Smart Contract Design Based on Block-chain Heperdeger Fabric

Jianwei Zhang[1[0000-0002-9291-3602]], Chunfeng Du[2[0000-0003-0097-5003]], Zengyu Cai[2[0000-0001-8824-0815]] and Zuodong Wu[1[0000-0003-1204-9381]]

[1] Zhengzhou Institute of Light Industry, School of Software, china
[2] Zhengzhou Institute of Light Industry, School of Computer and Communication Engineering, chain
lncs@springer.com

Abstract. Due to the characteristics of decentralization, transparency, security, and non-tamperability, the block-chain has received widespread attention since it was discovered. The key technologies of the block-chain are not only the main methods of block-chain network and system composition that include P2P core networking, distributed accounting, asymmetric encryption, smart contracts, and the proof mechanism and broadcast mechanism in the core mechanism. It is also the core technology that constitutes a new digital cryptocurrency. As a main feature of the block-chain, smart contracts are also the main reason why block-chain technology is called subversion of traditional technologies. This article briefly introduces the block-chain and smart contract technology, introduces the Hyperledger fabric as one of the three technologies of the block-chain, and uses the fabric project to develop a simple smart contract with transfer function. Based on the example of asset trading, this article briefly describes asset trading and fully reflects the characteristics of blockchain. Through this smart contract system, people can further learn and understand Heperledger's architecture and blockchain technology.

Keywords: Block-chain, Dyperledger, Fabric, Smart Contract.

1 Introduction

Blockchain technology originates from Bitcoin and is the underlying technology of Bitcoin. It is also the earliest application object of blockchain technology [1,2]. The blockchain itself has the characteristics of distributed data storage, point-to-point transmission, consensus mechanism and encryption algorithm, which is the core of the new generation of computer technology applications[3]. Its own decentralization feature transforms the “centralized mutual trust mode” in the traditional network into a new point-to-point “centralized mutual trust mode”. The concept of Smart Contract was first proposed by computer cryptographer Nick Saab in 1994. In simple terms, a smart contract is the underlying programming code of a computer, that is, pre-programmed program code, which needs to use external information to determine whether internal conditions are met. If the condition is met, the system activation program is triggered in time to execute the relevant code [4]. As a point-to-point distributed ledger technology[5], blockchain is very safe and effective for the issuance, management and transaction of assets. Therefore, it has been closely watched by the financial sector since it was proposed. The rapid development of blockchain technology has also created many different blockchain platforms. In 2015, the Linux Foundation launched an open source project called “Super Book” for the purpose of advancing blockchain technology, namely Hyperledger [6]. Hyperledger has innovated on traditional blockchain models. The old blockchain model has no security settings for the management participants, which will cause certain security problems. On the basis of this, Hyperledger adds identity verification and auditing, which improves the security of the previous blockchain model, thereby reducing the computational cycle and improving the scalability to meet the needs of different fields [7].

2 Block-chain and smart contracts

2.1 Block-chain

A block-chain is a distributed database that is almost impossible to change. Its "distributed" is not only reflected in the distributed storage of data, but also in the distributed recording of data (the collective maintenance of most network node participants). The decentralization of the block-chain is reflected in the ability to achieve distributed records of data, rather than a centralized record of participation by a central organization. From the results of the records, the block-chain is an unchangeable, trusted database that can generate a set of records in chronological order.

A block-chain is a data structure that connects blocks in a chain. It is suitable for storing reliable, sequential data that can be verified in the system. Its encryption and decryption algorithm based on cryptography ensures the immutability and forgery of data. It allows participants to reach a consensus on the order of transactions across the network and the state at this time.

2.2 Smart contract

Unlike the traditional sense of smart contracts, the emergence of block-chain has redefined smart contracts. From a user perspective, a smart contract is usually considered an automatic guarantee account. For example, when certain conditions are met, the program releases and transfers funds. From a technical point of view, smart contracts are considered to be web servers. These servers are not built on the Internet using IP addresses, but rather on the block-chain. So it can run a specific contract program on it. But unlike web servers, smart contracts can be seen by everyone because the code and state of these smart contracts are on the block-chain (assuming the blockchain is public). Moreover, unlike a web server, a smart contract does not depend on a particular hardware device. In fact, the code for a smart contract is executed by all devices involved in mining[8].

