Hindawi
Journal of Mathematics
Volume 2022, Article ID 2839566, 13 pages
https://doi.org/10.1155/2022/2839566

Research Article

Evolution Process and Supply Chain Adaptation of Smart Contracts in Blockchain

Yue Wu, Junxiang Li, Jiru Zhou, Shichang Luo, and Liwei Song

Business School, University of Shanghai for Science and Technology, Shanghai 200093, China

Correspondence should be addressed to Junxiang Li; lijx@usst.edu.cn

Received 29 August 2021; Revised 26 October 2021; Accepted 22 November 2021; Published 4 January 2022

Academic Editor: Jia-Bao Liu

Because of its unique decentralization, encryption, reliability, and tamper-proof, the block chain system makes smart contracts break through the shackles of the lack of trusted environment, and its application field keeps expanding. We read the source code and official documents of Bitcoin, Ethereum, and Hyperledger to explore the operation principle and implementation mode of smart contract. By analyzing the evolution process of smart contracts in blockchain and the sequence of its function expansion, according to the multirole business process of supply chain, we design a semipublic smart contract chain model based on Ethereum and Hyperledger in order to provide useful inspiration and help for the future research of smart contracts in blockchain applied in supply chain.

1. Introduction

Supply chain is a network of companies and departments which acquires and processes materials into middleware or finished products and then sends the finished products to customers. The entire process of supply chain includes multiple participating companies and multiple roles such as suppliers, manufacturers, and channel vendors. In this network, the biggest problem among the participants is the issue of trust, because only by building trust can the collaboration complete the overall product manufacturing and sales process. The primary problem faced by supply chain management is how to reduce the cost of trust so that each participant can effectively coordinate their internal and external resources to meet market demand.

The coordination mechanism of supply chain management information system includes process and consensus coordination management mechanism [1]. A safe and trusted environment can effectively process the supply chain information flow, which is essential for establishing an efficient and reliable supply chain program [2]. Innovation and sustainability have gradually become the core competitive dimensions of the supply chain [3]. The innovative features of blockchain technology have high disruptive potential for supply chain business models [4]. Blockchain technology provides a safe environment for the supply chain and realizes supply chain information sharing, traceability, and transparency [5]. Blockchain can be integrated into the supply chain architecture to create a reliable, transparent, trustworthy, and secure system [6]. Blockchain technology can effectively reduce the fragmentation, inefficiency, and incoordination of supply chain information and improve the efficiency of operations management [7]. Blockchain technology has a wealth of application scenarios in the supply chain, such as logistics monitoring systems [8] and container shipment management [9]. As the core function of the blockchain, smart contract inherits the characteristics of the blockchain, such as being tamper-proof, trustworthy, and decentralized [10]. Smart contracts have the characteristics of information disclosure, which can ensure that the content of the contract is transparent and trustworthy [11]. Users use smart contracts to achieve transparent and secure transactions between users [12]. Being embedded in blockchains, smart contracts enable the contractual terms of an agreement to be enforced automatically without the intervention of a trusted third party [13]. The collaborative business processes of parties that do not trust each other are compiled into smart contracts that can be deployed on the blockchain

platform [14]. Smart contracts allow verifiable operations to be executed in blockchains, bringing new possibilities for trust establishment in trustless scenarios [15]. Past studies have proved that smart contracts can be applied in many areas of the supply chain such as Logistics Service [16] and e-government [17].

Permissioned Blockchain means that every node participating in the blockchain system is licensed, and unauthorized nodes are not allowed to access the system. Permissioned Blockchain solutions adopt more efficient consensus algorithms and smart contracts. A role-based access control model provides controlled access of resources to members [18]. A Permissioned Blockchain implementation is used to ensure that only known agents are permitted to access the system [19]. The authors in [20] propose a framework for demand response registry and implemented it as a proof of concept on Hyperledger Fabric, using real assets in a laboratory environment, in order to study its feasibility and performance. Permissioned Blockchain has the following features: (1) Only certified members are allowed to participate. Avoid attacks from external users. (2) Consensus mechanism only needs to focus on efficiency without considering security. The system has faster transaction speed and higher transaction processing capacity. (3) No miners need to maintain the system, which can effectively reduce transaction costs. (4) The transaction process does not need to include user identity information that has been authenticated by the administrator, which can better protect user privacy information. Blockchain is a highly disruptive technology that is already remodeling the organizations and their supply chain business models [21, 22]. Blockchain can help firms achieve the following supply chain management objectives: (1) reducing costs, (2) assuring quality of products, (3) increasing speed, (4) increasing dependability, (5) reducing risks, (6) facilitating sustainable practices, and (7) enhancing flexibility [23].

However, the current research has the following problems: The application of blockchain technology ignores the restrictions of public chain and Permissioned Blockchain. The research using public chain technology such as Ethereum ignores the problem of how to manage roles [24, 25]. Research using Permissioned Blockchain technology such as Hyperledger ignores the problem of duplication of supply chain construction costs [26]. Therefore, we go deep into the evolution of smart contracts and propose a new semipublic chain smart contract model in order to achieve the following goals:

(1) Explore the blockchain and smart contracts from the bottom of the technology and discuss the evolution process of smart contracts

(2) Combine the advantages of the two types of blockchain technology to realize the user and role management functions in the public chain mode

(3) Lower the threshold and cost of user participation and attract more users to participate

2. Evolution of Smart Contract

Smart contract was first proposed by the cryptographer Szabo [27] in 1994; "smart contract is a set of commitments defined in digital form, including agreements on which contract participants can implement these commitments." Before the advent of Bitcoin [28, 29], smart contract was limited to a safe and reliable operating environment. Both the smart contract itself and its dependent data have the risk of being tampered by attacks, so it stayed in the theoretical research stage. Due to its features of decentralization, information disclosure, and tamper-proof, blockchain not only provides a trusted execution environment for smart contract but also reduces the difficulty of writing smart contract, while blockchain provides contract specification and standard. Therefore, smart contracts are once again developed in the blockchain. Smart contracts in blockchain are protocols defined in digital form, which are codes that execute commitments driven by events. The terms and contents of the contract are stored as code in the computer system. Blockchain system judges the conditions of the contract and the commitment to execute the contract according to the code. The whole process is completely automatic and cannot be interfered, and there is no need for participants to trust each other. The logical structure of smart contract is shown in Figure 1.

Instead of centralized server and ledger database, the blockchain system keeps the ledger in all nodes of the blockchain network and relies on the consensus mechanism to confirm the correctness of the ledger data and update the ledger. The tamper-proof of ledger data in blockchain avoids the risk of cheating or being attacked in a centralized system. At the same time, with the help of asymmetric encryption technology, different nodes of blockchain can verify each other's identity and cannot fake each other to ensure the security of transactions. With the release of Bitcoin by Satoshi Nakamoto, smart contract and blockchain started the evolutionary process.

2.1. Difference between UTXO and Account Mode. The address of Bitcoin is not an account in the traditional sense. There is no concept of account balance, only UTXO. Bitcoin transaction is composed of inputs and outputs. UTXO is a data structure in the transaction process, which is the most basic unit of transaction. The output of UTXO is indivisible once created and can only be consumed as input to a new transaction. The new transaction generates new UTXO, so that the value of the currency is transferred over and over again. So the balance seen in the Bitcoin wallet is actually the UTXO accumulated value in the Bitcoin address.

