Using Ethereum Blockchain in Internet of Things: A Solution for Electric Vehicle Battery Refueling

Haoli Sun[1] · Song Hua[1] · Ence Zhou[1] · Bingfeng Pi[1] · Jun Sun[1] and

Kazuhiro Yamashita[2]

1 Fujitsu research & development center , Suzhou, China
{sunhaoli, huasong, zhouence, winter.pi, sunjun}@cn.fujitsu.com
2 Fujitsu Laboratories, Kawasaki, Japan
y-kazuhiro@jp.fujitsu.com

Abstract. Internet of Things (IoT) technology has become more and more popular recently. However, due to the limited resources of IoT devices and the centralized system architecture, some severe issues remain difficult to solve, such as: overload of centralized server, single point of failure, and the possibility of malicious usage of personal information. Blockchain technology has achieved big success in cryptocurrency trading. It has many unique features, such as consensus mechanism, peer to peer communication, implementing trust without a trusted third party, and transaction based on smart contract. Blockchain appears to be suitable to help building a distributed and autonomic IoT system to overcome the aforementioned problems.

In this paper, we introduce an Ethereum blockchain based rich-thin-clients IoT solution to solve the problems caused by limited resources of IoT devices when adopting mining mechanism of blockchain in IoT scenarios. Rich clients and thin clients can both provide blockchain accessing and data collecting functions while only rich clients with more resources can perform mining process. Furthermore, based on the solution, we present an electric vehicle battery refueling system in which battery swapping approach is adopted. We also explain the rationality of our solution by experiments and compare our solution with other blockchain based IoT solutions. Our conclusion is that our blockchain-IoT solution is suitable for various IoT scenarios while avoiding the problems caused by the limited resources of IoT devices.

Keywords: Internet of Things (IoT), Blockchain, Ethereum, Battery Refueling.

1 Introduction

Blockchain was first introduced by Satoshi Nakamoto in 2008 as the underlying data structure of Bitcoin [1]. As its name suggested, a blockchain is a chain of blocks, in which each block contains a number of transactions which are hashed in a Merkle Tree [2]. By storing the hash value of the previous block, each block refers to its pre-

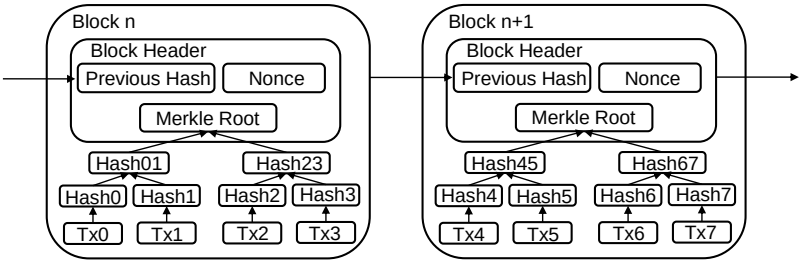

Fig. 1. Typical structure of a blockchain

vious block, forming a chain structure. Fig. 1 shows the typical structure of a blockchain. Together with peer-to-peer communication, consensus between miners such as Proof of Work (PoW), asymmetric encryption and digital signature, a blockchain system can provide a temper-proof and immutable value-transfer network which facilitates the booming of cryptocurrencies. This kind of blockchains which mainly used in cryptocurrencies can be defined as blockchain 1.0 [3].

Although Bitcoin supports scripts to define simple rules, the scripts are Non-Turing-Complete. In order to make blockchain suitable for more scenarios other than cryptocurrency, Ethereum [4] introduced smart contract which can be constructed with Turing-Complete programming languages (e. g. solidity [5]). Smart contracts are executable code stored on blockchain, defining what information to store and what transactions to execute. Theoretically, all transaction based state machines can be built by smart contracts. Fig. 2 shows the mechanism of smart contracts. If an application is built only by Ethereum-like blockchain without cyber-physical interaction or other external facilities, it is defined as blockchain 2.0, which means blockchain based economic, market, and financial related applications [3].

Blockchain 3.0 means blockchain based applications beyond the scenarios of blockchain 2.0 [3]. For example, blockchain based naming systems [6] [7], health caring systems, IoT systems [8] [9] [10] [11] [23] [24] and so on. As for blockchain based IoT systems, besides the blockchain related factors such as consensus method, smart contract support or not and state machine management, IoT related factors such

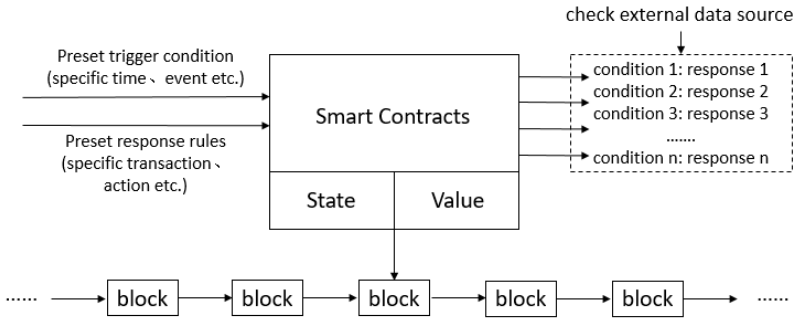

Fig. 2. The mechanism of smart contracts

as the constraint of computing power, memory of devices and network bandwidth, and security of devices must also be considered. However, the problem that the limited resources of IoT devices cannot well support the mining process of blockchain nodes occurs frequently. In this paper, we introduce a rich-thin-clients solution for blockchain based IoT systems to solve the problem.

