

Smart Contracts for Supply Chain Management

Abhishek Sharma, Dr. Budesh kanwar

Dept. Artificial intelligence and Data Science
Poornima Institute of Engineering and Technology Jaipur.

Abstract- The manufacture of raw materials to deliver the product to the consumer in a traditional supply chain system is a manual process with insufficient data and transaction security. It also takes a significant amount of time, making the entire procedure lengthy. Overall, the undivided process is ineffective and untrustworthy for consumers. If blockchain and smart contract technologies are integrated into traditional supply chain management systems, data security, authenticity, time management, and transaction processes will all be significantly improved. Blockchain is a revolutionary, decentralized technology that protects data from unauthorized access. The entire supply chain management (SCM) will be satisfied with the consumer once smart contracts are implemented. The plan becomes more trustworthy when the mediator is contracted, which is doable in these ways. The tags employed in the conventional SCM process are costly and have limited possibilities. As a result, it is difficult to maintain product secrecy and accountability in the SCM scheme. It is also a common target for wireless attacks (reply to attacks, eavesdropping, etc.). In SCM, the phrase “product confidentiality” is very significant. It means that only those who have been validated have access to the information. This paper emphasizes reducing the involvement of third parties in the supply chain system and improving data security. Traditional supply chain management systems have a number of significant flaws. Lack of traceability, difficulty maintaining product safety and quality, failure to monitor and control inventory in warehouses and shops, rising supply chain expenses, and so on, are some of them. The focus of this paper is on minimizing third-party participation in the supply chain system and enhancing data security. This improves accessibility, efficiency, and timeliness throughout the whole process. The primary advantage is that individuals will feel safer throughout the payment process. However, in this study, a peer-to-peer encrypted system was utilized in conjunction with a smart contract. Additionally, there are a few other features. Because this document makes use of an immutable ledger, the hacker will be unable to get access to it. Even if they get access to the system, they will be unable to modify any data. If the goods are defective, the transaction will be halted, and the customer will be reimbursed, with the seller receiving the merchandise. By using cryptographic methods, transaction security will be a feasible alternative for recasting these issues. Finally, this paper will demonstrate how to maintain the method with the maximum level of safety, transparency, and efficiency.

Index Terms- Blockchain, Smart Contracts, Supply Chain Management

I. INTRODUCTION

In today’s global market, supply chain management (SCM) is critical. It has a significant influence on the global economy. SCM is often defined as the movement of goods from producer to consumer. It is divided into numerous phases, starting with the supply of raw materials and ending with the client, and includes the producer, distributor, and retailer. It is a global process in which components are sourced from a single location, packaged, and supplied globally. Traditional supply chain management serves a broad goal but falls short of full compliance. Giving the final customer the ability to reverse the transaction and assuring the quality of the items

supplied has several limits. It usually corresponds to forward flows, or the flow of products from the sender to the recipient [1, 2]. Supporting the reverse flow of items and transactions for every consumer is also critical. The traditional supply chain management system might be disrupted by blockchain and smart contracts. The supply chain can benefit from the blockchain’s transparency and immutability [3].

By offering a secure mechanism for collecting data and developing and running programmed scripts or applications known as smart contracts, the blockchain aids in the modernization of the supply chain [4]. Smart contracts can help supply chain managers track the origin and security of

their products. We discussed the issues and came up with a solution.

The approach of developing a conceptual framework for a supply chain management system is included in this research. Its main objective is to leverage block chain and smart contracts to provide safe transactions and high-quality products. It will allow any client to return a product and receive a refund for the money spent on it, resulting in a trustworthy worldwide market. Most significantly, under our paradigm, the whole SCM system will undergo a significant transformation.

A blockchain is a continuously growing list of logs known as cryptographically connected and secured blocks. The blocks are connected cryptographically. The bulk of the nodes check the blocks in the blockchain network. The block will then be added to the chain that all network nodes share when it has been verified. Handling a single piece of data necessitates thousands for instances, each of which requires a significant amount of work and time to avoid. Access to information is associated with higher quality in various blockchain systems. There are a few qualities that distinguish blockchain from other technologies. Data on the blockchain, for example, is immutable, tamper-resistant, and based on a decentralized network, and it can be hacked and encrypted. In general, three forms of blockchain exist: public or unauthorized, private permitted, and consortium blockchain. Each one has specific characteristics because of the uniqueness of the network's geographic area [5, 6].

