



# Research on Cross-Domain Heterogeneous Information Interaction System in Complex Environment Based on Blockchain Technology

BaoQuan Ma<sup>1</sup>(✉), YeJian Cheng<sup>1,2</sup>, Ni Zhang<sup>1</sup>, Peng Wang<sup>1</sup>, XuHua Lei<sup>1,2</sup>,  
XiaoYong Huai<sup>1</sup>, JiaXin Li<sup>1</sup>, ShuJuan Jia<sup>1</sup>, and ChunXia Wang<sup>1</sup>

<sup>1</sup> National Computer System Engineering Research Institute of China, Beijing 102200, China  
15501210877@163.com

<sup>2</sup> School of Computer Science and Technology, Xidian University, Xi'an 710071, China

**Abstract.** In the field of information interaction, when a project involves a large amount of cross-domain heterogeneous information, it is difficult to transmit and update the required information in a timely, accurate, reliable and safe manner in such a complex environment, thereby maintaining synchronization. The decentralization and immutability of blockchain technology provide new ideas for solving this problem. Therefore, this paper takes this as a starting point and proposes a cross-domain heterogeneous information exchange system based on blockchain, which calls smart contracts to realize data transmission and other functions. This paper states the design idea of the system, introduces the system software framework and network topology in detail, shows the flow chart of the system operation, and conducts simple software testing and verification. Finally, this work is summarized, and the development direction of future work is prospected, hoping to provide inspiration for the solution of such problems.

**Keywords:** Blockchain · Smart Contract · Heterogeneous · Information Interaction

## 1 Introduction

In recent years, with the rapid development of information technology, blockchain technology has been widely used in information interaction control scenarios and cross-domain collaboration due to its unique characteristics such as distributed multi-party accounting, data immutability, and data traceability [1]. For interactive control, the current business scenarios that require multi-organization collaboration (such as user management, task management, data analysis, etc.) are mostly processed based on the systems, processes, and file systems built by each organization [2]. The system has poor openness, and there is a lack of a secure and credible resource sharing, distribution and display mechanism, and it is difficult to integrate domain information. In addition, the system has various security levels, and it is necessary to dynamically grant fine-grained permissions to each participant [3]. Large amounts of data are processed during transmission and security requirements are required. In the case of high performance,

there are also strict requirements for real-time performance, which constitutes a complex environment with obvious characteristics.

In order to complete unified collaborative scheduling in complex environments, ensure that all parties reach a consensus on the content and order of information exchange, and ensure a reliable and immutable record of proven operation history, this paper takes a task-oriented cross-domain collaboration system as an example [4] and proposes a trusted data exchange and logic execution system between cross-domain heterogeneous systems based on blockchain technology. Committed to helping agencies achieve multi-agency mission collaboration on the basis of minimizing system modifications [5].

In view of the above situation, this paper takes task-oriented interactive control business as an example, and uses blockchain technology to achieve four goals among multiple cross-domain heterogeneous systems [6]:

First, based on the mapping between the blockchain user system and cross-domain application users, it realizes user permission control, user access management, and consensus mechanism joint execution.

The second is to build a trusted data exchange system between cross-domain applications based on the characteristics of blockchain data that cannot be tampered with.

The third is to build a credible cross-domain logic execution mechanism, coordinate multi-party process flow, and intelligently execute multi-party agreements based on blockchain-based smart contracts, consensus algorithms [7], and remote call protocol data transmission functions.

The fourth is to realize the interaction of data flow, task flow and message flow between cross-domain applications on the basis of the above three points.

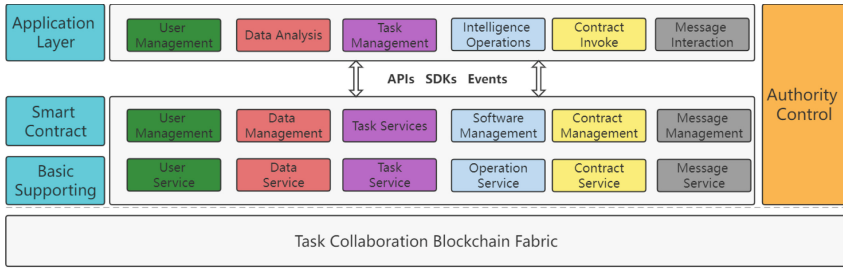
The structure of the paper is as follows: Sect. 2 describes the Blockchain Architecture Design for Task Management Business; Sect. 3 describes the Software Design; Sect. 4 summarizes work.