The remainder of this paper is organized as follows. In Section 2, we introduce some blockchain based IoT solutions. In Section 3, we propose an Ethereum blockchain based rich-thin-clients IoT solution and an implementation of EV battery refueling system using the proposed solution. A performance experiment is also introduced in Section 3 to prove the efficiency of the proposed solution. In Section 4, we compare our blockchain based IoT solution with others. In Section 5, we provide conclusions and future work.

2 Blockchain based IoT Solutions

IoT means the Internet to which "things" (devices, sensors, actuators, etc.) are connected. The data from the physical world is gathered by sensors, delivered through the Internet. The users (or control unit) of IoT systems can analyze the gathered data to discover trends or patterns, or change the status of actuators based on the data. IoT technology has been developing rapidly these years. Gartner, Inc. forecasted that by 2020, 20.8 billion IoT devices will be connected to the Internet [17].

However, due to the limited computing power, storage and network bandwidth of IoT devices [18] and the centralized system architecture, some severe issues remain difficult to solve. Minhaj and Khaled listed out the security issues of different layers in IoT systems [18], such as "jamming adversaries", "insecure physical interface", "insecure neighbor discovery", "sinkhole and wormhole attacks", "sybil attacks" and so on. Besides, Nir [19] summarized IoT challenges mainly caused by the centralized architecture of current IoT systems: "costs and capacity constraints", "deficient architecture", "cloud server downtime and unavailability of services", and "susceptibility to manipulation".

Marco et al. [20] suggested that a hierarchical architecture, in which additional applications should be built on top of an underlying blockchain, is suitable for building IoT systems, e.g., the architecture of Blockstack [7] which is proposed by Muneeb et al. Blockstack is a global naming and storage system, by which users can register name-value pairs. The name-value pairs are generated according to particular private keys, and only the owner of the private keys can perform writing or updating operations on the name-value pairs. Blockstack separates "control plane" from "storage plane", which makes Blockstack outperforms Namecoin [6]. Furthermore, the authors believe that various state machines can be built in the "Control plane" of BlockStack.

Ali et al. [8] proposed a blockchain based IoT solution for smart homes, and they conducted further researches about their proposed solution [21] [22]. The system architecture of their solution is also hierarchical, consisting of "smart home", "overlay network", and "cloud storage". "Smart home" contains all smart devices in the home, local blockchain, smart home manager (SHM) [22], and local storage. "Overlay net-

work" connects smart homes and cloud storages, and it provides distributed trust by overlay blockchain. "Cloud storage" is used to store data from smart home in order to provide data to other services on the Internet. Considering IoT devices do not have enough resources to support PoW, the authors designed their proposed system architecture without PoW. In a smart home, local blockchain is centrally managed by the owner through SHM. The nodes of overlay network could be SHMs and other devices with relatively high resource. The nodes are grouped in clusters to decrease network overload, and one of the nodes in each cluster is elected as a cluster head (CH). CHs are responsible for evaluating if other CHs are trustable by maintaining a trust rating based on direct and indirect evidence. The overlay blockchain is maintained by the CHs, since consensus mechanism is not adopted in this system, forking of blockchain is permitted. The authors came to the conclusion that the architecture they proposed keeps the benefits of traditional blockchain such as privacy and security while eliminating PoW of traditional blockchain for better performance.

Seyoung et al. [9] constructed an automatic electricity usage adjustment system based on Ethereum. The system consists of the Ethereum blockchain, a smart phone and three Raspberry Pis respectively representing an electricity meter, an air conditioner and a lightbulb. Since the smart contracts can define the behaviors of IoT devices, a user can set up policies of how the devices work using a smart phone. For example, user can define a threshold of electricity usage, once the threshold is reached, the air conditioner will change to energy saving mode. The authors used RSA algorithm to provide public key and signature functions instead of Ethereum accounts for fine-granularity.

Besides, Kamanashis et al. [10] proposed a general architecture for a blockchain based smart city solution, they utilized different blockchain systems in their solution, such as using both Ethereum and NXT in communication layer and using both permissionless and permissioned distributed ledgers in database layer, which may be an inspiring idea for utilizing blockchain systems in complex IoT scenarios.

In summary, the IoT issues which can be solved by blockchain are mainly the issues caused by the centralized architecture of current IoT systems. Some of the issues are listed:

- Central server failures: failures may be caused by fault of software or attacks.
- Single point of failure: a compromised device can cause failures of entire system.
- Lack of privacy: personal information saved in central server may be abused.

