Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2022, Article ID 8011525, 23 pages
https://doi.org/10.1155/2022/8011525

Research Article

Agriculture Supply Chain Management Based on Blockchain Architecture and Smart Contracts

Adil El Mane,[1] **Younes Chihab**,[1] **Khalid Tatane**,[2] **and Redouan Korchiyne**[1]

[1]*Computer Research Laboratory, Superior School of Technology, Ibn Toufail University, Kenitra 14000, Morocco*
[2]*Computer Science, Mathematics and Applications Laboratory, Institute of National School of Applied Sciences, Ibn Zohr University, Agadir 80000, Morocco*

Correspondence should be addressed to Adil El Mane; adil.elmane@uit.ac.ma

Received 7 July 2022; Revised 23 September 2022; Accepted 30 September 2022; Published 21 October 2022

Academic Editor: Imran Ashraf

Since the commercialization of agriculture technology, there has been a surge in interest in agricultural data. However, these data are notoriously chaotic, and analysts are concerned about their authenticity because there is a big possibility that others may have influenced data quality at various points along the data stream. This article suggests a new blockchain architecture to protect the integrity of agricultural data. The goal of this architecture is to provide farmers with safe storage. The agriculture data inserted cannot be modified without some rules. Many procedures are completed automatically using smart contracts to limit the danger of manipulation. One of the suggested architectures is the proof of concept. It connects a traditional farm system with the blockchain accompanied by smart contracts to facilitate the entire agri-supply chain. The conceptual architecture will eliminate the flaws discovered in prior studies. Sensors are used in this approach to provide us with environmental data. As a result, we store our data in blocks using the blockchain system. Then, we built some unique agricultural smart contracts to handle all transactions and automatize decisions based on the source code of these automated contracts. This strategy would be more efficient and secure.

1. Introduction

Agriculture is a big part of the economy of any country because it helps feed the entire population. It connects and communicates with all of the related industries. If the agriculture base is strong, it is generally regarded as a socially and politically stable society. Many modern farms make use of cutting-edge technology and scientific and technological ideas [1].

The following are some of the reasons for food supply chain problems and processing environment challenges.

The maximization of the profits relies on some farmers' vegetables and fruits with chemicals. Chemical fertilizers, insecticides, and other compounds are used in several plants and fruits.

As a result, pesticide residues in vegetables and fruits become excessive. It is a significant health risk.

Food gets contaminated with heavy metals. The irrigation water source of crops is polluted by the excessive intrusion of heavy metal elements such as lead, tin, mercury, and zinc, which are dangerous to human health.

Food additives are used excessively in food processing. Some nefarious enterprises use excessive food additives, antibiotics, hormones, and harmful substances [2].

The following are some of the most common blockchain applications [1]:

(i) Agribusiness insurance.
(ii) Smart farming.
(iii) Traceability.
(iv) Land registration.
(v) Food supply chain.
(vi) Security and safety farms.
(vii) Agricultural product e-commerce.

As a formal definition, the blockchain is a distributed ledger to share transactions or sensitive data across untrusted multiple stockholders in a decentralized network. The data are recorded in a sequential chain of hash-linked blocks that facilitate the data distribution to be more manageable than other traditional data storage formats. The blocks are verified and uploaded into the chain-like system by selected nodes via an agreed consensus protocol. This consensus mechanism allows all the parties to engage in the monitoring process when adding data flow. In addition, the duplicates of these data are stored in all involved nodes to ensure no tampering.

To make agricultural applications more efficient and reliable, we can divide blockchain applications into four categories. The first is the provenance of traceability and food authenticity. The second category is smart agricultural data management. The third category is trading finance in supply chain management. The last is the category of other information management systems [3].

In agriculture, collecting data is frequently prohibitively expensive. The blockchain provides a dependable source of truth about the state of crops, inventories, and contracts. Food provenance is tracked using blockchain technology, which aids in the creation of trustworthy food supply chains and develops trust between producers and consumers. It also enables timely payments among stakeholders generated by data changes when used conjointly with smart contracts [4].

Many characteristics of the blockchain make it unique and promising for future industrial applications. For example, blockchain is decentralized, transparent, immutable, irreversible, autonomous, open-source, ownership, provenance (authenticity and origin), and task automation.

Contract automation (smart contracting) eliminates the need for a traditional contract while improving security and lowering transaction costs. Smart contracts are designed with rules and actions that applied to all parties participating in the transaction [5].

E-agriculture, or smart farming, refers to building innovative methods to use modern information and communication technologies (ICTs), such as the Internet of Things (IoT), cloud computing, machine learning, big data, and blockchain, to move towards more feasible agricultural and farming practices. Blockchain technology in agriculture is gaining traction because of its ability to move away from the centralized approach that now governs the farm value chain. The new technologies have produced Agriculture 4.0 or smart farming [6].

Smart contracts help manage the challenges in implementing the revenue sharing algorithm and improve productivity, transparency, security, traceability, and full integration between supply chain levels.

Smart contracts are considered a flexible type of planning because they provide cost metrics that get used to accomplishing high productivity within plans for producing and delivering products in the context of current market restrictions and then executing the established programs [7].

All innovation results from an attempt to solve a problem, and blockchain technology is no exception. After learning about the origins of blockchain technology, it is evident that blockchain solves a flaw in existing centralized agricultural systems.

At the security level, we can never eliminate vulnerability; it can only be decreased and lessened. When parties sought to establish an agreement, groups have always functioned as third-party lawmakers to reduce suspicion. One party expects fair goods, while the other hopes to receive the negotiated cash. Even though the buyer and seller have no reason to trust one another, they complete the deal because they trust the third party. Blockchain claimed to overcome these issues by helping apps develop in a decentralized and safe way and ensuring some guaranteed level. One of the critical reasons for blockchain's widespread adoption was this. Implementing blockchain and smart contracts and profiting from their advantages is a big motivation to improve the agricultural system model and make it more secure.

On the blockchain, all transactions are securely recorded. The controller may use this technology with smart contracts and the Internet of Things to control the supply chain management, store farm data, and manage identity, among other things. On the blockchain, personal data are masked and need permission control to access them. Information gets stored over multiple computers (distributed ledger) rather than on a single server; this system makes it harder for hackers to alter data. By maintaining a track audit, blockchain can instantaneously trace commodities or goods, assisting in delivering proof and revealing weaknesses in any supply chain. Furthermore, smart contracts will automate transactions and enhance efficiency. Smart contracts eliminate the dependence on human intervention and ignore the reliance on third parties.

In our architectural scheme, IoT will play a role of collecting the environmental data. Smart contracts will play the role of data science analyzer, which means dealing with data for actionable insights, while blockchain records and validates data. This scheme uses algorithms created to govern interactions with various data segments. The next step of this system is to create a blockchain system automated by smart contracts and make the correct predictions after analyzing the collected data. This project will guarantee the enhancement of farm production. The application platform will allow all the participants on the network (providers, farmers, customers, and distributors) to visualize data and trace product growth. Blockchain validates data using a decentralized consensus algorithm and encryption, making it nearly hard to alter owing to the massive amount of computer power needed. As previously stated, blockchain's validated data are organized, comprehensive, and immutable.

When we talk about the tracked product, we discuss the collected environmental data in which the goods have grown. The network members will get all the growth humidity, temperature, light, and soil pH details. They have code that says "If x event occurs, perform y action." The participant will receive the updated data every period. When the customer knows all data about the product that he will buy, he will be satisfied with all this shared information and glad to be a part of this commercial deal. Also, when the farmer controls all the necessary conditions to grow the optimum quality, he will build the confidence to share information with customers and gain their trust. Another example of an advantage for the supply chain management provided by the scheme is allocating goods arrived and which container is in it. Blockchain allows tracking and storing information such as order receipts, product status, shipment details, and regulatory information to increase transparency and customer satisfaction.

Smart contracts are composed of codes that analyze collected data and display to members if the sensor results are on the optimum values. These contracts help track the product supply chain by keeping eyes on product identification and detecting the member that holds the product. Furthermore, the block cannot store the data until it is verified and validated by control members, so the case in which an entity can be malicious is significantly minimized compared to other centralized systems. In general, the truth is that blockchain systems and smart contracts can perform middleman duties independently.

Concerning the prospective research gap and research challenge, we noticed that many researchers built a scheme that consists of a single blockchain that stores data about farms, entities, products, financial business, deals, and trades all in one distributed ledger. We reckon that the data in our system need to be more structured. The research challenge is placing data related to individuals into "the user blockchain," data describing products into "the product info blockchain," and data related to deals between entities into "the transaction blockchain."

Also, the relationship between entities builds on trust, and each participant knows their responsibilities and rights. On the other hand, we have seen that some researchers do not automate specific tasks, such as detecting environmental sensor data without intervention. So after doing some studies, we have decided to work with smart contracts as a blockchain companion.

Generally speaking, the blockchain system aids by taking advantage of its solid security-protecting approach (simultaneously dispatching the last version of data to all network members). Since agricultural data are sensitive and essential, the blockchain helps to entirely transform how the information is seen by adding an end-to-end encrypted system to the blocks. It is important to remember that distributed data guard against fraud and illegal behavior. Participants in the network get secured data, and their personal information is anonymized. Attackers cannot access or alter the data because the information is not kept on a single server and the validated blocks have a very minimal possibility of being reversed.