## 2 Blockchain Architecture Design for Task Management Business

The overall framework designed in this paper is of cross-domain heterogeneous system information exchange software with layered encapsulation, bottom-up abstraction and top-level application-oriented. The entire information interaction software is divided into four layers from bottom to top, namely the task collaboration blockchain Fabric [5], the basic support layer, the smart contract layer and the application layer composed of functional modules corresponding to smart contracts. At the same time, strict authority control is implemented on the entire framework to enhance the overall security of the system. The overall framework is shown in Fig. 1. The proposed method implementation is based on a permissioned blockchain and the Hyperledger Fabric blockchain platform (HLF; [www.hyperledger.org](http://www.hyperledger.org)) [8].

### 2.1 Task Collaboration Blockchain Fabric Layer

The core part of this paper uses blockchain to achieve cross-domain heterogeneous information interaction. Fabric integrates three parts: data layer, network layer, and consensus



**Fig. 1.** System Framework

layer. The data layer of blockchain uses a chain structure to connect heterogeneous data blocks from different sources. In this paper, we define the corresponding data structure for tasks, message, etc. The data propagated on the chain is encrypted with a hash function, both of sender node and receiving node use asymmetric encryption at the same time to maintain data security; At the network layer, reflected in the P2P network architecture, the sender notifies the receiver to receive data through broadcast, and the receiver verifies the received data; The consensus layer is core of security and stability of the blockchain, the receiver determines whether the new module established by the sender can be accepted through a consensus algorithm. If the returned received information meets the specified standards, a consensus is reached, and the receiver adds the new module to its own library. The consensus mechanism is used to jointly maintain the stability of the blockchain.

## 2.2 Basic Support Layer

The basic support layer is the foundation that supports the entire software framework. It is divided into six modules: user service, data service, task service, operation service, contract service, and message services.

**User Service.** User information is stored in a separate database, and functions such as user registration and cancellation can be performed. Users can log in to a successfully registered account to use related functions.

**Data Service.** During the operation of the system, a large amount of initial data, process data and result data will be generated. Heterogeneous data from different systems, departments and even network domains need to be stored and processed in a timely, complete and secure manner. Data transmission and processing have high requirements on the reliability of network transmission. The data service combined with blockchain technology can avoid this problem well.

**Task Service.** With the help of the blockchain network, the task distribution can be notified in real time, and participants can also use it as a receiver to receive tasks in time. The distribution and execution of tasks, the allocation of appropriate resources and other necessary requirements for each task, each participant, etc., relies on the support of blockchain smart contract technology. All participants advance tasks in accordance

with smart contracts, avoid wasting resources, and maintain system operation jointly efficiently.

**Operation Service.** Operation service is the technical support and maintenance service provided for the healthy and stable operation of target application systems, operating environments, and business functions. It uses information technology methods and means which are combined with the actual needs of users. The functions it provided includes basic environment maintenance, software operation service, and security operation service, operation management service and other functions.

**Contract Service.** Contract service is the technical support for the normal and effective operation of smart contracts. It supports a variety of smart contracts defined on the blockchain to make promises in a digital form, and stipulates the rights and obligations of each participant. Operations on the blockchain need to invoke the relevant smart contracts and operate according to the contract content, which solves the problem of mutual trust between the participants and make all participants jointly maintains the operation of the system.

**Message Service.** Message services provide the underlying logic to update the message system in real time. Message services need to process a large amount of information, such as regional weather information, equipment resource information, task execution information, etc. A variety of information can be transmitted across domain to achieve real-time updates, providing the required data sources for data analysis, intelligent operation, etc.