The smart contract runs as follows, and encapsulates a predefined set of instruction operations, including states, rules, and conditions, based on the contract content [3]. After the contract is signed, it will be presented in the form of code. After being propagated and verified by the blockchain network, it will be recorded in the distributed ledger of each node as part of the underlying data. The entire smart contract that exists in the block at the same time is in a monitored state, and will be activated when certain trigger conditions are met, and then the contract content is executed [9].

3 Hyperledger fabric

The rapid development of block-chain technology has also created many different block-chain platforms. As a large-scale open source project, Heperledger has attracted the participation of different giants. The main members include: IBM, Intel, JP Morgan Chase and so on. The Hyperledger project is developing very rapidly, as shown in Fig 1.

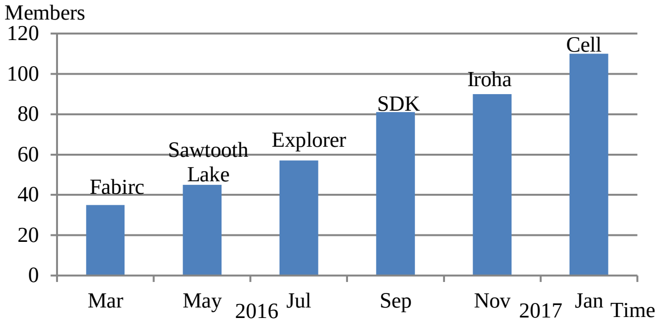


Fig. 1. Hyperledger project grows rapidly

At the same time, Chinese companies including Wanda Finance, Huawei, and Aiyi Digital Technology have also begun to pay attention to and invest in them. If the currency block-chain technology represented by B itcoin is version 1.0, the contract block-chain technology version 2.0 represented by Ethereum. Then the Hyperledger project, which implements complete access control and security, represents the arrival of the block-chain technology 3.0 era [10,11].

3.1 Fabric architecture

As an open source system, fabric is modular and extensible, primarily for deploying and operating managed Heperledger projects, while supporting scalable block-chain systems and consensus algorithms, and allowing systems to be customizable[11]. The architecture of the Hyperledger fabric version 0.6 is shown in Fig 2.

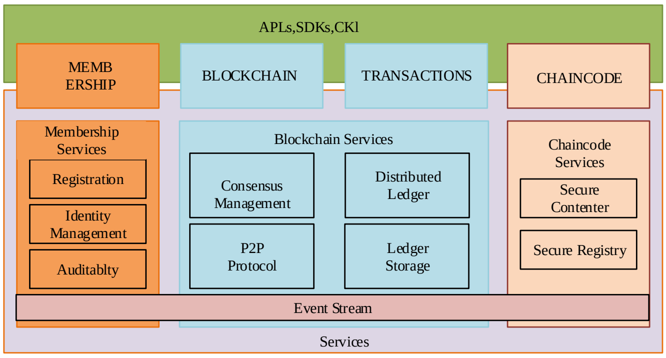


Fig. 2. Hyperledger fabric v0.6 overall architecture

Membership Services: This service is used to manage node identity, privacy, confidentiality, and trialability.

Block-chain Services: Block-chain services use a point-to-point protocol created over HTTP to manage distributed ledgers, providing the most efficient hashing algorithm to maintain a copy of the worldview.

Chaincode Services: Chaincode services operate in a sandbox mode, which is characterized by security and lightness and requires less resources [12].

Events: In a block-chain network, VP computer nodes and smart contracts send events. Its purpose is to trigger some behavior.

API and CLI: Different smart contracts have different API interfaces and rich API interfaces can realize many functions.

3.2 Fabric topology

In the block-chain network, there have a Membership service which is a simple VP computer node and an NVP computer node that can share the pressure of the VP node. One or more applications that form one or more chains, each with its own security and operational requirements [13]. A single VP node network and multiple VP node networks are shown in Fig 3 and Fig 4.

