ISSN 2616-7433 Vol. 6, Issue 10: 59-68, DOI: 10.25236/FSST.2024.061009

Research and Design of Campus Shared Trading Platform Based on Blockchain

Bo Gong[1,a], Guohua Xiong[1,b,*]

[1]*Guangdong Construction Polytechnic, Guangzhou, China*
[a] *gongbo@gdcvi.edu.cn,* [b] *xiongguohua@gdcvi.edu.cn*
[*]*Corresponding author*

Abstract: *With the development of the sharing economy, college students have an urgent need for how to realize the socialization of idle goods and improve the efficiency of resource utilization. However, there are relatively few researches on campus sharing trading platforms. The Traditional centralized information systems have not yet proposed effective solutions due to issues such as information asymmetry, lack of coordination among internal users, and difficulty in establishing effective trust mechanisms. Based on the research and analysis of blockchain technology, this paper constructs a decentralized shared transaction platform using Ethereum as the basic framework. The transaction process and order information are written into the blockchain, which is jointly maintained by all network nodes. This achieves decentralization of transactions and immutability of information, thereby solving the trust transaction problem between users in traditional information systems Data is easily tampered with, leading to issues such as inability to trace transactions.*

Keywords: *Blockchain, Ethereum, Campus, Sharing and Transaction*

1. Introduction

With the rapid development of Internet economy, e-commerce, as a new business application field, is affecting and changing all aspects of social and economic life. Universities that have always been at the forefront of information technology are also influenced by this business model, and online shopping has become a part of daily life for college students. However, when the purchased goods are not suitable for themselves and cannot be returned, or because of impulse consumption or replacement, the purchased goods become idle goods. The idle goods not only occupy a lot of living space, but also It takes time and energy to organize and maintain, so how to reuse idle resources has become a research hotspot in recent years.

At present, the most famous trading platform for idle goods is Alibaba's "idle fish" platform, but "idle fish" is for the whole social group, and does not provide specialized services for the special environment of campus; some colleges and universities have services to realize idle goods trading, such as WeChat Official Accounts, developing a traditional trading system for second-hand goods etc. , Such systems are often created by third parties and rely on centralized data storage services, which not only has the risk of overall data leakage and information asymmetry, but also occasionally results in arbitrary data tampering, and the credibility of data management is also questioned. Therefore, how to quickly develop a decentralized, secure and reliable Shared trading platform has become an urgent problem to be solved.

2. Key technologies

2.1 Blockchain

Bitcoin, as the first application of blockchain technology, is one of the most successful blockchain applications to date. In 2008, a scholar with the pseudonym Sakoshi Nakamoto published a groundbreaking paper on blockchain technology in the cryptography email group, titled "Bitcoin: A peer-to-peer electronic cash system." In this article, Sakoshi Nakamoto described it in detail as an electronic pair system. Essentially, blockchain is a distributed system or a shared and unchangeable ledger. This technical solution allows multiple nodes participating in the system to calculate and record all information and data in the system to one or more data blocks over a period of time through

ISSN 2616-7433 Vol. 6, Issue 10: 59-68, DOI: 10.25236/FSST.2024.061009

encryption algorithms, and generate a data "password" to verify the validity of their information and link to the next data block. All participating nodes in the system jointly determine whether the record is true[1-2].

In order to ensure the irreversibility, tamper resistance, and traceability of data, blockchain adopts some cryptography related technologies[3-6]. We mainly use three commonly used encryption techniques: hash algorithm, Merkle tree, and asymmetric encryption algorithm. Using hash algorithms to verify the integrity and authenticity of data, transaction information is stored in blocks in the form of a Merkle tree, which is a tree like data structure where all nodes that make up the Merkle tree are hash values. Each transaction has a hash value, and different hash values continue to hash upwards, ultimately forming a unique Merkle root. By digitally signing each transaction, it ensures that it cannot be tampered with, ensures consistency in information sharing decisions among multiple parties, and achieves the unforgeability, openness, and transparency of transaction identity and information. Asymmetric encryption algorithms can ensure the uniqueness and correctness of accounts.

2.2 Ethereum

In December 2013, Vitalik Buterin proposed the Ethereum blockchain platform[7-8], which is a smart contract and decentralized application platform based on blockchain technology. Like blockchain peer-to-peer networks, the Ethereum blockchain database is maintained and updated by many nodes connected to the network, ensuring Ethereum's fault tolerance and making the data stored in the blockchain tamper proof.