Ethereum chooses the account model, and the differences between the two models are shown in Figure 2. The account model contains the balance, and each transaction is recorded on the corresponding account. It is difficult to implement a Turing-complete smart contract based on the UTXO model, while the account model can be easily completed. For example, setting up a betting contract, multiple users need to enter funds for several times, and smart contract regularly takes out part funds from the fund pool to draw prizes. In order to manage accounts more easily, Ethereum uses the Merkle Patricia Trie (MPT) structure of the World State, and every transaction will change the MPT. Each account has a status associated with it

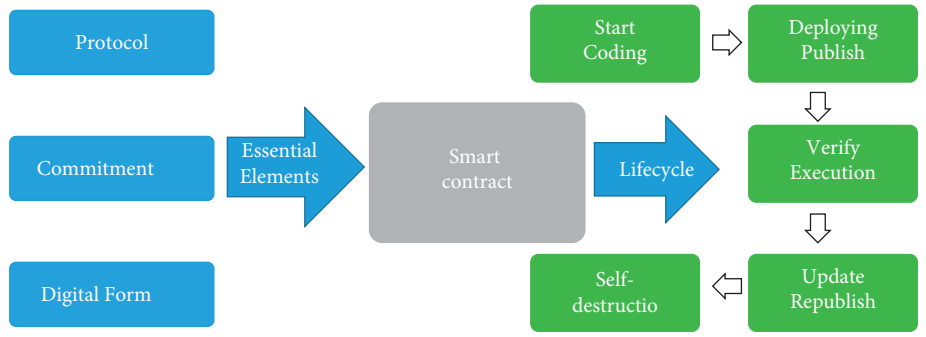

Figure 1: Three elements and life cycle of smart contract.

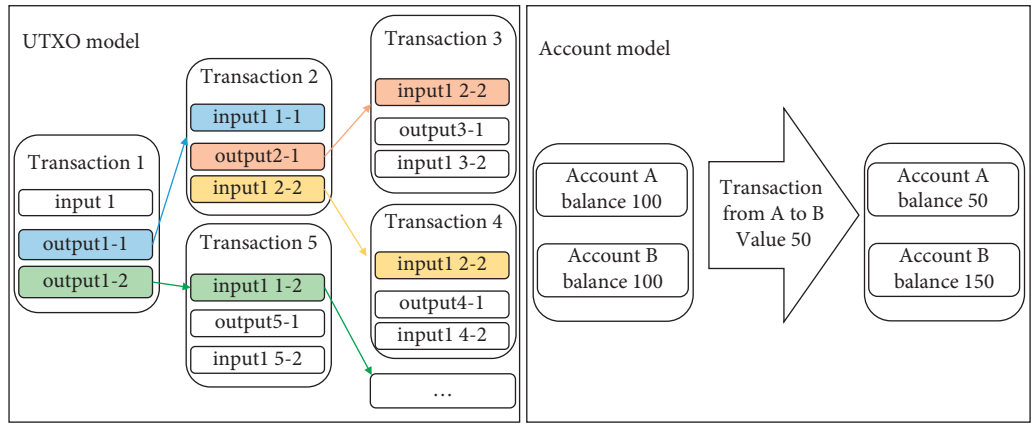

Figure 2: UTXO and account model.

and a 20-byte address. An address in Ethereum is a 160-bit identifier used to identify an account. The account model is more programmable, easier for developers to understand, and has a wider application scenario. There are two different types of accounts in Ethereum.

(1) User account: it is a regular account of an Ethereum user controlled by the user's private key without any code associated with it.

(2) Contract account: it is an account to which the smart contract belongs controlled by its own contract code and has a code associated with it. The contract is stored in the account in code form, and the account has its own state.

2.2. Upgrade of Contract. Smart contract differs from traditional software programs in that smart contract cannot be tampered once it is published on a blockchain. Even if a bug is found in the smart contract which needs to be fixed or the business logic is changed, it cannot be directly modified and rereleased on the original contract. Therefore, it is necessary to consider a reasonable upgrade mechanism based on a business scenario at the beginning of the design. The advantage of the tamper-proof mechanism is that the code of smart contract is open and the business rules are unchanged, which can attract more participants to use the smart contract. However, as the content of smart contract becomes more and more complex, even a seemingly insignificant decision may have serious or even dangerous consequences, and even if the development team finds a bug, it cannot be fixed in time. In the field of smart contract, Ethereum has carried out revolutionary innovations, breaking the shackles of the Bitcoin script and establishing a Turing-complete smart contract platform. Ethereum smart contract is limited by the public chain model and cannot solve the problems that transaction efficiency is limited by the block generation speed and the contract content cannot be kept secret. Hyperledger Fabric adopts the form of consortium blockchain, breaks through the application scenario limitation of public blockchain, and makes many functional evolutions. Hyperledger includes the following functions: (1) CA authentication mechanism, which can realize the identification of access nodes and achieve admission management; (2) permission mechanism, which divides nodes into different roles and different permissions to implement node management; (3) multichain mechanism, which can realize confidential communication of messages, data off-chain storage, off-chain transmission, and flexible deployment of smart contract.

2.3. Node Role and Multichain Channel. In the public blockchain model of Bitcoin and Ethereum, the blockchain system has only one main chain, and all nodes maintain a set of data ledgers, which cannot store different types of data in a distributed manner. With the increase of the running time of the blockchain system, the block data keeps expanding, and

the huge storage overhead increases the difficulty of node synchronization, storage, and processing. Due to the decentralized consensus mechanism of the public blockchain, all mining nodes are required to participate in maintaining the ledger, so the content of smart contract will be disclosed to all nodes, and the privacy of sensitive data cannot be guaranteed. In order to solve the above problems, Hyperledger has carried out two technological innovations. One is the role rights management of the nodes and the other is the use of channel technology to support multiple chains. Hyperledger has changed the authority management of all nodes in the public blockchain mode. According to the function authority, the nodes are divided into different roles, including Endorser node, Order node, Committer node, and Certificate Authority (CA) node. The system architecture of Hyperledger is shown in Figure 3. Nodes can join different channels, and smart contract can run on different nodes, which can better improve the system's parallel execution efficiency and throughput.

Channels are isolated from each other, and transactions sent in them are only visible to nodes that belong to the channel. Therefore, the channel can be regarded as the private communication subnet of some nodes in Hyperledger's network. In other words, establishing a channel is equivalent to establishing a subchain. In the Hyperledger network, there may be multiple channels isolated from each other. Each channel contains separate ledger data and a list of members allowed to participate. A node can subscribe to multiple channels and can only access transactions on the channels to which it subscribes, so a node can participate in multiple chains by accessing multiple channels to obtain data of different ledgers. The channels solve the following problems:

(1) Contract confidentiality: channel limits the scope of information dissemination. Smart contract in a channel can only be seen by members of the channel and can only be executed in the channel.

