*sustainability*

MDPI

*Article*

# A Study on the Transparent Price Tracing System in Supply Chain Management Based on Blockchain

**Minjae Yoo and Yoojae Won ***

Department of Computer Science and Engineering, Chungnam National University,
Daejeon 34134, Korea; vicerascal@cnu.ac.kr
* Correspondence: yjwon@cnu.ac.kr; Tel.: +82-42-821-6294

Received: 28 September 2018; Accepted: 1 November 2018; Published: 4 November 2018

**Abstract:** Over the past few years, the importance of improving the product distribution structure has increased worldwide. Therefore, distribution prices are now disclosed and product information is shared with consumers. However, distribution channels are complex and supply chain management (SCM) is autonomously executed in each company. For this reason, it has been frequently pointed out that the distribution process is not transparent and the distribution margin is high. In this paper, we propose a system that guarantees the transparency of the product distribution structure by applying blockchain and smart contracts to the price-tracking portion of supply chain management systems. This approach allows companies to track their trades by enhancing transparency in the SCM, thereby discouraging companies from pursuing excessive profits. In addition, companies can cut management costs by automatically storing distribution details in a blockchain network and managing information more securely.

**Keywords:** price tracing system; transaction transparency; supply chain management; blockchain; smart contract

## 1. Introduction

The recent contraction in consumer spending due to global fluctuations in distribution prices has increased the importance of improving the product distribution structure. The need for new methods to provide information to consumers has increased, and, as a result, a distribution price disclosure system has been introduced in many countries. However, distribution prices are unstable, and the distribution structure is often not transparently operated due to the closed and diverse supply chain management (SCM) operations of companies. The SCM framework was first introduced in 2000 as a new business model and a way to strategically manage relationships with key customers and suppliers [1]. In fact, many companies are using this system to manage their distribution structure and make profits. However, such a system with many stakeholders has a complicated distribution structure and a high distribution margin. In addition, there is no way for consumers to obtain information transparently about the margins of the product per distribution channel [2].

To solve this problem and develop this framework, we intend to introduce blockchain technology, which can enhance the security of data management and show the operation process transparently. Data management systems based on blockchain have been actively researched. A similar system has been used for the first time in finance, but this approach has been slowly spreading to various fields, such as manufacturing, distribution, medical care, cloud computing, and public services. A new paradigm of management decentralization has revolutionized the industry as a whole. We believe that the systems that apply will develop SCM further and give consumers confidence in its usage.

In this paper, we propose a path-based price tracking system for SCM based on blockchain and smart contracts. Many methods exist to manage data through the system [3–5]. The model proposed in this paper manages the information regarding the cost of the SCM using the Ethereum blockchain approach. Using this technique, companies can cut management costs because the processes are automated and handled through smart contracts. In addition, transaction data can be safely managed through a blockchain network that is difficult to modify [6,7]. Further, consumers can obtain information regarding the distribution route and distribution cost at any time; hence, they can buy goods knowing each transaction's history. This can reduce the incentive for companies to extract excessive profits. In addition, due to the characteristics of the blockchain technology, it is possible to protect the stored data against tampering and Distributed Denial of Service (DDoS) attacks thanks to the characteristic of Ethereum [8].

## 2. Materials and Methods

### *2.1. Related Studies*

#### 2.1.1. Supply Chain Management (SCM)

SCM is a system that registers the logistics flow from a product provider to a producer, distributor, or customer from a single value chain perspective and supports all necessary information for the flow to run smoothly. It is a technique to achieve overall process optimization between the components of a supply chain by avoiding sectorial optimization within an enterprise or individual enterprise unit optimization.

The importance of information sharing and delivery, mutual cooperation, and coordination of supply chain companies has been largely emphasized. The complexity of order management, production planning, information management, and tracking management can prevent a rapid decline in the efficiency of the supply chain. SCM is also a system capable of rational planning, management, and coordination control of the global supply chain and logistics and helps different stakeholders to accurately monitor and quickly respond to supply chain issues in real time. The flow of SCM is shown in Figure 1.

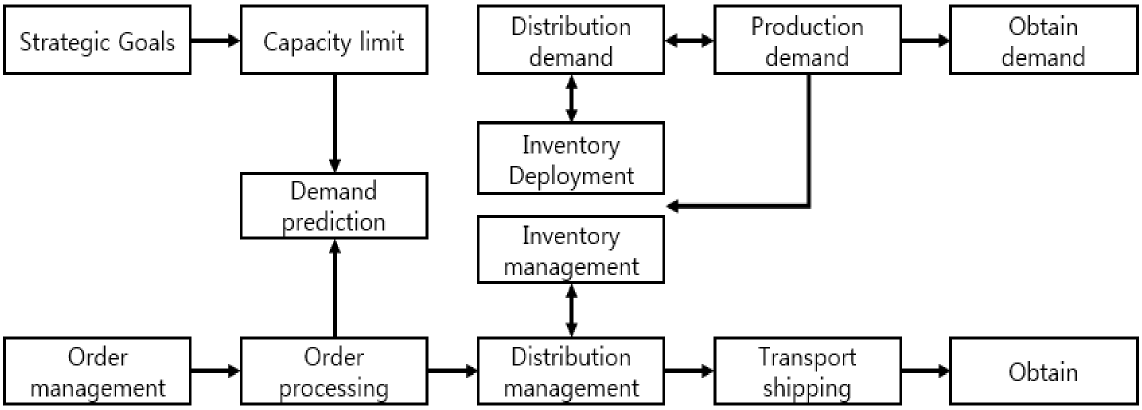


**Figure 1.** Flow of supply chain management (SCM).

#### 2.1.2. SCM Process

The process of SCM can be divided into the eight categories of customer relationship management, supplier relationship management, customer service management, demand management, order fulfillment, manufacturing flow management, product development and commercialization, and return management, as shown in Figure 2 [9].

The customer relationship management process is increasingly becoming a strategic, process-oriented, crossfunctional, value-creating, and financial means to both buyers and sellers. It also provides a structure for how relationships with customers are developed and maintained. The purpose of this process is to provide customized products and services to categorize customers over time based on customer value and increase the loyalty of their target audience. Supplier relationship management thus provides a structure for how relationships with suppliers are developed and maintained. The desired result is a win-win situation for both parties [10]. Customer service management monitors rules and detects threats that may prevent a company from meeting their commitments with customers, proactively tackling customer satisfaction. For example, a customer service manager can communicate with other process teams, such as supplier relationship management, manufacturing flow management, and order fulfillment, to ensure that appointments are implemented as planned [11]. Demand management is a process that balances the needs of customers with the capabilities of the supply chain. With the right processes, a company can actively support demand for executives and execute plans with minimal disruption [12]. Order fulfillment means all activities that allow a company to sign off the network and meet customer requests. This process maximizes profit, receiving feedback from crossfunctional and core vendors and customers [13]. Manufacturing flow management is an SCM process that includes all the activities needed to obtain the flexibility, implementation, and management of the supply chain to move products into and out of the factory. Manufacturing flow management assesses the current and desired flexibility of strategic resources, such as manufacturing plants, suppliers, distribution channels, IT, and human resources [14]. Product development and commercialization processes provide the structure for developing and bringing market products and customers and suppliers together. The effective implementation of this process not only enables executives to coordinate an efficient flow of new products across the supply chain, but also helps other members of the supply chain to strengthen the other activities needed to support manufacturing, logistics, marketing, and other commercial activities [15]. Finally, the returns management process is responsible for activities related to returns, return logistics, gatekeeping, and avoidance, both within the company and through key members of the supply chain. This can be an important opportunity to reduce costs, while avoiding administrative practices can help companies to reduce costs and increase their chances of becoming profitable.