Blockchain can solve the above issues because its decentralized architecture can prevent central server failures or single point of failure, and its public-private key encryption can provide pseudonymity [20] to protect personal information at some degree. However, the issues caused by the limited resources of IoT devices still remain difficult to solve: on one hand, most of the IoT devices do not have enough resources to support PoW, on the other hand, if a device is not a node of blockchain network, its security and identity is difficult to be guaranteed.

Besides, we found that the following factors are essential for constructing blockchain based IoT systems:

- What underlying blockchain system should be used?
  - What consensus method should be used?
  - Is Turing-Complete programming necessary or not? (Requiring smart contracts or not?)
  - How to manage state machines?
  - How should security and privacy be guaranteed?
- How to combine blockchain with IoT?
  - How to enable cyber-physical interaction?
  - If IoT devices need to be blockchain nodes or not?
  - How to design the system architecture? (hierarchical architectures are often adopted [7] [8] [10] [11] [24])
  - How to design the topology of IoT devices and blockchain nodes? (Such as clusters in [8], a cluster is a star topology. And the topology between CHs is P2P)

3 Ethereum Blockchain based Rich-thin-clients IoT Solution

In this section, we propose an Ethereum blockchain based rich-thin-clients IoT solution. We designed a rich-thin-clients architecture to solve the aforementioned dilemma between limited resources of IoT devices and the concerns for centralized architecture. The thin clients, which are responsible for user interaction and IoT data collection, can be considered as IoT devices with constrained resources; the rich clients, which are thin clients plus full blockchain nodes, can be considered as devices that have resources greater than or equal to personal computers. We use a private Ethereum blockchain network built by ourselves as our underlying blockchain system. Not only because it supports smart contracts by which we can design relatively complex interaction between different IoT devices and between human users and devices, but also due to the fact that it can generate a new block faster than Bitcoin [9] [25]. We also utilized the original consensus method: PoW, and the original encryption method: Ethereum account.

3.1 System Architecture and Topology

We designed a rich-thin-clients architecture which differs from the hierarchical architecture in that rich clients and thin clients have some overlapped functions while layers or components in a hierarchical architecture usually perform different functions.

Fig. 3 shows the rich-thin-clients architecture. Rich client and thin client can both provide a GUI for users, invoke blockchain (BC) APIs through BC interface which is deployed in a rich client, define business logics and collecting IoT data (optional for rich clients), but only a rich client contains a full BC node which can perform mining and contains all transaction records of the BC system .

Fig. 4 shows the overall architecture and the topology formed by rich clients and thin clients. Each rich client contains a fully functional Ethereum BC node which can

perform mining and run PoW consensus algorithm with other Ethereum BC nodes. The rich clients form a P2P network, just like public Ethereum BC nodes do. The thin clients connected to one same rich client form a star topology with the rich client. Since only the rich clients with high resources perform mining and consensus algorithm, our network should be similar to the public Ethereum BC network in performance, thus the dilemma between limited resources of IoT devices and the concerns for centralized architecture can be solved.