Because blockchain employs a distributed ledger, transactions are recorded identically in many places without the need for synchronization, and guaranteeing transparency is another significant benefit (improve speed processing). Additionally, this method aids in data tracking for analysis and reveals weak links in any supply chain. Smart contracts automate transactions, speeding up processing time and minimizing human involvement. In this study, the smart contract helps analyze the values, verify environmental data, and identify the crops with optimal growth status.

In the following sections, we will discuss how we can combine these technologies to create a robust scheme. Of course, the adoption of IoT is always advantageous since the data collected from these sensors are more exact and in real time. All these elements are unexplored areas, limit the scope, and define the conceptual boundaries in our research field, which will boost our chances of having a better research study.

We will provide further comparisons between the proposed and past approaches. Previous work will be listed in the upcoming paragraphs for sure. Still, the main point we are trying to convey is that the smart contracts developed in our architecture are innovative and have the potential to obtain sensors' values automatically, such as the optimum temperature, humidity, light composure, and soil pH values. The components of the smart contract's Solidity script are considered the source of the idea's novelty. Besides, decent data structure organization on the blockchain layer constantly improves data management. These details will guide us to generate new observations or insights based on prediction while incorporating other innovative technologies like artificial intelligence.

1.1. Comparison between Existing Agricultural Schemes and the Proposed Model.

Figure 1 shows the difference between the existing agriculture supply chain (using the centralized database), a standard blockchain-based agricultural supply chain, and the proposed blockchain-based agrarian architecture.

The key reason we chose to work with blockchain and incorporate its features into our architecture was the absence of need for third parties. Additionally, the control over a decentralized ledger stays with the user rather than a centralized authority.

Another benefit of blockchain is that there are no data breaches and hacks. However, the scalability of a centralized system with a single server is limited.

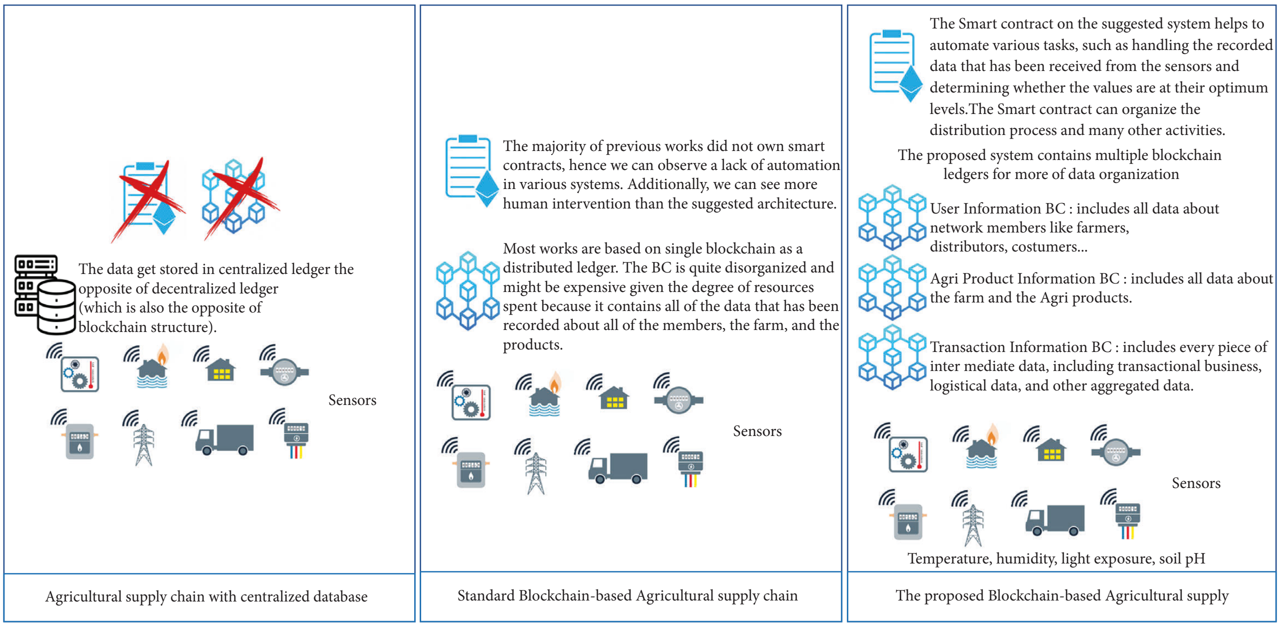
Implementing a blockchain system protects data so it cannot be changed or erased. But the recorded data in the standard blockchain architecture are not well organized, and many previous works did not use smart contracts, which facilitate some operations and tasks without the involvement of network participants.

The suggested model includes several blockchain ledgers that divide the data into user information, agri product information, and transaction information. Additionally, smart contracts are considered a significant factor in this model, which automate many tasks.

Most blockchain systems implement smart contracts provided by the Ethereum platform and its extension platform, Quorum: they compile using Solidity or Serpent into Ethereum virtual machine (EVM) bytecodes. Hyperledger Fabric and Sawtooth, the most active platforms in the Hyperledger family, use Golong, Java, Python, and JavaScript as the major programming languages for smart contract development [3].

1.2. Related Work.

The only way to verify and validate transactions in the system is to use IoT devices that are physical consortium members. With the RAFT consensus



The data get stored in centralized ledger the opposite of decentralized ledger (which is also the opposite of blockchain structure).

The majority of previous works did not own smart contracts, hence we can observe a lack of automation in various systems. Additionally, we can see more human intervention than the suggested architecture.

Most works are based on single blockchain as a distributed ledger. The BC is quite disorganized and might be expensive given the degree of resources spent because it contains all of the data that has been recorded about all of the members, the farm, and the products.

The Smart contract on the suggested system helps to automate various tasks, such as handling the recorded data that has been received from the sensors and determining whether the values are at their optimum levels.The Smart contract can organize the distribution process and many other activities.

The proposed system contains multiple blockchain ledgers for more of data organization

User Information BC : includes all data about network members like farmers, distributors, costumers...

Agri Product Information BC : includes all data about the farm and the Agri products.

Transaction Information BC : includes every piece of inter mediate data, including transactional business, logistical data, and other aggregated data.

Sensors
Temperature, humidity, light exposure, soil pH
The proposed Blockchain-based Agricultural supply

Figure 1: Comparison between existing agricultural schemes and the proposed model.

algorithm, the proposed blockchain platform becomes not only energy-efficient but also faster and scalable that can record thousands of confirmed transactions per second across multiple ledgers. Third, the blockchain with the RAFT consensus mechanism provides a transparent, secure, and trusted platform for faster exchange of all types of services among stakeholders. The RAFT consensus mechanism guarantees integrity if more than 50% of transacting nodes are honest. Several consensus algorithms are proposed for reaching a consensus among several action nodes, including proof of work (PoW), proof of stake (PoS), practical Byzantine fault tolerance (PBFT), and RAFT. The public blockchain generally uses PoW and PoS, although they lack the speed of confirmation. As a decision, our consensus mechanism decreases the use of the blockchain to connect various agri-based IoT devices. The RAFT consensus method is more convenient to employ in a private blockchain [8].

“A secure fish farm platform based on blockchain for agriculture data integrity” research [9] mentioned that smart farming necessitates scalable security. Therefore, various studies have focused on developing new paradigms based on blockchains.

The technology of SkuChain focuses on establishing direct contacts while also increasing trust in the supply chain. More than half of blockchain discussions are around retail and agricultural input and output tracking.

We can get agriculture data from many sources, such as soil sensors, weather satellites, drones, and farm equipment. It can be saved in a distributed store using the blockchain with secure long-term agriculture progress.

The distributed ledger records all of the activities of the farm. It is tied to the data collected by agricultural sensors. Furthermore, smart contracts are used to automate agriculture data processing, including outlier filtering, before generating records in the ledger. Smart contracts could activate and execute particular actions based on data recorded in the blockchain.

The blockchain has access control that determines who gets permission to reach network resources or perform actions on them.

The blockchain is a continuously expanding collection of documents known as blocks. A block often contained the hash value of the preceding block, a timestamp, transaction data, and much other information. It is impossible to tamper with the data without breaking the hash links. The blockchain network contains many peers, each of which has a smart contract.

A block must include signatures from a certain number of people. This technology eliminates the risk of data disclosure and guarantees that no unauthorized user can tamper with a transaction on the blockchain.

Blockchain functions are transaction verification, identity validation, and peer-to-peer communication. Also, some network services are available as web APIs, allowing external systems or client apps to connect.

The smart contract is distributed across a blockchain network in a single package. Once the contract is deployed, all of the smart contracts included inside it are available to applications.

The smart contract defines a collection of transactions. Participants in the blockchain network (for example, a farmer) or assets (anything of value) are used as resources (for example, water level data). The supported operations are CREATE, READ, UPDATE, and DELETE.

(i) The water level sensor can CREATE and Collect the Water Level transaction.

(ii) The farm owner can CREATE many transactions such as Predict Water Level, Energy Consumption, User Management, and Sensor Management.

(iii) The water pump has access to ALL operations for the Control Water Pump transaction.

(iv) The farm owner and the farmer can READ many transactions such as Water Level History, Predicted Water Level History, Energy Consumption History, and Water Pump History.