Problem Statement. On a daily basis, supply chain management systems encounter problems, and nearly all of them require immediate attention and action. The degree and intricacy of these issues may vary. Easy data access, quality and sustainability, supplier management, and managing consumer expectations are just a few of them. The Supply Chain Management system confronts ever-increasing challenges year after year. And, with the emphasis on stability, these concerns are now front of mind. After all, supply chains are at the heart of effective business operations, and problems will inevitably have an influence on a company's bottom line.

Motivation. Blockchain technology validates and stores data using blockchain data structures, generates and updates data using distributed node consensus methods, uses encryption to guarantee data transmission and access security, and employs intelligent script code. It is a novel computer and distributed infrastructure paradigm for programming and manipulating data.

The transaction data produced by each participating entity is packed into a data block, which is then organized in chronological order to form a chain of data blocks in the blockchain system. The main body has the same data chain

and cannot be tampered with unilaterally. Any changes to the information must be approved by a certain percentage of the topic, and only new material may be added. The old data cannot be changed or removed. The identification of each subject and the transaction information between entities are open and transparent and cannot be faked, thanks to information sharing and consistent decision-making.

II. METHOD AND METHODOLOGY

This section delves into the many methods and materials used to achieve the goal. The modeling approach is based on the concept of the blockchain as a cyber SC chain of information services that represents the operational fulfillment of the physical SC. When transaction activities begin and conclude, the blockchain maintains an account of them. As a result, the operations of logistics companies may be viewed as information services that they provide to the blockchain architecture. Smart contract design, in this sense, may be thought of as the computation of start and completion timings for information services in a blockchain-driven cyber environment that mirrors real SC activities.

Outline of the Full System. This system proposes an abstract concept after briefly explaining and reflecting on current definitions of smart contracts. Because of the concept's novelty and its sophisticated technological foundation, there is no common definition of smart contracts at this time. Given the lack of agreement on nomenclature, it appears that providing an overview of existing techniques and refining an appropriate description is of the highest relevance.

In Figure 1, the whole structure of the blockchain and smart contract working mechanism is depicted. A variety of logistics service providers can take up some operations in the SC that are progressively arranged as flow-oriented activities (i.e., intermediates, carriers).

Different operations may be assigned to several logistics service providers. Assigning operations to logistics firms will result in a variety of work lead times and pricing because different logistics service providers operate at different times and at different rates. The process of creating smart contracts includes assigning logistics businesses to jobs and arranging their activities in theocentric acts, which results in a Blockchain design structure. The start and finish of the execution of the operations of the transaction will be recorded on the blockchain



Figure 1: A minimal overview of the complete system.

As a result, the operations of logistics firms might be classified as information services provided by the blockchain architecture. In this sense, smart contract design may be described as the choice to start and stop information services in a blockchain-driven cyberspace that replicates physical SC activities. For blockchain python, for smart contracts and cryptocurrency purposes, solidity was used in this news supply chain strategy. My ether wallet and Ganache were also used since virtual transactions were performed here for a more realistic outcome. The Spider IDE for implementing Python, the remix IDE for solidity, and NodeJS for web pages are used in this system. Figure2 is the entire process of the system, from the start of the seller's transaction.

Whenever any seller intends to buy any product, he has to do it on the website and there will be no involvement of any third party which can manipulate the payment method. After the buyer gets the product, he will unblock the payment that he has made and then the seller will get his payment.

This transaction data will be auto updated by the website. If, anyhow, the buyer gets a damaged product or becomes misfired with the product he got, then he can return the product and take back his payment through the website. This will not cost any extra money to anyone. After they both make a clear statement review of the transaction, only then will the payment be updated.

Figure 3 is the diagram of the whole mechanism of this system. It will begin with the website receiving data from the user. After the transaction, the customer must approve if the goods are satisfactory or not. The administrator will be able to see the contents of this section.

It will expand in accordance with market circumstances. Any announcement may be made from here. All sales, customer information, and bounce rate data will be managed via this admin panel. The website will automatically update the transaction details.

If the customer receives damaged goods or is unhappy with the product, he may return it and get a refund via the website. Nobody will have to pay any additional money as a result of this.

Only when they have both completed a thorough statement review of the transaction can the payment be adjusted. Figure 4 is shown. This is the login page for the seller. The homepage features three web pages, which lead to three additional pages.

Another one of them is the login page for the seller. In order to log in, the seller will have to type in his e-mail address and password into his personal account. This page allows you to create three different types of accounts. This page's main

features are the admin account, seller account, and buyer account. If the user already has an account, the username and password will be required. However, if someone is creating one for the first time, they will need an e-mail address and a password. Every activity may be controlled and monitored by the admin account. The system requires that the seller and buyer accounts follow the same set of rules. For the development of this website system, it required JavaScript, CSS, HTML, and PHP. In Figure 5, the registration page for buyers is shown.