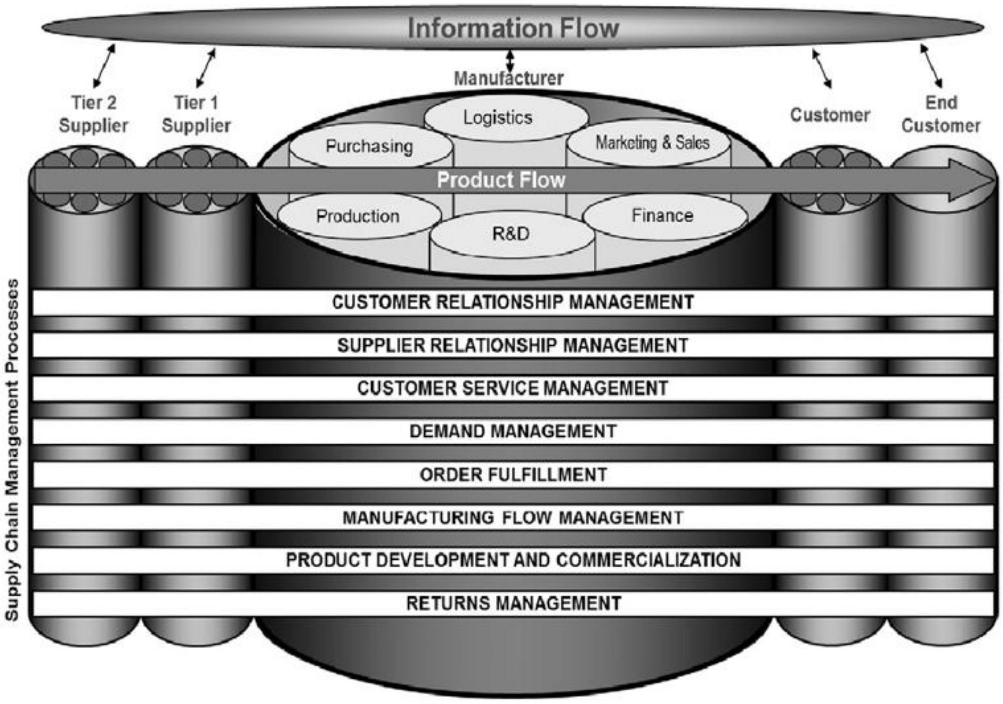


**Figure 2.** The supply chain management framework.

#### 2.1.3. Data Management on Blockchain

#### 2.1.3.1. Medical Care

Azaria et al. [16] have proposed a new distributed record management system that processes electronic medical records using blockchain technology. By providing comprehensive patient data and sharing medical information, the proposed system manages authentication, confidentiality, accountability, and data sharing, which are critical dimensions when dealing with sensitive information.

#### 2.1.3.2. Cloud Computing

Liang et al. [17] proposed a distributed and reliable cloud data source architecture based on blockchain technology, which provides antitamper records, ensures data account transparency in the cloud, and improves the privacy and availability of source data. In addition, they designed and implemented a ProvChain, which can help to transparently account for data across cloud storage and improve both privacy and data availability.

#### 2.1.3.3. Public Service

Yuan et al. [18] proposed a distributed and autonomous intelligent transportation system and ecosystem architecture based on blockchain technology. This architecture can better utilize intelligent transportation system infrastructure and resources and is effective in crowdsourcing technologies. It has been hitherto applied to the safety and reliability of parallel transportation management systems in the case of blockchain-based car sharing services [18].

#### 2.1.3.4. Others

Meng et al. [19] discussed the applicability of blockchain to intrusion detection systems (IDS) to tackle cyber threats and possible event identification. To improve the detection performance of a single IDS, the efficiency of the detection system is increased by applying blockchain to collaborative intrusion detection networks, where IDS nodes can exchange data with each other. Additionally, the authors also discuss the intersection of IDS and blockchain.

Kim et al. [20] proposed the Secure Authentication Management (SAM), which uses blockchain to authenticate mobile devices and verify the resource information on mobile devices participating in a Mobile Resource Management (MRM) resource pool. The SAM forms a blockchain based on the resource information of the subclient nodes around the master node of the MRM. Unauthorized MRM through SAM cannot access or forge data.

### *2.2. System Components*

#### 2.2.1. Blockchain

Blockchain is a distributed ledger technology, designed to store and verify data available to the network participants, and it was first discussed in a study by Satoshi Nakamoto [21], the developer of Bitcoin. Based on the proof-of-work system that allows two parties to trade without a third party being directly involved, the article suggested blockchain as core technology of the electronic payment system [21]. Typical blockchain-based virtual currencies include Bitcoin and Ethereum. However, blockchain technology has strengths and weaknesses, as presented in Table 1 [22].

**Table 1.** Strengths and weaknesses of blockchain.

| Strengths | Weaknesses |
|---|---|
| Transactions are publicly viewable to help protect against double spending | Irreversible transactions—no refunds unless the receiver issues a new transaction to send Bitcoins back |
| Transactions are secure—military-grade cryptology protection is theoretically immune to government interference or manipulation | Ostensibly anonymous users can be traced through network analysis |
| Users can establish their own financial system—developers can integrate a Bitcoin server directly into their applications | The balance of any user can be wiped out by a computer disk crash if a backup copy of the holdings does not exist |
| Trusted exchange | Subject to malware (for example, Mt. Gox14) |
| No costly regulation and overhead | Lack of regulation to protect consumers |

A blockchain has a structure that can implement various application services based on a distributed network that uses security technologies such as hash, digital signature, and encryption. Typical technologies include chaining technology and consensus techniques.

Chaining technology is a technique for connecting transactions to a blockchain when one block is created. The hash value of the preceding parent block is recorded in the connected block, and the blocks are connected to the hash-based tree and linked list structures. Since the modulation of the block takes longer than the verification, it is difficult to forge and falsify data.

Since blockchains use peer-to-peer (P2P)-based networks, trust between users is an important issue. To address the trust problem, various consensus algorithms are used, such as Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT). PoW and PoS are used in public blockchains, while PBFT is mainly used in private blockchains [23].

The processes of creating transactions, transferring, creating, and verifying blocks, and transferring blocks to the blockchain are shown in Figure 3.