The following research represents the article "Blockchain and smart contract for IoT enabled smart agriculture" [10]. It mentioned that seed storage, supply stores, producers, distributors, wholesalers, and retailers are the main actors in the system. In addition, the contract deployer is another actor here. The interaction method between the actors and the system is accomplished through several components. The sections below demonstrate the role of each actor.

(i) *Contract Owner*. The contract owner has more control over the system than anyone else. The owner enters the contract into the system and checks to see if the rule gets correctly implemented.

(ii) *Seed Storage*. Seed and other agricultural products are stored in the seed storage.

(iii) *Supply Shops*. They collect and sell a significant quantity of seed, fertilizer, and other agricultural materials to growers.

(iv) *Producers*. The farmers are considered the most basic rung in production. They are in charge of all tasks relating to agricultural planting and harvesting.

(v) *Distributors*. Distributors are in charge of safely transporting crops from one location to another.

(vi) *Wholesalers*. Wholesalers buy a decent quantity of crops and agriproducts and resell them to retailers.

(vii) *Retailers*. Retailers buy commodities and products from wholesalers and sell them to consumers on a small scale in open markets.

(viii) *Consumers*. They are a large group of individuals that rely on agricultural products. They play a big part in the system by constantly creating demand.

FarMarketplace is a novel digital marketplace [6]. It fully exploits the advantages of blockchain capabilities by proposing smart contract templates between farmers, consumers, and deliverers. FarMarketplace is considered innovative in three aspects.

FarMarketplace offers three main smart contracts coded in Solidity programming language. The first smart contract is "atomContract.sol." In this contract, the user buys the contract. The funds were added to the balance. The buyer gets identified. After that, the buyer receives the object of the contract, and the seller receives the money.

The second smart contract is coded in Solidity programming language. This file creates the smart contracts, sets their value, and monitors their status. This file can also modify the accessibility from available to unavailable or vice versa. Some methods determine if the conditions are met and, as a result, generate an error.

The second smart contract is "ownable.sol."

The ownable contract has an owner address and provides simple authorization control functions. This system makes user permissions easier to implement. The ownable constructor sets the original owner of the contract to the sender account. This file also can let the current owner transfer control of the smart contract to a new owner.

The article "Securing parked vehicle assisted fog computing with blockchain and optimal smart contract design" [11] mentioned that the blockchain provides complete security for parked vehicle assisted fog computing (PVFC). As a result, PVFChain, a specific blockchain system for computation offloading, is established to record and confirm crucial information about requesters and performers.

Without any third parties, the decentralized PVFC is built by relying on a majority of consensus nodes. Smart contracts organize the posting of requests, the completion of workloads, the grading of tasks, and the distribution of rewards.

Any registered entity to PVFChain is controlled to follow the contact-based agreement via smart contract execution. Due to the inbuilt transparency and accountability of smart contracts, fraudulent requesters and performers are exposed completely. Accessible records regarding the activities of requesters are tamper-resistant because of data auditing. Finally, PVFChain protects against network vulnerabilities by supporting identity authentication, request validation, computation verification, and reward integrity.

The feasibility of PVFChain depends on the identification of critical network entities. Requester, performer, and miner are the three main network entities.

When a mobile car runs a compute-intensive application on the road, it sends a computing job to PVFChain. This previous operation triggers a smart contract. PVFChain system transmits a task to a responsible administrative agent. At the vehicular network edge, the agent organizes the offloading service. In the PVFC environment, the agent looks for a fog server near a parking lot and uses that to complete cooperative tasks.

There are various business platforms available nowadays. For example, Ethereum is a common platform that facilitates smart contracts on the blockchain. On the PVFChain, each smart contract has its address, allowing us to describe business logic for constraining network behaviors in offloading activities.

In summary, the operations of the smart contract are as follows:

Step 1. The requester sends a computing task containing task requirements and a deposit to PVFChain. After that, the assignment gets allocated to a particularized agent.

Step 2. Through the clients, performers can see the offloading requests and the work criteria.

Step 3. Some performers respond positively to the requests and upload basic task execution information with their replies.

Step 4. The agent is authorized to recognize and pick suitable performances based on prior knowledge. The agent also assigns various rewards for performers.

Step 5. The client of the requester offers attractive compensation to the performers.

Step 6. The performers select a portion of the workload to process based on the rewards offered, and the agent provides the necessary input data for task execution.

Step 7. The actors distribute computing resources to obtain output outcomes. The smart contract contains their identities, node types, a total of processing workloads, and output results.

Step 8. A specific transaction in the smart contract authorizes miners and analyzes the output results. The miners choose suitable third-party methods to present evaluation findings.

Step 9. The agent collects all the output results and aggregates them into a final result before sending it to the requester.

Step 10. Finally, the requester offers additional payments. Qualified performers get promising prizes.

The article "Smart contract-based agricultural food supply chain traceability" [12] suggests that tracking and executing transactions in the agricultural food supply chain necessitates applying the Hyperledger Fabric to establish consortium chains and smart contracts called Chaincode.

The entire process of the agricultural food supply chain from farm to fork is as follows:

(i) *Agricultural Bureau*. It is an entity that keeps track of farmers, seed information, plot information, and yield data to assure the accuracy of source data. The data are saved in IPFS, and the blockchain conserves the hash value.

(ii) *Farmer*. He is in charge of sowing crops, utilizing sensors to monitor and record crop growth in the growing environment, such as water, air, sunlight, and soil quality. The farmer saves the information about the crop growth process in IPFS. Furthermore, the farmer is responsible for generating smart contracts and storing IPFS data hashes.

(iii) *Processor*. The farmer gathers the crop and sells it to the processor, who transforms the raw crops into products for the final customer and records batch information, quantity, and other information in IPFS. The data hash is saved in the blockchain, and the data label is created and put on the product package.

(iv) *Quality Supervision Bureau*. It is in charge of overseeing qualities and conducting mandatory inspections of manufacturing companies. All the tasks help respect regulations concerning standardization and quality and penalize rule violators.

(v) *Distributor*. Before reaching the retailer, the finished product may go through several tiers of distribution. The distributor is in charge of storing and selling processed agricultural products to retailers in batches. Company information, product selling times, prices, and other data are kept in IPFS. The hash value is stored in the blockchain, just as it is for the quality supervision bureau, to ensure that the succeeding data are unchangeable.

(vi) *Retailer*. He purchases products from distributors and then sells them to consumers in modest amounts. IPFS stores on the blockchain some basic information about the retailer. For example, it stores the time of sale, the quantity sold, and the hash value.

(vii) *Consumers*. These are the people who buy and eat the finalized agricultural food, and they can get the complete supply chain information about it by scanning the barcode, RFID, or QR code on the product package, which makes tracking food easier.

The smart contracts interact with the blockchain participants. Our network participants and their objectives are listed in the "Blockchain-based soybean traceability in agricultural supply chain" research [13].

(i) Seed company maintains local records and produces seeds. These seeds get labeled by the EAN-UCC global standard.

(ii) Farmer purchases recognized seeds from seed companies, harvests the crop, and sets up the smart contract. Data are also saved on the decentralized file system IPFS by the farmer.

(iii) Grain elevator: a grain elevator is an agricultural institution. These organizations decide the quality of the product by managing the sensing data and storage duration of the harvested crop.

(iv) Grain processor purchases grain from the elevator, refines it, removes foreign matter, and produces the finished product.

(v) Distributor is in charge of the product distribution procedure.

(vi) Retailer purchases traceable items from manufacturers and sells them in small quantities to consumers.

(vii) Customer is the person who uses the product and consumes it.

Any smart contract code can be handled using the Ethereum virtual machine (EVM). This EVM is in charge of comprehending every command code and ensuring that the smart contract is executed on all nodes.

The research employs Ganache as a local Ethereum test network rather than the whole Ethereum network. The purpose of this movement is to make the test stage easier.

The following are some of the most significant smart agriculture notions:

(i) Sensors and surveillance cameras.

(ii) The network participants like farmers, suppliers, customers, and so on.

(iii) Computing equipment like microcontroller systems and cloud computing.

(iv) Applications and libraries include mobile device apps, machine-to-machine communication, API, and a blockchain web app.

The agricultural zone is equipped with an IoT controller, sensors, and cameras. The controller will encrypt the environmental (sensing) data before sending them to the blockchain network. At this point, the controller will use machine-to-machine communication to connect with the blockchain network [14].

Precise measurement is critical in agriculture. The soil type and the climate influence the monitored metrics on a field. The following are a few of them [15]:

(i) *Air Temperature and Humidity*. Seed germination gets relative to the optimum temperature because the phases of biochemical mechanisms depend on enzyme activation and hydration. Also, humidity gets defined as the water vapor pressure in moist air.

(ii) *Soil Temperature and Humidity*. As a carbon storage tank, the soil is a considerable component in irrigation scheduling. The watering period is determined by the driblet irrigation technology employed or the techniques of deep wells.

(iii) The fundamental goal of this type of irrigation is to improve plant water quality standards. Sunlight and humidity have a significant impact on soil temperature. The temperature of the soil is generally higher than that of the air.

(iv) *Evapotranspiration*. It is a component of the hydrologic process that is the most elemental. Consumptive usage includes plant transpiration and water evaporation from the soil.