(2) Data expansion: channel solves the problem of data storage in the single main chain. Different data is stored in different channels to avoid the expansion of ledger such as Bitcoin and Ethereum, and ordinary nodes cannot be stored.

(3) Cross-chain interaction: channel solves the problem of untrustworthy external data. Users can formulate more complex smart contract by reading data on different chains, such as using DApp to monitor *B*-chain data, and execute smart contract through *A*-chain when conditions are met.

2.4. System and User Chaincodes. Chaincode is a piece of program code that is deployed on Hyperledger network nodes and can be called to interact with the distributed ledger. It can be considered as a smart contract on Hyperledger. Chaincode is a program that supports multiple programming languages. It implements some of the interfaces predefined by Hyperledger and runs in Docker. Hyperledger breaks through the limitation that Ethereum smart contract can only be written by users and cannot support the system contract management in the administrator mode and divides the chaincode into system chaincode and user chaincode. User chaincode is a code written and published by ordinary users and contains the processing logic of contracts, which, like Ethereum, is event-driven and has its own state. The system chaincode is embedded in the Hyperledger system to implement smart contract management functions. The system chaincode includes lifecycle system chaincode (LSCC), querier system chaincode (QSCC), configuration system chaincode (CSCC), endorser system chaincode (ESCC), and validator system chaincode (VSCC). Compared to Ethereum, Hyperledger's chaincode is separated from the underlying ledger. When upgrading the chaincode, it is not necessary to migrate the ledger data to the new chaincode, which truly realizes the separation of logic and data.

However, there are also some disadvantages of smart contract. Smart contract is expensive to learn and difficult to maintain. Different blockchain products use different programming languages and different coding rules. Writing smart contract code requires the user to have the programming foundation and master the contract code language. Smart contract has the tamper-proof feature that requires users to fully test the code. If a bug or unforeseen situation occurs after the smart contract is released, there will be problems that cannot be fixed in time. Every time Bitcoin upgrades its code, it must obtain the unanimous approval of all nodes in the system. If all nodes cannot reach an agreement, the system will fork and form two independent sets of ledgers, for example, Bitcoin fork into Bitcoin (BTC) and Bitcoin Cash (BCH) on August 1, 2017.

2.5. Functions and Advantages of Smart Contracts. As a core component of the blockchain, smart contract has been enriched with the development of blockchain technology. Initially, Bitcoin hoped to solve the security problems in the transaction process through transaction scripts so as to realize a trusted transaction system. With the increasing influence of Bitcoin, the continuously evolving Bitcoin transaction script has realized data storage, multiparty transactions, and even the expansion of subtokens through protocolized OP_RETURN content. In the Bitcoin era, people mainly focus on the digital currency. With the continuous development of technology, people start to pay attention to the blockchain technology itself. Turing-complete blockchain system represented by Ethereum opened the era of smart contract and triggered a new revolution. On the basis of Bitcoin, Ethereum has been continuously developing and realizing functions such as EVM, Gas, ERC20, and various DApps based on Ethereum. The account model and Turing-complete blockchain system solve the problems of low flexibility and poor scalability of the Bitcoin script. A large number of tokens and DApps are also released based on the Ethereum platform such as CryptoKitties which conform to the ERC-20 standard and visualize token graphics. Hyperledger expands the smart contract function on the basis of the consortium blockchain and implements a

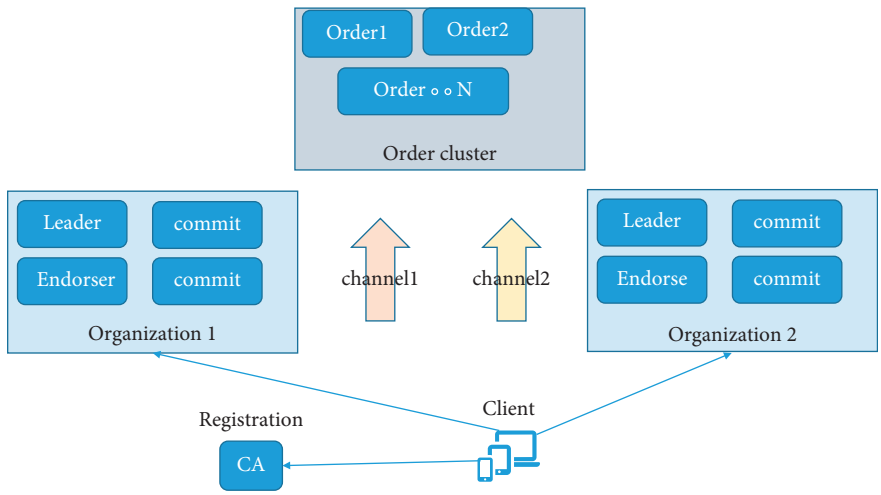

Figure 3: The system architecture of Hyperledger.

multichain model through node role management, channel, and other technologies, breaking through the problems of single main chain data and efficiency of public blockchain. Hyperledger implements system-level management such as online upgrade of chaincode through the system chaincode and solves the risks caused by incomplete smart contract codes of the public blockchain. The technology evolution process of smart contract is shown in Figure 4.

Supply chain often includes many stakeholders, such as producers, processors, wholesalers, retailers, and consumers. There are many complicated problems in the supply chain, such as dynamic coordination among multiple parties, limited predictability, data compatibility between parties, high cost of commodity tracking, and blind spots. Blockchain has the ability to replace complex, error-prone processes with simplified smart contract and is expected to change the supply chain processes between suppliers, retailers, and consumers. As technology develops and matures, it will be used to open new doors for cross-organizational collaboration and enable new business models in the supply chain. With the continuous improvement of smart contract technology, the application scenarios continue to expand, and the functions of applications also continue to be innovated. At the same time, new requirements also promote the continuous evolution of smart contract technology. The application of blockchain technology to supply chain system has the following advantages:

(1) Security and trustworthiness: blockchain technology has unique characteristics that help build trust, transparency, and accountability among multiple parties in the supply chain. Smart contract can serve as a shared data platform enabling participants' access management and participants to track the status of assets and share information in a secure manner.

(2) Data processing and integration: data is the core of business processes in the supply chain. Today, the organizations that can provide data include not only some core enterprises but also logistics companies, e-commerce platforms, and Enterprise Resource Planning (ERP) vendors. Smart contract can process data in accordance with business logic, promote the "four flows in one" of business flow, logistics, capital flow, and information flow in the supply chain, promote the effective transfer of information and resources of supply chain enterprises, and thus improve the efficiency of supply chain enterprises.

