Blockchain-Based Cross-Organizational Integrated Platform for Issuing and Redeeming Reward Points

Conference Paper · December 2019

DOI: 10.1145/3368926.3369676

Chia-Hung Liao
National Chiao Tung University

Shyan-Ming Yuan
National Yang Ming Chiao Tung University

Blockchain-Based Cross-Organizational Integrated Platform for Issuing and Redeeming Reward Points

Chia-Hung Liao
Department of Computer Science,
National Chiao Tung University,
Hsinchu 300, Taiwan, ROC
aiallen.cs07g@nctu.edu.tw

Ya-Wen Teng
Department of Computer Science,
National Chiao Tung University,
Hsinchu 300, Taiwan, ROC
yuka071099@gmail.com

Shyan-Ming Yuan[†]
Department of Computer Science,
National Chiao Tung University,
Hsinchu 300, Taiwan, ROC
smyuan@cs.nctu.edu.tw

ABSTRACT

For the sake of enhancing repurchase rate and customer loyalty, plenty of enterprises have issued reward points. However, as there are more enterprises issuing reward points, customers become less and less interested to collect them. Even though there are companies forming a business alliance to make reward points more attractive, it is usually managed by a central leading company. In other words, there still exists a security concern for a centralized system. In this paper, we propose a blockchain-based cross-organizational integrated platform (BCOIP) for issuing and redeeming reward points. BCOIP provides customers an electronic reward point system that is easier to collect and manage and could circulate in many enterprises. Besides, since BCOIP deploys the smart contract to operate the reward point and make all point transactions stored on the permissioned blockchain, it provides more interoperability and security to enterprises and more credibility for costumers. Moreover, goods providers joining the BCOIP could not only improve the diversity of the reward market but also reach more costumers. Finally, we conduct two experiments to evaluate the efficiency and stability of the BCOIP. Results indicate that points transactions are stable regardless of the number of parallel processes and the other shows the number of nodes has nothing to do with the efficiency of this system.

KEYWORDS

Blockchain, Ethereum, Smart contract, DApp, Electronic Reward Point, Business Alliance

ACM Reference format:

Chia-Hung Liao, Ya-Wen Teng, Shyan-Ming Yuan. 2019. Blockchain-Based Cross-Organization Integrated Platform for Issuing and Redeeming Reward Points. *In SoICT '19: The Tenth International Symposium on Information and Communication Technology, December 4–6, 2019, Hanoi – Ha Long Bay, Viet Nam. ACM, New York, NY, USA*, 6 pages. https://doi.org/10.1145/3368926.3369676

[†]Corresponding authors.

SoICT 2019, December 4–6, 2019, Hanoi - Ha Long Bay, Viet Nam

ACM ISBN 978-1-4503-7245-9/19/12…$15.00
https://doi.org/10.1145/3368926.3369676

1 INTRODUCTION

Many companies succeed in offering loyalty reward programs to enhance repurchase rate [1], for example, the mileage of airlines. However, as there are more and more companies issue their own reward points, the benefits from the loyalty reward programs become limited. It is a fact that loyalty programs no longer provide a competitive advantage in the current market because all companies run alike programs [2]. Customers are tired of competitive reward marketing recently and become less interested in gathering reward points [3]. Moreover, few goods choices and high redeeming points often make reward points easily get expired. In addition, customers usually receive many kinds of reward points from different stores, it ultimately becomes a mess and hard to manage in a good way. Although there are already some business alliances issuing the digital reward points that could circulate in their joined companies, their systems are often dominated by a central company, that still exists security concerns such as malicious attack and tampering transaction.

The objective of this paper is to propose an architecture of blockchain-based cross-organizational integrated platform (BCOIP) for issuing and redeeming reward points. BCOIP integrates numbers of companies into a consortium and issues electronic reward points which are more convenient to collect and able to circulate in these joined companies. Besides, the BCOIP system makes all the member companies collaborate with equal rights and power. In this work, Ethereum Blockchain is chosen as the system backend. Ethereum features the decentralization, openness and tamper resistance and supports Ethereum Virtual Machine and smart contract functionality so that the system will be secure and transparent [4]. It also allows all the member companies are capable of supervising and maintaining the platform and eliminates the need for central control.