AgriOnBlock [16] ensures non-repudiation and confirms the authenticity by encrypting and decrypting the transaction with the private key and public key of a cryptography mechanism like RSA.

(1) When a retailer wishes to get a product from the warehouse, it communicates with AgriOnBlock and the bank, specifying the identity product, the item code, the quantity, and the rate per unit.

(2) The retailer has adequate funds, and its credentials are verified. The required amount gets transferred from the account of the retailer to AgriOnBlock.

(3) The bank informs AgriOnBlock that the payment transaction was successful.

(4) Once AgriOnBlock receives the proof of payment from the bank, it directly notifies the warehouse to transfer the requested item in the stated amount to the retailer.

(5) The transaction gets recorded in the AgriOnBlock.

The steps for the interactions between some agricultural supply chain entities are as follows:

(1) The farmer lodges an insurance claim to AgriOnBlock.

(2) AgriOnBlock delivers the information to the insurance carrier when it gets verified.

(3) The insurance company sends the claim to a surveyor for physical inspection.

(4) The surveyor physically visits the site to do the survey.

(5) The surveyor submits a report to the insurance company, accompanied by the amount of the insurance claim for payment.

(6) The insurance company notifies the bank to reimburse the farmer after obtaining the report from surveyor.

(7) The bank pays the farmer.

(8) The insurance company informs AgriOnBlock of the transaction.

Smart contract algorithm for AgriOnBlock supply chain management:

Step 1. If the farmer does not have the ready harvest, stay on step 1; otherwise, proceed to step 2.

Step 2. If the farmer prepares a crop invoice and sends it to the retailer, the request to pick up the crop gets forwarded to the shipping company.

Step 3. If the retailer confirms the invoice received from the farmer, it will transmit a confirmation to the shipping company (and proceed to phase 4); otherwise, it will go to the error step (error code = 1).

Step 4. The shipping company picks up the produce from the farmer and delivers it to the retailer and then updates the transaction on the AgriOnBlock after receiving the invoice from the farmer and confirmation from the retailer.

Step 5. If the transaction updated by the shipping company on AgriOnBlock is verified, the smart contract between the retailer and the farmer gets executed to inform the bank to transfer the amount from the retailer to the farmer, and the transaction gets updated on AgriOnBlock; otherwise, go to error step (error code = 2).

Step 6. The retailer creates a crop invoice and sends it to the distributor.

Step 7. If the distributor confirms the invoice sent by the retailer, he submits a request to the shipping company, and then the transaction gets added to the AgriOnBlock. Otherwise, you will be directed to the error step (error code = 3).

Step 8. If the shipping company validates the bill received from the distributor, the shipping company produces the crop from the retailer to the distributor and updates the transaction on the AgriOnBlock. If the shipping company does not validate the invoice received from the distributor, go to the error step (error code = 4).

Step 9. Go to step 11.

Step 10. Error: if the error code is 1, the retailer informs the farmer of the reasons for refusing to accept the invoice. If the problem code is 2, the retailer and the farmer notify about a failed bank transaction. If the problem number is 3, the distributor informs the retailer of the reasons for not accepting the invoice. Finally, if the problem code is 4, the shipping company contacts the distributor about the reasons for not approving the invoice.

Step 11. Stop.

"Blockchain and edge computing technology enabling organic agricultural supply chain: a framework solution to trust crisis" research proposed that the physical layer, edge data layer, smart contract layer, cloud/blockchain layer, and user layer are the five levels that make up the concept. This viewpoint is from the standpoint of data flow [17].

(i) *Physical Layer*. A smart contract gets placed in this layer, consisting of numerous sensors, controllers, and IoT devices. These devices are either encapsulated with the smart contract address of the client or discovered by a discovery service. Furthermore, numerous wireless protocols such as Wi-Fi, Zigbee, or LoRa are commonly used in agricultural farms.

(ii) *Edge Data Layer*. The edge nodes make up this layer for deploying containerized microservices, data infrastructure, IoT devices, and QoS control. The edge data layer takes data from the physical layer. It analyzes, compresses, transforms, and splits the data into local and cloud ones. The data rights or the identification of the data creator is initially completed by this layer. As a result, this layer enables off-chain verification of tracking data from a cloud-based blockchain. Local servers are in charge of storing off-chain data. This testing overcomes the blockchain implementation problem, privacy, transmission bandwidth, energy usage, and latency, to name a few.

(iii) *Smart Contract Layer*. This layer is responsible for assembling a group of smart contracts. It enables effective, distributed, and heavily automated OASC workflows. Smart law contracts, decentralized autonomous organizations (DAO), and application logic contracts are included in this tier of smart contracts, which go beyond the transfer of simple currency values. The law contracts specify strict legal remedies to prevent contracting parties from carrying out their obligations. DAO is a blockchain-based community that can design a list of norms expressed in smart contract code. Each participant should respect these rules and have the right to seek recourse if the program gets stopped.

(iv) *Cloud/Blockchain Layer*. This layer combines a cloud storage repository with a blockchain-like ledger that supports three types of blockchains: public, alliance, and private. This layer takes advantage of the alliance chain to include all stakeholders in OASCs. This type of chain combines public and private chains. Cloud/blockchain layer introduces an InterPlanetary File System (IPFS) and BigchainDB. IPFS is new content-addressable storage. On each computer, the same file has the same name, and any change in the data file causes the modification in the file name. Due to data storage constraints, the process only keeps the hash value of the file content in the cloud blockchain, and the file itself gets saved at the edge. This layer also utilized BigchainDB, a data storage and search engine, to suit the query specifications of files.

(v) *User Layer*. The user layer is the main gateway for anyone interested in tracking organic products or maintaining OASCs. The blockchain ecosystem gets linked to this layer via the blockchain bridge, which resembles an Ethereum bridge Metamask and is available as a browser add-on. Furthermore, this layer offers a variety of APIs.

The article "Blockchain for Internet of Things: a survey" [18] mentioned that the Internet of Things (IoT) is a significant part of this revolutionary transformation by bridging the gap between the physical industrial system and the cyber-physical system (CPS). Because of its increased connectivity, safety, protection, dependability, and flexibility, blockchain is essentially the ideal companion to IoT.

IoT devices are capable to interact with one another without the assistance of a reliable third party. Smart contracts can help us reach this flexibility. Contract provisions included in smart contracts, in particular, will be done automatically in a specific circumstance.

Many IoT devices make up the perception layer, such as sensors, QR code tags, RFID tags, smart meters, actuators, regulators, and other wireless tools. These gadgets can perceive and gather information about the zone.

IoT gateways, Wi-Fi access points, small base stations, and macro-base stations may link with various wireless sensors, RFIDs, regulators, and other tags to establish an industrial network at the communication layer. Different communication protocols facilitate the connection, including Bluetooth, low-power wireless personal area network (LoW-PAN), near-field communication, and wireless highway addressable remote transducers.

Concerning traceability, a timestamp gets appended to each transaction stored on the blockchain. After examining the blockchain data with matching timestamps, users may quickly check and track the provenance of the last data.

Integrating IoT systems with blockchain technologies (such as smart contracts) can strengthen the network's security by continuously refreshing IoT device firmware to fix security flaws. However, developing smart contracts is dynamic and involves several agreement cycles. Meanwhile, it also involves several groups, including stakeholders, regulators, and software developers.

Software-defined networking (SDN) technology can deliver distributed IoT with adaptability. Networks may now be more agile and flexible to serve different application and performance needs thanks to SDN technology and network function visualization (NFV). The mixture of blockchain and

SDN can eliminate the drawbacks. However, various conditions in the hybrid environment are required to enhance and deploy network and computer assets.

One of China's famous supply chain architectures implements both RFID and blockchain. Data on the food supply chain's tracing are maintained by this approach. The research demonstrates how blockchain might sweeten food quality by making traceability of goods available. Also, situations showed that customers could follow the entire product manufacturing process if blockchain formed in the food supply chain.

The research "The performance evaluation of blockchain-based security and privacy systems for the Internet of Things: a tutorial" [19] discussed security levels and how we can implement blockchain systems and IoT in one architecture.

The validator gathers new transactions, checks that they adhere to the rules, and then adds them to the block to prepare it. After several durations, the network participants can no longer change the information stored in the blocks. When a transaction appears in a validated block, it is regarded as proved. Each block has a unique hash code stored in it as its identification.

The application of blockchain technology in IoT agriculture concentrates on three aspects:

(i) Helping farmers track down the source of their fresh produce.

(ii) Enabling peer-to-peer agro-based transactions via smart contracts.

(iii) Ensuring the integrity of agriculture data by boosting sales of fresh vegetables and addressing the problem of environmental pollution.

Given the overall mobility and group-based behavior, IoT devices in blockchain applications furnish meaningful confidentiality.

The four levels that made the software-defined network (SDN) architectures and network function virtualization (NFV) are the perception layer, data plane, virtualization and control plane, and blockchain layer.

Data collection from the environment completes at the perception layer. Data flow to and from the perception layer via IoT bridges is achieved by the data plane. Resources are typically assigned to controllers using the virtualization and control plane. The blockchain layer's duties include data storage, broadcasting transactions, and P2P network management via a consensus mechanism.

We can explain some keywords that are required to be defined.