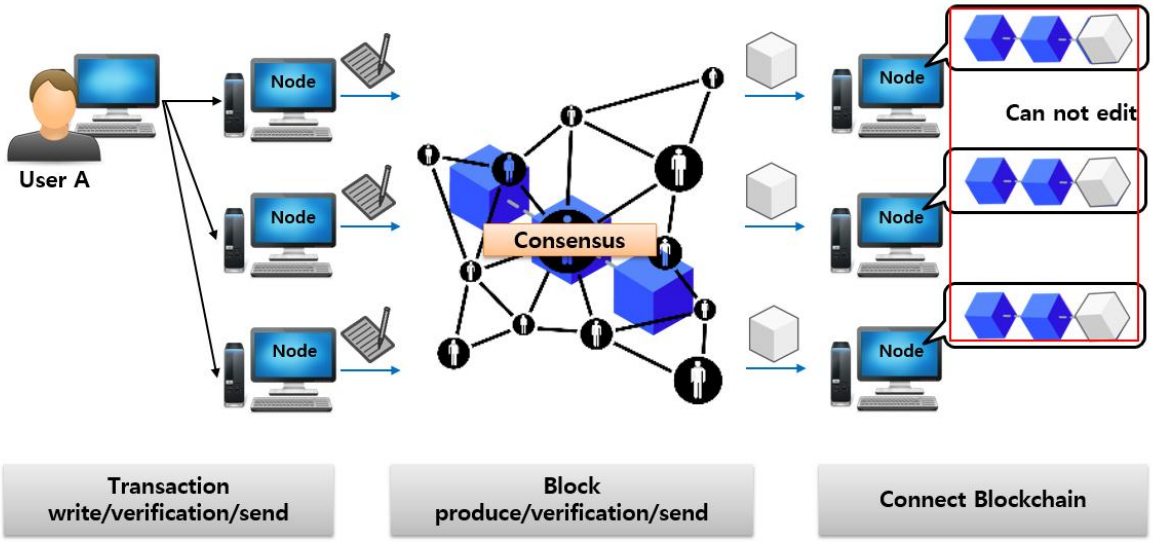


**Figure 3.** Blockchain operation.

### 2.2.2. Smart Contract

A smart contract is a protocol that automates and replaces the necessity of a contract, such as negotiation, expediting performance, and confirmation, and contract clauses on execution [24]. It usually has a user interface and provides better security than traditional contract laws with respect to general contract clauses. Its purpose is to reduce contract-related transactional costs. The concept of smart contracts traces back to the introduction of the Ethereum project.

The Ethereum smart contract refers to the bytecode working in an EVM (Ethereum Virtual Machine) and operates using a programming language called Solidity [25]. A smart contact in Ethereum is composed of a program code, a storage file, and the account balance. The program code is fixed when a contract is created and cannot be changed [26]. Its basic operating method is shown in Figure 4.

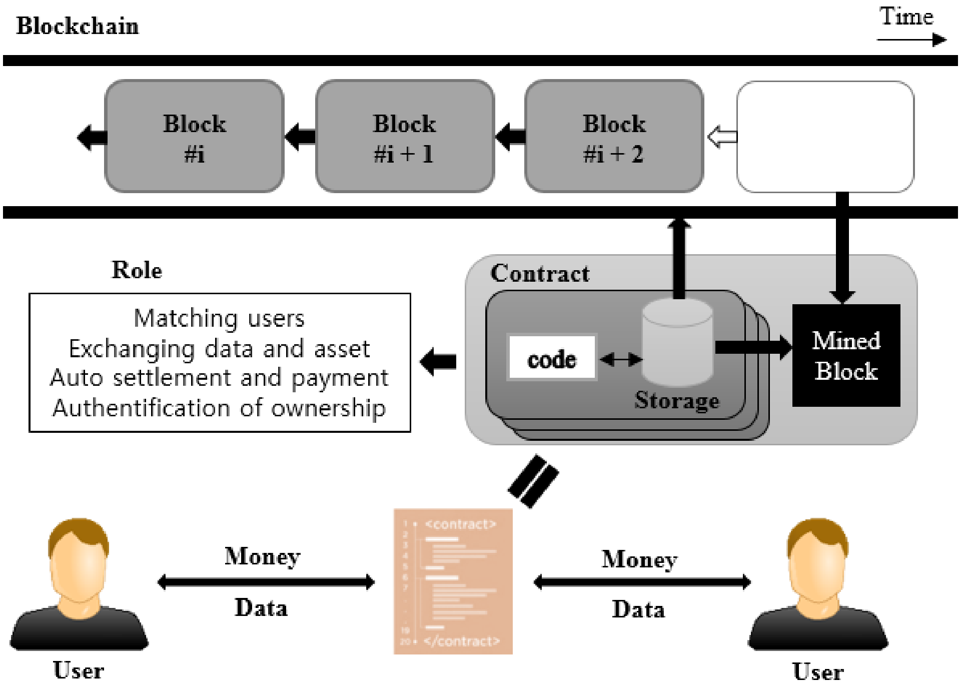


**Figure 4.** Basic operation method and roles of smart contracts.

### 2.3. *Transparent Price Tracing System in Supply Chain Management*

The proposed transparent price tracing system in SCM can provide benefits to both consumers and businesses. At present, consumers become aware of the price and origin of a product only when purchasing that product, and the details of the margin price in the distribution channel and the distribution channel of the product [27,28] are not known. However, the proposed system allows users to trace information about the raw material composition of the product and the manufacturer through blockchain records and disclose the data on the supply chain. All participants, including the end consumer, will thus be able to record, provide, and share prices on the supply chain for the product. Since this information is stored in a blockchain network, companies can make any changes to take advantage of this, delivering honest information to consumers, and the information is delivered through to web site [29].

Second, companies can obtain information regarding the purchase intentions of the buyer through a system that can track product history from the sales point to the final consumer. The personal information of the blockchain can be anonymized to perform marketing operations for exploiting customers' propensity to consume without disclosing personal information. Smart contracts can be used to process automated trading contracts within a certain period, and their history is also stored in the blockchain.

This system can significantly improve the processes of the SCM introduced in Section 2.1.2. Monitoring with blockchain makes customer service management more efficient and enables quick and convenient demand management. In addition, due to the management of manufacturing flows and customer relationships, supplier relationships are automated through smart contracts and data are securely stored and improved through blockchains.

This type of system allows to register and sell products in a distributed form, as well as track the seller's identity and awareness. This can create a new type of shared economy. In addition, the characteristics of the blockchain can prevent DDoS attacks and cut automation costs.

#### 2.3.1. System Structure

We propose a smart contract structure and correlation for the price tracing system in the SCM in Figure 5.

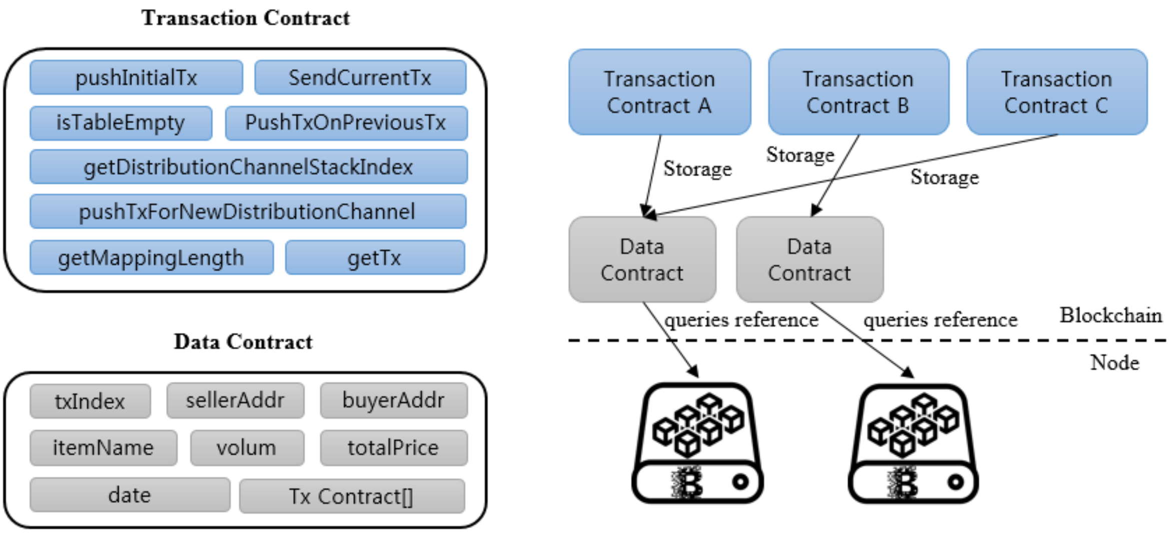


**Figure 5.** Smart contracts on the blockchain and data references for price tracing.

##### 2.3.1.1. Data Contract

A data contract is a smart contract that manages transaction details for each item. txIndex is the identifier of the transaction, SellerAddr is the seller's account, and buyerAddr is the buyer's account. itemName refers to the target of the transaction. Volum is the transaction volume, and totalPrice is the total transaction volume. Finally, Date represents the date of the transaction, and Tx Contract[] creates a stack that represents the distribution price path based on the current transaction information received from the device.