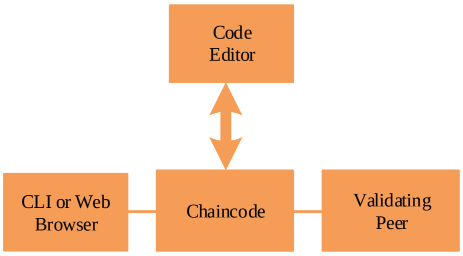


Fig. 3. Schematic diagram of a single node network

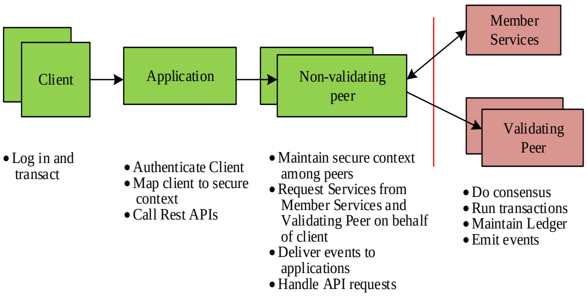


Fig. 4. Schematic diagram of multiple VP nodes

3.3 Fabric protocol information

The Fabric protocol mainly implements P2P communication through Grpc, and its important feature is bidirectional flow information transmission. At the same time, Protocol Buffer is used to serialize the data structure to be passed.

Message information: Fabric's messages mainly have the following types: Discovery, Transaction, Synchronization, and Consensus. The payload is an opaque array of bytes that contain information about transactions, such as transaction information or feedback. If type is CHAIN_QUERY, payload is the object of a transaction query.

Probe node(discovery): A newly started peer, if the performance state is CORE_PEER_DISCOVERY_ROOTNODE, the discovery protocol will be start ran. ROOTNODE is the discovery node for which the IP is specified, and then the ROOTNODE node discovers all the nodes in the blockchain network. DISC_HELLO is the protocol information of Discovery. Its payload is a HelloMessage object, and contains the information of the information sending node [14].As shown in Tab 1, the different fields in the node indicate different meanings.

Table 1. Attribute Meaning in Node Information

Attributes	meaning
PeerID	The name of the node defined at the beginning of the startup or defined in the configuration file
PeerEndpoint	Describe the node and determine if it is an NVP and VP node
PkiID	The encrypted ID of the node
Address	ip：port
BlockNumber	The height of the Blockchain currently owned by the node

A single peer implements a synchronization protocol message through the DISC_HELLO message, and then implements full network synchronization in the blockchain. After the node completes the DISC_HELLO message delivery, it periodically sends DISC_GET_PEERS to discover other nodes joining the network. At the same time, in order to reply to DISC_GET_PEERS, the node will send DISC_PEERS.

Synchronization information: When a peer discovers that the status of the block is inconsistent with other peers, the synchronization information protocol is triggered at this time. The node broadcasts three kinds of information: SYNC_GET_BLOCKS , SYNC_STATE_GET_SNAPSHOT , or SYNC_STATE_GET_DELTAS. At the same time, three kinds of information are received: SYNC_BLOCKS, SYNC_STATE_SNAPSHOT or SYNC_STATE_DELTAS.

SYNC_GET_BLOCKS will request a series of contiguous blocks, and the payload in the sent data structure will be a SyncBlockRange object.

4 Briefly describe the design of smart contract

4.1 Docker program installation

Smart contracts that want to run on a computer must rely on a suitable platform [15]. This article selects the Hyperledger Fabric project as the development platform, and runs the system in the Ubuntu system with the help of Docker container. Get root privileges before installation, enter the following command in Ubuntu open terminal:

- Docker installation

```
sudo apt-get update
sudo apt-get install docker
```

Docker info for docker information version query

- Docker-sompose installation

`apt-get install python-pip` --First need to install the python-pip installation tool
`pip uninstall docker-compose` --If there is an old version of docker-compose, uninstall
`pip install docker-compose` --Install docker-compose
`docker-compose –vesion` --Query docker-compose version information

- Pull the mirror

`docker pull hyperledger/fabric-peer:latest` --Pull the mirror image of the peer
`docker pull hyperledger/fabric-membersrvc:latest` --Pull the mirror of memberserver

4.2 Intelligent contract system design

The system uses the go language to write the Hyperledger fabric[15], which mainly implements the following functions.

- Initialize two accounts a, b, and assign initial asset values to both accounts.

- Ability to trade assets between two accounts a and b
- Check the balances of the two accounts a and b respectively to confirm the success of the transaction.
- Delete the account.