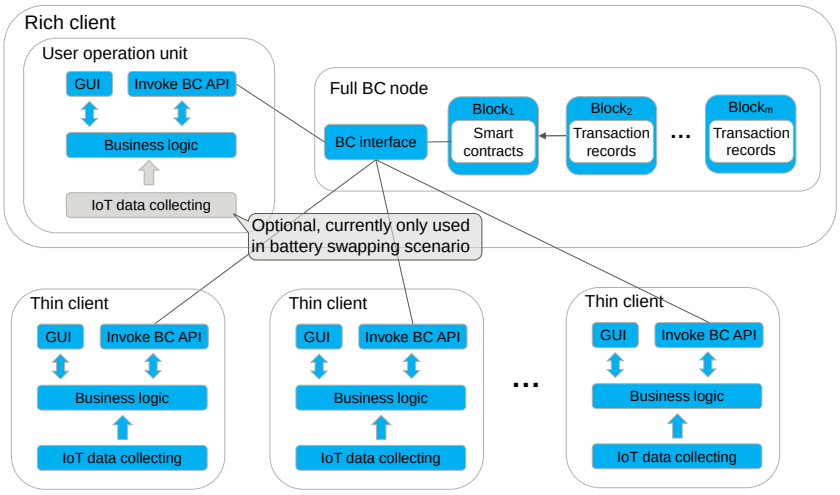

Fig. 3. Rich-thin-clients architecture

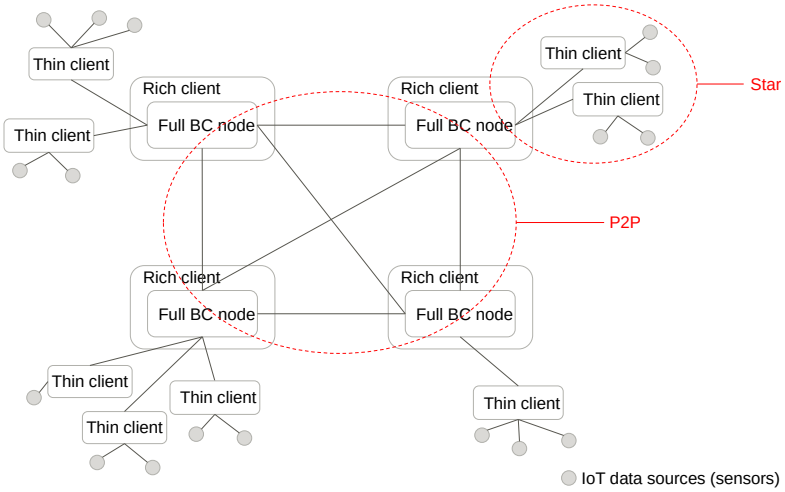

Fig. 4. Overall system architecture

3.2 Privacy and Security

We use Ethereum account mechanism to identify each client and to encrypt transactions. We assign unique Ethereum account for each client, so that they can be uniquely identified in our system. Each client gets its unique public key and private key along with its Ethereum account. And since the accounts are not directly connected to personal information in the real world, this could provide privacy to users.

In an IoT system, IoT devices (thin clients) are the most vulnerable parts facing attacks. If an IoT device is hacked, the attacker may adopt three operations:

1. Steal the property of the account which was used on this device. Unfortunately, this problem is unsolvable by using centralized architecture nor decentralized architecture.
2. Hack other devices through this device. This situation may cause severe issues in a centralized system, because the central servers may get attacked through an IoT device, but it will not happen in a decentralized architecture.
3. Fake an account or send faked data through this device. Since only the rich clients can generate a valid account, the faked account cannot be validated in our system. And thanks to blockchain's validation mechanisms when generating blocks, invalid transactions or accounts will be refused.

Furthermore, even if a rich client is hacked, since it's just a single node in the blockchain network, other rich clients will reject the invalid requests sent from it.

3.3 Implementation of Ethereum based Cyber-physical Battery Refueling System

Electric Vehicle Battery Refueling. Due to the development of battery technologies and the environmental awareness, EV technologies have been developing rapidly in last decades. With large-scale utilization of EV technologies, the release of greenhouse gases can be reduced, and the energy utilization can be more efficient. However, battery refueling is still a problem which have not been well solved. There are three major EV battery refueling methods: Alternating Current (AC) charging, Direct Current (DC) charging and battery swapping. AC charging can be adopted in an EV owner's garage. It's convenient but time-consuming. It will cost more than eight hours to fully charge a depleted EV battery by AC charging. DC charging can be provided by charging station. It will cost 1~2 hours to fully charge a depleted EV battery by DC charging. However, DC charging may harm EV battery because of the large power. Battery swapping is the least time-consuming one among the three EV battery refueling methods. It will cost only a few minutes to swap a depleted battery for a fully charged one by a battery swapping station. [12]

Tesla Inc. and NIO Inc. introduced their EV battery swapping technologies in 2013 and 2017 respectively [13] [14], suggesting that battery swapping may be a promising solution for EV battery refueling. In our previous work [12], a blockchain based EV battery swapping system (in the form of a web application) was proposed to evaluate the batteries to be swapped fairly by smart contracts and to manage the batteries'

information such as its manufacturer, brand, power capacity, price and refueling history.

It is actually an IoT scenario to use our previous blockchain based EV battery swapping system in real situation, since the battery swapping stations and EVs have to be connected to the Internet, and the cyber-physical interaction of the battery information has to be involved. In this section, we verify if our blockchain based IoT solution can be utilized in the EV battery swapping scenario.

Architecture of the Battery Swapping System. We implemented the battery swapping system based on our proposed rich-thin-clients architecture. Fig. 5 shows how the system is composed.

We use Raspberry Pi (RPi) [26] as the hardware of a thin client, each thin client represents an EV. In each EV, "truffle" [27] is used to invoke Remote Procedure Call (RPC) service of blockchain, and a local "express" [28] server is used to control the cyber-physical-interaction. We use USB disks which can be connected to RPi to represent real batteries, the information of each battery is stored in a file in each corresponding USB disk. Fig. 6 shows the static information and the dynamic information of a typical battery information file. (Static information and dynamic information is introduced later in "Smart Contracts" of 3.3.)

A station consists of a battery swaping interface (an RPi works like an EV) and a full Ethereum node. We use "Geth" [29] as the command line interface for running full Ethereum nodes and providing RPC service for invokers.