Every member company is suggested to run one node for increasing the security and the credibility of the system. The more nodes in the blockchain, the higher potential to prevent double spending attack [5]. More nodes indicate more copies of data in different places. If any node crashes, the whole system won't crash consequently. Also, member companies could have a look at all block contents, and mine on their node. Then, all of these nodes form a permissioned blockchain. User data and goods information are stored in the database. A web server is launched so that each user or company could access the system by the web browser.

The rest of this paper is organized as followed. In Section 2, we present some related works. In Section 3, the system design and implementation are shown in detail. Section 4 evaluates the efficiency and stability of the BCOIP and makes discussions. Eventually, we make conclusions of this research in Section 5.

2 Background

First conceptualized by Satoshi Nakamoto in October 2008, blockchain technology emerged as the public transaction ledger of the cryptocurrency Bitcoin to solve the double-spending problem [6]. Blockchain is a distributed ledger which is preserving a growing list of immutable records called blocks that are bound to each other and secured using cryptography. As a distributed ledger, a blockchain is governed by a peer-to-peer network and every participant in the network stores a copy of blockchain data. Participants are able to dominate and supervise transactions. New generated blocks are validated by consensus algorithm to become the records in the ledger. The data in any given block are immutable and permanently recorded in the blockchain [7]. Permissioned blockchain is essentially a private blockchain [8]. In contrast to public blockchain such as bit coin blockchain, which has entirely no access restrictions, permissioned blockchain requires permission from the network owners to join. Only allowed members could verify transactions and validate blocks. It may even hold its own consensus mechanism.

Ethereum [4] was initially introduced by Vitalik Buterin in late 2013. Ethereum is an open source, public distributed computing platform built on top of a blockchain. Additionally, Ethereum provides the capability of computation to the blockchain with a built-in Turing-complete programming language, allowing anyone to deploy smart contracts and decentralized applications. Transactions in Ethereum could either be the Ether transfer or the function call of the smart contract. The Ethereum Virtual Machine (EVM) [9] is a sandboxed virtual machine embedded in each full Ethereum node, responsible for executing EVM code. The EVM is totally isolated from the network, file system or any processes of the host computer. An Ethereum node could establish a peer-to-peer communication channel with other nodes, validate blocks, sign and broadcast transactions, mining, deploy and execute smart contracts, etc [10].

Smart contracts are a term that was coined by Nick Szabo in 1994. Smart contracts aim to perform as a contract digitally and support the operation of transactions without third parties. Smart contract could help developer deploy decentralized and secure blockchain application, and is one of the most promising technologies for modern Internet of things (IoT) ecosystem today [11]. An Ethereum smart contract is a program running on top of an Ethereum blockchain and is generally written in a high-level programming language such as Solidity [9]. Smart contract is the main part of our proposed system which defines the entire functionalities of the electronic reward points and executes on the permissioned blockchain.

Decentralized applications (DApps) [12] are designed to run on a distributed computing system to avoid any single point of failure. DApps are stored on the Ethereum Blockchain, executed by the EVM and its smart contracts. In contrast to the traditional applications need to get data from a database through an API, the DApp gets data from the blockchain by invoking functions of the smart contract. At present, more than 90% of the world's DApps are hosted by Ethereum. For example, IDEX [13] is a decentralized exchange for trading ether and ERC-20 tokens which are offered by Aurora, a firm that has developed a series of financial services DApps. CryptoKitties [14] is an online game which allows the players to collect, sell and breed unique, non-replicable digital kittens. The kittens are actually ERC-721 tokens.

