Proceedings of the Federated Conference on Computer Science and Information Systems pp. 565–568
DOI: 10.15439/2019F296
ISSN 2300-5963 ACSIS, Vol. 18

Using Blockchain to Access Cloud Services: A Case of Financial Service Application

Min-Han Ruby Tseng
Graduate Institute of Technology Management
National Chung Hsing University
Taichung City, Taiwan
Email:cv727320@gmail.com

Shuchih Ernest Chang*
Graduate Institute of Technology Management
National Chung Hsing University
Taichung City, Taiwan
Email:eschang@dragon.nchu.edu.tw

Tzu-Yin Kuo
Taipei Fubon
Commercial Bank Co., Ltd.
Taipei City, Taiwan
Email:tzuyin9118@gmail.com

***Abstract*—Most cloud providers use centralized servers to manage data. However, centralized servers still suffer the risks of single point of failure and data theft. We add a blockchain to the cloud service and propose a new architecture to manage data. Using blockchain as a connector to utilize the tamperproof, traceable, and data-sharing features of the blockchain to ensure that the transaction data are properly stored in each node. We use the stock simulation trading service to extend and divide the research design into two levels, namely, system and application services. First, we directly write the data into the blockchain. Second, we alternatively store the data in the cloud and then write it into the blockchain. Finally, the two versions are compared and analyzed to investigate their feasibility and performance. At the application service, we implement the smart contract for the existing stock transaction process to achieve real-time settlement.**

I. Introduction

WITH the cloud service, the cost of equipment maintenance within the company is converted into the cost of service operation [1]. And the maintenance of the system becomes simple and also increases flexibility.

However, while availing of the convenience of cloud services, enterprises' internal data or even highly sensitive data are stored in the data center of a third-party. If sufficient security measures are not taken, then security risks, such as data leakage and tampering, will occur. In recent years, well-known cloud service providers have frequently reported cloud vulnerability incidents [2].

In particular, many small- and medium-sized enterprises (SMEs) adopt centralized third-party cloud services . As the size of the enterprise increases, the vertical expansion of the cloud service database becomes prone to the risk of single point of failure [3]. In the case of a single point of failure or single-path disconnection, the cloud service provider will interrupt the network service and even the entire production line. This situation can cause considerable losses for companies [4].

The centralized environment of enterprises is increasingly unable to adapt to the needs. Thus, they are gradually moving from the original centralized database to the decentralized database. The blockchain is a large decentralized database. The data structure of the blockchain ensures that the transactions in the network are traceable, immutable, and tamperproof. In this study, the information stored in the cloud is encrypted, and the user's digital assets and transaction records are distributed in different nodes in the network through the P2P network, thereby reducing the risk of being stored only in a single node. Based on the blockchain, the data stored in the cloud is encrypted and stored in the network's block. The user's file data will not be exposed to the risk of being leaked or stolen during cloud server failure [5]. Users can securely access data, and privacy is well protected.

The design of this study will be divided into two levels, namely, system and application services. We classify the level discussed in the previous paragraph as the system service level and analyze the problems that cloud storage may encounter. We hope to reduce the risks that cloud storage may encounter by using the blockchain. At the application service level, we design the blockchain smart contract for the existing stock transaction process. The traditional stock transaction employs the T+2 settlement cycle. However, through the blockchain, users can achieve real-time delivery of stocks and the trading become more secure.

II. Research Background

This study is based on the cloud service and stock market transaction implementation services proposed by Wang and Chang [6]. We use the concept and technology of the blockchain to extend and improve the establishment of the blockchain as a connector for cloud data services.

A. Stock market simulation trading system architecture

We design a network-based stock market simulation trading system (hereinafter called SMSTS). Using ASP.NET web development including HTML, CSS, Bootstrap, `C#` and Python to design the SMSTS. The database is designed using the Microsoft SQL Server.

B. Blockchain

Satoshi Nakamoto published a paper entitled "Bitcoin: A Peer-to-Peer Electronic Funds Transfer System" in 2008 [7], proposing the concept of bitcoin and its underlying technology.

This work was supported by the Ministry of Science and Technology, Taiwan, under contract number MOST-106-2221-E-005-053-MY3.
*Corresponding author: eschang@dragon.nchu.edu.tw

The blockchain is a large global decentralized ledger database that records all transaction records [8]. Each node uses the proof-of-work hash function to determine who verifies these transactions. The node that obtains the verification right would broadcast the block to all of the nodes. Until the first successful node confirms the verification, the block quickly connects to the parent blockchain.

C. Smart contract

The concept of a smart contract was first proposed by Nick Szabo in 1994 [9]. He advocated that the trading conditions could be automated by the program. When the conditions are met, the value can be transferred. All of these are performed automatically by the computer program, and no third party is involved.