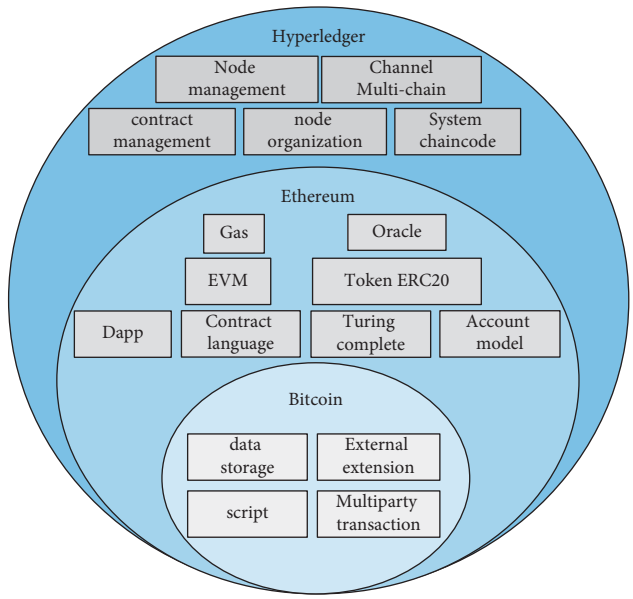

Figure 4: The evolution diagram of smart contract.

(3) Trust and collaboration: blockchain technology provides a decentralized and equal collaboration platform for all participants, which can greatly reduce the risk of credit collaboration and transaction costs among participants. Based on the information on the blockchain, participants can achieve real-time synchronization and verification of data. Smart contract can use tokens to solve problems that traditional supply chains cannot solve, such as establishing a credit model based on tokens. Smart contract can allow credit to penetrate the entire

chain, cover every participant, and thereby solve the problem of multiagent trust relationships. Smart contract can also use tokens to implement supply chain financial functions such as credit and financing.

(4) Reducing risk: smart contract can promote the formation of a fair and credible trading environment, allow multiple institutions to coexist in a scenario of mutual cooperation and mutual supervision, and avoid private transactions or collusion in the traditional supply chain financial model. In addition, smart contract can urge all parties to fulfill their obligations, ensure that transactions are automatically executed when conditions are met, and use token for liquidation of funds, which can effectively control performance risks and ensure capital security.

3. Design of Semipublic Chain Smart Contract

The supply chain business model has the following characteristics: First of all, there are multiple roles in a supply chain, and each role may be composed of multiple participating users. Secondly, users may participate in multiple supply chains at the same time and may play different roles in different supply chains. Finally, the supply chain is not led by a traditional central organization, and participants randomly join or exit in a distributed manner. How to realize the identity authentication, authority management, participation access, and cost management and lowering the participation threshold of the supply chain participants have become the current problems to be solved urgently.

3.1. Business Process Architecture. Supply chain is a network chain structure formed by upstream and downstream participants in the activities of providing products to end users during the production and circulation of products. A complete supply chain is composed of multiple roles, each role is composed of multiple participants, and each participant will participate in multiple supply chains at the same time. As shown in Figure 5, the seven roles of raw materials, factories, wholesalers, logistics, warehousing, retail, and consumers form a supply chain, and each supply chain is intertwined with other supply chains.

Public chain and Permissioned Blockchain, the two types of blockchain technologies, have their own advantages and disadvantages. Public chains such as Ethereum have the advantages of easy joining and unified rules, but there is no role permission management, and all participants have the same permissions, which is not suitable for enterprise-level solutions. Permissioned Blockchain, such as Hyperledger, is convenient for management and has clear roles and permissions. However, different supply chains need to establish different Permissioned Blockchains, and there is a problem of duplication of construction. Ethereum has programmable functions, which can realize the functional logic of specific application scenarios through the development of smart contracts, with low latency and scalability. Hyperledger also supports smart contract writing, but the smart contract of Hyperledger needs to run in an additional configured isolation environment Docker sandbox, which is cumbersome to deploy and has a high threshold for use. Therefore, from the perspective of business processes, we choose the Ethereum solution.

3.2. Comprehensive Cost Analysis. The use of blockchain technology by supply chain users will also incur new costs. Users not only need to purchase hardware equipment to connect to the blockchain network but also need to write smart contracts to use blockchain functions. User costs include hardware equipment costs, network equipment costs, development costs, maintenance costs, and blockchain access costs. The user cost is as follows:

$$C = C_{\text{hardware}} + C_{\text{network}} + C_{\text{development}} + C_{\text{maintenance}} + C_{\text{access}}, \tag{1}$$

where C is the summary cost of various costs of supply chain users using the blockchain system. Different types of costs are marked with subscripts; for example, C_{hardware} stands for hardware equipment costs.

This article assumes that there are $i, i \in [1, N]$ users and $j, j \in [1, M]$ supply chain systems in the market. When multiple users N access a single supply chain system j, the construction cost of each user is independent of the others, there is no duplicate construction cost, and the total market cost is the sum of user costs. According to formula (1), the total market cost of a single system with multiple users is as follows:

$$TC_j = \sum_{i=1}^{N} C. \tag{2}$$

When a single user i accesses multiple supply chain systems M, the cost of the first access system is the basic cost, and the system cost for additional access is the cost of repeated construction. The cost of repetitive construction is usually lower than the basic cost. Because hardware facilities can be reused, software costs are also reduced due to experience. The total market cost of a single user with multiple systems is as follows:

$$TC_i = \begin{cases} C, & j = 1, \\ \sum_{j=2}^{M} \left(\alpha_i C_{\text{hardware}}^{j-1} + \beta_i C_{\text{network}}^{j-1} + \delta_i C_{\text{development}}^{j-1} + \lambda_i C_{\text{maintenance}}^{j-1} + \omega_i C_{\text{access}}^{j-1} \right) + C, & j \geq 2, \end{cases} \tag{3}$$

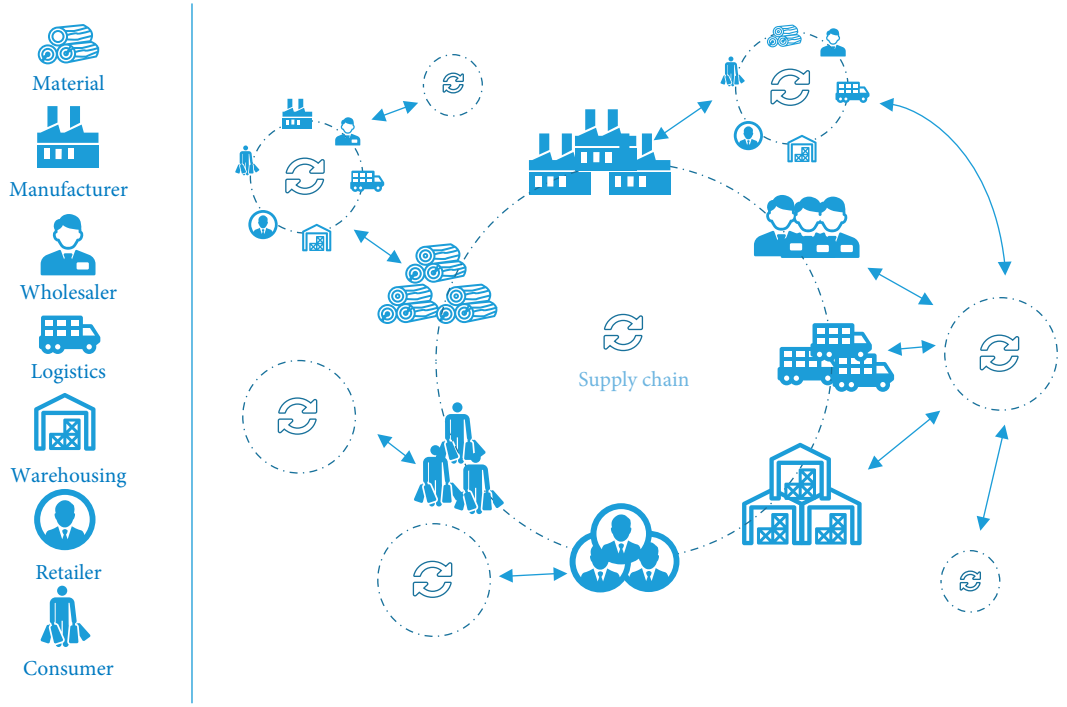

Figure 5: Supply chain process architecture diagram.

where α_i, β_i, δ_i, λ_i, and ω_i are user coefficients. The values of coefficients are between [0, 1] (zero represents no repetitive construction costs, and one represents repetitive construction costs equal to new construction costs).