3 System Design and Implementation

3.1 System Design

The following introduces the design idea of the proposed BCOIP system and details of each component. The research objective is to construct a system which offers electronic reward points that can circulate in different companies and support cross-organizational redeeming. Review the data management for a traditional centralized system, it usually stores participant information, point transactions and goods data in a centralized database. If there is a malicious attack happen, the system may crash or the exist point transactions may have tampered. Since the point transaction records are the most important part to sustain the ecosystem, our proposed system moves them to the blockchain distributed ledger to attain more safety. Furthermore, every participant can transfer reward points by invoking our designed point contract on the Ethereum blockchain. As for user information and goods data, they are still stored and protected in a database. This system does not involve real cash flow. Every real money transaction happens outside the BCOIP. In addition, all joined companies are informed of the request to launch a new Ethereum full node to join this permissioned blockchain. Because the more company nodes join, the more platform credibility will be. Also, member companies are encouraged to mine, the BCOIP system could design a relative return to those devoting ones. By implementing the above ideas, we construct a more credible system which offers electronic reward points that can circulate in different companies and support cross-organizational redeeming. Thus, this proposed system could facilitate the new type of business alliance to develop more effective loyalty reward program.

Figure 1 shows the conceptual architecture of the proposed BCOIP system. Basically, it contains three components: Permissioned blockchain, BCOIP web server, and database. Permissioned blockchain is formed with the nodes which are run by the platform and all the companies joining the system. It executes the smart contract that defines all functions and variables of reward points, and stores point transaction records and event logs generated by the platform. The BCOIP web server deals with the

HTTP requests, returns the responses, interacts with the permissioned blockchain, and operates data management with MongoDB. As for the selected toolkits, this web server is written by Node.js and Express, and web3.js is used to interact with the blockchain. The front-end webpage is written by HTML, CSS and Javascript, and ethjs works with the Ethereum blockchain. We choose Geth as Ethereum full node and MongoDB as a database to stores all user account information and goods data.

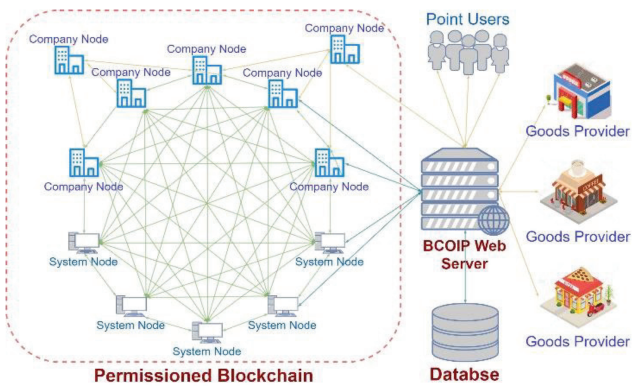

Figure 1: System overview for the BCOIP.

3.2 Participant Roles and Point Contract

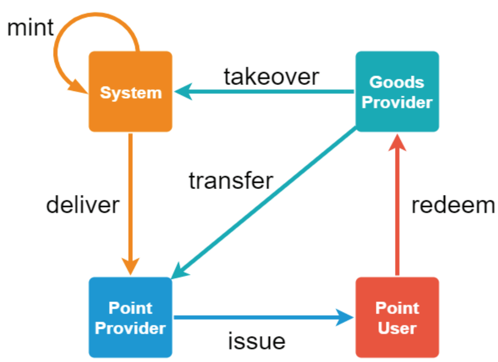

Figure 2: The reward point flow for the four main roles inside the platform.

Figure 2 shows the circulation for reward point flows between different roles in the BCOIP. The four main roles are the system, point user, point provider, and goods provider. System represents the BCOIP system and plays a role in providing reward points to point providers and buying reward points from goods providers. Point user gathers reward points given from companies and redeems reward points for goods which platform provides through the BCOIP website. Point provider acquires reward points from the BCOIP, issues reward points to point users and can purchase reward points from goods providers. Goods provider supplies goods for point users to redeem and sell the reward points they earned to point providers or the BCOIP.