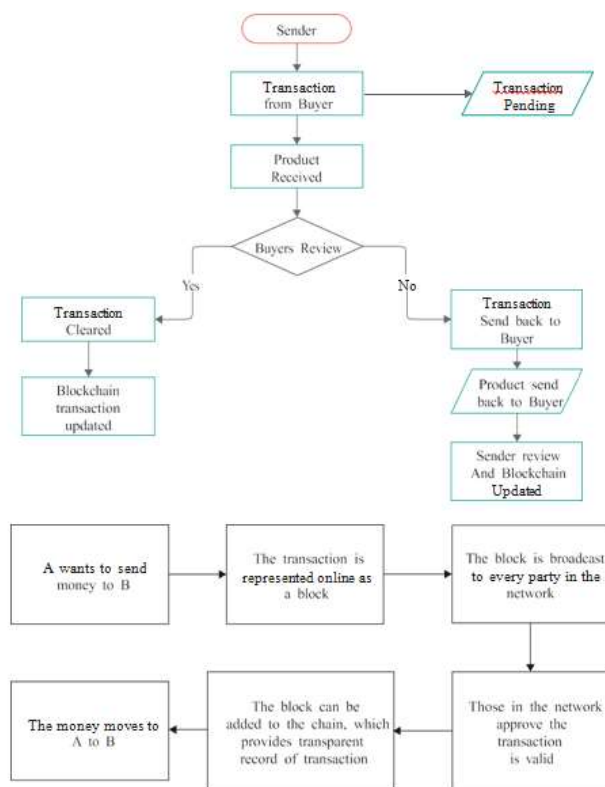


Figure 3: Block diagram of the whole system.



Figure 4: Login page for a seller account.

The buyer will fill out the registration form with various pieces of information, like their username, e-mail address, and password. They will also type the password they used to

confirm it again. The information n will be typed and stored in the local database. Using the administrative control panel, it can be seen.

Data Security and Transaction in the Blockchain and Smart Contract. Transactions are not born in the block.

When one transaction occurs, the list needs to be empty. Otherwise, it will not get all the separate data. Blocks cannot have the same data in each block

A transaction is put into a block if it is authorized by a majority of the nodes. Each block references the one before it, forming the blockchain. A miner must examine if a transaction is eligible to be processed according to the blockchain history before adding it to their block. The transaction is legitimate and can be included in the block if the sender's wallet balance has sufficient funds according to the current blockchain history. In Figure 6, it is shown that in the traditional system, if we want to buy any big asset, we have to pay for it and, after paying the amount, we get a deed that is the proof of the asset.

blockchain class, there is a genesis block, a chain function, and a block function that will add a new block and will mine a block. The create block function will take two argument s: proof and the previous hash number. In Figure 9, proof_of_function has two arguments. One is self, which is for using an instance object that will be created by the class. The other one is previous proof, for creating a way for miners to solve the problem.



Figure 7: Immutable ledger with high security

```

import datetime
import hashlib
import json
from flask import Flask, jsonify, request
import requests
from uuid import uuid4
from urllib.parse import urlparse
  
```

Figure 8: Imported function for the blockchain in spider.

```

def proof_of_work(self, previous_proof):
    new_proof = 1
    check_proof = False
    while check_proof is False:
        hash_operation = hashlib.sha256( str( new_proof**2 - previous_proof**2).encode()).hexdigest()
        if hash_operation[:4] == '0000':
            check_proof = True
        else:
            new_proof += 1
    return new_proof
  
```

```

def add_transaction(self, sender, receiver, amount):
    self.transactions.append({'sender': sender,
                              'receiver': receiver,
                              'amount': amount})
    previous_block = self.get_previous_block()
    return previous_block['index'] + 1
  
```

Figure 10: Function of transaction in spider



Figure 5: Sign-up page for a buyer account

Chain and Data Component. This section introduces the basic notion of smart contracts by discussing the nature and types of contracts. From a legal and economic standpoint, we first outline the basic aspects of contracts and their various roles throughout the relationship lifecycle. Then, after examining several ways of defining smart contracts, we offer a broad definition. This part concludes with a critical examination of the role of distributed ledger technology in the idea of smart contracts. In Figure 8, it shows all the functions which have been imported. The date time function is used for each block to have its own timestamp when the block is created and mined. The hash function will need to hash the blocks because the hash function will be used here.