Ganache is a private Ethereum blockchain that executes scripts and tests, builds smart contracts, and creates apps. The Ganache testbed is a blockchain testing platform designed to execute data trades and assess the charges of smart contracts. Both desktop program and command-line tool are available.

A decentralized framework called Ethereum Geth uses the Ethereum protocol to operate smart contracts. The private blockchain network and smart contract deployment are carried out via the Geth Ethereum client using the Go programming language. In addition, for calculating the time required to pack data into the blockchain system, Ethereum Geth is employed.

The Rinkeby Ethereum testnet is a development and testing condition for Ethereum, a proof-of-authority consensus-based platform. The efficacy of the blockchain-based fog computing system is assessed using this testnet.

For Ethereum, Truffle is a development and testing environment that includes an asset workflow and network management for deploying public and private networks.

Remix Ethereum IDE is a browser-based compiler that helps users construct Ethereum contracts in the Solidity programming language and inspect transactions.

The article "Security challenges and opportunities for smart contracts in Internet of Things: a survey" [20] says that the smart contract system is still not advanced or reliable. Distributed autonomous organizations (also known as DAO), the largest blockchain initiative to be crowdfunded with assets worth roughly USD 60 million, are responsible for the most well-known smart contract accident. The DAO was hijacked in 2016, causing a significant loss because of a recursive call that the attackers had cruelly changed its smart contract. Given that research on blockchain-based IoT has been gaining steam recently, it is critical to investigate the potential threat as soon as attainable.

Another frequent security issue with smart contracts is programming vulnerabilities, particularly with a new language like Solidity. Hackers would likely use these flaws to target IoT and smart contract systems.

For instance, the unchecked call/send functions return a Boolean result to represent the success or failure of the call. The transaction performing these functions will not get rolled back if the external call fails, an error will arise, and there will not be a rollback if the return value has not yet been under complete verification. As a result, the attacker gains illegal entry into the smart contract's protected operations.

When Solidity types are used for direct calls, the calling party must define the interface of the recipient and cast it to the recipient's address.

As an integer variable may only represent a specific range of values, the EVM provides a fixed-size data format for integers.

Every Ethereum smart contract has an address object and can contact other addresses. The address of the account that initially made the global variable tx.origin returns the query or transaction. This variable fully traverses the call stack. Utilizing this variable to approve IoT users is prohibited as it increases the risk of phishing attacks.

Through constructors in smart contracts, programmers may initialize contract objects. However, since the constructor's name differs from the contract's name and is a public type, anybody may call the absent constructor due to its vulnerability. Furthermore, constructors get often used when initializing the contract to specify the administrator's address, the number of tokens, and other details. As a result, several parts of the IoT infrastructure, such as network access and token control, will be essentially destroyed.

Many owners' interests have been gravely impacted by the abovementioned problems, particularly in modern IoT setups. Thankfully, other academics are now looking at ways to address these problems. They looked for solutions such as fixing block timestamps' dependence, the inherently vulnerable particularities, and security audits for programming vulnerabilities (using SCaaS, for example, a smart contract audit engine via signature matching and machine learning algorithms).

Smart contract encryption is a brilliant concept to ensure the privacy and security of the private data used in smart contracts. For IoT service providers and customers, the transaction details of the trade entities, the contract code, and the blockchain processing mechanism are precious. Therefore, a more dependable and cost-effective IoT system is predicted in the real world due to the adoption of privacy-preserving smart contracts. This kind of security is achieved via various encryption mechanisms, including TEEs.

Adopting appropriate communication protocols is a valuable element in increasing the effectiveness of data transmission for sporadic IoT networks. Named data networking (NDN) is a new form of network design that includes useful features.

Because of its unique properties like data immutability and transparency, blockchain can accomplish tasks to make clinical trials traceable and increase public confidence in an open and equitable process, including all stakeholders. The smart contract safely automates these clinical trial processes while ensuring traceability and preventing a probability of reconstruction. The viability of the intended solution depends on a proof of concept of Hyperledger Fabric in clinical management research for numerous clinical studies. This model [21] is a Hyperledger Fabric type of blockchain with permissions, smart contracts, and good security performances.

Clinical trials produce sizable clinical research data to establish novel medications, devices, and medical or surgical procedures. Investigators take vital signs, variations in symptoms, and side effects in the study of all these recorded data. In general, evaluating the data quality is a difficult task and keeping the data quality at an acceptable level needs knowledge about the business and data modeling.

Many stakeholders are certified with blockchain technology certification to ensure secure data transformation across many parties. Without a third party, clinical trial procedures are automated using a smart contract.

Numerous clinical trial-related service scenarios are demonstrated to show how the smart contract can manage tasks such as medical data gathering and audit queries. Using Hyperledger Fabric, a proof of concept shows how well the suggested solution works by carrying out the connected processes to clinical trials among various organizations for various clinical trials. Additionally, a web-based application lets users communicate with the blockchain platform.

The model's architecture includes three layers: the physical layer, which contains all the necessary equipment, the medical pillbox, and blood analysis tools. This layer is connected to the service layer, composed of the blockchain network and its services such as smart contracts, consensus algorithms, distributed ledger, and certificate authority. The ledger maintains replicated and shared data across the whole platform. The smart contract defines the business logic concerning all operations of clinical trials. The last layer is the application layer, which manages devices, eCRF, and users and deals with audit queries.

All transactions are generally available to apps after the smart contract launches and configures the network. Remarkably, the network administrator can initiate or pause the network and its functions. By calling the relevant transaction, it can manage a subject's profile. Additionally, the smart contract provides a specific rule list to determine whether or not a user is permitted to access or alter network resources.

The experiment starts utilizing ten clients in a two-channel network of 8 organizations and 12 endorser peer nodes using the simulation program Hyperledger Caliper. The ordering service is in solo mode, and the block size sets to 10 transactions per block, creating a new block every 250 ms.

Though it includes the medical area, this model is the closest research to my architecture. A multi-blockchain system with smart contracts and a physical layer consists of the tools required to collect priceless data, such as pillboxes, blood pressure monitors, and airflow sensors. This medical research and our research share many similarities. However, while our research focuses on agriculture, the other is in the medical field.

According to Hang et al.'s [22] research results, the throughput, latency, number of transactions, and scalability were some of the indicators used to analyze Hyperledger Fabric's efficiency. The increased transaction has a considerable influence on network performance, notably latency. However, the throughput closes on zero as the network hits its capacity. The authors tested the effectiveness of Hyperledger Fabric in terms of transaction throughput and network latency using various scenarios. Based on the network configurations, performance constraints vary somewhat at each level. In these cases, the effects of factors like batch-timeout, batch size, and the number of peers are examined.

Node.js is the programming language utilized to develop this blockchain, with smart contract implementations and a maximum of 5 clients or network participants. The block contains a maximum of 10 transactions, and its frequency is a maximum of 250 ms.

This study introduces a unique fuzzy logic-based transaction traffic management method that helps boost blockchain speed. The model automatically carries out various operations on transactions received following current network circumstances; the fuzzy controller gets embedded in the smart contract. Without the involvement of a third party, the fuzzy-based transaction traffic controller in the smart contract may autonomously manage the traffic flow following the observed network circumstances.

The system architecture comprises the Hyperledger Fabric network, the admin, a blockchain adapter, a benchmark database, and a transaction traffic measurement analyzer. The distributed ledger and the smart contract get duplicated by different peers that constitute the Fabric network. The administrator can set up network files and benchmarks for performance analysis. A network configuration file summarizes the tested system and the conditions for connecting to the network.

A benchmark configuration file defines the user-specified test files and the performance benchmark workload. The client where the workload occurs transmits transactions to the blockchain adapter, receiving them and sending instructions to start the blockchain. By calling the smart contract functions, many clients can send transactions to the network and receive replies to those transactions. The benchmark findings are stored in the benchmark DB by the transaction traffic measurement analyzer once it has read predetermined statistics. The fuzzy controller modifies the transaction approval rate by evaluating the acceptance rate, transaction throughput, and transaction delay.

Both transaction throughput and latency are the fuzzifier's input parameters. The inference engine assesses rules. The acceptance rate is transformed into non-fuzzy values by the defuzzifier. The transaction control module obtains the output value, which then modifies the transaction acceptance rate. The procedure gets repeated, allowing the Fabric network to keep a proper degree of transaction processing capabilities dynamically. Transaction throughput and transaction latency are specified as the fuzzy variables for membership functions on the approach, and the acceptance rate is evaluated as "low," "acceptable," and "high."

The network model of Kumar et al.'s [23] research has eight core elements. Trusted authority, IoT devices, unmanned aerial vehicles, intrusion detection systems, cloud servers, InterPlanetary File Systems, smart contracts, and blockchain networks are these eight elements.

Before installation, the trusted authority must register IoT devices, unmanned aerial vehicles, and smart contracts. Using the established session keys, the authentication and key management phase of the Internet of Things (IoT) begins with mutual authentication and key agreement between two IoT devices, between an IoT device and any associated unmanned aerial vehicle, and between associated unmanned aerial vehicles and the cloud server.

The safe contact between the involved entities is maintained throughout this phase. The IoT device installed in each flying zone has the potential to extract agricultural values. Unmanned aerial vehicles (UAVs) are connected to each flying area to gather data from IoT devices. The cloud server saves transmitted transactions. In further detail, every cloud server mines valid transactions using smart contracts. This process keeps them in IPFS and maintains the returned transaction hash in the global blockchain network.