The Ethereum blockchain mainly includes two different types of accounts: External Owned Account (EOA) and Contract Account. All actions on the Ethereum blockchain are initiated by transactions triggered by EOA accounts. Every time the contract account receives a transaction, its code will be executed according to the input parameter instructions and sent as part of the transaction. The contract code is executed by EVM on all nodes participating in the network, and the account execution process is shown in Figure 1.

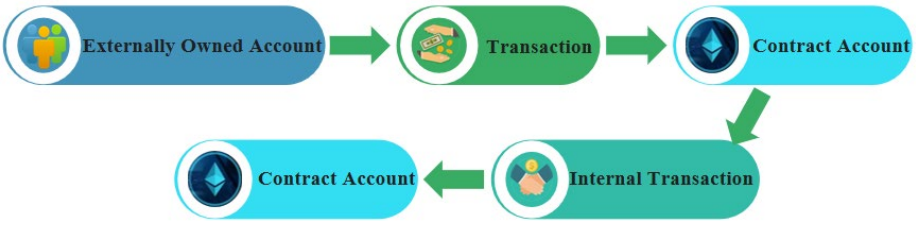

Figure 1: Account Execution Processth

The Ethereum blockchain platform provides the necessary conditions for developing DApps, including smart contracts written in Solidity programming language, EVM, graphical clients, and command-line terminals. Among them, Solidity is Turing's complete programming language; EVM is the executor of smart contract code, allowing developers to execute any program, stored on various computers like blockchain, and can achieve the jumping and execution of various functional codes. Therefore, through the Ethereum blockchain platform, developers can write the required smart contracts using Solidity language according to actual needs, rely on EVM to automatically execute, and build and develop various DApps.

If Bitcoin is used as blockchain technology 1.0, then Ethereum is a typical representative of blockchain technology 2.0. Due to its first use of blockchain systems to implement Turing, it makes it possible to develop and use applications in blockchain systems, and can ensure effective program execution. The specific reasons are as follows:

2.2.1. Support Smart contracts

The concept of smart contracts was proposed by Nick Szabo in 1994, which is a computer protocol that uses computer language to execute contract terms. Smart contracts use digital forms to enable contract participants to execute the content of the protocol in Ethereum without the need for human intervention. Once the protocol of a smart contract is released and deployed, its code cannot be modified and can only be modified by publishing a new contract, thus ensuring the authenticity of the running results. The languages used to develop Ethereum intelligent contracts are Serpent, Solidity, Mutan, and LLL etc. Among them, Solidity supports inheritance and polymorphism, and provides

ISSN 2616-7433 Vol. 6, Issue 10: 59-68, DOI: 10.25236/FSST.2024.061009

direct access to the global variables that have parameters related to the blockchain nodes. Blockchain, at the same time, Solidity has all the built-in features of the Serpent language, its' syntax rules are similar to JavaScript and easy to learn, so it is officially recommended for use.

2.2.2. Faster transaction speed

By adopting new consensus algorithms (such as POS, DPOS, etc.), smart contracts are divided into different logical regions, and the smart contracts in each region are executed sequentially. The throughput can be improved by parallel execution between different regions, thus meeting the transaction speed requirements of the vast majority of application scenarios.

2.2.3. Support information encryption

Advanced cryptographic technologies such as Zero-Knowledge Proof, ring signature, and homomorphic encryption are used to encrypt the information that needs to be sent and received, thereby protecting the privacy of both parties involved in the transaction.

2.2.4. Zero consumption

The emergence and application of new consensus algorithms have made it possible for nodes in the blockchain to reach consensus without consuming computing power, achieving zero resource consumption and thus enabling green deployment.

3. Requirements Analysis and Architecture Design of Shared Trading Platform

3.1. Platform requirement analysis

The purpose of this study is to design a blockchain based campus shared trading platform, which needs to address the following two issues: firstly, a lack of trust between the trading parties; Secondly, traditional transaction price negotiation mechanisms are time-consuming and cumbersome, resulting in higher communication time costs for both parties. The goal of this platform is to solve these two problems, and it is expected that the platform can provide a flexible total price system based on blockchain, rather than simply fixing the price of idle items.