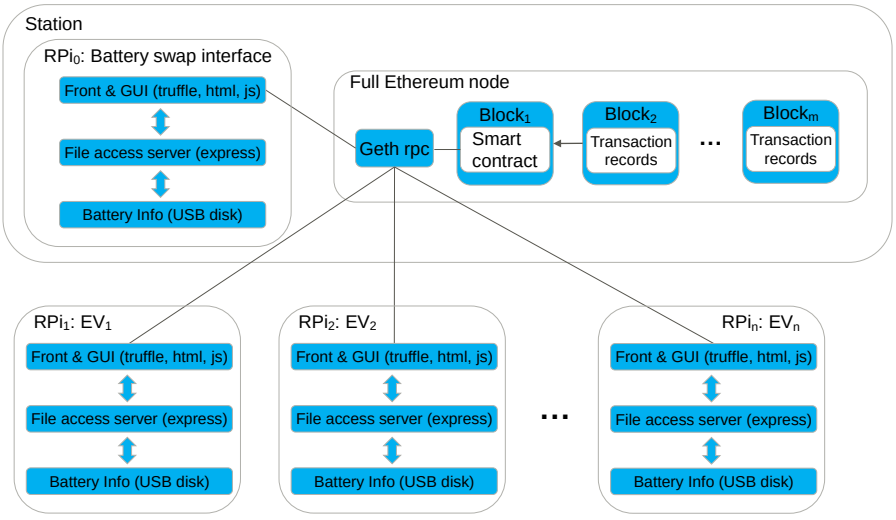

Fig. 5. Architecture of the battery swapping system

```
{
    "batteryID": 1,
    "ownerAddress": "",
    "filePath": "/media/usb0/battery_information.json",
    "staticInfo": {
        "brand": "Samsung",
        "maxChargingCount": 200,
        "maxDischargingCount": 200,
        "maxYearLimited": 20,
        "maxChargingTotalTime": 2000,
        "maxDischargingTotalTime": 2000,
        "manufactureYear": 2017,
        "manufactureMonth": 11,
        "manufacturePrice": 200
    },
    "dynamicInfo": {
        "price": 164,
        "energy": "20",
        "SOC": "100",
        "chargingCount": "1",
        "dischargingCount": "1",
        "chargingTotalTime": "2",
        "dischargingTotalTime": "2"
    }
}
```


Fig. 6. An example of battery information file

Smart Contracts. We implemented three smart contracts to manage the state machine of the battery swapping system [12]:

- "BatteryProcess" smart contract is used to operate and store battery information. It stores the batteries' static information and dynamic information. Static information is determined since a battery was produced and cannot be altered, such as brand, production time, manufacture price, and so on. Dynamic information is used to show a battery's status, such as charge times, state of charge (SOC), price, owner's account, and so on.
- "BalanceProcess" smart contract is used to manage value transfer between accounts. Given that it is not rational to require every EV user to have Ether [4], we defined a token called E-coin as the currency in our system.
- "BatteryInterface" smart contract provides API interfaces for three types of terminal users: station operator, EV owner and super account. EV owners can discharge the battery in their own EV, and they can send a battery swapping request to a battery station and wait for the confirmation. Station operators are the employees of battery station. They can charge, discharge, and recycle batteries belong to the battery station, and they can approve or deny a battery swapping request sent by EV owners. Super account is the system manager of this battery swapping system, he can create other types of accounts, grant E-coins, define the GAS required to invoke smart contracts, and so on.

System Flowchart. Fig. 7 shows how the system works. At first, the EV has a depleted battery which belong to the brand of TOY, and the station has two fully charged

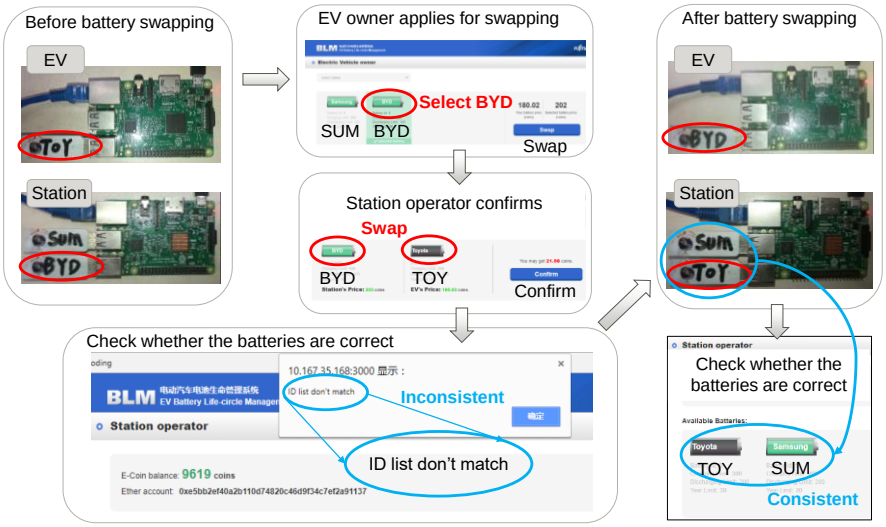

Fig. 7. System flowchart

batteries which belong to the brand of SUM and BYD respectively. Then the EV owner accesses the GUI of battery swapping system and send a swapping request to swap his TOY battery with station's BYD battery. After the station operator confirms the swapping request, the ownership information stored in blockchain is changed. However, at this point of time, the corresponding "real" batteries (USB disks) have not been swapped yet, so the inconsistency between cyber information and physical information is alerted. Finally, after the swapping of corresponding batteries, there is no alert any more, and the current battery status can be checked.