D. Ethereum

The concept of Ethereum was first proposed by Vitalik Buterin: A next-generation smart contract and decentralized application [10]. Ethereum is an application platform based on blockchain technolocy that allow many different applications to be built on the Ethereum platform.

E. Stock settlement

Settlement is the end of a creditor–debt relationship. If it cannot be completed in time, then it may cause the next transaction to be unsuccessful, which will affect other business activities. The stock settlement cycle is T+2. Thus, if the settlement speed can be improved, then the efficiency of the capital market operation can be improved and the cost of verification by the settlement institution personnel can be reduced.

III. Architecture and Design

In recent years, cloud computing has become even more popular. However, when confidential information is in third-party cloud services, the risk of leakage increases. The blockchain database consists of several nodes and all participate in data management. Any data added to the blockchain database must be agreed upon by most nodes in the blockchain network to be successfully recorded in the block and cannot be controlled by a single entity. Such mechanism ensures that data are secure, transparent, and permanently recorded, thus making it difficult to tamper with the content.

This study divides the blockchain into two levels, namely, system and application service levels. In terms of system services, two versions are proposed for writing data into the blockchain database. The first version involves directly writing the data into the blockchain. The second version involves storing the data in the cloud, and the blockchain acts as a connector to encrypt and decrypt the location where the cloud stores data. In terms of application services, the details of the use of smart contract automation for SMSTS are described in the subsequent paragraphs.

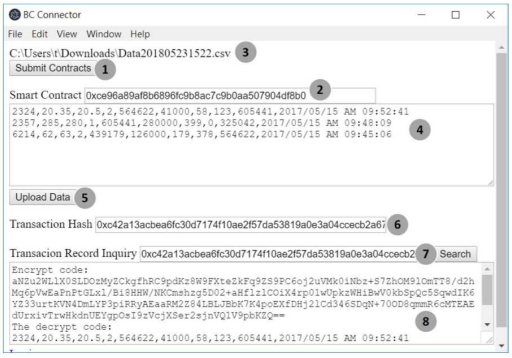

Fig. 1. User Interface of BC Connector

A. Data directly stored in the blockchain

We use SMSTS that we have set up on the local server before, then set up the blockchain environment on the notebook. In this program, we designed the user interface to gradually write the data into the blockchain. Besides, we use symmetric key cryptography to encrypt the data.

1) Blockchain environment setting: We use the Ethereum platform to set up a private chain network and Solidity to write the smart contracts. Before the smart contract is submitted to the blockchain network, it needs to be compiled and deployed.

2) Download the file: We add the function button for downloading files in the SMSTS which enables users to download the transaction records from the SMSTS to their own computers and then write into the blockchain through the blockchain connector.

3) Compile and deploy the smart contract: To ensure easy operation by the user, this study designed a user interface that can be operated step by step, shown in Fig. 1. First, after pressing the (1) Submit Contracts button, it will generate an address in the (2) Smart Contract field, then click on File in the upper function bar and click on Open File, and then select the file to be written into the blockchain. After confirming the file, the file path will be displayed in (3). The file content will be displayed in the data column of (4). After pressing (5) Upload Data, the data will be written into the block. When the data are written into the blockchain, the encrypted Transaction Hash will be generated in (6). The user can also query the data of the blockchain through the (7) Transaction Record Inquiry, and the query data will be displayed in (8) Area. To ensure that the data are secure, we use the AES-128-CBC symmetric key encryption method to encrypt the data to be written into the blockchain. The Encrypt Code displayed in the query data bar is the encrypted garbled code. Even if the user obtains the Transaction Hash, the file content will still be invisible in the blockchain.

B. Write the data address stored in the cloud into the blockchain

Most users use the centralized cloud server to store data, which is vulnerable to hackers; thus, the security of data is

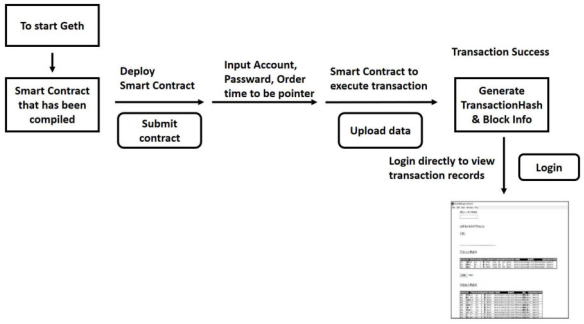

Fig. 2. Using the BC Connector to enter SMSTS query data

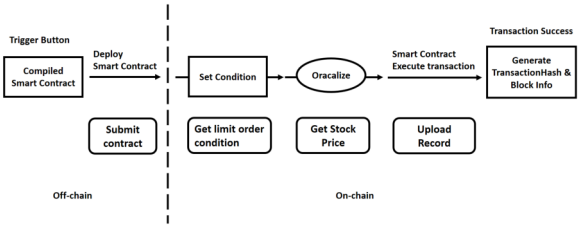

Fig. 3. Limit trading desigh flow

threatened. Therefore, this study integrates the web server, database, stock price program, and transaction processing program into the cloud system, allowing users to buy and sell stocks through the browser, thereby collecting data and using these data as the test data for writing the blockchain.