This platform is aimed at both the supply and demand sides of idle items on campus, and analyzes the demand for a shared trading platform based on blockchain technology. The data on the blockchain has immutable and traceable properties, and its natural properties can ensure the security and trustworthiness of idle item transactions. In addition, the platform also needs to provide flexible trading methods for both supply and demand parties through informal bidding; Secondly, it is necessary to ensure the safe execution, convenience, and security of post auction transactions; Finally, analyze the demand for differentiated storage of data on idle item sharing and trading platforms[9-10].

To better design a shared trading platform, this article divides the platform's functions into two aspects: functional requirements and non functional requirements. The business application scenarios are shown in Figure 2.

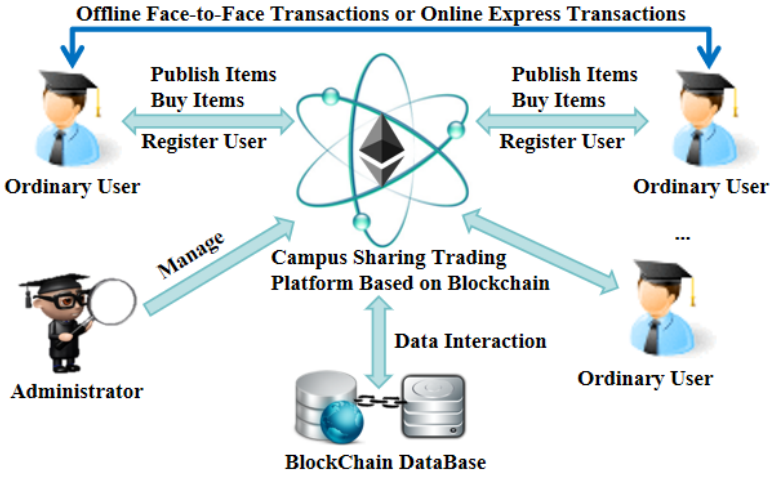

Figure 2: Business application scenarios

ISSN 2616-7433 Vol. 6, Issue 10: 59-68, DOI: 10.25236/FSST.2024.061009

The main functions of the trading platform include user registration and identity verification, transaction creation and execution, user comment and evaluation system, security protection mechanism, etc. Non functional requirements:

3.1.1. Robustness

Robustness refers to the ability of a computer system to continue running normally in the event of user input errors, user failure to follow normal logic, or system anomalies. This trading platform will verify the user's relevant input in the smart contract, and if there is an incorrect input, it will provide corresponding error information to guide the user to input the correct information. Meanwhile, due to the P2P nature of blockchain, there is no need to worry about a single point of failure leading to system crashes.

3.1.2. Usability

In order to adapt to users with different computer application capabilities, this trading platform has a simple and clear front-end module design, with good interactive pages, simple and easy to understand user operations, and clear functional logic between each module. The trading platform ensures that users can basically use the functions of various modules in the platform after reading the page prompts.

3.1.3. Scalability

Following the development and design principles of "high cohesion and low coupling", through extensive requirement analysis and design, each module is made as independent as possible and the coupling between modules is reduced as much as possible. Communication between modules is carried out through a unified API, while retaining some expandable interfaces for data or other information. When users have other new requirements, it is convenient to expand. This is not only beneficial for improving the development efficiency of the system, but also for reducing the development and maintenance costs of the entire platform.

3.1.4. Security

The platform will store a large amount of user information. In order to ensure the security of user data information, the main consideration is to use custody contracts to ensure the security of user fund transactions.

3.2. System architecture design

The overall system adopts a distributed architecture based on Ethereum, with the top layer being the application layer, which mainly displays the relevant interfaces with user operations, such as system management, transaction management, reputation management, tracking and tracing management modules; The smart contract layer includes virtual machines that execute scripting languages and distributed application development, to achieve a complete combination of smart contracts and front-end interaction interfaces to provide services to users; The consensus layer adopts a consensus mechanism based on proof of workload to achieve consensus in a decentralized decision-making system; The incentive layer enables each node to solve a mathematical problem through computing power competition, and rewards the first solved node, thereby promoting the entire network to achieve computing power competition; The network layer is a distributed network that primarily achieves communication between Ethereum nodes through P2P networks [13-14], message propagation mechanisms, and data validation; The data layer encapsulates the data structure of blocks and the content related to data encryption; The storage layer mainly uses the interstellar file system to store the data of the entire system, as shown in Figure 3.