##### 2.3.1.2. Transaction Contract

A transaction contract is a contract that collects functions for trading. PushInitialTx is used to store transactions when the distribution route table is empty and SendCurrentTx operates in a different manner, depending on whether transactions are stored in the distribution route table. If there is no previous transaction, a stack is created and stored in the distribution route table. If there is a transaction, it is stored on the stack that represents the distribution channel in which the previous transaction exists. isTableEmpty verifies whether the table is empty, pushTxOnPreviousTx judges whether there is a previous transaction, and if so, that transaction is saved. getDistributionChannelStackIndex retrieves information about the item and displays it to the consumer when inputting the item or when inquiring the retail price. pushTxForNewDistributionChannel stores the user's current transaction. getMappingLength stores the transaction after determining the size of the table to add a new distribution path. getTx retrieves the transaction information at the specified location in the distribution route table and displays it to the consumer.

### *2.4. Transaction Type*

The proposed tracing system for price data deals with two types of transactions: Between the raw material producers and the wholesale and retail dealers and between the wholesale and retail dealers and the final consumers. When a transaction takes place, its type is shared in the network of the tracing system for distribution price data. Figure 6 proposes a type of tracing system for price data.

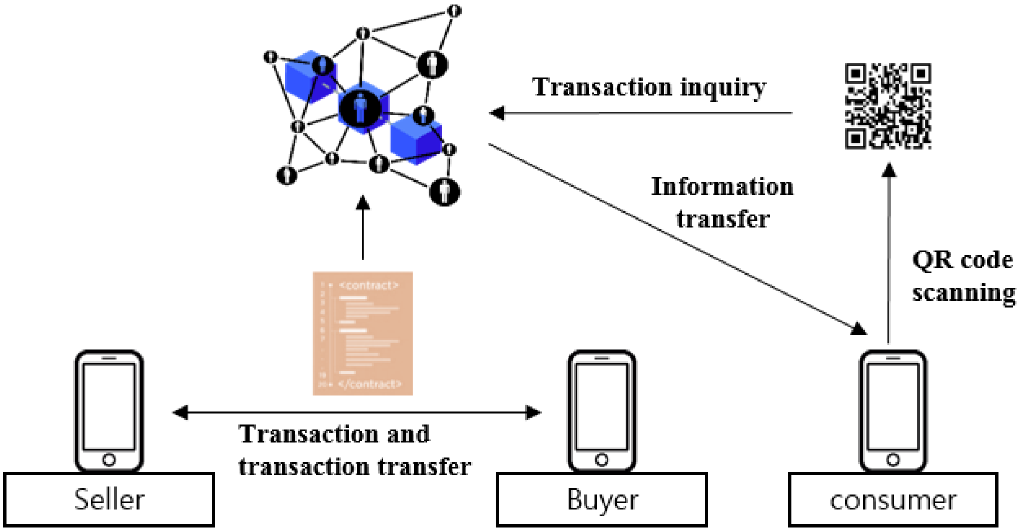


**Figure 6.** Transaction type for price tracing.

As shown in Figure 6, the tracing system is based on the Ethereum network, which has been in operation since 2013. Many projects are running on this network [30]. The stability of the nodes can be guaranteed, and the sellers, buyers, and consumers can access the implemented system on the Ethereum network via applications. The applications save or share data in the Ethereum network to process them and show the relevant information when the consumers, wholesalers, or retailers search for data. The producers, wholesalers, or retailers can be the sellers in such cases. The buyer refers to the wholesale and retail dealers, whereas the consumer refers to the final product purchaser and the wholesale and retail dealers who want to search for product information.

## 2.5. Nodes and Network Operations

In Figure 7, we can see the structure in which the nodes operate during the buyer and seller transactions.

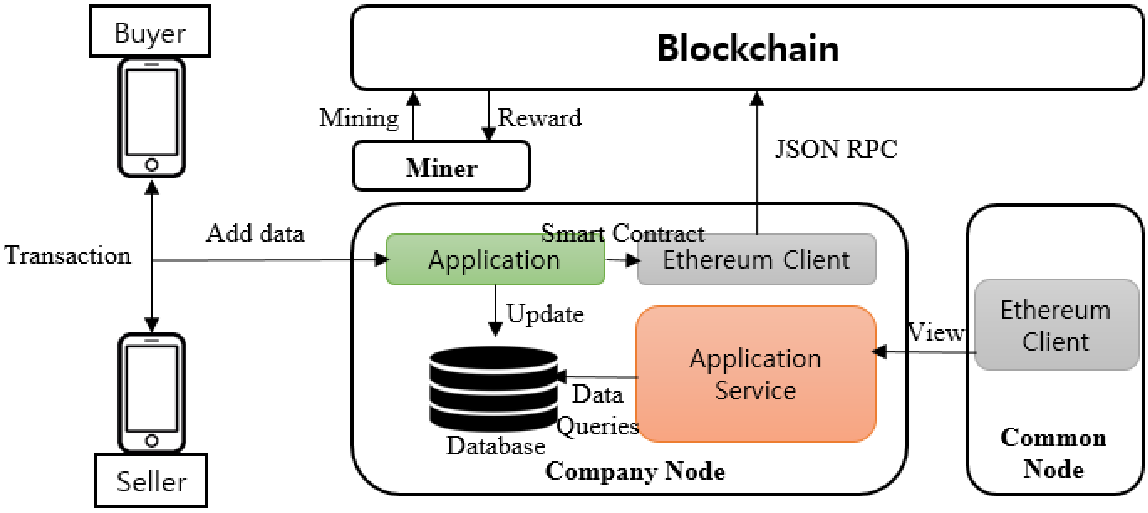


**Figure 7.** Nodes and network operations.

### 2.5.1. Transaction

Sellers and buyers enter transaction details using a device such as a cell phone, and when the transaction is transmitted, the data are passed over to the operating node in the company.

2.5.2. Company Node

Companies participating in a price tracing system in a blockchain-based SCM must operate at least one node. Each node has a database to store application information and provides application services to allow consumers to access the data. The transactions received are divided into contents to be stored in Smart Contract and contents to be stored in Database, and the parameters to be stored in Smart Contract are managed through Ethereum Client. Smart Contracts that have been verified by Ethereum Client are registered in the Blockchain Network through JSON RPC, and the relative data are managed.

2.5.3. Common Node

Consumers who are not participating in the system can download and own the entire dataset. Using the API (Application Programming Interface) of the application service provided by the enterprise, it is possible to access the Ethereum Network and synchronize data for all items.