Here, the new proof value is 1 because after every iteration, it needs to increment by one until it gets the right proof. check proof will do the checking part to see if it is correct proof or not. The scope contains four leading zeros, which will make mining difficult for the miners, and it is a string of 64 characters. The encode function will encode the string in the right format, which is expected by the sha256 function. In Figure 10, the function of the transaction has been declared. This add transaction method will conduct the process through the self, sender, and receiver keys of the argument.

Here, using the Json function, we will encode the blocks before we hash them. The flask function will be used by the flask library. We will need a flask class because this will use a web application and jsonify, which will take the message and interact with the blockchain with the postman. In the

This will service the transaction before adding it to the block. This list of data will use the append function to add a new transaction. For every new transaction on, it is necessary to add the previous data, which will be done by the previous_block function. Before returning to the previous

block function, it will add +1. As a result, the number will automatically rise, the list will grow, and data will be stored.

2.4. Transaction Component. After defining the version of solidity, there will be some functions we need to add to connect with my ether wallet and Ganache. We have declared the total amount that people who will mine can use from here. And after that, the coin value of our created coin named “had coin.” In Figure 11, the transaction method has been initiated. There will be two mining. One is for equity had coins and the other is for equity USD, which is for dollars.

Mapping is like a function, but this data will be stored in an array. It is not like a function that takes a variable and returns it. It is more like an array that will have an input variable which will be the investor’s address. It will return the equity in had coins and the other will be in USD. In solidity, address is a type of work that works as a function. Modifiers can check if the investor can buy or sell any had coins or not. In Figure 12, the control panel for the entire system is shown. All of the transactions that have occurred will be added here automatically. It has the ability to monitor and regulate the operations of miners as well as data.

```
contract hadcoin_ico {
    uint public max_hadcoins = 1000000;
    uint public usd_to_hadcoins = 1000;
    uint public total_hadcoins_bought = 0;
    mapping(address => uint) equity_hadcoins;
    mapping(address => uint) equity_usd;
}
```

Figure 11: Hadcoin creations and adjusting USD in remix

There will also be information on the buyer and seller, as well as data from Ganache containing transaction details. Details about the buyer’s account, the sell the first step in the process, it has no prior hash number. It will include evidence of labor, which will be the same for all blocks. Transactions will be placed here using the “hadcoin” coin. As soon as all of the blocks are added to the system, the length will grow and the system will grow too. It will also contain a timestamp, which is a record of the moment when it was first added to the list. In Figure 14, the fourth block is mined. After mining, it will automatically be added here and will show the message text. The difficulty is always set at a certain time interval and then modified every two weeks, such that a block may be instructed at a predetermined time period. As a result, the difficulty of the network sets a predetermined time period between building two blocks, which is roughly 10 minutes (in Bitcoin).

Previous hash is the hash number of its immediate previous hash and proof is the unique number of miner’s account, and the transaction will be logged here. This section’s information will be visible to the administrator. Its growth will naturally reflect market conditions. From here, an announcement may be issued. This admin panel will manage all sales, customer

information, and bounce rate information. New buyers will be able to see the past information of any previous buyers, making this method extremely trustworthy.



Figure 12: Control panel of the website.

III. RESULT AND DESIGN ANALYSIS

This is the get chain output. Here, index means the number of the block, as this is the first block, so it has no previous block. Proof indicates the proof of work, which is different for every single block. A Timestamp indicates when the block was created, the exact time, and date with a number. As it is the first block, no transaction happens here. In Figure 13, this is a random block from all the blocks created for mining. Here the index number is different. It has a previous hash which is the hash value of its previous block.

The proof number is different from the first block, which means mining is happening here. The genesis block is the initial block in any blockchain. As this is block.

The Timestamp function will show the exact time of when it was mined. The most crucial aspect of this system is mining. This mining is done using a fixed algorithm. Many blocks may be mined at once, but only one will get the proper code and be added to the list. This block will get the prize, while the others will not. The procedure, with the exception of mining, will not be particularly genuine. In Figure 15, a block has been mined and this will be added to the main chain.

```
1 |
2 |
3 |   "chain": [
4 |     {
5 |       "index": 1,
6 |       "previous_hash": "0",
7 |       "proof": 1,
8 |       "timestamp": "2021-05-23 00:55:28.684967",
9 |       "transaction": []
10 |    },
11 |  ],
12 |   "length": 1
```

Figure 13: Genesis block and details.

```
"index": 4,
"previous_hash": "f972c9ab8ff358c576c2613c567d7035170335f670d044d16e4c05ed81f4c68",
"proof": 21391,
"timestamp": "2021-05-23 04:03:32.996029",
```

Figure 14: Adding blocks in the list.