### 2.3 Smart Contract Layer

There are six major contracts in the smart contract layer, which correspond to the six major sectors provided by the basic support layer. The smart contract mechanism is the core key technology of blockchain technology, and the interaction standards on the chain are unified through this mechanism. By accessing the smart contract, the data is classified by different purpose modules, and the connection between the upper and lower layers is completed relatively quickly. It is the link between the layers and provides data and other support for the application layer.

### 2.4 Application Layer

The application layer is the software functions that are finally provided to users, such as User Management, Data Analysis, Task Management, Intelligence Operation, Contract Invoke, and Message Interaction. The six modules are independent of each other, coupled with each other, and support each other. Task management is the most commonly used and the most important function. Nodes can accept tasks. They can also formulate tasks and submit them to the blockchain through the analysis of initial message and the current needs of the department, coordinate task allocation, and promote tasks. The Message Interaction function allows users to grasp the latest developments in real time and realize

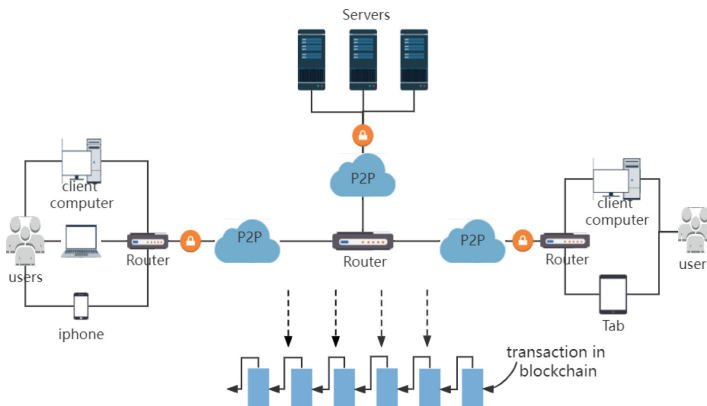
real-time message updates within or across domains. The data analysis function can quickly and deeply analyze the latest intelligence, dig out more useful information, and allow users to proceed more rationally. Based on the existing operation and maintenance data, the intelligent operation system can analyze and summarize the rules from the data based on the existing operation data, and provide solutions, improve the operation and efficiency, and solve the problems that the traditional operation system cannot solve.

## 2.5 Rights Management

Fabric proposes the concept of Member Service Provider (MSP), which abstractly represents an authentication entity. MSP can be used to verify the authority of identity certificates for different resources. Specifically, users who join the system will get the initial minimum permission by default after the registration review. The relevant personnel of authority management can open the corresponding authority for specific registered users according to their computing power, network, equipment platform and other resources' access restrictions, as well as their actual levels. The permissions that users have are not fixed. The blockchain records various resource authorization operations in cross domain collaboration as a trusted authorization record to ensure the authenticity and reliability of authorization operations, and it serves as a trusted basis for restructuring information relationships. Managers have the right to adjust the permissions of relevant personnel according to changes in the situation, and increase or close the corresponding permissions. The strict implementation of authority management ensures the security of data and makes the system run smoothly and efficiently.

## 3 Software Design

### 3.1 Network Topology Design



**Fig. 2.** Network topology design diagram of cross-domain heterogeneous information interaction

In the case of large-scale project tasks involving data processing and participants from different organizations, the environment is complex and changeable, which brings great difficulties to the execution of the task. The equipment systems used by each participant are heterogeneous and not unified, such as servers, desktop computers, portable computers, and mobile phones and other terminals (see Fig. 2), all of which are connected in their own network domains. Compared with devices in other network domains, it processes data in a cross-domain heterogeneous system. Therefore, we use blockchain technology to build distributed network structures in complex environments and connect them to P2P [9]. All transactions between them take place in the blockchain. Through the access to the blockchain and the scheduling of smart contracts, the policy of mutual data access by all parties is achieved.