Each miner packs the IPFS hashes of the validated transactions into the current block while generating the block hash and Merkle root for the next block.

This model has specific verification procedures to guard against different kinds of threats. The attackers impersonate a real user by creating a temporary identification and a partial private key. However, a session-based technique confirms each device's identification. Access rights are given if all credentials match; connections are also promptly terminated. As a result, this strategy guards against spoofing attacks.

The attackers may be insiders and have access to all credentials, including timestamps, pseudo-identities, and IoT device identification. However, the model can only grant access following session-based entity verification. Thus, the method avoids insider attacks by not allowing access without authorization.

With the strikes of man-in-the-middle (MITM) through unsecured channels and communications, the attacker may be able to obtain information about IoT devices. Attackers may provide information to UAVs, so they can carry out specific tasks. The UAVs, however, verify the session and look up the timestamp. As a result, the attacker is unable to conduct the MITM threat.

Fertilizers are crucial in smart agriculture by improving profit and decreasing waste. This paper [24] discussed the research difficulties of monitoring and managing farming land by enhancing crop yield and lowering resource, energy, fertilizer, and human interaction wastage. Information from the agricultural field is gathered using the multi-modal sensor. IoT-based agricultural systems assist in minimizing human contact and developing new valuable methods.

Any farmer may join the blockchain network without permission, a type of permission-less programmable blockchain, and they can carry out numerous transactions in a database. Different users connect to knowledge support systems through an intelligent interface device. Data about the land environment is gathered from various sensors and sent to the database server in the cloud environment. The MYSQL cloud database stores the acquired data. Data are compiled by the system at multiple intervals and sent to a cloud database. The machine learning algorithm creates knowledge patterns.

Every edge node in the network has a unique identity on the Hyperledger blockchain. The member service provider (MSP), which issues cryptographic certificates using a public key primitive, issues certificates to every network participant. The user of the network receives a login and password. The user gets enrollment and transaction certificates to send transactions. In this approach, the chain code includes the business logic that establishes the state of the transaction. The ledger states that matching bock numbers get stored in the state database. Hyperledger offers a secure, scalable, and adaptable blockchain technology for agricultural knowledge systems.

2. The Research

All old concepts are in service of the suggested theoretical architecture (see Figure 2). It is a multi-layer design, which is the cause why it will be easier to control. The Internet of Things (IoT) layer is the first. It consists of essential sensors like humidity, temperature, pressure, acceleration, and other variables connected to ARDUINO boards. Bluetooth or a wireless network is used to communicate between the devices. The Arduino features will provide us with data, which will be exploited in the subsequent layers. The first layer can supply us with reliable environmental data.

The blockchain layer contains three particular blockchains, and the agri-product information blockchain is the

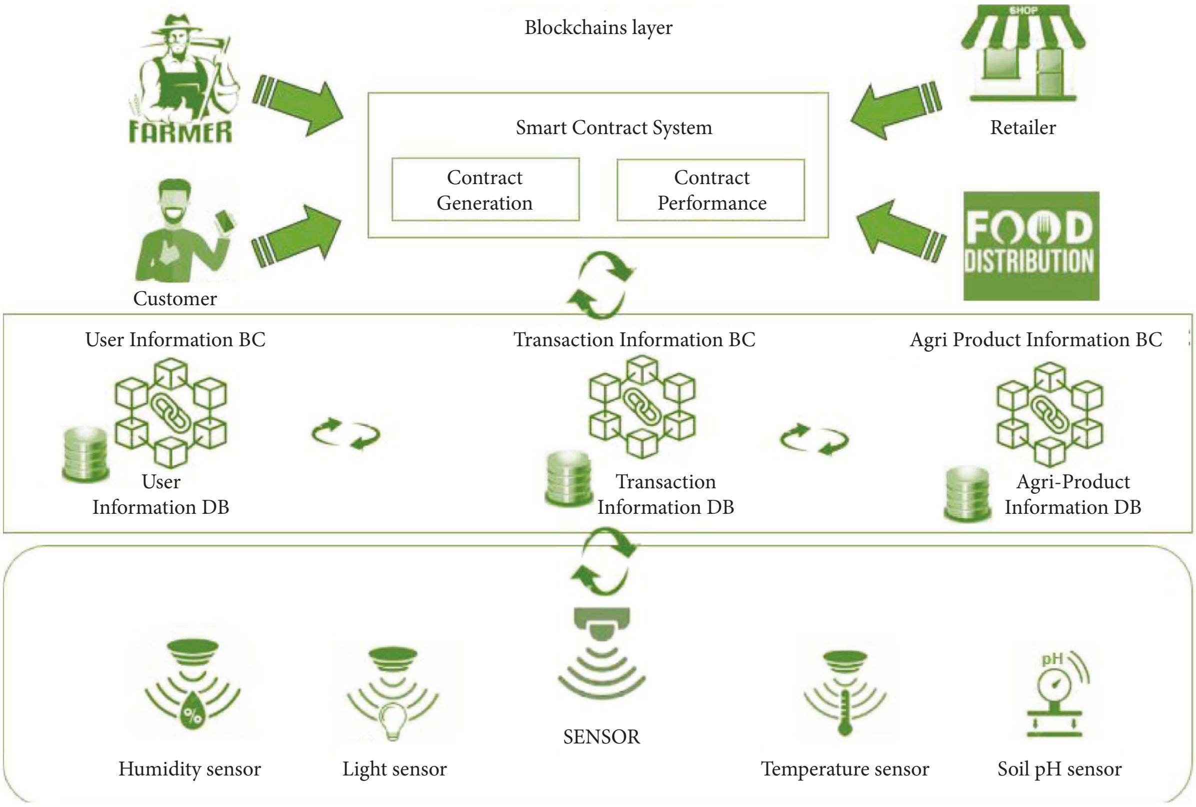


Figure 2: The proposed smart contract and blockchain architecture.

first. The user information blockchain is the second one. It has all the information about the network participants.

The transaction information blockchain is the last one on the list. It holds a wealth of information, including user personal information, intermediate information, transactions, logistics, and agricultural product data. Smart contracts are used in this paradigm. Those automated contracts use the highest level of data encryption currently available in the security industry. As a result, it will confirm the quality of the decision and guarantee clarity and effective communication between entities [25].

Many sides will attempt to check the average transaction fees because the whole system comprises a multi-blockchain system. But let us see the other meaning of the blockchain model. The suggested architecture concurrently distributes secure data to all network participants. It is regarded as significant advantage that can increase the efficacy agricultural system.

The block will be full of transactions containing recorded data, and we will get multiple verified transactions at once. Combining numerous transactions into a single block each period is cost and space-efficient. So, batching allows for reducing per-transaction fees by aggregating various transactions into one. Following this protocol benefits you and keeps the fees low across the board for everyone. We can use cryptocurrencies for frequent transfers across platforms if we want to add crypto money to our model. Their transaction fees are virtually nothing, for example, Ripple or Litecoin, because generally, cryptocurrencies are constructed for lowering or eliminating fees.

The research does not focus on cryptocurrencies or transaction fees because the goal of our blockchain is just storing and transmitting data simultaneously to members, and that is all. But, if it is obligatory to mention some info, in this case, everything on Ethereum's transaction fees was based on "gas," and the payment system will adopt the Ethereum payment process based on the calculated procedure. In 2021, the average bitcoin fees were between 2 and 5 dollars. The average ETH transaction fee ranges from 2 to 7 dollars or 0.00056 to 0.002 ETH.

The block size is crucial because it can significantly impact the speed and capacity of the network. After discovering some related research, the average block size needed in our model over 24 hours is between 0.75 and 1.25 megabytes. The average transactions per block required in our model over 24 hours are between 1200 and 2200. The average number of payments per block needed to get seen in our model over 24 hours is between 2500 and 5500.

The members of this system are farmers, providers, distributors, retailers, clients, and controllers that help as a processing node for validating transactions, for example, to administrate the data provided by sensors and check if data (temperature, humidity, light exposure, and soil pH) are reasonable. There is no problem with hardware devices. Controllers can check the data traceability and the warehouse stock and quantity.

The class diagram in the Unified Modeling Language (UML) (see Figure 3) explains the structures, classes, attributes, methods, and relationships between objects to provide extra information about our model. Class user stores information about the network's participants, including farmers, distributors, and customers. The members who are on the blockchain network can visualize the database as well as place orders for the products they desire. The sensors send environmental information to ProductStatus, providing values and conditions to determine whether they are at their optimum level or not. These products belong to a unique category or sub-category. We added a class named OrderDetails since the order class has a lot of data.

This use case diagram (see Figure 4) represents the actors in the system model. It displays the context and requirements of our system. It functions as an outsider's perspective on a system that defines how users engage with its services. The farmer, distributor, and customer types of users are described in the figure. For example, farmers may provide a product that persuades a consumer to place an order for it. Distributors transported the goods from the farms to the consumer. The user can view a wide range of information on crop growth conditions using the soil pH, humidity, temperature, and light exposure sensors.

