the blockchain-empowered software system. IEEE Access **6**, 53019–53033 (2018)
3. Chun, B.G., Ihm, S., Maniatis, P., Naik, M., Patti, A.: CloneCloud: elastic execution between mobile device and cloud. In: Proceedings of The Sixth Conference on Computer Systems, pp. 301–314. ACM (2011)
4. Corcoran, P., Datta, S.K.: Mobile-edge computing and the internet of things for consumers: extending cloud computing and services to the edge of the network. IEEE Consum. Electron. Mag. **5**(4), 73–74 (2016)
5. Decker, C., Wattenhofer, R.: A fast and scalable payment network with bitcoin duplex micropayment channels. In: Pelc, A., Schwarzmann, A.A. (eds.) SSS 2015. LNCS, vol. 9212, pp. 3–18. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21741-3_1
6. Kosta, S., Aucinas, A., Hui, P., Mortier, R., Zhang, X.: ThinkAir: dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In: 2012 Proceedings IEEE INFOCOM, pp. 945–953. IEEE (2012)
7. Liu, M., Yu, F.R., Teng, Y., Leung, V.C.M., Song, M.: Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing. IEEE Trans. Wirel. Commun. **18**(1), 695–708 (2019). https://doi.org/10.1109/TWC.2018.2885266
8. Mach, P., Becvar, Z.: Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun. Surv. Tutor. **19**(3), 1628–1656 (2017)
9. Palattella, M.R., et al.: Internet of things in the 5G era: enablers, architecture, and business models. IEEE J. Sel. Areas Commun. **34**(3), 510–527 (2016). https://doi.org/10.1109/JSAC.2016.2525418
10. Poon, J., Dryja, T.: The bitcoin lightning network: scalable off-chain instant payments (2016)
11. Rahman, M.A., et al.: Blockchain-based mobile edge computing framework for secure therapy applications. IEEE Access **6**, 72469–72478 (2018). https://doi.org/10.1109/ACCESS.2018.2881246
12. Satyanarayanan, M., Bahl, V., Caceres, R., Davies, N.: The case for VM-based cloudlets in mobile computing. IEEE Pervasive Comput. **8**, 14–23 (2009)
13. Shrouf, F., Ordieres, J., Miragliotta, G.: Smart factories in industry 4.0: a review of the concept and of energy management approached in production based on the internet of things paradigm. In: 2014 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 697–701, December 2014. https://doi.org/10.1109/IEEM.2014.7058728
14. Utsunomiya, H., Kobayashi, N., Yamamoto, S.: A safety knowledge representation of the automatic driving system. Procedia Comput. Sci. **96**, 869–878 (2016). https://doi.org/10.1016/j.procs.2016.08.265. http://www.sciencedirect.com/science/article/pii/S1877050916320816. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 20th International Conference KES-2016
15. Xiao, B., Fan, X., Gao, S., Cai, W.: EdgeToll: a blockchain-based toll collection system for public sharing of heterogeneous edges. In: 2019 IEEE Conference on Computer Communications Workshops (INFOCOM 2019 WKSHPS) (2019)

16. Xu, J., Wang, S., Bhargava, B.K., Yang, F.: A blockchain-enabled trustless crowd-intelligence ecosystem on mobile edge computing. IEEE Trans. Ind. Inform. **15**(6), 3538–3547 (2019). https://doi.org/10.1109/TII.2019.2896965
17. Xu, L.D., He, W., Li, S.: Internet of things in industries: a survey. IEEE Trans. Ind. Inform. **10**(4), 2233–2243 (2014). https://doi.org/10.1109/TII.2014.2300753