When multiple users N access multiple supply chain systems M, according to formulae (2) and (3), the total cost of the market is as follows:

$$TC_{i,j} = \begin{cases} C, & i = 1, j = 1, \\ 2C, & i = 2, j = 1, \\ \alpha_i C_{\text{hardware}} + \beta_i C_{\text{network}} + \delta_i C_{\text{development}} + \lambda_i C_{\text{maintenance}} + \omega_i C_{\text{access}} + C, & i = 1, j = 2, \\ \sum_{i=2}^{N} \sum_{j=2}^{M} \left(\alpha_i C_{\text{hardware}}^{j-1} + \beta_i C_{\text{network}}^{j-1} + \delta_i C_{\text{development}}^{j-1} + \lambda_i C_{\text{maintenance}}^{j-1} + \omega_i C_{\text{access}}^{j-1} \right) + C, & i \geq 2, j \geq 2. \end{cases} \tag{4}$$

When a user participates in multiple supply chains, the supply chains of different blockchain technology models will generate different input costs. In the public chain Ethereum technical solution, multiple supply chains coexist in Ethereum, and users only need to access a set of blockchain systems and access different supply chains through different smart contracts. In the Permissioned Blockchain Hyperledger technical solution, different supply chains exist in different blockchain systems, and the systems of each supply chain are independent of each other, and users need to access multiple blockchain systems. The cost impact of different solutions is compared as follows:

(1) Hardware cost: the costs of both Ethereum and Hyperledger are basically the same, and both require the purchase of additional hardware equipment. When the number of supply chains is large enough, Hyperledger needs to connect to multiple systems, and there may be insufficient performance of hardware equipment, and more hardware resources need to be invested. Ethereum can also be used for hardware expansion such as parallel use of multiple devices.

(2) Network cost: the network needs to purchase network service provider services. The cost is similar to the hardware cost. The costs of the Ethereum and Hyperledger solutions are basically the same.

(3) Development cost: this part of the cost mainly includes the cost of learning the development platform and the cost of writing smart contract code. In the Ethereum model, the rules for writing smart contracts are relatively uniform, and users have lower learning costs when accessing multiple supply chains and lower secondary development costs. In Hyperledger mode, rules cannot be unified. When users

access multiple supply chains, the cost of learning is higher, and the cost of secondary development is higher.

(4) Maintenance cost: this part of the cost includes the smart contract inspection and upgrade cost, which is similar to the development cost. When multiple blockchain systems are connected, the cost of the Ethereum solution is less than the cost of the Hyperledger solution.

(5) Access cost: this part of the cost includes system access and business access costs. The system access cost depends on the hardware cost. The costs of Ethereum and Hyperledger are basically the same. Business cost access depends on development and maintenance costs, and Ethereum has lower repetitive construction costs than Hyperledger.

Assume that the number of users in the market N is 10, and the number of supply chain systems M is 10. In order to simplify the calculation, we assume that all the user coefficients are the same, and the basic value of each cost is 1. The correlation coefficient values of formula (4) are shown in Table 1.

The total cost of the Ethereum solution is less than the total cost of the Hyperledger solution, as shown in Figure 6. At the initial stage, the total market costs of the two schemes are almost the same. With the increase in the number of users and supply chain systems, the total market cost is also increasing. The Ethereum solution has a cost advantage compared to the Hyperledger solution. Therefore, we choose the Ethereum solution as the blockchain technology platform of the semipublic chain smart contract and use Solidity as the programming language.

4. Implementation of Semipublic Chain Smart Contract

Smart contract has different advantages and disadvantages under different blockchain models. Ethereum's public blockchain model has the advantages of decentralizing and allowing any node to participate but also has the disadvantages of data disclosure. Contract content cannot be kept secret, and data cannot be stored in different chains. Hyperledger's consortium blockchain mode has the advantages of node management, network access participation management, and contract management. However, it also has the disadvantages of system centralization and higher development complexity. In order to strengthen enterprises' application level of blockchain, reduce enterprise costs, and realize node and smart contract management in a decentralized supply chain environment, this paper builds a semipublic chain smart contract based on Ethereum and Hyperledger to achieve the functions of node role management, multichain management, and contract upgrade management in a decentralized public blockchain model. The logical architecture of semipublic chain smart contract is shown in Figure 7.

Ethereum account addresses are unique. In this paper, address permission verification is used to manage user role permissions, so as to build a semipublic chain model. This model makes it unnecessary for supply chain enterprises to rebuild the block chain basic platform and uses public blockchain smart contract to implement roles and authority management functions similar to the consortium blockchain model. Supply chain companies can participate in multiple semipublic chain contracts, and each contract can have a different authority. For example, a manager of a company in smart contract A can formulate A's supply chain contract rules, while a manager of the company in smart contract B is an ordinary user. The system functional architecture of semipublic chain smart contract is shown in Figure 8.

The architecture of the upgradeable smart contract is shown in Figure 9. The contract is divided into three parts: the immutable agent contract, the data storage contract, and the replaceable logical contract. The owner of the contract calls the different logical contracts by updating the logical contract address of the proxy contract and thereby implementing the upgrade of contract. The "owner" of the smart contract has been full controlled, meaning that the upgradeable smart contract is centralized and loses the decentralized nature of the blockchain. Modifications made to the smart contract by the "owner" do not require permission from other nodes.

4.1. Role Authority Management. In order to realize the user role setting, authority management, and other functions of the Hyperledger consortium blockchain, nodes must first be distinguished, so this model uses the address of Ethereum as the unique identifier. In this paper, a smart contract model is designed to implement the authority management of different roles based on address role control and address verification. The flow of the algorithm is shown in Algorithm 1.

4.2. Subcontract Storage and Privacy Protection. Hyperledger's multichain mainly implements data classification storage, privacy protection, and other functions. Ethereum has only one main chain, which cannot achieve true multichain storage and privacy protection. This paper constructs a new independent contract for this purpose, which is also controlled by the commander node and is used to store data for semipublic contracts. In order to implement the privacy protection mechanism, the contract address is not open to the public, and the commander node owns all authorities such as modified data. The flow of the algorithm is shown in Algorithm 2 .