ISSN 2616-7433 Vol. 6, Issue 10: 59-68, DOI: 10.25236/FSST.2024.061009

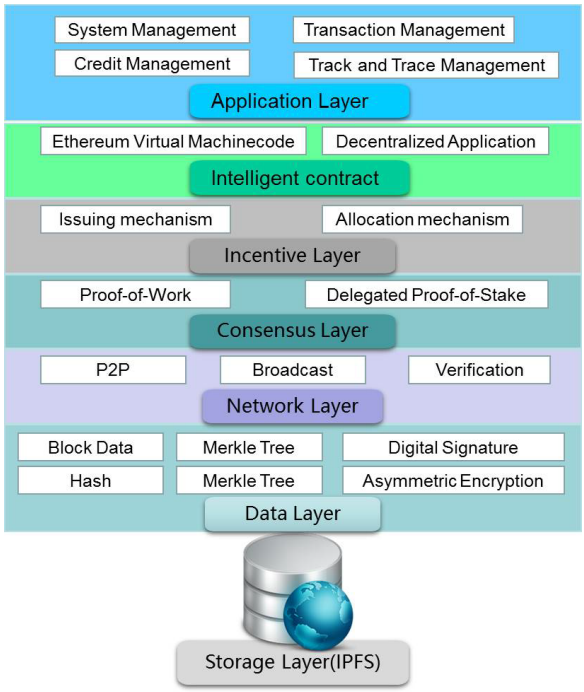

Figure 3: System architecture

3.3. Smart Contract Design

Design a smart contract model for this shared trading platform based on the transaction process of traditional e-commerce, as shown in Figure 4.

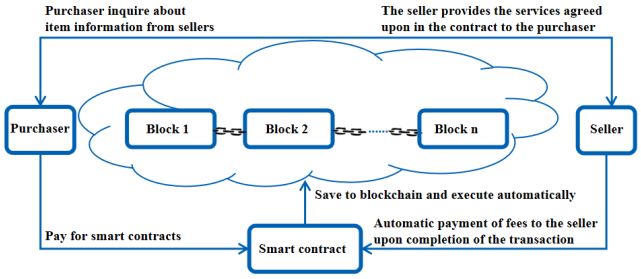

Figure 4: Smart contract model

After the user registers as an official user on the shared trading platform, the blockchain will distribute public and private keys to each user. The public key is the user's account address on the blockchain, and the private key is the user's encrypted key. Then, the two parties to the transaction agree on a smart contract based on their needs. The two parties to the transaction sign and confirm the contract information with their respective private keys to ensure the authenticity of the contract and avoid malicious tampering of the contract. The contract content is transmitted to the blockchain through P2P networks and verified by authentication nodes. After consensus is completed, the legitimate contract is stored in the data block and automatically executed. This trading model achieves decentralization of the system, with open, transparent, and tamper proof trading and contract information, as well as automatic storage and execution of contracts[11-13].

When designing the smart contract template, we thoroughly consider the characteristics and requirements of campus transactions to ensure that the contract can flexibly adapt to different types of transactions while safeguarding the rights and interests of both parties. The following is a detailed elaboration on the design of the smart contract template:

1) Basic Information Module

Party Information: Includes unique identifiers (such as student IDs, blockchain addresses) for both the buyer and seller, along with necessary contact details.

ISSN 2616-7433 Vol. 6, Issue 10: 59-68, DOI: 10.25236/FSST.2024.061009

Item Details: Describes the name, brand, model, condition (new/used), and image links of the transaction item, ensuring both parties have a clear understanding of the item.

Transaction Terms: Specifies key information like transaction price, payment method, delivery time, location, whether shipping costs are included, etc.

2) Deposit & Payment Module

Deposit Setting: Establishes a reasonable deposit amount based on the item's value, which the buyer must pay into an account controlled by the smart contract before the transaction begins.

Payment Process: Enables the buyer to pay the transaction amount through the blockchain network into a temporary account controlled by the smart contract. Upon completion of the transaction, the funds are released to the seller according to predefined conditions.

3) Transaction Execution & Confirmation Module

Transaction Execution: Sets conditions that trigger the execution of the transaction, such as the buyer confirming receipt of the item or the seller confirming receipt of payment.

Automatic Execution: When these conditions are met, the smart contract automatically performs subsequent operations, like releasing the deposit to the seller and transferring the transaction amount to the seller.

4) Breach & Dispute Resolution Module

Breach Clauses: Clearly defines breach behaviors and corresponding penalties, such as delayed delivery or damaged items.