### 3.2 Cross-Domain Information Interaction Process

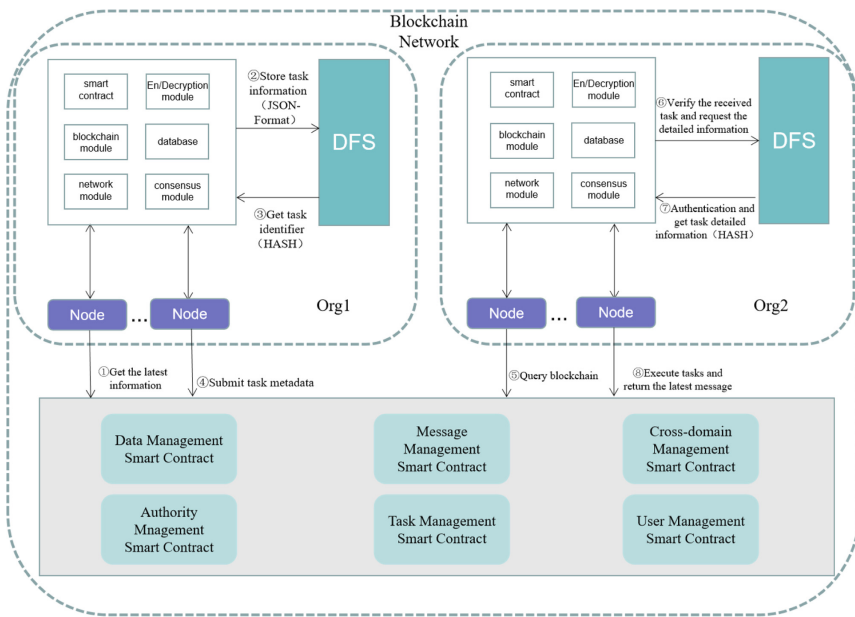


Fig. 3. Flow chart of peer-to-peer task information Interaction

At the system logic execution layer, the system mainly uses the blockchain to invoke smart contracts to achieve cross-domain logical trusted execution [10] (Fig. 3). First, each smart contract will be released in the entire blockchain, and the publisher, time, version, HASH and other certificates are stored on the chain, and then called by the smart contract through the system server.

In the cross-domain execution stage, the smart contract will write the cross-domain call protocol data into the blockchain, and after the cross-domain application obtains the cross-domain call protocol data from the blockchain, the entire cross-domain execution is

completed according to the contract initiator program [11]. And write the execution status to the blockchain according to the cross-domain call contract. Finally, the system server will further invoke the smart contract according to the execution result obtained from the blockchain until the execution of the smart contract is completed. In the meantime, the system will provide an interface so that each participant can directly invoke the smart contract [12].

This paper designs a proof-of-concept model for decentralized data, focusing on smart contracts and query components, and provides a functional overview of the task system. The process is as follows:

- The task sender node obtains the latest message, analyzes the message, and then formulates tasks.
- Store detailed task information including data structures, user identities, etc. The task information is stored in the Distributed File Storage System (DFS).
- When task information is successfully stored in DFS, the file storage framework returns a storage identifier (HASH).
- The sender node must store the task metadata including the storage identifier and contract type into the blockchain and broadcast it to all other nodes.
- The consensus nodes use the PBFT consensus algorithm to check the block information, execute the smart contract and compare the results. It will not pass until they confirm that the transaction is the latest and the signature information is consistent. Other nodes can then query the blockchain after that.
- The receiver node verifies the received task and checks it against the information received by the DFS and blockchain.
- The receiver node obtains the task details through the storage identifier and DFS verification
- The receiver node completes the task and returns the latest message information of the blockchain.

### 3.3 Case Study

To describe the specific business processes within the framework of the HLF platform, many concepts are used, mainly assets, participants, transactions and events. In our case, the most important assets are data files in distributed storage, and their attributes are traceability metadata [13]. Based on the development status and best practices of international open data traceability, we have extracted the provenance metadata in relevant standards and specifications for project tasks. Participants are members of a task collaboration. They can own assets and make transaction requests, that is, when the assets (equipment, personnel, etc.) required by organization 1 are insufficient, they can make transaction requests to organization 2 to meet the resources required to complete the task by trading assets. Transactions are the mechanism for participants to interact with assets. All transactions take place in the blockchain. When all nodes reach a consensus and generate results, transactions cannot be tampered with. Every operation with data contains at least two types of transactions: one for client requests and one for server responses. Event messages can be sent by transaction processors to notify the changes of

external software components in the blockchain. Applications can subscribe to receive event messages via HLF's API.