There are three kinds of major operations in the working procedure:

- **Initialization**. EV and Station perform initialization respectively. The system on RPi will read the information of all the batteries inserted on RPis' USB interfaces, and the smart contracts will create a battery record based on the static information of each "real battery" and generate a unique battery ID, meanwhile, the generated battery ID will be written into the battery information file inside the USB disk.
- **Battery information consistency checking**. Every time an EV owner or a station operator accesses the index page which shows the information of batteries belong to the EV or station, the system will check whether the batteries' static information recorded in USB disks are consistent with the batteries' static information on blockchain. If any inconsistency is found, an alert box will pop up. After an EV owner submits a swapping request, and a station operator confirms the request, the corresponding USB disks must be changed to remove inconsistency.
- **Battery charging and discharging**. When a user performs charging operation or discharging operation on a battery, the system will change the dynamic information of the battery on blockchain while changing the dynamic information recorded on USB disk according to the battery ID.

3.4 Experiments

In order to find a proper way to utilize Ethereum in the rich-thin-clients architecture, we have tried three different kinds of implementations, which are presented in Fig. 8.

The 1st implementation (Fig. 8(a)) is deploying a Geth client and a full Ethereum node on a Raspberry Pi which is utilized as the rich client, and starting mining on the Raspberry Pi. The 2nd implementation (Fig. 8(b)) is deploying Geth client without mining on each Raspberry Pi and deploying Geth and a full Ethereum node on a PC, the Ethereum nodes on Raspberry Pis synchronize data from the mining node. The 3rd implementation (Fig. 8(c)) is deploying a Geth client and a full Ethereum node only on the PC of a rich client, the Raspberry Pi of the rich client works only as a battery swapping interface.

The 1st implementation caused system crash soon after the starting of mining, because the resources of a Raspberry Pi cannot support the mining of a full Ethereum node, so it was excluded from the performance comparison.

We conducted quantitative comparison between the 2nd implementation and the 3rd implementation. As for experimental environment, Raspberry Pi 3B+ with ARMv7 1.2GHz CPUs, 1GB RAM and Raspbian OS, and PC with Intel 2.0GHz 64bit CPUs, 66GB RAM and Ubuntu 16.04.2 OS were used. We monitored the CPU and memory usage percentages of the Raspberry Pi in thin clients of the 2nd implementation and the 3rd implementation. Furthermore, we timed the EV discharging as an index to show the efficiency of each implementation. We conducted the experiments by repeating the battery swapping system's typical scenario which consists of the processes of initialization, EV discharging batteries, battery swapping and station charging batteries. Fig. 9 shows how the CPU, memory and network usage change during one time of experiment, indicating that the 2nd implementation clearly requires more CPU and memory usages than the 3rd implementation. Fig. 10 shows the statistical results of 10 times of repeated experiments, indicating that the 3rd implementation requires less resources and costs less time when performing EV discharging process. We can tell from the results that the 3rd implementation outperforms the 2nd implementation, thus we used the 3rd implementation in our previously proposed solution.