Dispute Resolution: Provides a mechanism for resolving disputes, such as introducing arbitration nodes or a smart arbitration algorithm, which adjudicates based on transaction records and statements from both parties.

5) Extension & Customization Module

Transaction Type Extension: Supports the extension of multiple transaction types (e.g., book borrowing, equipment leasing, second-hand goods trading).

Term Customization: Allows both parties to customize contract terms within a certain scope to meet specific needs.

This design ensures that the smart contract template is adaptable, secure, and efficient in facilitating campus transactions, promoting trust and facilitating dispute resolution when necessary.

4. Ant Colony Optimization-base Smart Contract Algorithm

4.1. Design of Improved Ant Colony Optimization

The Ant Colony Smart Contract Algorithm is an innovative application that combines Ant Colony Optimization (ACO) with smart contract technology. The Ant Colony Algorithm solves optimization problems by simulating the pheromone communication mechanism observed in ants during their foraging process. When foraging, ants release pheromones along the paths they traverse, and other ants choose their paths based on the concentration of these pheromones. The shorter the path, the faster the accumulation of pheromones and the higher their concentration, which attracts more ants to choose that path, ultimately leading to the discovery of the optimal path. By continuously iterating and updating the pheromone levels, the Ant Colony Algorithm gradually steers the search process towards the global optimal solution.

The Ant Colony Optimization (ACO) algorithm randomly selects one buyer from all participating in the transactions as the starting point to initiate the transaction matching process. In the improved search strategy, the kth ant starts from buyer i as its initial point, and the probability of selecting seller j to match with buyer i is $p_k(i, j)$:

ISSN 2616-7433 Vol. 6, Issue 10: 59-68, DOI: 10.25236/FSST.2024.061009

$$p_k(i,j)=\begin{cases}\arg\max\{[\tau(i,j)]\bullet[\eta(i,j)]\}, j\in J_k(i)\\ 0 \qquad\qquad\qquad\qquad, otherwise\end{cases} \tag{1}$$

In Formula (1), $\tau(i,j)$ represents the pheromone level between buyer i and seller j. $j_k(i)$ is the set of all sellers who have not yet participated in a transaction matching when the kth ant arrives at buyer i. Initially, this set contains all sellers participating in the transactions. $\eta(i,j)$ is the heuristic factor that influences the matching between buyer i and seller j, and the heuristic function is expressed as follows:

$$\eta(i,j)=(z_i+z_j)/2 \tag{2}$$

In Formula (2), z_i and z_j represent the utility value functions for the buyer and seller, respectively. The improved pheromone update formula is as follows:

$$\begin{aligned}&\tau(i,j)=(1-\rho)\tau(i,j)+\rho(\Delta\tau(i,j)+\Delta\tau^*(i,j))\\ &\Delta\tau(i,j)=\begin{cases}C, if(i,j)\in NS\\ 0, otherwise\end{cases}\\ &\Delta\tau^*=\frac{|\max(f(S_{(i,j)}))-\Delta\tau_{ij}|}{\Delta\tau_{ij}}\end{aligned} \tag{3}$$

In Formula (3), $\tau(i,j)$ represents the pheromone on the effective match (i,j); ρ is the evaporation coefficient of the pheromone, with $\rho\in(0,1)$; $\Delta\tau(i,j)$ is the incremental pheromone left by ant k when exploring the effective match at (i,j); C is the number of effective matches in the current matching, where $C\in[0,n]$, and n is the total number of matches in the current scenario. The term $\Delta\tau^*(i,j)$ is the introduced pheromone adjustment increment. $\max\left(f(S_{i,j})\right)$ refers to the maximum value of the matching evaluation function. When the pheromone $\tau(i,j)$ of a certain match becomes excessively large, $\Delta\tau^*(i,j)$ performs a fine adjustment to the pheromone of the next match at the current position, thereby avoiding falling into a local optimum[14].