2.5.4. Miner

Miners in the system encourage participation in the network to leverage computing resources and ensure that the data are reliable. In the Ethereum's unique incentive model, transactions are processed through Ether, the network currency unit. Companies are motivated to participate in Mining to provide data. In addition, Ether is required to enable transactions to be made automatically at regular intervals through Smart Contract using Ether. This allows the system participants to require Ether and maintain the network.

### *2.6. Scenario for Price Tracing System*

2.6.1. Transaction Scenario

The scenario in Figure 8 assumes the distribution phase as one step in a simplified form. Sellers and buyers can take the roles of producers and wholesalers, small and medium marts, large marts, and so on, as well as retailers. For this reason, they are represented in a simple step-by-step procedure because they are representative nodes.

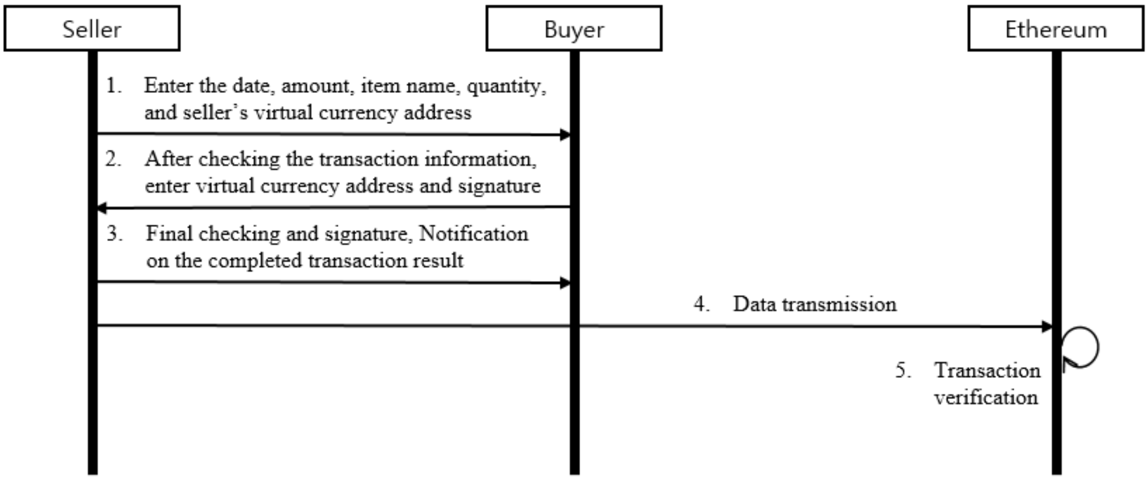

Seller
Buyer
Ethereum
1. Enter the date, amount, item name, quantity, and seller's virtual currency address
2. After checking the transaction information, enter virtual currency address and signature
3. Final checking and signature, Notification on the completed transaction result
4. Data transmission
5. Transaction verification

**Figure 8.** Transaction scenario between the seller and buyer.

The transaction scenario proposed in this study is shown in Figure 8. The seller and buyer are assumed to have Ethereum accounts. The seller logs into the application and enters the transfer date, transfer amount, item name, quantity, and seller's virtual currency address. Further, the buyer enters the virtual currency address and signature before the seller confirms the final transaction and signs to complete the transaction. The transaction created by the seller and buyer is transmitted to the Ethereum network, and the approved transaction is registered.

2.6.2. Scenario for Transaction Verification

The consumer can install the application to a device and track the product price. After running the application and scanning the QR code placed next to a product, the information on the product's distribution channel is obtained from the Ethereum network and displayed on the screen. From the obtained information, the user can know how much margin was marked up in each distributional transaction and verify whether any transaction is suspected of excessive profit.

## 3. Results

### *3.1. Application*

3.1.1. Development Environment

An application was implemented to test the system proposed in this paper. As you can see in Figure 9, Geth, an Ethereum-network-based client program, was used to implement this application and React Native was used to work, using JavaScript in an Android environment. For communication, Web3.js, a JavaScript library, was used to implement the test bed to use JSON RPC. Solidity, a scripting language, was used to control the functions of the smart contract.

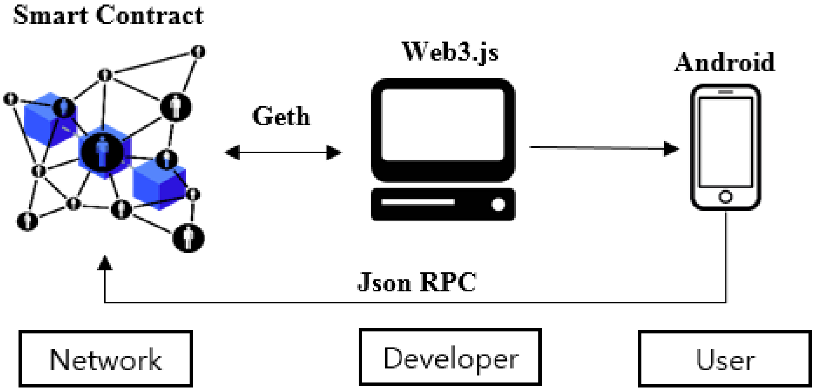


**Figure 9.** Development environment and communication environment.

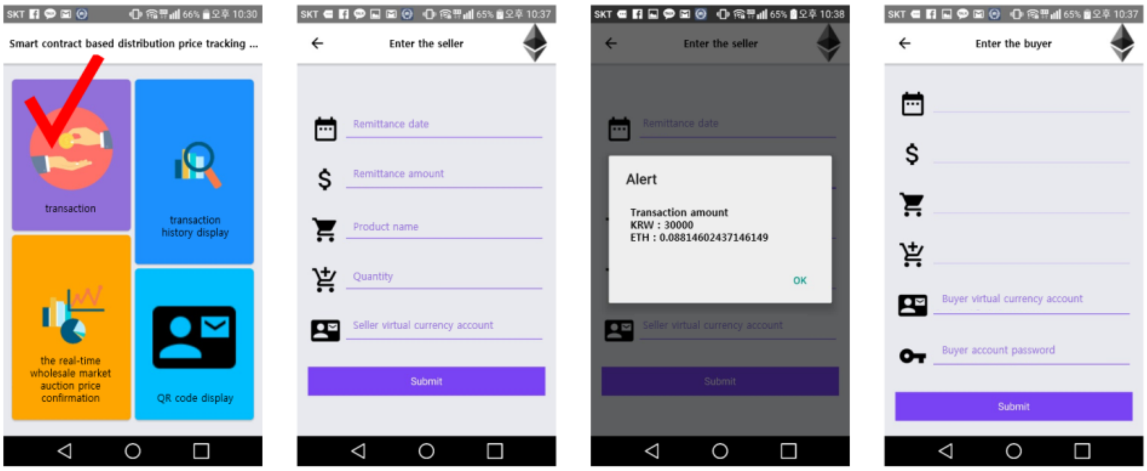
3.1.2. Implementation of Application

The implemented application was created on Android and consists of several functions: Trading, distribution history inquiry, showing My QR Code, and real-time wholesale market auction price confirmation tab.

As in Figure 10, in the transaction menu, companies and wholesalers can trade according to the scenario proposed in this paper. When one clicks on the transaction, the seller and buyer enter their information and sign it, and then the transaction is concluded.

As in Figure 11, One can track the price of the registered goods in the transaction history display menu. If the virtual currency address of the seller is inputted, the goods to be sold by the seller are listed, and the user can track the price of the desired product by clicking the desired product's picture or by searching.