The task leader can publish specific task information through the interface, including task ID, task title, invited participants, etc. (see Fig. 4). The middle module is the task information interaction module, which will update the execution of the task and obtain other information which is collected or stored by the application system and called real-time intelligence information, in real time. The commander conducts data analysis according to all the data in the module through the data analysis function, and formulates new tasks and releases them to the participants. The nodes participating in the task can view the specific task information through the individual user's task list, and can also update the information in real time to the task information interaction module for all nodes to view. The task description template data structure is displayed (see Fig. 5).

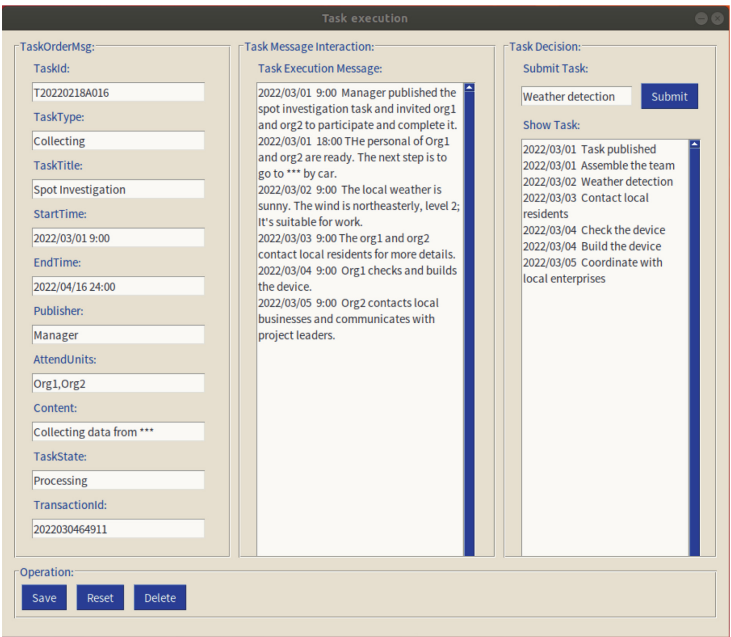


Fig. 4. Task execution interface diagram

### 3.4 Performance Analysis

In the traditional information interaction system, resource discovery, information interaction, etc. all need to follow multi-level processes. In this way, processes are solidified, and failure at any level in the middle may cause overall process failure, causing task interruption. Compared with the traditional way, the delay is longer and fragile. Table 1 lists the qualitative comparison between the performance of traditional hierarchical interaction and the performance of cross domain heterogeneous information interaction in a complex blockchain based environment in the case of task management-oriented business.



```

Task Description Template
{
  " TaskId" :T20220218A016
  " TaskType" : " Collecting"
  " TaskTitle" : " Spot Investigation"
  " StartTime" : " 2022/03/01 9:00"
  " EndTime" : " 2022/04/16 24:00"
  " Publisher" : " Manager"
  " AttendUnits" : " org1;org2"
  " Content" : " Collecting data from"
  " TaskState" : " Processing"
  " TransactionId" : " 2022081164911"
}
    
```

**Fig. 5.** Description of a task data struct in JSON format

**Table 1.** Qualitative comparison of information interaction system performance

Parameter	Traditional information interaction system	Cross domain heterogeneous information interaction system based on blockchain
Number of data forwarding	More	Less
Number of instructions forwarding	More	Less
Access control mode	Control circulation scope	Password technology authorization
Data request steps	More	Less
Emergency response speed	Slow	Fast
Single point decision speed	More	Less
Data storage space	Small	Large
Network traffic	Small	Large

For the performance of the cross domain heterogeneous information interaction mode of the blockchain, this paper makes a preliminary illustrate as follows.