4.3. Contract Upgrade Management. To solve the modification, upgrade and start-stop management after the release of smart contract. The semipublic chain designed the role of commander. A new version of the contract address can be written and released through the commander node, and participants can migrate to the new contract. If only part of the subcontract is upgraded, the commander node can directly update the subcontract address. The flow of the algorithm is shown in Algorithm 3.

Table 1: Cost coefficient value table.

Blockchain type	Hardware cost α	Network cost β	Development cost δ	Maintenance cost λ	Access cost ω
Ethereum	0.01	0.02	0.2	0.3	0.6
Hyperledger	0.015	0.03	0.9	0.8	0.8

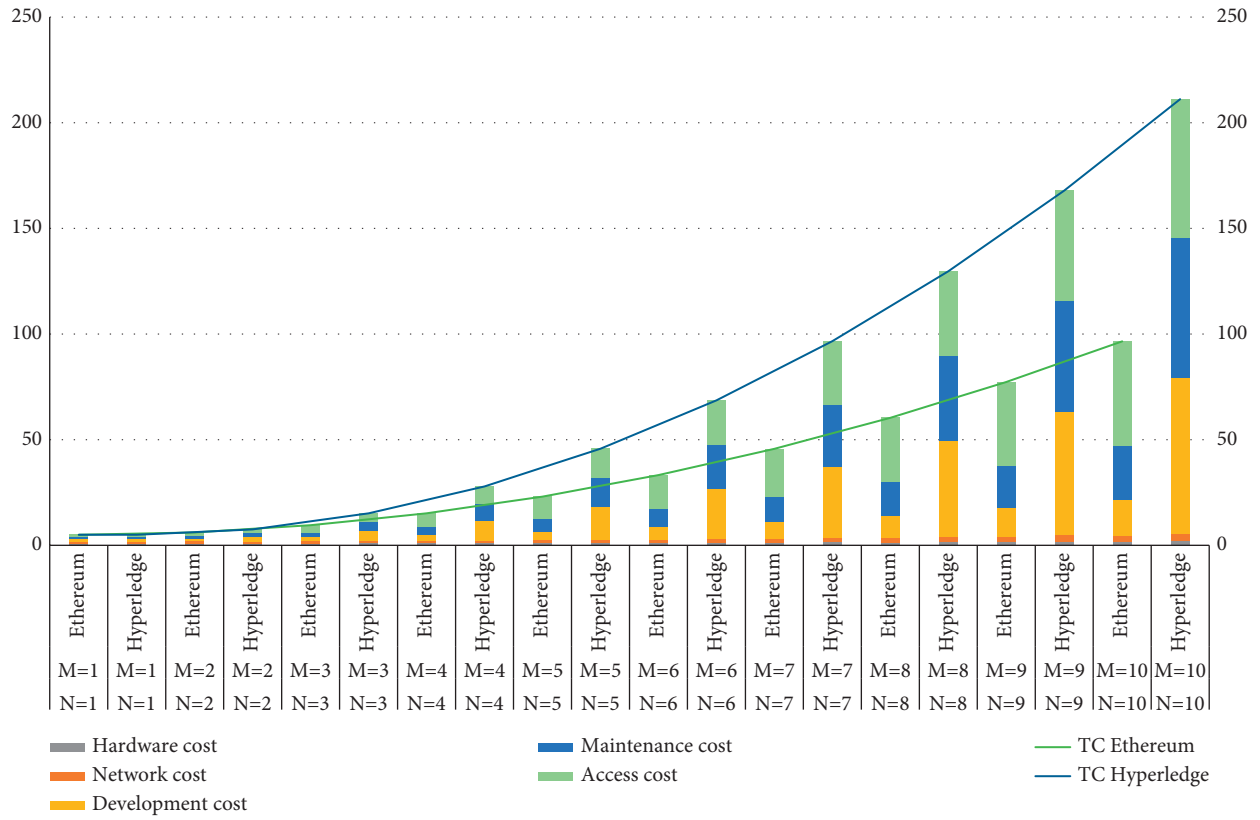

Figure 6: Supply chain cost structure diagram.

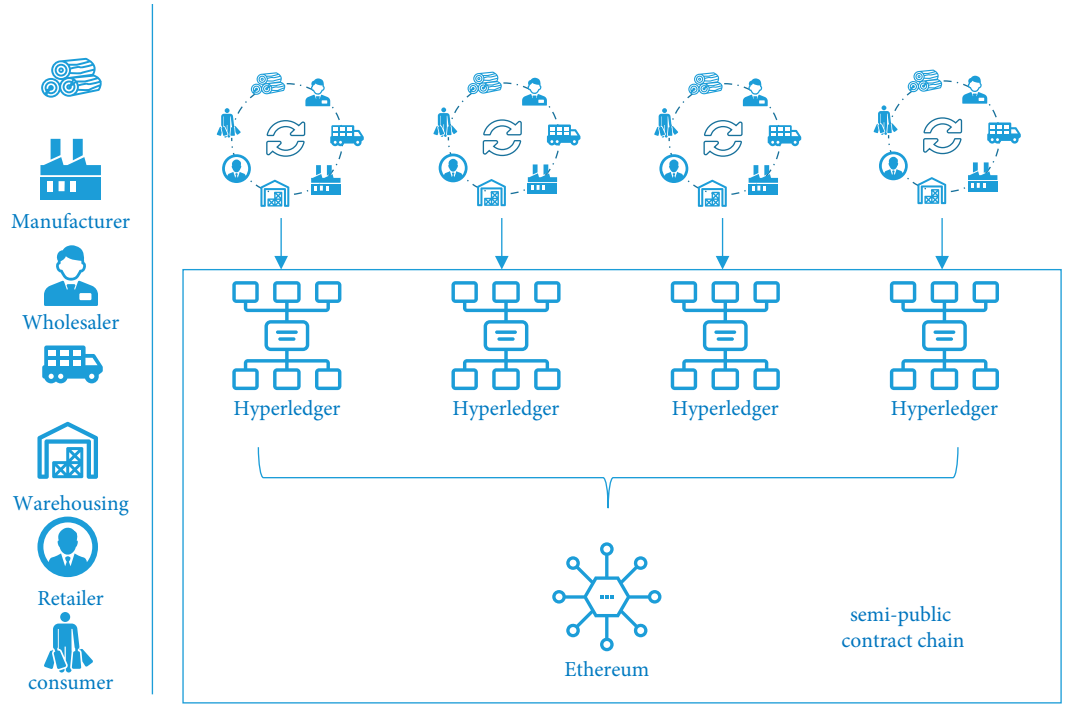

Figure 7: Logical architecture diagram of semipublic chain.

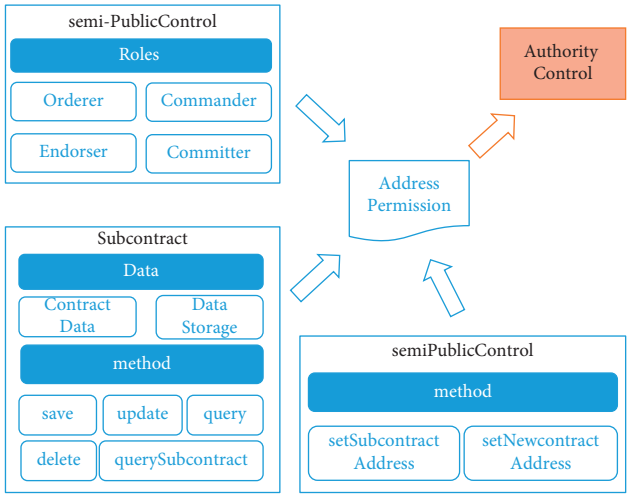

Figure 8: The system architecture of semipublic chain smart contract.