Table 1: View and Modification Functions in The Point Contract

Type	Name	Description
View Function	totalSupply()	Returns the total point supply.
	quotaOf()	Returns the account quota.
	balanceOf()	Returns the account balance.
	numberOf()	Returns the number of the transactions.
Modification Function	mint()	The system will mint ‘value’ new reward points.
	deliver()	The system will transfer ‘value’ reward points to a point provider, address ‘to’, from its own account.
	redeem()	A point user, address ‘from’, transfers ‘value’ reward points to a goods provider, address ‘to’.
	takeover()	A goods provider, address ‘from’, transfers ‘value’ reward points to the system.
	issue()	A point provider, address ‘from’, transfers ‘value’ reward points to a point user, address ‘to’.
	transfer	A goods provider, address ‘from’, transfers ‘value’ reward points to a point provider, address ‘to’.

Table 1 shows the two kinds of functions of the “Point Contract”. One is the view function which just returns the value of state variables, and the other is the modification function which modifies the states of reward points. The view function “balanceOf(address who)” is usually invoked for displaying the account balance of a user account on the web page. The other view functions totalSupply(), quotaOf() and numberOf() are designed for the joined companies to get the information from the system. As for the modification functions mint(), deliver(), redeem(), takeover(), issue(), and transfer(), they will alter the states of reward points. Figure 2 also reveals the point flow when these modification functions are executed. In addition, to avoid some malicious users executing these functions to get reward points illegally or destroy the balance of the whole system, we set the modifier “onlyOwner”. If any function is equipped with the modifier “onlyOwner”, it could be only invoked by the creator of this smart contract, that is, the account of the system.

3.3 BCOIP Web Server

This part illustrates some used toolkits and the functionalities of the proposed system server. Express is a popular web framework which is hosted within the Node.js runtime environment and is designed for building web applications and APIs. Web3.js is a collection of JavaScript libraries which helps our web server to interact with an Ethereum node. MongoDB provides API for Node.js to operate data management.

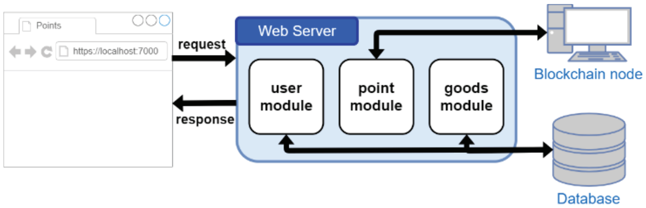

Figure 3: Architecture of BCOIP web server.

Figure 3 shows the architecture and workflow of the BCOIP web server. Basically, this server has three modules, which are user module, point module, and goods module. When any user sends requests via web browsers to the server, these three modules will deal with different kinds of requests. User module and goods module mainly work with the database, and point module chiefly interacts with the permissioned blockchain. When the server carries out the requests, the responses will be sent back to the user web browsers. Besides, we have provided an on-line demonstration [15] that illustrates the whole procedure for interaction with our BCOIP website, such as registration, login, issuing points, redeeming goods and deleting goods, etc.

3.3.1 User Module. User module is used for dealing with all the requests related to user management, such as registration, login, and identity confirmation and it should interact with the MongoDB. For registration, after newcomers select their role and fill out the form, the web browser transforms user data into json format and sends it to the BCOIP server. We apply the POST request method to send the registration data since it contains the user password. Then, the server checks if the user exists already. If the user has not involved in the platform yet, the server invokes the insert function to save the data into the "users" collection in the MongoDB. About login, after a user fills out its account and password in the homepage, the server checks the user data in the database. If the login request is permitted, the browser will redirect to a corresponding URL. The third is about Identity confirmation. For example, when point providers want to issue reward points to a user, they should input the user account. Then, the server will verify if the account is admissible. If so, the corresponding blockchain address will be sent back to the web browser. Thus, the system could continue the follow-up work.