The main function:

- Init: Initialize two accounts a and b.
- Invoke: Implement the balance on the a and b accounts.
- Query: Query the balance on the a and b accounts.
- Delete: delete the account.

Dependent package:

```
import(
"errors" "fmt" "strconv"
"github.com/hyperledger/fabric/core/chaincode/shim"
        )
```

Stronv implements the conversion between int and string type conversion. In the invoke function, there is:

```
X,err=strconv.Atoi(args[2])
Aval=Aval-X
Bval=Bval+X
```

When args[2]<0, the balance of the A account increases, otherwise the balance of the B account decreases.
The a and b values are initialized to operate the flow chart, as shown in Fig 5.

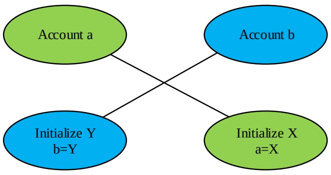


Fig. 5. a, b initialization flow chart

Then account a, account b is based on the smart contract fabric system for trading, the transaction amount is Z. The amount after the transaction and the information at the time of the transaction are shown in Fig 6.

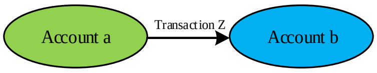
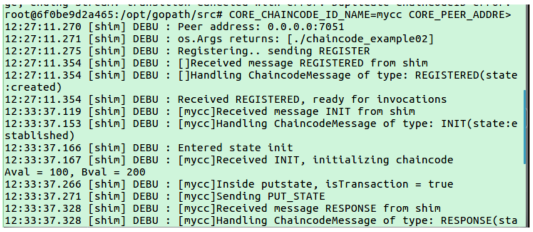


Fig. 6. transaction flow chart

5 System test

Run docker-compose in the /home directory on the Ubuntu system, and enter docker-compose exec vp0 bash on the command line to enter the fabric virtual machine [16,17].

Publish the chaincode to run under GOPATH/src CORE_CHAINCODE_ID_NAME=mycc CORE_PEER_ADDRESS=0.0.0.0:7051./chaincode_example02, as shown in Fig 7.

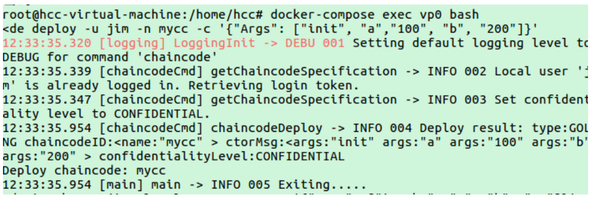


```
root@6f0be9d2a465:/opt/gopath/src# CORE_CHAINCODE_ID_NAME=mycc CORE_PEER_ADDRE>
12:27:11.270 [shim] DEBU : Peer address: 0.0.0.0:7051
12:27:11.271 [shim] DEBU : os.Args returns: [./chaincode_example02]
12:27:11.275 [shim] DEBU : Registering.. sending REGISTER
12:27:11.354 [shim] DEBU : []Received message REGISTERED from shim
12:27:11.354 [shim] DEBU : []Handling ChaincodeMessage of type: REGISTERED(state
:created)
12:27:11.354 [shim] DEBU : Received REGISTERED, ready for invocations
12:33:37.119 [shim] DEBU : [mycc]Received message INIT from shim
12:33:37.153 [shim] DEBU : [mycc]Handling ChaincodeMessage of type: INIT(state:e
stablished)
12:33:37.166 [shim] DEBU : Entered state init
12:33:37.167 [shim] DEBU : [mycc]Received INIT, initializing chaincode
Aval = 100, Bval = 200
12:33:37.266 [shim] DEBU : [mycc]Inside putstate, isTransaction = true
12:33:37.271 [shim] DEBU : [mycc]Sending PUT_STATE
12:33:37.328 [shim] DEBU : [mycc]Received message RESPONSE from shim
12:33:37.328 [shim] DEBU : [mycc]Handling ChaincodeMessage of type: RESPONSE(sta
```

Fig. 7. Loading a smart contract

The Hyperledger fabric rights management mechanism needs to log in to the official jim user to log in.