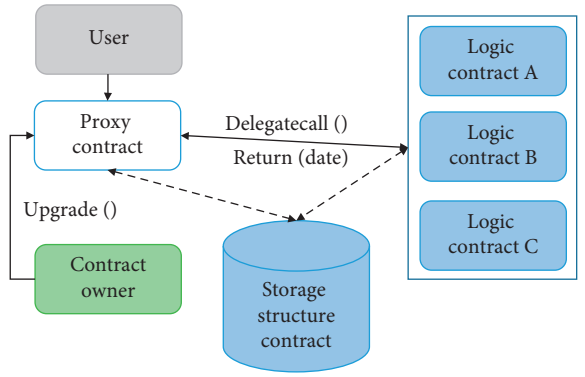

Figure 9: Architecture diagram of the upgradeable contract.

```
pragma solidity ^0.4.11;
contract Semi-PublicControl {
   //Declare addresses for different roles.
   address public CommanderAddress; address public EndorserAddress;
   address public OrdererAddress; address public CommitterAddress;
   //Commander node authority control and verification
   modifier onlyCommander() {
      require(msg.sender == commanderAddress);}
   //Endorser node authority control and verification
   modifier onlyEndorser() {
      require(msg.sender == EndorserAddress);}
   //Orderer node authority control and verification
   modifier onlyOrderer() {
      require(msg.sender == OrdererAddress);}
   //Committer node authority control and verification
   modifier onlyCommitter() {
      require(msg.sender == CommitterAddress);}
   //Permission node authority control and verification
   modifier onlyPermission() {
      require(
         msg.sender == OrdererAddress||msg.sender == CommanderAddress||msg.sender == EndorserAddress||
msg.sender == CommitterAddress);}
```

Algorithm 1: Continued.

```
//Set up a new Commander, only the Commander has authority function setCommander(address_newcommander) external
onlyCommander {
//Address cannot be null check
    require(_newCommander != address(0));
    CommanderAddress = _newcommander;}
 //Set up a new Endorser, only the Commander has authority function setEndorser(address_newEndorser) external
onlyCommander { require(_newEndorser != address(0));
    EndorserAddress = _newEndorser;}
//Set up a new Orderer, only the Commander has authority function setOrderer (address_newOrderer) external onlyCommander
{
    require(_newOrderer != address(0));
    OrdererAddress = _newOrderer;}
//Set up a new Committer, only the Commander has authority function setCommitter (address_newCommitter) external
onlyCommander {
    require(_newCommitter != address(0));
    CommitterAddress = _newCommitter;}
//Permission node universal permission, only Permission node has permission function TODO (address_newCommitter) external
onlyPermission {
    //This method verifies the Permission node address before executing the code}}
```

Algorithm 1: The main management process of the semipublic chain.

```
pragma solidity ^0.4.11;
contract Subcontract{//Subcontract
struct Container{//Declare a stored data structure
  uint64 CreateTime; uint32 SireId;
  uint16 DataStorage1; uint32 DataStorage2
  uint64 DataStorage3; address DataOwner}
Container [] container; //Declare an array of stored data
function save (address_from, data1, data2, data3){
  return SireId }//New data method, return SireId unique identifier
function update(address_from, sireId, data1, data2, data3){}//Update data method
function query(address_from, sireId){}//Query data method
function delete(address_from, sireId){}//Delete data method}
contract Semi-PublicControl {
function querySubcontract(address_from, sireId){//Semi-public chain contract query subcontract method
  Subcontractdate = Subcontract.query(address_from, sireId);}}
```

Algorithm 2: The subcontract function management of the semipublic chain.

```
pragma solidity ^0.4.11;
contract semiPublicControl {
//Set up new address of Subcontract, only the Commander has authority
function setSubcontract Address(address_address) external onlyCommander {
  Subcontract = Subcontract Interface(_address);}//Set up new address of Contract, only the Commander has authority
function setNewcontract Address(address addressnew) external onlyCommander {address Newcontract = addressnew;}}
```

Algorithm 3: The subcontract management process of the semipublic chain.

5. Conclusion and Discussion

The emergence of Bitcoin has revived the vitality of smart contract. With the continuous development and evolution of blockchain and smart contract technology, smart contract and blockchain are now inseparable. Smart contract is one of the most important functions of the blockchain and the main reason why the blockchain can be called a disruptive technology. Compared with the rapidly evolving blockchain commercial products, the academic research on smart contract is still in its infancy, and the key technologies need to be researched and followed up. Based on the Hyperledger architecture model and the Ethereum development environment, this paper not only designed a semipublic chain smart contract model but also verified the feasibility and implementability of the model. We select

two representative blockchain products, Ethereum and Hyperledger, and systematically introduce the evolutionary process and functional enhancement of the evolution of smart contract. In the future, with the continuous development of blockchain technology in the supply chain business scenarios and application fields, smart contract technology will also continue to evolve.

However, the system could be improved in several ways. First of all, semipublic chain smart contract only considers two blockchain products, Ethereum and Hyperledger, and does not consider other blockchain products. Secondly, we did not consider the issue of transaction fees. As the price of Ether rises, transaction fees may be an important part of the cost. Finally, we assume that the supply chain process is consistent and does not consider the supply chain of different processes; the actual supply chain process may be more complicated.

The analysis of this manuscript shows that semipublic chain smart contract can effectively reduce user costs on the basis of user management, and the simulation results have also been verified. The limitations of this model can be extended. First of all, semipublic chain smart contract is only considered from the perspective of cost and can be expanded from the perspective of revenue, so that the user utility of the smart contract solution can be better evaluated. Secondly, this article only considers the situation where there is no central organization, and the main body of the central organization can be expanded, such as introducing an independent third party as a service platform for the central organization [30, 31]. Thirdly, we only consider the main processes of the supply chain; we can expand the supply chain business scenarios and introduce other modes of supply chain such as closed-loop supply chain (CLSC) [32] and environmental supply chain dynamics (ESCD) [33]. Finally, this article only considers the Ethereum technology, which can extend the application technology, such as the use of blockchain technology to achieve Radio Frequency IDentification (RFID) management [34] and blockchain of double-chain architecture technology [35]. This not only expands the limitation of business scenarios proposed in this paper but also provides a direction for future research.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (71572113 and 71871144), Matching Project of the National Natural Science Foundation of China (1P16303003, 2020KJFZ034, 2019KJFZ048, and 2018KJFZ035), and Innovation Training Program of University of Shanghai for Science and Technology (XJ2021150, XJ2021160, XJ2021165, XJ2021191, and XJ2021206).