In addition, there is a QR code display and real-time wholesale market auction price confirmation menu. As Figure 12, in the My QR Code display menu, the seller's account is displayed in the QR code and information on the products sold by the seller can be obtained through the QR code scan without inputting the seller's virtual currency address. In addition, the real-time wholesale market auction price confirmation menu can be linked to a web page for confirming real-time distribution information and knowing the general distribution price of the product [31].


Smart contract based distribution price tracking ...
transaction
transaction history display
the real-time wholesale market auction price confirmation
QR code display
Enter the seller
Remittance date
Remittance amount
Product name
Quantity
Seller virtual currency account
Submit
OK
Enter the buyer
Buyer virtual currency account
Buyer account password
Submit

Alert

Transaction amount
KRW : 30000
ETH : 0.08814602437146149

**Figure 10.** Trading menu screen.

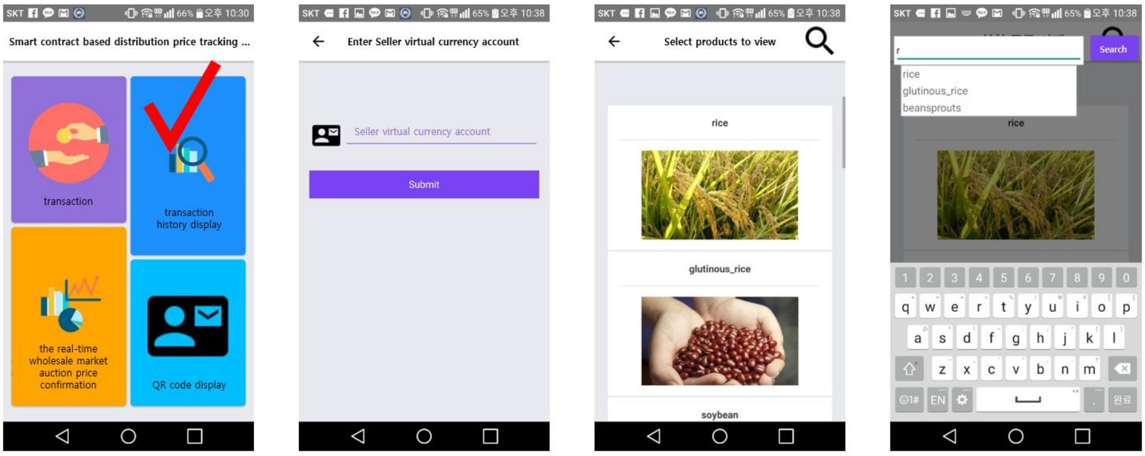
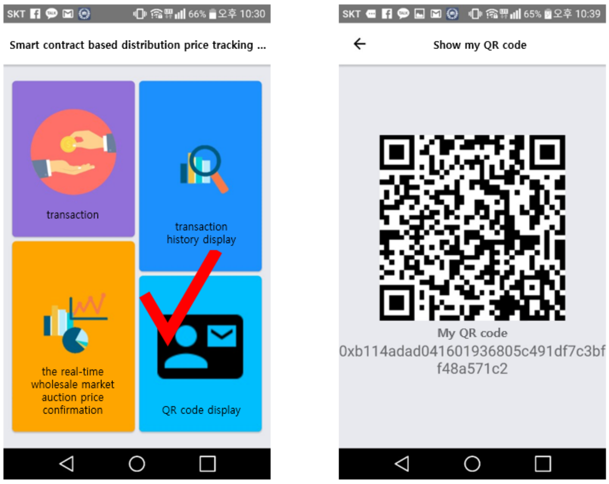

Smart contract based distribution price tracking ...
transaction
transaction history display
the real-time wholesale market auction price confirmation
QR code display
Enter Seller virtual currency account
Seller virtual currency account
Submit
Select products to view
rice
glutinous_rice
soybean
Search
rice
glutinous_rice
beansprouts

**Figure 11.** Distribution history search menu screen.


Smart contract based distribution price tracking ...
transaction
transaction history display
the real-time wholesale market auction price confirmation
QR code display
Show my QR code

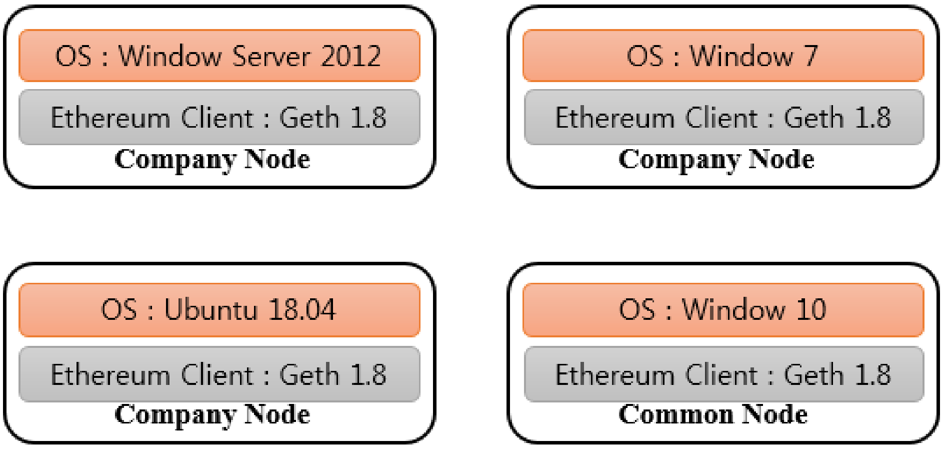
My QR code
0xb114adad041601936805c491df7c3bff48a571c2

**Figure 12.** View My QR Code menu screen.

## 3.2. Blockchain System

### 3.2.1. Development Environment

We have constructed an environment for testing a blockchain network, which is the backend of the proposed system. When operating a node, we assigned a total of three company nodes and one common node to test on various operating systems (OSs). One of the company nodes is operated in a Linux environment (Ubuntu 18.04, Chungnam University, Daejeon, Korea), one in a Windows server environment (Windows server 2012, Chungnam University, Daejeon, Korea), and the last one in a Windows environment. In the case of the Common Node, the node is operated in a Windows environment. The configured environment is shown in Figure 13.


OS : Window Server 2012
Ethereum Client : Geth 1.8
**Company Node**

OS : Window 7
Ethereum Client : Geth 1.8
**Company Node**

OS : Ubuntu 18.04
Ethereum Client : Geth 1.8
**Company Node**

OS : Window 10
Ethereum Client : Geth 1.8
**Common Node**

**Figure 13.** Node configuration environment.

3.2.2. Operation Test

We tested the behavior of a smart contract when a transaction is generated through an application implemented in a built environment. We also tested whether there is a difference between the synchronization of each node and the OS. Figure 14 shows the situation of a blockchain network when a transaction occurs.