Blockchain is mining correctly or not, then with the `is_valid` command, it can be checked. If it stops working, then it will not be valid.

If any sort of unauthentic behavior is detected, the chain will be immediately disrupted. The validation text will not be displayed if this is the case. It will display the phrase "There must be a problem." As a result, the system's working process must be halted, and data must be double checked. As previously said, nothing can be altered manually, and each piece of data will have its own unique identification. This is an advantage of the system. Figure 18 is from Ganache after a transaction occurs in a smart contract.

Before the transaction happened, the balance of ethereum was 100 ETH and the TX COUNT was 0. After the transaction occurs, it becomes 99.97 ETH and the TX COUNT becomes 3 because we have mined three times. Here we need to use its private key code, which is just right after the index. In Figure 19, it is shown that all the transaction data will be stored in the transactions section of Ganache.

```

{
  "index": 4,
  "message": "Congratulations, you just mined a block!",
  "previous_hash": "2a3f663164fe14ac8feb2b4c6139d9afbd9b0861e718e45c8217d94db8e23ef",
  "proof": 21391,
  "timestamp": "2021-05-24 20:48:06.213044"
}

```

Figure 15: After mining text in postman

```

{
  "transaction": [
    {
      "amount": 1,
      "receiver": "hadcoin5001",
      "sender": "9f76ed3cb56b4824be8689a91afc2148"
    }
  ]
}

```

Figure 16: Transaction occurs in postman.

```

{
  "message": "All good. The blockchain is valid"
}

```

Figure 17: Valid or invalid checking in postman



Figure 18: Proof of the transaction data in Ganache



Figure 19: Adding the transaction data in Ganache

Transaction	Status	Timestamp
Transaction	Success	2021-05-24
Transaction	Success	2021-05-24
Transaction	Success	2021-05-24
Transaction	Success	2021-05-24
Transaction	Success	2021-05-24

Figure 20: Record of all the transactions

Table 1: Comparison table between this paper and other papers

Points of this paper	Points of other papers
(1) Third-party interventions are removed as a result of the (1) Third-party interventions are frequent in SCM management deployment of the SCM system described in this article, and strong systems are not every paper proposes their own coin as a currency relationships between peers are formed. [4].	(2) The transaction system is broken. Both the vendor and the (2) The transaction procedure is transparent and highly secure. Buyer's trust are affected [5].
(3) Absence of an immutable ledger on the other paper. If there is (3) This paper contains the strongest immutable ledger technology any, then it is not secure enough. As a result, the website was hacked to prevent cyberattacks, which results in a safe website.	

Additionally, the date and They all have their own distinct worth that cannot be duplicated. The whole system will be separated and secured as a recorded here, and the list will become larger as more result of this. Had coin's approach provides greater security transactions are made. Only 5 transactions have been made thus far, according to this section. When the status is clear it indicates that both the buyer and the seller have completed their transaction. The Date now option will provide time information. It contains all of the transaction information. Comparison with Other Papers. Table 1 clearly shows the comparison between this paper's materials and the flaws of other papers.

In this article, several very strong security measures have been added, making the system very safe and trustworthy. Other papers, on the other hand, have

IV. CONCLUSION

The goal of this article is to make the supply chain management more intelligent, current, and secure. This framework is immutable and tends to give total transaction on transparency. It protects our website from unauthorized access and data manipulation. Furthermore, smart contracts cut the amount of time spent on tedious paperwork. In conventional supply chain management, a lot of documentation is usually necessary. The blockchain keeps the information as proof, making smart contracts immutable. The transaction, immutability, and refundable processes in supply chain management are primarily influenced by this paradigm. This study proposed an end-to-product supply method. It also allows all customers to return a product if they are unhappy with it and receive a refund for their purchase. Every actor's

function and role has been specified. It also means that our framework may be used for a variety of reasons. Smart contracts are also discussed in terms of the structure. The difficulties that individuals experienced with old procedures will be permanently eliminated because of the findings of this paper. Among its numerous benefits (the most important of which is the capacity to keep data safe), this research promises to speed up and decrease transaction costs, as well as increase financial inclusion by offering more possibilities for people who do not have easy access to financial services