Task instructions are recorded in the form of transactions as reliable regulatory records and objective basis. The tasks in the information interaction system are strictly standardized and distributed, and the feedback information of each task completed is also realized in the form of transaction. Under the scale of 1000 nodes, assuming that each node issues 30 instructions to each unit under its jurisdiction within 1 min, the TPS required to be provided by the blockchain system is  $30/5 * 1000 * 2 = 1000$  TPS. At present, the consensus mechanism blockchain system such as PBFT can fully meet such TPS speed requirements.

## 4 Summary

Aiming at the need to complete unified collaborative scheduling in complex environments, to ensure that all parties reach a consensus on the content and order of information

exchange, and to ensure reliable and immutable recording of proven operation history, this paper proposes a task-oriented trusted data exchange and logic execution architecture between cross-domain heterogeneous systems based on blockchain technology and the system architecture design and system networking scheme are analyzed. We have shown some preliminary work, including data structure design, business interface and Proof-of-Concept Demonstration. In the future, by using emerging technologies, such as edge computing, virtual private network and other embedded blockchain platforms, the security guarantees of data security and business supervision in various application scenarios can be enhanced.

## References

1. Lanlan, G., Yijing, L., Ran, L., Jian, L.: Design and application prospects of task-oriented cross-domain collaboration system. In: 2nd Information Technology and Internet Security, February **40**(2), pp. 19–23 (2021)
2. Qiu, M., Zhang, L., Ming, Z., Chen, Z., Qin, X., Yang, L.T.: Security-aware optimization for ubiquitous computing systems with SEAT graph approach. *J. Comput. Syst. Sci.* **79**(5), 518–529 (2013)
3. Qiu, H., Qiu, M., Lu, Z.: Selective encryption on ECG data in body sensor network based on supervised machine learning. *Inf. Fusion* **55**, 59–67 (2020). <https://doi.org/10.1016/j.inffus.2019.07.012>
4. Pathak, J., Peterson, K., Kanabar, M., & Schmeelk, S.: Electronic health records and blockchain interoperability requirements: a scoping review. *JAMIA open*. **5**(3), oaac068. (2022)
5. Qiu, M., Gai, K., Xiong, Z.: Privacy-preserving wireless communications using bipartite matching in social big data. *Fut. Gener. Comput. Syst.* **87**, 772–781 (2018). <https://doi.org/10.1016/j.future.2017.08.004>
6. Chen, C., Zhu, F., Liu, G., Zhou, J., Li, L., Wang, Y.: A unified framework for cross-domain and cross-system recommendations. *IEEE Trans. Knowl. Data Eng.* (2021)
7. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N.: Algorand: Scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th Symposium on Operating Systems Principles, pp. 51–68, October 2017
8. Androulaki, E., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference. Article No.30, 23–26, Porto, Portugal, April (2018)
9. Demichev, A., Kryukov, A.: Complete Decentralization of Distributed Data Storages Based on Blockchain Technology. arXiv preprint [arXiv:2112.10693](https://arxiv.org/abs/2112.10693) (2021)
10. Wei, F., Ru, H., Tao H., Jiang, L., Shuo, W., Shiqin, Z.: A review of blockchain technology research: principles, progress and applications. *J. Commun.* (2020)
11. Ramachandran, G.S., Radhakrishnan, R., Krishnamachari, B.: Towards a decentralized data marketplace for smart cities. In: 2018 IEEE International Smart Cities Conference (ISC2), pp. 1–8. IEEE, September 2018
12. Raikwar, M., Gligoroski, D., Velinov, G.: Trends in development of databases and blockchain. In: 2020 Seventh International Conference on Software Defined Systems (SDS), pp. 177–182. IEEE, April 2020
13. Nagayama, R., Banno, R., Shudo, K.: Trail: a blockchain architecture for light nodes. In: 2020 IEEE Symposium on Computers and Communications (ISCC), pp. 1–7. IEEE, July 2020