Research on Digital Smart Contract Based on Blockchain

Shiyao Zhang[1], Cheng Cheng[2], Zhimin Chen[3], Yuxuan Shi[3] and Dingli Sun[3, *]

[1] Department of investment, School of finance, Nanjing University of Finance and Economics, China
[2] Department of finance engineering, School of finance, Nanjing University of Finance and Economics, China
[3] Department of finance, School of finance, Nanjing University of Finance and Economics, China
[*]Corresponding author e-mail: aldrich1013sdl@163.com

ABSTRACT. *With the continuous improvement of the popularity of blockchain in recent years, smart contracts have also gradually attracted people's attention. Smart contracts are a section of code that can automatically run, allowing developers to develop personalized according to specific protocols, business or logic, and blockchain. The characteristics of decentralization and distribution provide a good platform for the application of smart contracts.*

KEYWORDS: *blockchain, smart contract, super ledger, missile; data storage*

1. Blockchain open source platform

1.1 Basic architecture of blockchain

According to the degree of openness of the blockchain network, blockchain can be divided into the following three categories, what are, public chain, alliance chain and private chain. Blockchain technology has come into view with the proposal of Bitcoin, and there have been many areas in recent years Blockchain researchers, foundations, enterprises, alliances, and active blockchain enthusiasts have done a lot of work on the blockchain open source platform, and many excellent blockchain platforms have emerged.

1.2 Ethereum

The breakthrough achieved by the Ethereum platform is that it supports users to write smart contracts and issue their own digital currencies. The Ethereum platform allows users to write smart contracts using the Sodility programming language.

A consensus mechanism is introduced in the blockchain. The role of the consensus mechanism is to enable each node to reach consensus efficiently in a decentralized system with highly dispersed decision-making power, and to keep all distributed ledger data in sync.

1.3 Super Ledger

The Linux Foundation opened the most active version of the Fabric community in the original organization. It adopts a modular architecture design to support the development and use of pluggable components. The main components of Fabric include: client and network node (Peer), CA (Certificate Authority) node and ordering service node (Orderer).

1.4 Other platforms of blockchain open source

Ethereum and Hyperledger are two epoch-making projects in the development of blockchain technology. The emergence of Ethereum marks the entry of the blockchain into the 2.0 era, and the Hyperledger project represents the era of blockchain 3.0. In addition, there are many blockchain platforms, such as BitShares, Ripple, and China's Hyperchain, which are active on the world Bitcoin stage.

2. Smart contract

2.1 Introduction to Smart Contract

The essence of a smart contract is a modular, reusable, and automatically executed script code that runs on the network. At the same time, it also has two important features, what are, Turing Complete and Sandbox Isolation. It is also because of the characteristics of smart contracts, in the early days, it did not receive widespread attention and application. With the emergence of Bitcoin, blockchain, and various blockchain open source platforms, smart contracts have gradually been implemented by concepts, and now smart contracts have become an important part of blockchain applications.

2.2 Smart contracts in Hyperledger

In Fabric, the smart contract is also called chain code, which is mainly divided into system chain code and user chain code. The chain code designed and written in blockchain application development refers to the user chain code. In Fabirc, access, modification, and querying ledgers are done by calling the chain code of the deployment. It is generally written in the Go language, implementing code for the specified interface through the API provided by Fabric. These APIs are located in the Shim package in the Fabric source code. At the same time, chain codes can also call each other, which are the same as smart contracts in Ethereum. Currently, Fabric supports the development of chain codes using Go, Java, and Node.js, among which Go is the best and most stable.

3. Design of a smart contract for the digital registration of an aviation missile business

At present, the form of military aviation missile business registration is mainly manual registration. There are many problems and hidden safety hazards in the manual registration method. Blockchain technology has the characteristics of decentralization, data traceability, and tamper resistance. Therefore, based on the blockchain technology, digitizing the registration of aviation missile business can largely solve the problems of the current manual registration method. The aviation missile business is an application scenario, which carries out blockchain architecture design, smart contract design and development. According to the basic architecture of the blockchain, the contract layer is deployed on the data layer, network layer and consensus layer. Therefore, before designing, you need to design the blockchain architecture first.

3.1 Blockchain architecture design

Compared with the traditional blockchain architecture, the digital registration of a certain type of aviation missile business studied in this paper does not require a concept similar to the currency in the Bitcoin system, so there is no need for an incentive layer, and the contract layer is the application layer. Connected through the client (CLI / SDK). Specific design and improvement based on the traditional blockchain architecture.

In Fabric, there are two kinds of databases that support KVS storage to choose from. I chose CouchDB here, because Fabric supports Rich Query to CouchDB, and CouchDB also has a graphical management interface, which provides great convenience for developers, and CouchDB supports JSON data format, which is more suitable for table data storage. P2P network topology is shown in the following figure.

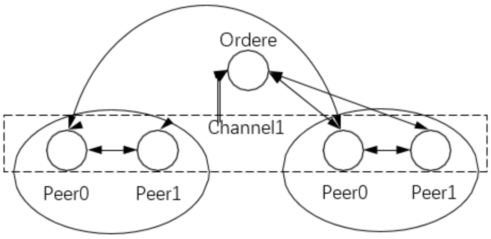

Figure. 1 P2P network topology

The consensus layer uses the PBFT consensus algorithm. PBFT is a consensus algorithm that considers Byzantine fault tolerance.It can tolerate up to one-third of malicious nodes in the network. Compared with PoW, PoS and other consensus algorithms, PBFT is more efficient. At the same time, because the scale of the alliance chain is much smaller than that of the public chain, malicious nodes have a greater impact on account synchronization, and the use of PBFT can better ensure that each node in the blockchain effectively reaches consensus.