3.3.2 Point Module. Point module deals with all the requests related to reward points, such as issuing reward points to point users, redeeming reward points for goods and obtaining the balance of an account. In fact, the reward points in our system is a smart contract deployed to the permissioned blockchain. Therefore, the point module requires web3.js to interact with the permissioned blockchain. To reach the instance of our "Points contract", we need the ABI of the smart contract and the address where the contract deployed. After we get the contract instance, we could use the instance to call functions in "Points contract". In web3.js, there are two methods to call the functions of the smart contract which are the 'call' method and the 'send' method. The 'call' method is for the view functions such as 'balanceOf' in Point Contract. In contrast, when we need to call functions such as 'issue' and 'redeem' which alter the smart contract state, we must use the 'send' method to invoke them.

3.3.3 Goods Module. Goods module interacts with MongoDB for dealing with all the requests associated with goods management. All the goods on the reward point platform will be listed on the web page after point users log in. On the other hand, when goods providers want to increase new goods on the platform, they have to fill out the information and choose a goods picture. Then, the data will insert to the "goods" collection in the database, and the picture of goods will be uploaded to the local directory of the web server. Goods providers could also delete the goods they no longer provide. The record of the goods will be deleted from the database.

4 Evaluation and Discussion

In this section, we design two experiments to evaluate system efficiency and capability. The first experiment is about the total running time and the maximum, minimum and average running time of single process separately under different number of processes executed simultaneously. Since there are two kinds of function calls in smart contract, we examine two different function calls respectively. The second experiment is to evaluate the efficiency under different numbers of nodes in permissioned blockchain.

4.1 Experiment 1

In the first experiment, a blockchain with 5 nodes is set up. To examine the efficiency of dealing with different numbers of requests simultaneously, we execute 1000, 2000 and 3000 processes separately by sending the requests to the server to invoke the point contract functions. There are two kinds of requests to make function calls. One is the 'call method' for invoking the view functions of the point contract, and the other is 'send method' for invoking the modification function to alter the point state. Here we choose one view function 'balanceOf()' for the 'call method', and one modification function 'issue()' for the 'send method'.

Table 2: The function running time under different numbers of processes executed simultaneously (millisecond)

	call method			
processes	total	max	min	average
1000	8076	71	4	13.78
2000	37851	84	4	9.6
3000	66779	124	4	9.25

	send method			
processes	total	max	min	average
1000	15401	427	4	27.8
2000	67519	472	4	29.81
3000	121627	361	4	25.82

Table 2 demonstrates the function running time under different number of processes executed simultaneously. In the first column of both call and send method, the total running time of 'send method' is approximately twice as long as the 'call method' regardless of the number of processes. The second and third column of both call and send method shows the maximum and the minimum running time for the single process during processing. The results indicate that although both of the minimum running times are the same, the running time of the 'call method' is generally much shorter than the 'send method'.

4.2 Experiment 2

In the second experiment, we test the efficiency under the blockchains with 10 nodes. Then, we execute 1000 processes of 'call method' to invoke the 'balanceOf()' function in the point contract on both of blockchain respectively. Such experiments are repeatedly conducted 5 times.

Table 3: The function running time of 1000 processes executed simultaneously under a blockchain with 10 nodes (millisecond)

Exp. #	total	max	min	average
1	8357	106	4	15.414
2	7742	140	4	15.763
3	8305	103	4	14.447
4	7860	107	4	14.071
5	8429	122	4	16.894
mean	8138.6	115.6	4	15.3178

Table 3 shows the function running time of 1000 processes executed simultaneously under the blockchains with 10 nodes. Results in the first column indicates that the total running time are all around 8 seconds. The second and third column demonstrate the maximum and the minimum running time of the single process among 1000 processes. The fourth column reveals the average running time of 1000 single processes.

5 CONCLUSIONS

In this paper, a design of a blockchain-based cross-organizational integrated platform (BCOIP) for issuing and redeeming reward points is proposed. Every company joining the platform runs an Ethereum node to form a permissioned blockchain. The implementation of reward points is the smart contract running on the blockchain. Whenever a request for transferring reward points arises or getting the information about the point state, the function calls are invoked to trigger the corresponding functions in the point smart contract.