The food supply chain can be used to monitor a product from its raw material (e.g., tomato) on the farm to the finished product (e.g., tomato sauce) on the shelf in the retail store. This solution depends on many Solidity files. The first file contains Farmer Struct (Struct types are used to represent an organized structure). This structure is composed of the ID and the name of the farmer. The second struct is the expected yields which contain the location of the land, the corps, the amount, the estimated price, and the expiration date. The deal struct is the third one, containing the terms, the amount, and the addresses of both the farmer and the customer. This contract can control farmers, manage their yields, organize deals, and recharge wallets. The farmer does not need to wait for a bank loan or other types of financing to get started. The consumers could provide the fund with zero interest. Furthermore, the farmers are not susceptible to lengthy and complicated bank financing procedures.

The food supply chain is controlled by the second file, which contains three Solidity structs. The first is the goods quality report, which includes the address of the inspector, the quantity, the sample size, and any remarks about the goods. The processor report is the second struct, containing the notes, shipment data, and the received quality report. The last struct is the retailer report, which comprises the name of products, the raw material, remarks, the manufactured date, the quantity produced, and the processor report received earlier. Companies can use the system to track unsafe products back to their source and discover where they got dispersed. These advantages can save lives and money by preventing diseases. In addition, no one can tamper with the details of the quality reports (which get stored on the blockchain network) for their gain. The history of some products can get tracked in real time from their inception to their current state. Consumers may obtain high-quality items at lower prices because they are sponsoring the crops from the start. Consumers in the low-income category can fund crops according to their requirements and avoid market volatility in product costs. There is no need for vast farmlands. Even small-scale and family farmers can sell their products and make more money.

On the distributed public ledger, farmers can list their prospective crops. Consumers can look over the information and assess the credibility of the farmers based on previous cultivation and supply. This system establishes a transparent and tamper-proof digital market platform for farm products. As a result, the contract can be reached between the farmer and the customer, allowing the consumer to support particular crops or fields and acquire the production or a profit percentage of its market value. Based on previous agricultural production experiences, a ranking system gets adopted to enhance farmer and consumer trust. Eventually, the system will create a decentralized agro-market where farmers can raise funds for production while also having clients buy their produce. Alternatively, customers can secure quality products at a lower cost by investing in crops early. Customers and farmers will benefit from establishing a trusting atmosphere for future collaboration. Farmers will make the most money from their produce, and the investors (consumers) will be able to feed their families high-quality food.

Next, there is the ownable contract, which has an owner address. It provides sample authorization control. It makes user permissions easier to implement. The OWNABLE constructor sets the sender as the OWNER (the user who owes the product). The existing owner can also transfer the ownership of his contract to a new owner under this contract. The migration file facilitates the distribution of contracts to the network and serves as a staging area for your deployment activities. They get developed with the notion that your deployment requirements will evolve simultaneously with the blockchain evolution.

Additionally, we can embrace the product delivery file and drop it on our architecture. The food delivery file is a smart contract for a decentralized food delivery service.

The system uses some features of Solidity code, such as the unsigned integer UINT, which corresponds to the product ID and cannot have a negative value. The mapping is used to organize the order of products. The contract allows for the creation of products and the option of receiving them. Various functions check the contents of the smart contract, which the client will review and trigger if it is correct. Only customers will be able to inspect it, and only customers will queue outside to obtain the number of products. The contract adds or removes traces from a product and calculates a total of product traces for statistical purposes. The system also manages and controls temperature, retrieves an array of temperatures, and returns the number of traced places.

We must not overlook the contract that entails self-monitoring the humidity, temperature, soil Ph, and light exposure. Also, if these environmental data are beyond the threshold, below the threshold, or ideal, this file detects violations and triggers alarms.

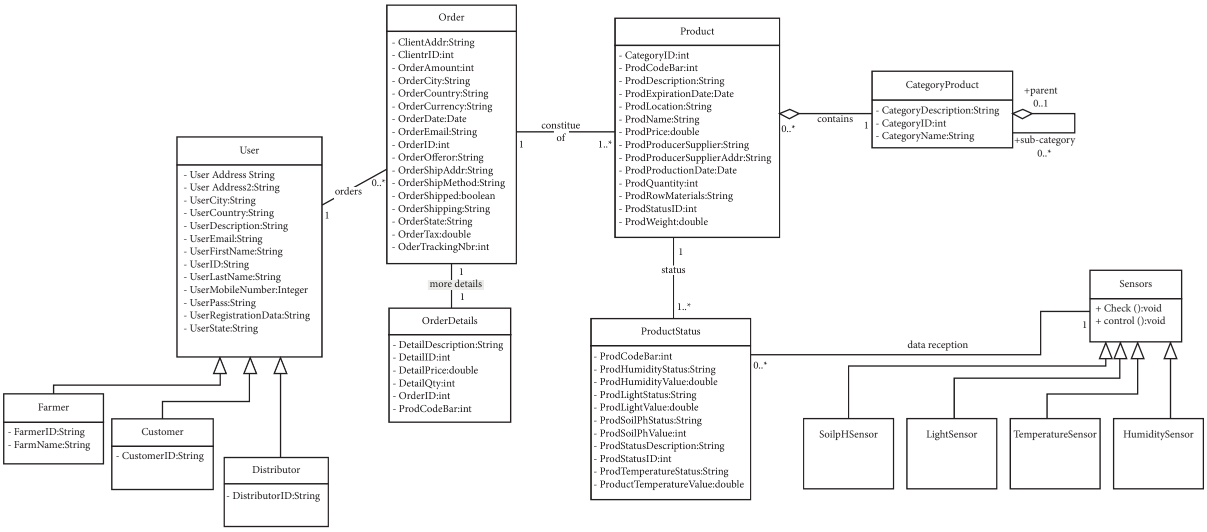


Figure 3: The class diagram of the proposed model.

The next step is upgrading the model to Ethereum 2.0, an update to the present Ethereum blockchain. It intends to improve the Ethereum network's speed, effectiveness, and sustainability by boosting the number of transactions. When compared to the prior version, it features several significant structural transformations. "Proof of stake" and "sharding" are these effective modifications.

Validators, rather than miners, are used in the proof of stake consensus method. Their primary function is to propose new blocks and offer computer power, storage, and bandwidth for transaction validation. Validators are paid in ETH regularly (deposit contract of 32 ETH to validators).

Sharding is the process of breaking up a single blockchain into numerous shards. As a result, the network becomes more solid since a single validator does not need to handle the entire architecture.

The consensus process type is the main difference between Ethereum 2.0 and its predecessor. Ethereum 2.0 employs a proof of stake (PoS) method. In contrast, Ethereum utilizes a proof of work (a power procedure in which miners decode complicated mathematical challenges using high processor quality and computer equipment).

Because Ethereum 2.0 will use fragment chains, it can process up to 10,000 transactions per second, whereas Ethereum can only handle 30.

Because the cryptographic approach is more straightforward in Ethereum 2.0, it consumes significantly less energy. Furthermore, because of its effectiveness, the improvement is projected to lower transaction fees inside the network, allowing for smaller transactions.

Agricultural production is influenced by various external elements such as climate and harvest quality. The system links all of these participants with the end purchasers, allowing anyone to intercommunicate in the profits while ensuring that customers get their healthy crops. Ethereum 2.0 can improve supply chain transparency and assist producers and customers in eliminating inefficient workflow. Validators can provide a perfect environment from the farm to the market. As a result, the goods' transaction costs will be decreased. The proof of stake can easily locate and track a product from a local store. This mechanism can also determine the farm that provided a crop as well as the date of harvest. The sharding process in Ethereum 2.0 can allow us to maximize the blockchain organization level by classifying user information from product information. The mechanism did not design validators in Ethereum 2.0 to resolve mathematical problems to validate transactions; instead, they must check data integrity.

A technical stack, sometimes referred to as a technology infrastructure or solutions stack, is currently crucial for creating scalable, maintainable applications. A technology stack is a collection of technologies layered on top of each other to develop any application. The technology stack influences the sort of apps you may design, the degree of customization you can make, and the resources you require to build your application.

Proof of concept (PoC) is an essential component of every software project since they reduce possible risks and demonstrate the project's perspective.

Proof of concept should be the foundation of every software development process since it has several benefits for projects. PoC takes care of everything, from reducing failure risks to evaluating the possibility of scalability. To improve the overall research, we have collected all the advantages of this procedure.

Risk minimization is by far a proof of concept's most significant benefit. The development team analyzes potential problems before beginning the procedure.

It is just as crucial to plan your scaling approach as it is to release a top-notch product. In the long term, you need to know how to build your solution technically and commercially.

During a seed round, supporting the idea with concrete evidence of its potential is essential. Nobody pays for abstract ideas.