1) Cloud erection: We transfer the SMSTS to the Azure cloud platform and collect the data generated by the user through the simulation program for stock trading as the experimental basis.

2) Using the blockchain as a connector: We build a Electron, to design and develop our blockchain connector user interface. After launching Geth, another Git Bash window opens, with the electron command to execute our project. The account, password, and order time are used as pointers for the user on how to store the transaction data. After being encrypted and written into the blockchain, the user can directly log in to the transaction record page of the SMSTS to query the transaction data of the specified order date, as shown in Fig. 2.

C. Smart contract automation

Smart contract automation is applied to the stock limit trading service. Based on the design structure of the SMSTS, the system automatically generates the smart contract and deploys it to the P2P network environment. We have added a trigger button that allows users to link to their own blockchain wallet to perform limit trading services. The system design flow for this architecture is shown in Fig. 3

TABLE I
APPLICATION LOAD TIME TEST RESULTS SET ON THE LOCAL AND CLOUD

Load time	SMSTS is set up on the local server (sec)	SMSTS is set up in Azure cloud server (sec)
1	2.07s	1.03s
2	2.54s	1.02s
3	2.05s	1.23s
4	2.01s	1.11s
5	2.52s	1.02s
6	2.40s	1.08s
7	2.31s	1.15s
8	2.23s	1.24s
9	1.99s	1.09s
10	2.36s	1.15s
Avg time	2.48s	1.12s

IV. EXPERIMENT AND EVALUATION

This study conducts a series of tests and analyses based on the architecture and design at the system service level described in the previous section, and the application service level will be compared with the existing platform.

A. Results from stock market simulation trading system test results

To test the performance, we use the Pingdom Website Speed Test, a free website tool that detects website speed and performance, to understand and evaluate the performance of the proposed service and to measure the load time of the website through experiments. We have designated San Jose(Califonia,USA) as the test area.

The test results of the page load time and the time of writing data into the blockchain are shown in Table I. From the table, we can see that the cloud server is faster than the local server. This finding can be plausibly attributed to the fact that the cloud service providers focus more on optimization of the cloud storage service. This difference is also affected by the network speed, test location, and device capabilities between the cloud service provider and the research computer device.

B. Results from blockchain application system test results

We first compare the difference between (1)directly storing the accessing data in the cloud and (2)storing/accessing via cloud addresses recorded on blockchain. Table II shows that the time of transaction data generated by the SMSTS, which is directly stored in the cloud, is shorter and more average than that of other systems. Storing of the cloud data address to the blockchain increases the step of writing the blockchain; thus, it takes a long time. Compared with storing only the data in the cloud, writing into the blockchain is time consuming but more secure.

C. Discussion of cons and pros of the proposed blockchain system services

Traditional decentralized systems generally use a Replicated state machine [11] to implement a fault-tolerant mechanism. The blockchain uses similar approach,but it does not rely on a single entity to complete the service because there exits consensus agreement in the blockchain. When a transaction conflict occurs, only one transaction is approved to avoid double-spending. In order to solve the problem of Byzantine failure [12], we write the data encryption program in the smart

TABLE II
COMPARISON OF DATA STORED IN THE CLOUD AND WRITE CLOUD DATA ADDRESS INTO THE BLOCKCHAIN

Load time	Data directly stored in the Azure cloud platform (sec)	Writes into the blockchain after the Azure output data address (sec)
1	21.14s	51.42s
2	20.31s	64.36s
3	15.92s	40.55s
4	14.54s	52.31s
5	24.64s	36.96s
6	18.51s	88.49s
7	16.61s	37.12s
8	19.92s	35.71s
9	23.16s	21.33s
10	17.18s	46.40s
Avg time	19.19s	47.45s

contract and enable it to be executed automatically, and then use the blockchain as a connector to achieve decentralized storage as the core of the cloud storage [13].

Through the solutions proposed in this study, the blockchain will be continuously extended, and the nodes can be connected to each other [14]. Once the data are written into the block, it cannot be tampered with, which helps the cloud provider in ensuring the security of the user data. Moreover, the nodes that are distributed in the network can reduce the cost of network transaction, authentication, and collaboration and can effectively solve the synchronization problem of the traditional distributed database. However, the current transaction speed of Ethereum is still very slow, only 10˜20 transactions per second use sharding or plasma as a blockchain expansion solution.