References

[1] H. Yuan, H. Qiu, Y. Bi, S.-H. Chang, and A. Lam, "Analysis of coordination mechanism of supply chain management information system from the perspective of block chain," *Information Systems and E-Business Management*, vol. 18, no. 4, pp. 681–703, 2020.

[2] C. K. Wu, K. F. Tsang, Y. Liu et al., "Supply chain of things: a connected solution to enhance supply chain productivity," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 78–83, 2019.

[3] S. Golrizgashti, S. Piroozfar, and A. Dehghanpoor, "Product innovation and supply chain sustainability," *IEEE Engineering Management Review*, vol. 47, no. 4, pp. 128–136, 2019.

[4] S. Tnnissen and F. Teuteberg, "Analysing the impact of blockchain-technology for operations and supply chain management: an explanatory model drawn from multiple case studies," *International Journal of Information Management*, vol. 52, Article ID 101953, 2020.

[5] Z. Wang, T. Wang, H. Hu, J. Gong, X. Ren, and Q. Xiao, "Blockchain-based framework for improving supply chain traceability and information sharing in precast construction," *Automation in Construction*, vol. 111, Article ID 103063, 2020.

[6] R. Azzi, R. K. Chamoun, and M. Sokhn, "The power of a blockchain-based supply chain," *Computers & Industrial Engineering*, vol. 135, pp. 582–592, 2019.

[7] A. D. Vaio and L. Varriale, "Blockchain technology in supply chain management for sustainable performance: evidence from the airport industry," *International Journal of Information Management*, vol. 52, Article ID 102014, 2020.

[8] P. Helo and Y. Hao, "Blockchains in operations and supply chains: a model and reference implementation," *Computers & Industrial Engineering*, vol. 136, pp. 242–251, 2019.

[9] H. Hasan, E. Alhadhrami, A. Aldhaheri, K. Salah, and R. Jayaraman, "Smart contract-based approach for efficient shipment management," *Computers & Industrial Engineering*, vol. 136, pp. 149–159, 2019.

[10] M. Li, L. Shen, and G. Q. Huang, "Blockchain-enabled workflow operating system for logistics resources sharing in E-commerce logistics real estate service," *Computers & Industrial Engineering*, vol. 135, pp. 950–969, 2019.

[11] A. Carvalho, "Bringing transparency and trustworthiness to loot boxes with blockchain and smart contracts," *Decision Support Systems*, vol. 10, Article ID 113508, 2021.

[12] S. Myung and J.-H. Lee, "Ethereum smart contract-based automated power trading algorithm in a microgrid environment," *The Journal of Supercomputing*, vol. 76, no. 7, pp. 4904–4914, 2020.

[13] Z. Zheng, S. Xie, H. N. Dai et al., "An overview on smart contracts: challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2019.

[14] O. López-Pintado, M. Dumas, L. García-Bauelos, and I. Weber, "Controlled flexibility in blockchain-based collaborative business processes," *Information Systems*, vol. 8, Article ID 101622, 2020.

[15] W. Shao, Z. Wang, X. Wang, K. Qiu, C. Jia, and C. Jiang, "LSC: online auto-update smart contracts for fortifying blockchain-based log systems," *Information Sciences*, vol. 512, pp. 506–517, 2020.

[16] H. Baharmand and T. Comes, "Leveraging partnerships with logistics service providers in humanitarian supply chains by blockchain-based smart contracts," *IFAC-PapersOnLine*, vol. 52, pp. 12–17, 2019.

[17] M. Kassen, "Blockchain and e-government innovation: automation of public information processes," *Information Systems*, vol. 103, Article ID 101862, 2021.
[18] M. Y. Khan, M. F. Zuhairi, T. Ali, T. Alghamdi, and J. A. Marmolejo-Saucedo, "An extended access control model for permissioned blockchain frameworks," *Wireless Networks*, vol. 26, no. 7, pp. 4943–4954, 2020.
[19] S. Saxena and H. E. Z. Farag, "Distributed voltage regulation using permissioned blockchains and extended contract net protocol," *International Journal of Electrical Power & Energy Systems*, vol. 130, Article ID 106945, 2021.
[20] A. Lucas, D. Geneiatakis, Y. Soupionis, I. Nai-Fovino, and E. Kotsakis, "Blockchain technology applied to energy demand response service tracking and data sharing," *Energies*, vol. 14, pp. 1–17, 2021.
[21] S. F. Wamba and M. M. Queiroz, "Blockchain in the operations and supply chain management: benefits, challenges and future research opportunities," *International Journal of Information Management*, vol. 52, Article ID 102064, 2020.
[22] A. A. Mukherjee, R. K. Singh, R. Mishra, and S. Bag, "Application of blockchain technology for sustainability development in agricultural supply chain: justification framework," *Operations Management Research*, 2021.
[23] N. Kshetri, "1 blockchain's roles in meeting key supply chain management objectives," *International Journal of Information Management*, vol. 39, pp. 80–89, 2018.
[24] Z. Xu, J. Zhang, Z. Song, Y. Liu, J. Li, and J. Zhou, "A scheme for intelligent blockchain-based manufacturing industry supply chain management," *Computing*, vol. 103, pp. 1771–1790, 2021.
[25] F. Longo, L. Nicoletti, A. Padovano, G. d'Atri, and M. Forte, "Blockchain-enabled supply chain: an experimental study," *Computers & Industrial Engineering*, vol. 136, pp. 57–69, 2019.
[26] N. Hazbiy, F. A. Ekadiyanto, and A. Ratna, "Blockchain based warehouse supply chain management using hyperledger fabric and hyperledger composer," *International Journal of Information Technology*, vol. 9, pp. 147–151, 2020.
[27] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, pp. 1–9, 1997.
[28] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008, http://www.bitcoin.org/bitcoin.pdf, https://academic.oup.com/jamia/article/24/6/1211/4108087#210313959, and https://www.sciencedirect.com/science/article/pii/S0264275120311987.
[29] S. Nakamoto, "Bitcoin source code," 2018, https://github.com/bitcoin/bitcoin.
[30] Q. Long, J. Lin, and Z. Sun, "Modeling and distributed simulation of supply chain with a multi-agent platform," *The International Journal of Advanced Manufacturing Technology*, vol. 55, no. 9–12, pp. 1241–1252, 2011.
[31] J. P. Skudlarek, T. Katsioulas, and M. Chen, "A platform solution for secure supply-chain and chip life-cycle management," *Computer*, vol. 49, no. 8, pp. 28–34, 2016.
[32] E. U. Olugu, K. Y. Wong, and A. M. Shaharoun, "A comprehensive approach in assessing the performance of an automobile closed-loop supply chain," *Sustainability*, vol. 2, no. 4, pp. 871–889, 2010.
[33] J. Hall, "Environmental supply chain dynamics," *Journal of Cleaner Production*, vol. 8, no. 6, pp. 455–471, 2000.
[34] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, "A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain," *IEEE Access*, vol. 5, Article ID 17467, 2017.
[35] K. Leng, Y. Bi, L. Jing, H.-C. Fu, and I. Van Nieuwenhuyse, "Research on agricultural supply chain system with double chain architecture based on blockchain technology," *Future Generation Computer Systems*, vol. 86, pp. 641–649, 2018.