We conduct two experiments to evaluate the efficiency and stability of the BCOIP system. In the first experiment, the result shows that the executing time of the 'send method' to call the modification functions of point contract to achieve reward point transfer is longer than that of 'call method' to call the view functions which just read the point state regardless of the number of parallel processes.

The BCOIP achieve a win-win-win situation for consumers, companies and goods providers. For consumers, they get a more convenient type of reward points to collect and manage. Besides, since electronic reward points issuing by the BCOIP system can circulate in various companies, consumers could attain more goods choices to redeem. For companies, the BCOIP create a new type of business alliance for companies to cooperate with equal rights and power since this is a decentralized loyalty reward program. The more companies join this BCOIP permissioned blockchain, the more credibility platform will attain because the point transaction records are safer on the blockchain than in a centralized database. For goods provider, their products can reach more users after joining the BCOIP. Last but not least, this design concept is not only limited to issuing reward points but also adapted to other scenarios such as amusement parks or shopping malls. For example, they can run a BCOIP-like system to provide digital tokens which circulate among various shops, restaurants, and facilities, etc.

REFERENCES

[1] Leenheer, J.v.H., Harald J. Bijmolt, Tammo H. A. Smidts, Ale, *Do loyalty programs really enhance behavioral loyalty? An empirical analysis accounting for self-selecting members.* International Journal of Research in Marketing, 2007. **24**(1): p. 31-47.
[2] Zhang, J.B., Els, *The Impact of an Item-Based Loyalty Program on Consumer Purchase Behavior*. Vol. 49. 2012, Journal of Marketing Research. 50-65.
[3] Dowling, G.R.U., Mark D. Hammond, Kathy, *Customer loyalty and customer loyalty programs.* Journal of Consumer Marketing, 2003. **20**(4): p. 294-316.
[4] Buterin, V., *A next-generation smart contract and decentralized application platform.* white paper, 2014. **3**: p. 37.
[5] Kroll, J.A.D., Ian C. Felten, Edward W. *The economics of Bitcoin mining, or Bitcoin in the presence of adversaries*. in *Proceedings of WEIS*. 2013.
[6] Nakamoto, S., *Bitcoin: A peer-to-peer electronic cash system.* 2008.
[7] Nofer, M.G., Peter Hinz, Oliver Schiereck, Dirk, *Blockchain.* Business & Information Systems Engineering, 2017. **59**(3): p. 183-187.
[8] Joshi, A.P.H., Meng Wang, Yan, *A survey on security and privacy issues of blockchain technology.* Mathematical Foundations of Computing, 2018. **1**(2): p. 121-147.
[9] Dannen, C., *Introducing Ethereum and Solidity*. 2017: Apress, Berkeley, CA.
[10] Wood, G., *Ethereum: A secure decentralised generalised transaction ledger.* Ethereum project yellow paper, 2014. **151**(2014): p. 1-32.
[11] Liu, X., et al., *Elastic and cost-effective data carrier architecture for smart contract in blockchain.* Future Generation Computer Systems, 2019. **100**: p. 590-599.
[12] Raval, S., *Decentralized applications: harnessing Bitcoin's blockchain technology*. 2016: " O'Reilly Media, Inc.".
[13] IDEX. *0.00018634 IDEX/ETH - IDEX - Decentralized Ethereum Asset Exchange*. [cited 2019 July 15]; Available from: https://idex.market/eth/idex.
[14] CryptoKitties. *CryptoKitties | Collect and breed digital cats!* [cited 2019 July 15]; Available from: https://www.cryptokitties.co/.
[15] DCS-Lab. *A Demonstration for the BCOIP system*. [cited 2019 Aug 20]; Available from: https://github.com/nctu-dcs-lab/Blockchain-Based-Cross-Organizational-Integrated-Platform-for-Issuing-and-Redeeming-Rewrd-Points/tree/master/Demonstration.