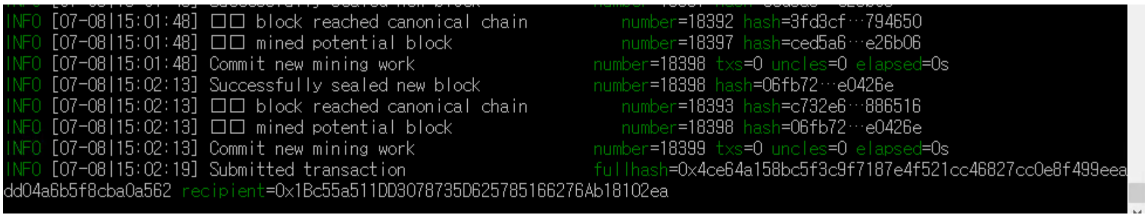

```
INFO [07-08|15:01:48] □□ block reached canonical chain      number=18392 hash=3fd3cf…794650
INFO [07-08|15:01:48] □□ mined potential block              number=18397 hash=ced5a6…e26b06
INFO [07-08|15:01:48] Commit new mining work              number=18398 txs=0 uncles=0 elapsed=0s
INFO [07-08|15:02:13] Successfully sealed new block       number=18398 hash=06fb72…e0426e
INFO [07-08|15:02:13] □□ block reached canonical chain      number=18393 hash=c732e6…886516
INFO [07-08|15:02:13] □□ mined potential block              number=18398 hash=06fb72…e0426e
INFO [07-08|15:02:13] Commit new mining work              number=18399 txs=0 uncles=0 elapsed=0s
INFO [07-08|15:02:19] Submitted transaction               fullhash=0x4ce64a158bc5f3c9f7187e4f521cc46827cc0e8f499eea
dd04a6b5f8cba0a562 recipient=0x1Bc55a511DD3078735D625785166276Ab18102ea
```

**Figure 14.** Blockchain network when transactions occur.

Figure 15 shows the screen that appears when a Company that operates on a Windows server as OS creates and compiles a proposed smart contract in the company node and registers it in the blockchain network. The screen informs that the smart contract is created, creates a transaction, and broadcasts it to other nodes. At this time, it is possible to know in which block the smart contract generated is registered.

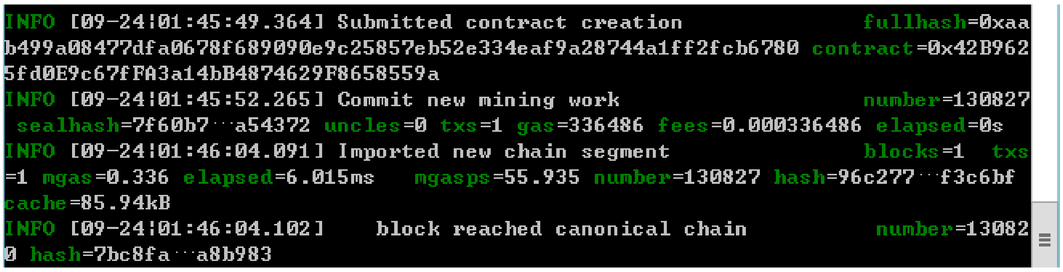
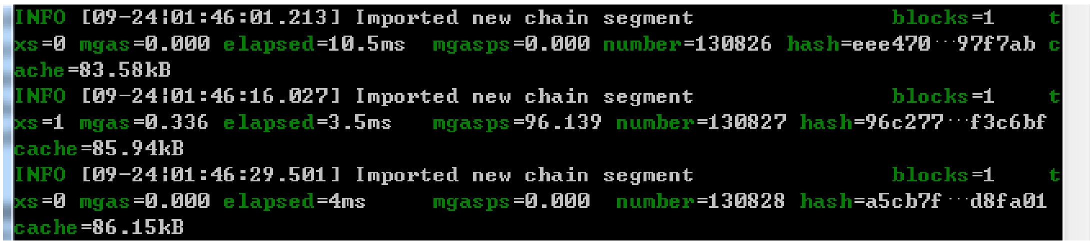

```
INFO [09-24|01:45:49.364] Submitted contract creation             fullhash=0xaa
b499a08477dfa0678f689090e9c25857eb52e334eaf9a28744a1ff2fcb6780 contract=0x42B962
5fd0E9c67fFA3a14bB4874629F8658559a
INFO [09-24|01:45:52.265] Commit new mining work                  number=130827
 sealhash=7f60b7…a54372 uncles=0 txs=1 gas=336486 fees=0.000336486 elapsed=0s
INFO [09-24|01:46:04.091] Imported new chain segment              blocks=1   txs
=1 mgas=0.336 elapsed=6.015ms   mgasps=55.935 number=130827 hash=96c277…f3c6bf
cache=85.94kB
INFO [09-24|01:46:04.102]    block reached canonical chain         number=13082
0 hash=7bc8fa…a8b983
```

**Figure 15.** Upload smart contract at company node.

Figure 16 shows the situation when the scenario shown in Figure 15 occurs in the common node synchronized through the application service of the company node. In this case, it is not immediately known what transaction has taken place, but one can see a transaction in the newly connected block. By checking the information inside the block through the command, one can obtain information regarding the transaction that has occurred.



```
INFO [09-24|01:46:01.213] Imported new chain segment               blocks=1     t
xs=0 mgas=0.000 elapsed=10.5ms   mgasps=0.000 number=130826 hash=eee470…97f7ab c
ache=83.58kB
INFO [09-24|01:46:16.027] Imported new chain segment               blocks=1     t
xs=1 mgas=0.336 elapsed=3.5ms    mgasps=96.139 number=130827 hash=96c277…f3c6bf
cache=85.94kB
INFO [09-24|01:46:29.501] Imported new chain segment               blocks=1     t
xs=0 mgas=0.000 elapsed=4ms      mgasps=0.000  number=130828 hash=a5cb7f…d8fa01
cache=86.15kB
```

**Figure 16.** Common node that is synchronizing the situation in Figure 15.

### 3.2.3. Test Results

When operating a node, there was no difference according to the OS used, and it took 0.003–0.01 s to synchronize between the company and common node. It took 0.005 s to generate a block, and 200 TPS, which is the number of transactions per second, is sufficient, but there is room for improvement. In addition, there may be some differences when the nodes are configured with the same class IP and applied to the actual system.

## 4. Discussion

In a blockchain-based SCM, the price tracing system provides consumers with transparent and reliable price information. Companies can understand consumers' tastes, and consumers can obtain price information, which encourages consumption. Permission management for blockchains allows nodes to operate and exchange data. The blockchain maintains a record of transaction information available to businesses, consumers, and wholesalers. Companies that run SCM can upload information from existing data stores to the blockchain to make use of existing system information. Data can also be accessed from portable devices through an Android application. Since the built-in system stores the price information in a distributed manner, if a point is attacked, its information can be recovered by comparison with surrounding nodes. However, for individual nodes, the local system administrator must guarantee appropriate security standards. The seller's name can be kept private, but a form of price fixing can be created, for instance by repeatedly trading with a specific provider or repeating the trade. In the case of blockchains, information can be fully verified; auditing the data is possible so that participants in the ecosystem can verify all information through consensus. In the initial prototypes, the existing Ethernet network has been used, which can affect the system. Hence, this risk has been addressed by Ethereum's policy change.

The proposed system can disclose the data generated in the supply chain to record, provide, and share product price history to all participants, from the initial manufacturer to the final consumer. If one does not want some information to be disclosed, the relevant parameters can be changed using encryption. This means that marketing activities can be carried out without leaking consumers' personal information. In addition, we need to deal with more detailed transactions at each stage, such as 1 tier supplier, 2 tier supplier, 3 tier supplier, customer, end customer. We cannot see the same transaction at each stage, and producers such as production, purchasing, logistics, marketing and sales, finance, and R and D can react differently.

When applying the proposed approach to real SCM, several factors will need to be considered. We are researching the best way to expand the proposed technology when high transaction volumes occur. In this scenario, one will also need to calculate the Ether that is consumed as a commission in the transaction. The rewards provided to miners should be kept constant to assure that the network operates without problems. The proposed system will enable the emergence of new data economies, matching the demand and supply between data producers and consumers. The prototype needs to be further developed by analyzing the agreement protocols and mining schemes, network maintenance methods, and incentives that can be given to those who participated in the early ecosystem for the blockchain-based SCM.