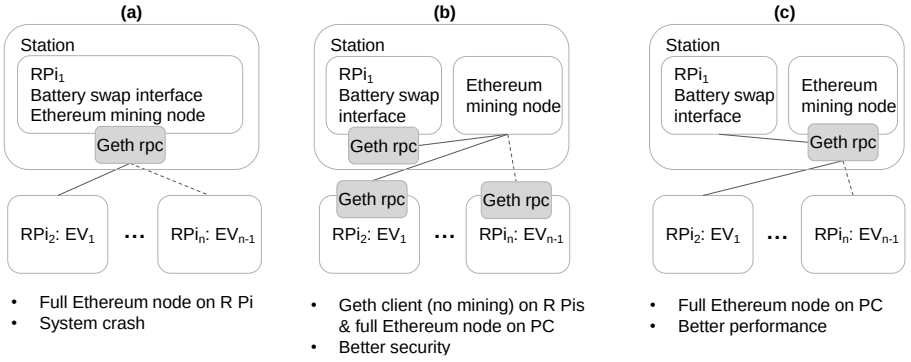

Fig. 8. Different implementations of rich-thin-clients architecture

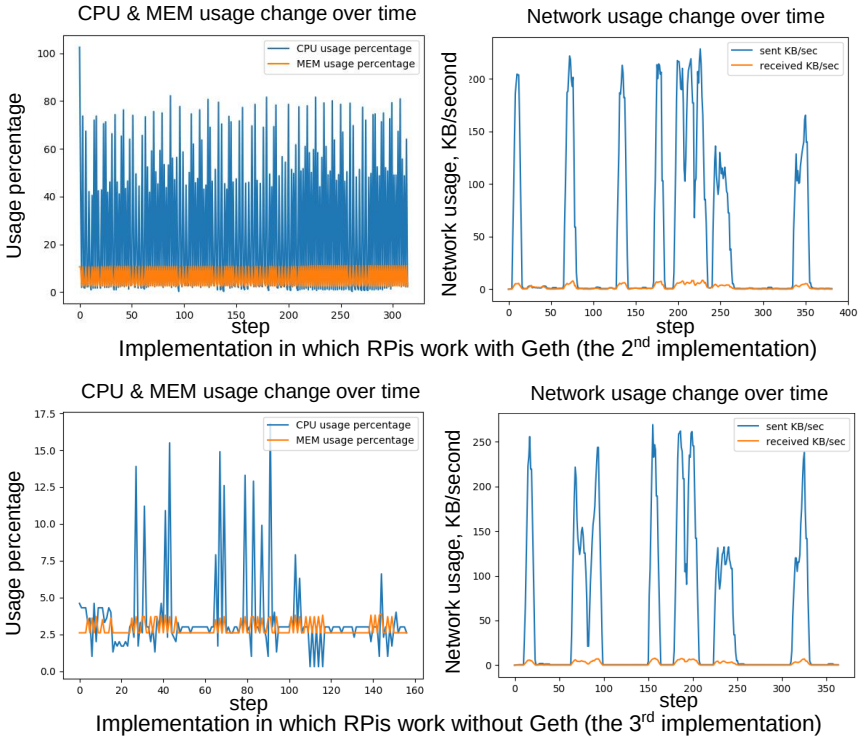

Fig. 9. Comparison of CPU, memory and network usages for two different implementations

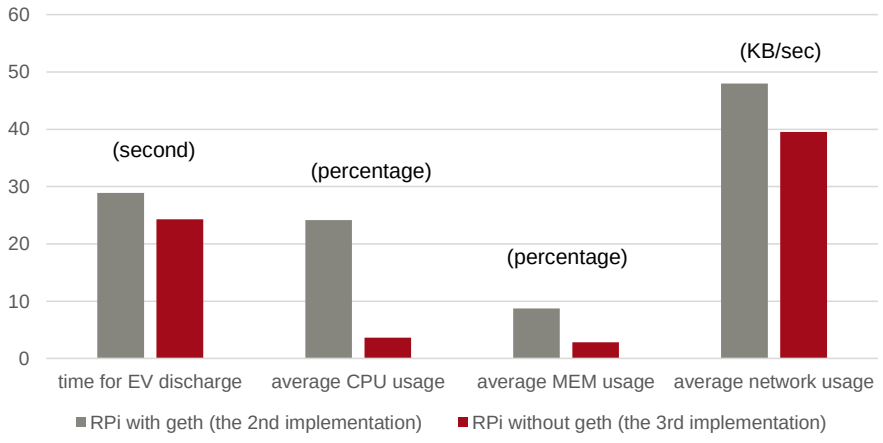

Fig. 10. Statistical performance comparison between two different implementations

Table 1. Comparison between blockchain based IoT solutions

Solution or Application	BC related features					Major data storage	Architecture
	Underlying BC system	Consensus method	Support smart contract?	State management	Security & Privacy		
Battery swapping (Our solution)	Non-public Ethereum	PoW	Yes	By Ethereum	Ethereum account mechanism	In Ethereum	• Rich-thin-clients • P2P+star topology
Automatic electricity usage adjustment	Public Ethereum	PoW	Yes	By Ethereum	Public key & private key based on RSA algorithm	In Ethereum	P2P topology
Smart city	Hybrid (Ethereum + other BC, e.g.NXT)	Unspecified	Yes	Unspecified	• By blockchain protocols • Recommend private ledgers	On BC	Hierarchical
Blockstack (A global naming and storage system)	Public Bitcoin	PoW	No	By "control plane"	Depend on the underlying blockchain	In "storage plane"	Hierarchical
BC-based smart home	Bitcoin-like	No consensus	No	By BC and Policy headers	• Public/private keys, shared key • By PK lists	*Local storage *Cloud storage	Hierarchical
LoRaWAN IoT (A low power wide area technology)	Bitcoin-like	PoW	No	By BC	• LoRaWAN is already safe • Public-key cryptography and digital signature	On BC	Hierarchical

4 Comparison between Blockchain based IoT Solutions

We made a comparison between our solution and other blockchain based IoT solutions. Table 1 shows the comparison result of some major features. Due to the usage scenarios of the solutions are totally different, so that the performance comparison between different solutions is absent in this paper.

Since we use Ethereum as underlying blockchain system, our solution surpasses the solutions which use Bitcoin as underlying blockchain system [7] [8] [11] in terms of flexibility of constructing state machines (supporting smart contract). Although Blockstack [7] also supports various state machines, there is no evidence to indicate that its state machine constructing mechanism is more convenient than smart contract of Ethereum. As for the automatic electricity usage adjusting system proposed by Seyoung et al. [9], although Ethereum blockchain is used, there is no value-transfer mechanism involved in their system, so Ethereum's advantages are not fully utilized in their solution.