4.2. Simulation Experiments and Experimental Analysis

1) Experimental Environment

Hardware Environment:

CPU: i7-8565U CPU @ 1.80GHz

Memory: 16.00GB

System Environment:

Operating System: Ubuntu 16.04

Development Languages:

Java, Solidity

ISSN 2616-7433 Vol. 6, Issue 10: 59-68, DOI: 10.25236/FSST.2024.061009

2) Dataset

This system utilizes the order information from a campus-based shared trading platform to construct a dataset of 1,200 order services for experimentation. The commodity transaction information in Table 1 and Table 2 is as follows: In these tables, Ca represents the account address on the blockchain, Cp denotes the price of the commodity, Cd signifies the time required for the service, and Cq indicates the type of commodity. A portion of the data is presented in the tables below:

Table 1: Buyer transaction order information table

	Ca (Wallet Address)	Cp Price/eth	Ct Time /month	Cq Categories /items
X1	0x6f6421e630fa62b1798151d7296446767522b301	0.2315-0.4261	1-2	1-2
X2	0x1a07540eb2b11902e3a3a45ced3f8b7ea92d3242	0.6405-0.7243	2-3	2-3
X3	0xd2ecdab6a248b2dbc25ebde7adcdaccc507558dd	0.7432-1.1426	2-5	2-4
X4	0x65c26319c9b34418aeb6ee13a8b8ea40eed82ec4	1.1627-1.3554	3-5	6-8
X5	0xdadcc96e78b9bec5891dba90aab5aa7510ee200d	1.4473-1.7161	2-6	4-6
X6	0x82a978b3f5962a5b0957d9ee9eef472ee55b42f3	0.7618-0.9269	3-6	2-3
X7	x4b41d3e1bb9daba173ab13062b3abed234c20d7dc	1.4263-2.2852	1-7	4-7
X8	0xa21cc3be08c8e10d7cc37611ac930ecc026aebd0	1.9345-2.0864	2-8	5-8
X9	0x6dcdc2e7e6c4dd4dc65cce02b241acda5c0b6d82	2.3257-2.6477	3-8	3-4
X10	0x0aa6aa50a1e5b9b8dd8de1b63341b0cac9c065ae	2.0954-2.2257	2-9	6-9
…	…	…	…	…

Table 2: Seller Transaction Order Information Table

	Ca (Wallet Address)	Cp Price /eth	Ct Time /month	Cq Categories /items
Y1	0x1e5b9b8dd8d0aa6aa50ae1c065aeb63341b0cac9	0.4221-0.7221	1-2	3
Y2	0xe71eae7b3703aca3dd338a436d480ea4ebe93c3a	0.8402-1.4751	2-3	4
Y3	0x8bb7dd07ca8ec083e9de3ccb9d9a3e0c4a2b838f	1.1658-1.3423	2-5	5
Y4	0x20c0e7225ad4450ee15b1d0d0bc4e502b1bd9bbe	2.2236-2.9628	3-5	6
Y5	0xe3505abb67d4e98c9eb7e52c6d2cce1a0c40b3d7	1.7892-2.2318	2-4	4
Y6	0x865bd3e8bbd5bd51ae4dc924e8da7ad5b83b8a3b	1.4673-1.8667	3-6	3
Y7	0x2ed21c84aa1c8bd569f5bea0ad73d6c76a55be88	2.2522-2.4429	2-7	5
Y8	0xc4b1bd9a0beb6d50b411804de70c7b5a7ddd3e8e	1.2673-1.7667	2-8	3
Y9	0xb6d82dc2e7e6c4dd4dc656dce02b24cc1acda5c0	2.3411-2.6431	2-3	2
Y10	0x5cf0c26cdcd726d7d577a597b2742b498cb39de6	3.0942-3.3245	3-6	7
…	…	…	…	…

3) Simulation Results and Analysis

Based on Tables 1 and 2, a multi-objective matching for platform transactions is conducted, and the transaction matching results, pairings, and overall matching utility values are presented in Table 3. The data in the table indicates that both parties are relatively satisfied with the matching results. Subsequently, smart contracts are generated based on these matching results.

Table 3: Matching Results

Matching Results	Satisfaction
X1-Y1	0.453
X1-Y2	0.356
X1-Y3	0.343
X1-Y4	0.283
X1-Y5	0.145
X1-Y6	0.164
X1-Y7	0.131
X1-Y8	0.258
X1-Y9	0.412
X1-Y10	0.372
…	…

After the improved algorithm performs the matching and obtains the matching data results, the contract is created based on the demand conditions of both the buyer and the seller, and a smart

ISSN 2616-7433 Vol. 6, Issue 10: 59-68, DOI: 10.25236/FSST.2024.061009

contract is generated through a function. Table 4 shows the signing status of the smart contracts as follows:

Table 4: Smart Contract Signing Status

Date	Buyer	Seller	Successful Matches	Failed Matches
2023.9.12	20	20	20	0
2023.9.13	91	110	86	8
2023.9.14	430	424	424	16
2023.9.15	662	672	655	17
2023.9.16	757	711	746	21
2023.9.17	700	531	688	22
2023.9.18	842	798	832	30
total	3502	3266	3451	114

As can be seen from Table 4, the experimental results show that a total of 3,565 contracts were signed, among which 3,451 were successfully signed as smart contracts, 114 were invalid, and the effective contracts accounted for approximately 96.80%. Based on the above experimental results, it can be concluded that the proposed ant colony smart contract algorithm and transaction mechanism for the campus sharing trading platform are feasible in solving problems such as the intellectualization of shared trading platforms, the centralization of transaction data, and the lack of multi-transaction smart contracts.

5. Conclusion

Based on the analysis of blockchain technology and Ethereum architecture, this article proposes a decentralized shared transaction platform architecture design. This design utilizes the characteristics of blockchain technology to solve the problems of information leakage, information asymmetry, and data tampering that traditional shared transaction systems face, ensuring the reliability of the system. At present, the design is still in the development and improvement stage, and the next work will continue to improve the smart contract function, and further explore how to achieve the integration and development of blockchain and various database management systems, thereby expanding the application fields of blockchain technology.

Acknowledgment

This paper is Supported by Educational Planning Project of Guangdong Province in 2021 (Grant No. 2021GXJK534), Supported by 2023-2024 Academic Year Research Project on Graduate Employment and Entrepreneurship in Higher Education Institutions (Grant No.GJXY2024N103), Supported by universities characteristic innovation project of Guangdong Province in 2022 (Grant No. 2022KTSCX248).

References

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[EB/OL], available: https://bitcoin.org/ bitcoin. pdf, 2019.
[2] Lin Xiaochi, Hu Yeqianwen. Research Review on Blockchain Technology[J]. Financial Market Research, 2016(02):97-109.
[3] NButerin V. A next-generation smart contract and decentralized application platform [EB/OL], GitHub: ethereum/wiki, 2014.https:// github.com/ethereum/wiki/wiki/%5BEnglish%5D-White-Pager, 2014.
[4] Antonopoulos A M. Mastering Bitcoin: Unlocking Digital Crypto-currencies.USA:O'Reilly Media Inc., 2014.
[5] Becker J, Breuker D, Heide T, et al. Can we afford integrity by proof-of-work? Sceinarios inspired by the Bitcoin currency [M].The Economics of Information Security and Privacy. Springer Berlin Heidel berg, 2013:135-156.
[6] Wrigllt A, De Filippi P.Decentralized Blockchain Technology and the Rise of Lex Cryptographia [J]·Social Science Electronic Publishing, 2015.
[7] Ametrano FM.Bitcoin, Blockchain, and Distributed Ledger Technology[J]. Social Science

ISSN 2616-7433 Vol. 6, Issue 10: 59-68, DOI: 10.25236/FSST.2024.061009

Electronic Publishing, 2016.
[8] Dennis R, Owen G. Rep on the block: A next generation reputation system based on the blockchain [C].Intemational Conference for Internet Technology and Secured Transactions. IEEE, 2015.
[9] Grishchenko, I, Maffei, M, & Schneidewind, C. A. Semantic Framework for the Security Analysis of Ethereum Smart Contracts [J]. International Conference on Principles of Security and Trust, 2018(21): 243-269.
[10] Mavridou, A, & Laszk, A. Designing Secure Ethereum Smart Contracts: a Finite State Machine Based Approach [J]. International Conference on Financial Cryptography and Data Security, 2018(11): 1-17.
[11] Gao Hang, Yu Xuemai, Wang Maolu. Blockchain and the New Economy: The Digital Currency 2.0 Era [M].Beijing: Electronic Industry Press, 2016.
[12] Shentu Qingchun. Blockchain Development Guide [M]. Beijing: Machinery Industry Press, 2017.
[13] Stoica, Ion, Morris, Rober, Liben-Nowell, David.et al.: Chord:a scalable peer-to-peer lookup protocol for internet applications[J].IEEE/ACM Transactions on Networking, 2003, 11(4):149-160.
[14] Herbert J, Litchfield A.A Novel Method for Decentralised Peer-to-Peer Software License Validation Using Cryptocurrency Blockchain Technology[C].Australasian Computer Science Conference. 2015.