## 5. Conclusions

In this paper, we propose a price tracing system that can prevent wholesale firms from making extra profits by automating some transactions and integrating them, thus making the pricing information in the SCM transparent. This approach not only gives consumers easy access to retail price information, but also ensures that each transaction is valid and transparent. In addition, companies and wholesalers can easily identify consumer trends, are more secure than when using existing electronic records, and can automate some functions to reduce administrative costs. The proposed system is convenient for data management and can protect companies against DDoS and data forgery attacks.

In the future, we will apply the proposed system to SCM in actual operation and verify whether transaction registration and inquiry speed match those measured in the test bed, and design a blockchain network with an optimized algorithm and data transmission system. Therefore, we will consider ways to complement the weaknesses of this paper and develop the proposed model into a practical system.

**Author Contributions:** M.Y. designed the research framework, analyzed the data, and wrote the paper. Y.W. guided this work and provided extensive revisions during the study. All authors have read and approved the final manuscript.

**Funding:** This research received no external funding

**Acknowledgments:** This research was supported by MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2016-0-00304), supervised by IITP (Institute for Information and Communications Technology Promotion).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lambert, D.M.; Cooper, M.C. Issues in supply chain management. *Ind. Mark. Manag.* **2000**, *29*, 65–83. [CrossRef]
2. Kim, Y.D. Study on distribution margin by agricultural products' distribution channels. *J. Assoc. Food Distrib.* **2011**, 9–33.
3. McHugh, J.; Abiteboul, S.; Goldman, R.; Quass, D.; Widom, J. Lore: A database management system for semistructured data. *SIGMOD. Rec.* **1997**, *26*, 54–66. [CrossRef]
4. Bernstein, P.A.; Giunchiglia, F.; Kementsietsidis, A.; Mylopoulos, J.; Serafini, L.; Zaihrayeu, I. *Data Management for Peer-to-Peer Computing: A Vision*; University of Trento: Trento, Italy, 2002.
5. González-Aparicio, M.T.; Ogunyadeka, A.; Younas, M.; Tuya, J.; Casado, R. Transaction processing in consistency-aware user's applications deployed on NoSQL databases. *Hum.-Cent. Comput. Inf. Sci.* **2017**, *7*, 1–18. [CrossRef]
6. Idelberger, F.; Governatori, G.; Riveret, R.; Sartor, G. Evaluation of logic-based smart contracts for blockchain systems. In *International Symposium on Rules and Rule Markup Languages for the Semantic Web, Stony Brook University, New York, USA, 5–8 July 2016*; Springer: Cham, Lecture Notes in Computer Science, 2016; Volume 9718.
7. Sharma, P.K.; Moon, S.Y.; Park, J.H. Block-V.N: A distributed blockchain based vehicular network architecture in smart City. *J. Inf. Pros. Syst.* **2017**, *13*, 184–195.
8. Rodrigues, B.; Thomas, B.; Burkhard, S. Enabling a cooperative, multi-domain DDoS defense by a blockchain signaling system (BloSS). In Proceedings of the 42nd IEEE Conference on Local Computer Networks 2017 (LCN 2017), Singapore, 9–12 October 2017.
9. Lambert, D.M.; Enz, M.G. Issues in supply chain management: Progress and potential. *Ind. Mark. Manag.* **2017**, *62*, 1–16. [CrossRef]

10. Freytag, P.V.; Clarke, A.H. Business to business market segmentation. *Ind. Mark. Manag.* **2001**, *30*, 473–486. [CrossRef]
11. Bolumole, Y.A.; Knemeyer, A.M.; Lambert, D.M. The customer service management process. *Int. J. Logist. Manag.* **2003**, *14*, 15–31. [CrossRef]
12. Croxton, K.L.; Lambert, D.M.; García-Dastugue, S.J.; Rogers, D.S. The demand management process. *Int. J. Logist. Manag.* **2002**, *13*, 51–66. [CrossRef]
13. Croxton, K.L. The order fulfillment process. *Int. J. Logist. Manag.* **2003**, *14*, 19–32. [CrossRef]
14. Goldsby, T.J.; García-Dastugue, S.J. The manufacturing flow management process. *Int. J. Logist. Manag.* **2003**, *14*, 33–52. [CrossRef]
15. Aarikka-Stenroos, L.; Sandberg, B.; Lehtimäki, T. Networks for the commercialization of innovations: A review of how divergent network actors contribute. *Ind. Mark. Manag.* **2014**, *43*, 365–381. [CrossRef]
16. Azaria, A.; Ekblaw, A.; Vieira, T.; Lippman, A. MedRec: Using blockchain for medical data access and permission management. In Proceedings of the 2nd International Conference on Open and Big Data (OBD), Vienna, Austria, 22–24 August 2016; pp. 22–30.
17. Liang, X.; Shetty, S.; Tosh, D.; Kamhoua, C.; Kwiat, K.; Njilla, L. A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Madrid, Spain, 14–17 May 2017.
18. Yuan, Y.; Wang, F.Y. Towards blockchain-based intelligent transportation systems. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016.
19. Meng, W.; Tischhauser, E.W.; Wang, Q.; Wang, Y.; Han, J. When intrusion detection meets blockchain technology: A review. *IEEE Access* **2018**, *6*, 10179–10188. [CrossRef]
20. Kim, H.W.; Jeong, Y.S. Secure authentication-management human-centric scheme for trusting personal resource information on mobile cloud computing with blockchain. *Hum.-Cent. Comput. Inf. Sci.* **2018**, *8*. [CrossRef]
21. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 26 September 2018).
22. Hurlburt, G.F.; Irena, B. Bitcoin: Benefit or curse? *IT Prof.* **2014**, *16*, 10–15. [CrossRef]
23. Nguyen, G.T.; Kim, K.A. Survey about consensus algorithms used in blockchain. *J. Inf. Process. Syst.* **2018**, *14*, 101–128.
24. Szabo, N. Formalizing and securing relationships on public networks. *First Monday* **1997**, *2*. [CrossRef]
25. Solidity. 2017. Available online: https://solidity.readthedocs.io/en/v0.4.23/ (accessed on 4 May 2018).
26. Delmolino, K.; Arnett, M.; Kosba, A.; Miller, A.; Shi, E. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In Proceedings of the International Conference on Financial Cryptography and Data Security, 18 November 2015; Springer: Heidelberg, Germany; pp. 79–94.
27. Kim, C.S.; Pack, M.Y.; Kim, J.H.; Jin, K.J. Empirical testing on distribution channels of Korea's leading enterprises. *Korea Soc. Transp. Conf. Pap.* **2007**, *57*, 219–228.
28. Jang, Y.S. Study on distribution efficiency of large-scale discount stores' marine products. *J. Fish. Manag.* **2004**, *35*, 169–191.
29. Chuang, H.M.; Chen, Y.S.; Lin, C.Y.; Yu, P.C. Featuring the e-service quality of online website from a varied perspective. *Hum.-Cent. Comput. Inf. Sci.* **2016**, *6*. [CrossRef]
30. Buterin, V. Ethereum White Paper. 2013. Available online: https://github.Com/ethereum/wiki/wiki/White-Paper (accessed on 26 Septembe 2018).
31. Korea Agricultural Marketing Information Service. Available online: https://www.kamis.or.kr/ (accessed on 24 September 2018).