5 Conclusions and Future Work

This paper proposes an Ethereum blockchain based IoT solution, introduces a battery swapping system implemented based on the solution and experiments to test the system's performance indexes, and compares our solution with other blockchain based IoT solutions. We use Ethereum (a blockchain system which supports smart contracts) to solve the issues caused by traditional centralized IoT architecture. And by

adopting a rich-thin-clients architecture, the dilemma between limited resources of IoT devices and the concerns for centralized architecture can be solved. Besides, the usage of our solution is not limited to battery swapping application, other IoT applications such as trading systems for sensor data or other digitalized property [23] [24] may also benefit from our proposed architecture.

In future studies, we would like to build a more practical battery swapping system with actual vehicle system instead of Raspberry Pi and to include multiple stations to simulate real usage scenarios.

6 References

1. Satoshi, N.: A peer-to-peer electronic cash system. (2008).
2. Merkle, R, C.: Protocols for public key cryptosystems. In: 1980 IEEE Symposium on Security and Privacy, pp. 122-122. IEEE (1980).
3. Melanie, S.: Blockchain: blueprint for a new economy. O'Reilly Media, Inc. (2015)
4. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper (2014).
5. Solidity Documentation, http://solidity.readthedocs.io/-en/latest/index.html, last accessed 2018/02/07.
6. Namecoin Homepage, https://namecoin.org, last accessed 2018/02/07.
7. Muneeb, A., Jude, N., Ryan, S., & Michael, J, F.: Blockstack: A global naming and storage system secured by blockchains. In: USENIX Annual Technical Conference, pp. 181-194. (2016).
8. Ali, D., Salil, S., & Raja, J.: Blockchain in internet of things: challenges and solutions. In: arXiv preprint, vol. 1608.05187. (2016)
9. Seyoung, H., Sangrae, C., & Soohyung, K.: Managing IoT devices using blockchain platform. In: Advanced Communication Technology (ICACT), 2017 19th International Conference on, pp. 464-467. IEEE (2017).
10. Kamanashis, B., Vallipuram, M.: Securing smart cities using blockchain technology. In: High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on, pp. 1392-1393. IEEE (2016).
11. Jun, L., Zhiqi, S., & Chunyan, M.: Using Blockchain Technology to Build Trust in Sharing LoRaWAN IoT. In: Proceedings of the 2nd International Conference on Crowd Science and Engineering, pp. 38-43. ACM (2017).
12. Song, H., Ence, Z., Bingfeng, P., Jun, S., Yoshihide, N., & Hidetoshi, K.: Apply blockchain technology to electric vehicle battery refueling. In: Proceedings of the 51st Hawaii International Conference on System Sciences. (2018).
13. Tesla Bttery Swap Event, https://www.tesla.com/videos/battery-swap-event, last accessed 2018/02/08.
14. NIO Power, https://www.nio.com/en/nio-power, last accessed 2018/02/08.
15. Owen, W., Diego, K.: Optimization of battery charging and purchasing at electric vehicle battery swap stations. In: Vehicle Power and Propulsion Conference (VPPC), pp 1-4. IEEE (2011).
16. Jun, Y., Hao, S.: Battery swap station location-routing problem with capacitated electric vehicles. In: Computers & Operations Research, 2015, 55, pp. 217-232. (2015).

17. Gartner Says 6.4 Billion Connected “Things” Will Be in Use in 2016, Up 30 Percent From 2015, https://www.gartner.com/newsroom/id/3165317, last accessed 2018/02/11.
18. Khan, M, A., Salah, K.: IoT security: Review, blockchain solutions, and open challenges. In: Future Generation Computer Systems (2017).
19. Nir, K.: Can blockchain strengthen the Internet of Things? IT Professional, 19(4), 68-72 (2017).
20. Marco, C., Antonio, V., & Juan, C., D., M.: Blockchain for the Internet of Things: a systematic literature review. In: Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of, pp. 1-6. IEEE (2016).
21. Ali, D., Salil, S., Raja, J., & Praveen, G.: Blockchain for IoT security and privacy: The case study of a smart home. In: Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on, pp. 618-623. IEEE (2017).
22. Ali, D., Salil, S., and Raja, J.: Towards an optimized blockchain for IoT. In: Proceedings of the Second International Conference on Internet-of-Things Design and Implementation, pp. 173-178. ACM (2017).
23. Dominic, W., von Bomhard, T.: When your sensor earns money: exchanging data for cash with bitcoin. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, pp. 295-298. ACM (2014).
24. Yu, Z., Jiangtao, W.: An IoT electric business model based on the protocol of bitcoin. In: Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on, pp. 184-191. IEEE (2015).
25. 4 cryptocurrencies with much faster block times than bitcoin, https://themerkle.com/4-cryptocurrencies-with-much-faster-block-times-than-bitcoin, last accessed 2018/02/24.
26. Raspberry Pi Homepage, https://www.raspberrypi.org, last accessed 2018/02/27.
27. Truffle, https://github.com/trufflesuite/truffle, last accessed 2018/02/27.
28. Express, https://github.com/expressjs/express, last accessed 2018/02/27.
29. Geth, https://github.com/ethereum/go-ethereum/wiki/geth, last accessed 2018/02/27.