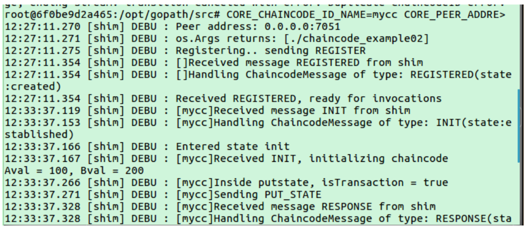
After the login is successful, we also need to perform initial operations on the a and b accounts. Enter the following command:CORE_SECURITY_ENABLED=true CORE_SECURITY_PRIVACY=true peer chaincode deploy -u jim -n mycc -c '{"Args": ["init", "a","100", "b", "200"]}' as shown in Fig 8.



```
root@hcc-virtual-machine:/home/hcc# docker-compose exec vp0 bash
<de deploy -u jim -n mycc -c '{"Args": ["init", "a","100", "b", "200"]}'
12:33:35.320 [logging] LoggingInit -> DEBU 001 Setting default logging level to
DEBUG for command 'chaincode'
12:33:35.339 [chaincodeCmd] getChaincodeSpecification -> INFO 002 Local user 'ji
m' is already logged in. Retrieving login token.
12:33:35.347 [chaincodeCmd] getChaincodeSpecification -> INFO 003 Set confident
ality level to CONFIDENTIAL.
12:33:35.954 [chaincodeCmd] chaincodeDeploy -> INFO 004 Deploy result: type:GOL
NG chaincodeID:<name:"mycc" > ctorMsg:<args:"init" args:"a" args:"100" args:"b"
args:"200" > confidentialityLevel:CONFIDENTIAL
Deploy chaincode: mycc
12:33:35.954 [main] main -> INFO 005 Exiting.....
```

Fig. 8. initialization operation

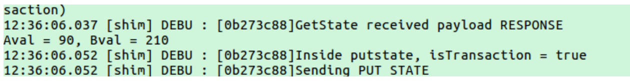
The initialization operation is successful, Aval=100, Bval=200, as shown in Fig 9.



```
root@6f0be9d2a465:/opt/gopath/src# CORE_CHAINCODE_ID_NAME=mycc CORE_PEER_ADDRE>
12:27:11.270 [shim] DEBU : Peer address: 0.0.0.0:7051
12:27:11.271 [shim] DEBU : os.Args returns: [./chaincode_example02]
12:27:11.275 [shim] DEBU : Registering.. sending REGISTER
12:27:11.354 [shim] DEBU : []Received message REGISTERED from shim
12:27:11.354 [shim] DEBU : []Handling ChaincodeMessage of type: REGISTERED(state
:created)
12:27:11.354 [shim] DEBU : Received REGISTERED, ready for invocations
12:33:37.119 [shim] DEBU : [mycc]Received message INIT from shim
12:33:37.153 [shim] DEBU : [mycc]Handling ChaincodeMessage of type: INIT(state:e
stablished)
12:33:37.166 [shim] DEBU : Entered state init
12:33:37.167 [shim] DEBU : [mycc]Received INIT, initializing chaincode
Aval = 100, Bval = 200
12:33:37.266 [shim] DEBU : [mycc]Inside putstate, isTransaction = true
12:33:37.271 [shim] DEBU : [mycc]Sending PUT_STATE
12:33:37.328 [shim] DEBU : [mycc]Received message RESPONSE from shim
12:33:37.328 [shim] DEBU : [mycc]Handling ChaincodeMessage of type: RESPONSE(sta
```

Fig. 9. Initialization assignment assignment

CORE_SECURITY_ENABLED=true CORE_SECURITY_PRIVACY=true peer chaincode invoke -u jim -l golang -n mycc -c '{"Args": ["invoke", "a", "b", "10"]}'. After we saw a to b for 10 b, we can see that a became 90 bucks and b became 210 bucks, as shown in Fig 10.



```
saction)
12:36:06.037 [shim] DEBU : [0b273c88]GetState received payload RESPONSE
Aval = 90, Bval = 210
12:36:06.052 [shim] DEBU : [0b273c88]Inside putstate, isTransaction = true
12:36:06.052 [shim] DEBU : [0b273c88]Sending PUT_STATE
```

Fig. 10. Transaction Information

6 Conclusion

Blockchain technology will have a huge impact on real life in the near future, so standardized blockchain technology is necessary to promote its application in different fields. In the future world, distributed databases and blockchains are booming, and they are numerous and interconnected. Each distributed database and blockchain will meet the needs of specific users, and can communicate with other books to form a huge network [18,19].