D. Discussion of the benefit of blockchain application services

The proposed application is based on the limit trading service in the SMSTS developed in this research. Through the characteristics of the blockchain, real-time delivery of stocks can be achieved. The settlement cost can be reduced and the stock or payment time can be shortened [15].

V. CONCLUSION

This research is divided into three parts. First, we set up the web server and database system of the stock market simulation program in our Internet data center, collect transaction information from it, and store the data directly in our private blockchain. Moreover, the storage, verification, transmission, and communication of network data are performed through distributed nodes. Our private chain can record, sort, and encrypt every transaction. Participants use the verification code to link the transaction records and then use the characteristics of the blockchain to save records of all transactions and ensure the integrity of the data. Thus, the transaction history cannot be falsified.

Secondly, as more SMEs turn to cloud computing services, the blockchain can create secure, effective, tamperproof, and democratic computing networks. In this study, the location of each data block is recorded in the blockchain. When the file needs to be accessed, the system will verify the identity according to the private key of the user and assemble the file. We found that combining data in a blockchain with a decentralized cloud is safer than storing it in a centralized system. One device does not contain complete files, which makes it almost impossible for hackers to steal data, thus improving security and reliability.

Third, the smart contract can automate of the stock limit trading service. The actual stock transactions do not need to wait for the T+2 settlement cycle. The blockchain increases the flexibility of trading strategy execution and reduces the labor cost of transaction clearance and settlement.

The blockchain can effectively reduce the cost of authentication, network transactions, and collaboration [16], can control the reading and modification of data through public and private keys by taking into account transparency and privacy security, and is reliable. This study conducts prelimilary experiments in using the blockchain as a connector to store cloud data, and in the future, it can be applied to other applicable cloud services, even in different application scenarios.

REFERENCES

[1] J. Gibson and R. Rondeau and D. Eveleigh and Q. Tan , "Benefits and challenges of three cloud computing service models," *in 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN) IEEE,* 2012, pp. 198–205. DOI: 10.1109/CASoN.2012.6412402

[2] H. Tianfield , "Security issues in cloud computing," *in 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC) IEEE* 2012, pp. 1082–1089. DOI: 10.1109/ICSMC.2012.6377874

[3] W.A. Jansen , "Cloud hooks: Security and privacy issues in cloud computing," *in 2011 44th Hawaii International Conference on System Sciences IEEE* 2011, pp. 1–10. DOI: 10.1109/HICSS.2011.103

[4] A. Kirar and A.K. Yadav and S. Maheswari, "An efficient architecture and algorithm to prevent data leakage in Cloud Computing using multi-tier security approach," *in 2016 International Conference System Modeling & Advancement in Research Trends (SMART) IEEE* 2016, pp. 271–279. DOI: 10.1109/SYSMART.2016.7894534

[5] M. Dai and S. Zhang and H. Wang and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access,* vol. 6, 2018, pp. 22970–22975. DOI: 10.1109/ACCESS.2018.2814624

[6] C.-W. Wang and S.E. Chang, "Cloud service in stock trading game: Service virtualization, integration and financial application," *in 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN) IEEE,* 2016, pp. 857–862. DOI: 10.1109/ICUFN.2016.7537158

[7] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008, Retrieved Nov 20, 2018 from https://bitcoin.org/bitcoin.pdf

[8] M. Swan, Blockchain: Blueprint for a new economy, 2015, " O'Reilly Media, Inc."

[9] N. Szabo, Smart contracts, 1994, unpublished. Retrieved Dec 15, 2018 from http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

[10] V. Buterin, "A next-generation smart contract and decentralized application platform," white paper, 2014.

[11] F. B. Schneider, "Implementing fault-tolerant services using the state machine approach: A tutorial," *ACM Computing Surveys (CSUR),* vol. 22, no. 4, 1990, pp. 299–319. DOI: 10.1145/98163.98167

[12] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS),* vol. 4, no. 3, 1982, pp. 382–401. DOI: 10.1145/357172.357176

[13] X. Xu et al., "The blockchain as a software connector," *in 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), IEEE* 2016, pp. 182–191. DOI: 10.1109/WICSA.2016.21

[14] D. Drescher, "Blockchain Basics: A Non-technical Introduction in 25 Steps," 1st edn, Apress, Frankfurt am Main, 2017

[15] K.-H. Huang, Application of blockchain in the field of securities trading: settlement delivery, 2017, Retrieved March 3, 2019 from http://www1.cof.nkfust.edu.tw/ezfiles/12/1012/img/1938/729871942.pdf

[16] N. Herbaut and N. Negru, "A model for collaborative blockchain-based video delivery relying on advanced network services chains," *IEEE Communications Magazine,* vol. 55, no. 9, 2018, pp. 70–76. DOI: 10.1109/MCOM.2017.1700117