3.2 Smart contract design

The main difference between the applications of blockchain in traditional applications is the use of smart contracts to implement the main business logic. Smart contracts are both the main body of program logic and the main body of data storage. The aviation missile business is divided into many aspects. The airtightness check in preparation is an example for smart contract design.

The air tightness check records the internal pressure at a certain moment, then inflates, and then maintains a certain pressure value for a period of time, and then records the pressure value. If the difference between the two pressure values is less than a certain threshold, it means that the gas spring. The conclusion of the tightness check is qualified, otherwise it is unqualified.

The smart contract must implement the Init interface and the Invoke interface. Init is to initialize the smart contract. The result of the execution is to create a database in the state database to store the data written to the ledger through the smart contract. There is no consideration to enter data in the database in advance, so the Init interface does not need to operate during the implementation process, only need to return a Success message; The Invoke interface is the interface to execute the transaction, the role in this article is to call the written function.

The Invoke interface requires the client to enter two parameters including the function and the required parameter. Invoke in a smart contract only needs to receive the function and required parameters passed in by the client through the method of *GetFunctionAndParameter* in the Shim package. Then by writing the determination method inside Invoke, you can specify the written function by entering the corresponding function as needed.

According to the operation process and actual business requirements of a certain air missile airtightness check, the custom functions that need to be called in the Invoke interface are as create, Query, queryTime, queryJl and queryCreator, etc.

4. Development, testing and security analysis of smart contracts

4.1 Smart contract development

Choose Ubuntu16.04 as the development and running environment, the Fabric version is the most stable v1.2, and the Go language uses the stable version 1.11.4. During the airtightness check in the preparation of a certain type of air missile technology, during the airtightness check in, the information to be recorded is: missile number, inspection date, inspector, start time, start pressure (0.1 ± 0.005 Mpa), end time, end pressure (Mpa), pressure difference (Mpa) and conclusion (pressure difference (≤0.01 Mpa is qualified), a total of 9 data items. The pressure difference and conclusion needs to be judged by the function of Create inside the smart contract, first calculate the pressure difference, and then judge the conclusion based on the relationship between the pressure difference and the threshold.

4.2 Smart contract testing

Use the API provided by the Shim package to write each custom function function, and then deploy and debug. Enter the corresponding command in the CLI client to get the running result. Firstly, enter the command of docker exec –it ci bash to the client CLI container, and use *peer chaincode install* to install the chain code. You need to specify the channel during installation. The channel created in this article is named channel1; and then use the command of *peer chaincode init* to initialize the smart contract. Since there was no requirement when implementing the Init interface before The CLI passes in any parameters, so it is sufficient to enter the Function parameter and Args parameter arrays. After the initialization is complete, a database about the smart contract is created in CouchDB. Finally, use the command of *peer chaincode invoke* to enter the channel and smart contract. And you need to pass in the Function parameter and Args parameter array in order to get the query result,

Through the design, development and demonstration of smart contracts, it can be seen that compared with manual registration in paper, the use of blockchain technology has the advantages of traditional digital registration system, which is conducive to storage, format specification, and fast registration speed. The centralized distributed storage method ensures that the registration data can be traced and cannot be tampered with. However, to further apply the blockchain technology to the field of aviation missile business registration, further research and exploration is needed.

4.3 Security analysis

The system designed in this article uses a NoSQL database. CouchDB as the storage environment, which can effectively prevent SQL injection attacks compared to traditional data logging software that uses SQL databases.

A smart contract is a code implementation of related business or logic. Its purpose is not to rely on a trusted third party to perform various operations. If it is maliciously modified, it will cause business or logic confusion, and even damage the system. Therefore, the smart contract Protection is important content. The smart contract privacy protection measures proposed in this article are mainly to install and run smart contracts on Docker. Docker is a container that can be virtualized but is different from virtualization. Containers are completely sandboxed there is no interface between the mechanisms. At the same time, there are also functions in the Linux system kernel that implements container resource isolation and resource limitation. When the smart contract runs on the Docker container, it can ensure the independence of the smart contract. The smart contract already installed on Docker is also invisible to the outside world, even if the smart contract code is modified, the installed smart contract will not be affected.

5. Conclusion

Aiming at the problems and hidden dangers in the registration of the traditional aviation missile business of the army, this paper puts forward the plan of digitizing the traditional registration method based on blockchain technology, and completes the design of the smart contract, and finally carries on the development and testing of the intelligent contract for the flight missile air tightness detection registration. It also makes a preliminary exploration of the application of blockchain technology in the work of the army, and proves that the scheme has good feasibility and effectiveness.

Acknowledgements

This paper is supported by Key Project of Jiangsu Provincial Undergraduate Innovation and Entrepreneurship Training Program (201910327022Z), Jiangsu Provincial Research Fund for Philosophy and Social Sciences in Universities and Colleges (2018SJA0245) and Fintech Course Team Develepment Fund in School of Finance at NUFE.

References

[1] Zhang Jun, Gao Wenzhong, Zhang Yingchen, etc. Intelligent distributed power system running on blockchain: demand, concept, method and outlook. Journal of Automation, 2017, 43 (9): 1544.

[2] Tuo Xiaozhong. The application of blockchain in authentication. Science and Technology Economics, 2017, (3): 26-27.

[3] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf, 2018-12-23.

[4] Lee K, James JI, Ejeta TG, etal. Electronic voting service using block-chain. Journal of Digital Forensics, Security and Law, 2016, 11 (2): 8.