Adapted to the unknown development of the future, Heperledger's design must meet the modularity and high scalability. Easy to use, practical and robust is the basic requirement for the development of Heperledger. For future long-term development, Heperledger should also include a large number of easy-to-use interface APIs and customizable core development modules, in order to have more people involved in the Heperdeger development and design process. This makes the Heperdeger system more complete and more powerful [20]. The Hyperledger project is still in the incubation phase and its prospects are still very clear. I believe that in the future, blockchain technology will become popular. Hyperledger, as the leader of the blockchain, plays a very important role in the development of blockchain.

Acknowledge.

This work is supported by National Natural Science Foundation of China (No.61672471, No.61502436), Key Technologies R & D Program of He'nan Province (No.172102210059 and No.172102210060), He'nan Province University science and technology innovation team(No.18IRTSTHN012) and Plan For Scientific

Innovation Talent of Henan Province (No.184200510010), Cernet Network (NGII20160103).

References

1. Narayanan, A. & Clark, J. (2017), 'Bitcoin's Academic Pedigree', Communications of the ACM 60 (12), 36-45.
2. Bitcoin Sourcecode [EB/OL]. https://github.com/bitcoin/bitcoin/. 2017/5/31.
3. Y. Yuan, F. Y. Wang, “Development Status and Prospects of Blockchain Technology,” Journal of automation, vol. 42, no. 4, pp. 481-494, 2016.
4. Szabo N.Smart Contracts[R].http://szabo.best.vwh.net/.1994.
5. Nakamoto S. Bitcoin: A peer-to-peer Electronic cash system [EB/OL]. [2016-10-07]. https://bitcoin. org/ bitcoin. Pdf.
6. L He. Preliminary Study on Blockchain 2.0 Architecture and Its Application in Insurance Industry[J]. Golden Card Project, 2017(z1):45-49.
7. H Zheng, L Xiangxue, L Xuejia, C Kefei. Blockchain Technology and Its Application[J]. Information Security Research, 2017, (03): 237-245.
8. Bailis, P(Bailis, Peter), Narayanan, A (Narayanan, Arvind), Miller, A (Miller, Andrew), Han, S (Han, Song). Research for Practice: Cryptocurrencies, Blockchains, and Smart Contracts; Hardware for Deep Learning[J]. Communications Of The ACM. 2017,(7): 60(5), 48-51.
9. S Xin, Z Qingyu, L Xuefeng. Overview of Blockchain Technology[J]. Journal of Network and Information Security, 2016, 2(11): 12-20.
10. X Jian. Blockchain or Leading the Wave of Subversive Innovation[J]. China Strategic Emerging Industries, 2016, (16): 67-69.
11. Clare Sullivan,Eric Burger. E-residency and blockchain[J]. Computer Law & Security Review: The International Journal of Technology Law and Practice,2017,(16):76-79.
12. Sue McLean,Simon Deane-Johns. Demystifying Blockchain and Distributed Ledger Technology – Hype or Hero?[J]. Computer Law Review International,2016,17(4):135-138.
13. Harald Vranken. Sustainability of bitcoin and blockchains[J]. Current Opinion in Environmental Sustainability,2017,16(5):20-24.
14. Steve Mansfield-Devine. Beyond Bitcoin: using blockchain technology to provide assurance in the commercial world[J]. Computer Fraud & Security,2017,2017(5):30-36.
15. Mark Giancaspro. Is a ‘Smart Contract’ Really a Smart Idea? Insights from a Legal Perspective[J]. Computer Law & Security Review: The International Journal of Technology Law and Practice,2017,8(6):124-128.
16. Androulaki E, Barger A, Bortnikov V, et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains[J]. 2018.
17. Yin Z C, Hang L I, University S N. Solution of Electronic Contract System Based on Hyperledger[J]. Modern Computer, 2018.
18. Cong L W, He Z. Blockchain Disruption and Smart Contracts[J]. Social Science Electronic Publishing, 2018.
19. Marsal-Llacuna M L. Future living framework: Is blockchain the next enabling network?[J]. Technological Forecasting & Social Change, 2018, 128.
20. Dhillon V, Metcalf D, Hooper M. The Hyperledger Project[J]. 2017.