REFERENCES

1. C. Prahinski and C. Kocabasoglu, "Empirical research opportunities in reverse supply chains," *Omega*, vol. 34, no. 6, pp. 519–532, 2006.
2. L. Xiaoming and F. Olorunniwo, "An exploration of reverse logistics practices in three companies," *Supply Chain Management International Journal*, vol. 13, no. 5, pp. 381–386, 2008.
3. S. Apte and N. Petrovsky, "Will blockchain technology revolutionize excipient supply chain management?" *Oe Journal of Excipients and Food Chemicals*, vol. 7, no. 3, pp. 76–78, 2016.
4. S. Alqahtani, X. He, R. Gamble, and M. Papa, "Formal verification of functional requirements for smart contract compositions in supply chain management systems," in *Proceedings of the Hawaii International Conference On System Sciences*, pp. 5278–5287, Maui, USA, January 2020.
5. O. Alfandi, S. Otoum, and Y. Jaraweh, "Blockchain solution for IoT based critical infrastructures: byzantine fault tolerance," in *Proceedings of the 2020 IEEE Network Operations and Management and Symposium*, pp. 1–4, Budapest, Hungary, April 2020.
6. D. Magazzeni, P. Mcburney, and W. Nash, "Validation and verification of smart contracts: a research agenda," *Computer*, vol. 50, no. 9, pp. 50–57, 2017.
7. N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (SoK)," *Lecture Notes in Computer Science*, in *Proceedings of the International Conference on Principles of Security and Trust*, pp. 164–186, +essaloniki, Greece, April 2017.
8. F. Idelberger, G. Governatori, R. Riveret, and G. Sartor, "Evaluation of logic-based smart contracts for blockchain systems," in *Proceedings of the International Symposium on Rules and Rule Markup Languages for the Semantic Web*, pp. 167–183, New York, USA, June 2016.
9. L. Luu, D. H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the ACM SIGSAC Conference On Computer And Communications Security*, pp. 254–269, Vienna, Austria, October 2016.
10. F. Q. Z. Yao, "BlockChain based supply chain financial risk management research," in *Proceedings of the 2020 International Conference On Mechanical Automation And Computer Engineering (MACE)*, pp. 170–180, Nanchang, China, 2020.
11. Y. Fu and J. Zhu, "Big production enterprise supply chain endogenous risk management based on blockchain," *IEEE Access*, vol. 7, no. 8626088, pp. 15310–15319, 2019.
12. S. K. Dwivedi, R. Amin, and S. Vollala, "Blockchain based secured information sharing protocol in supply chain management system with key distribution mechanism," *Journal of Information Security and Applications*, vol. 54, no. 102554, pp. 1–15, 2020.
13. E. H. Alfonso-Lizarazo, J. R. Montoya-Torres, and E. G. Franco, "Modeling reverse logistics process in the agroindustrial sector: the case of the palm oil supply chain," *Applied Mathematical Modelling*, vol. 37, no. 23, pp. 9652–9664, 2013.
14. 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.
15. Z. Nehai, P. Piriou, and F. Daumas, "Model- checking of smart contracts," in *Proceedings of the IEEE Smart Data (SmartData)*, pp. 980–987, Halifax, NS, Canada, July 2018.
16. N. Kshetri, "1 Blockchain's roles in meeting key supply chain management objectives," *International Journal of Information Management*, vol. 39, pp. 80–89, 2018.
17. S. E. Chang, Y.-C. Chen, and M.-F. Lu, "Supply chain reengineering using blockchain technology: a case of smart contract based tracking process," *Technological Forecasting and Social Change*, vol. 144, pp. 1–11, 2019.
18. I. Ahmed and S. Dixit, "Role of technologies in revamping the supply chain management of kirana stores," *Blockchain Applications in IoT Ecosystem*, pp. 275–287, EAI/Springer Innovations in Communication and Computing, Europe, 2021.
19. R. Kamran, N. Khan, and B. Sundarakani, "Blockchain technology development and implementation for global logistics operations: a reference model perspective," *Journal of Global Operations and Strategic Sourcing*, vol. 14, no. 4, pp. 360–382, 2021.
20. Z. Liu and P. Guo, "Supply chain decision model based on blockchain: a case study of fresh food E-commerce supply chain performance improvement," *Discrete Dynamics in Nature and Society*, vol. 2021, Article ID 5795547, 14 pages, 2021.
21. M. Yoo and Y. Won, "A study on the transparent price tracing system in supply chain management based on blockchain," *MDPI AG*, vol. 11, p. 4037, 2018.