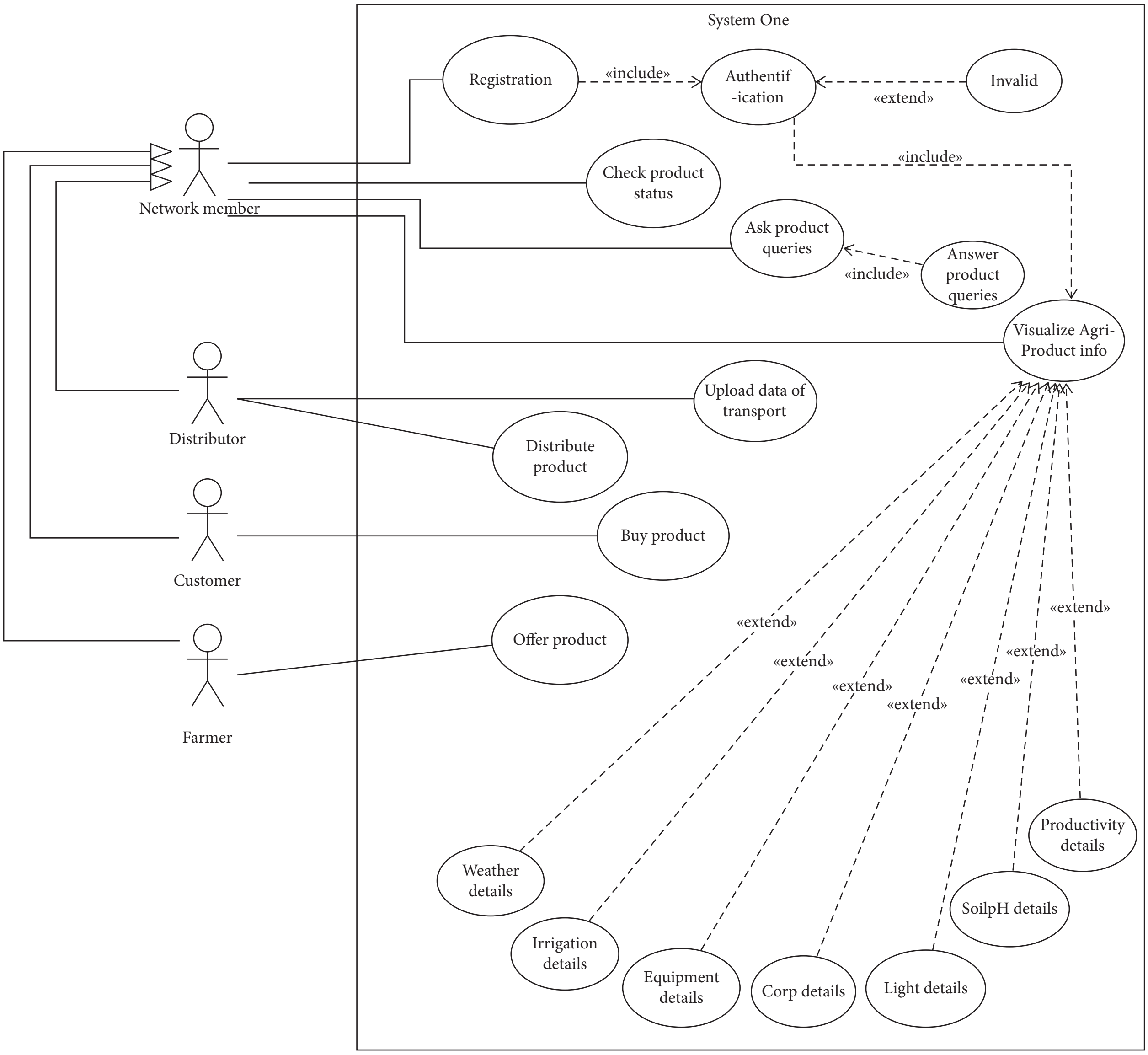


Figure 4: Use case diagram of the proposed model.

Building a PoC to test concepts in actual settings and obtain early feedback involves selecting the right technology stack and creating the architecture.

Figure 5 shows the technology stack of the proposed model.

The physical or infrastructure layer assists in gathering environmental data, such as soil pH, temperature, humidity, and light exposure measurements. We also have nodes, which are network members. A node is any device that has permission to access blockchain network.

Blockchain, as a concept, organizes data into user data, agricultural product data, and transaction data. Blockchains employ consensus methods to make sure the nodes reach an agreement. It is a fantastic procedure to increase the network's efficiency by introducing a new degree of dependability and achieving information security.

Large portions of the model are at the level of smart contracts. The middleman got just eliminated. You will not have to worry about trust issues since there are standards protecting everyone's legal rights. Since everything gets computerized, higher authorities cannot have any impact, and the procedure is quite open.

The Kaleido interface serves as a functional testing platform. It assists in facilitating access to the decentralized ledger. The users receive great help from this layer. This platform enables access to all data. Peer-to-peer server networks allow users to connect to the blockchain network.

We have selected the blockchain network and its tools/services. The system software, database, server, and frameworks comprise most of the backend or server-side technology stack. These technologies are appropriate for our planned research. For the automation of various functions, we also have the opportunity of dealing with smart contracts.

The front end of choice for most online apps stays the same. The research will use a cutting-edge web application

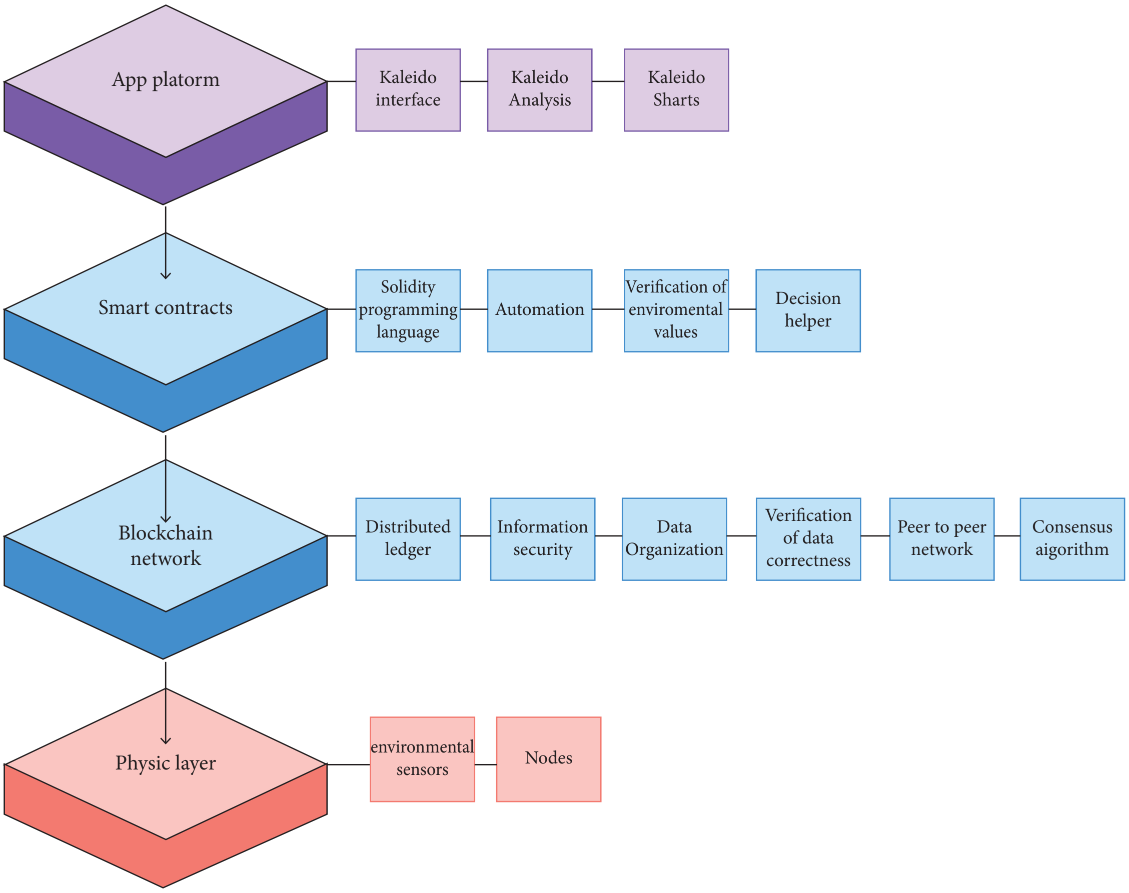


Figure 5: The technical stack of the proposed model.

built on the foundations of HTML, CSS, JavaScript, and TypeScript to deliver additional functionalities.

Programming languages handle the crucial business logic required by the applications. It is necessary to utilize the Solidity programming language and create source code that will automate various functions, such as identifying the circumstances of crop growth. However, no technology can provide complete 100% security. The blockchain network consistently adheres to best practices to guarantee the highest data and transaction protection level and reduce risks.

Future testing will focus on the new Web 3.0 blockchain technology stack, which differs from earlier iterations. However, transitioning from client-server architecture to a decentralized web will not be radical. The changes are significant and distinct.

Web 3.0 blockchain technology stacks are still developing and maturing step by step. Even though they are safer, they are often a little bit slower. As a result, the transition would initially include building a partially decentralized network before going decentralized.

3. Testing and Analysis

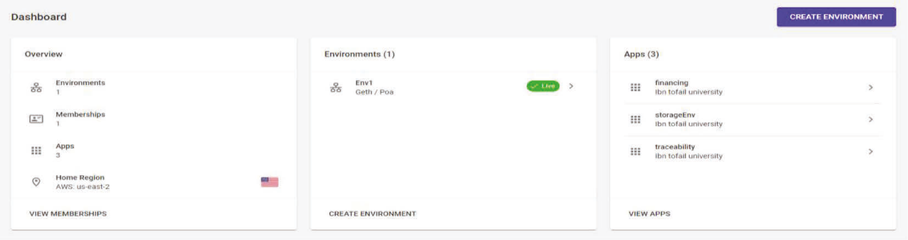
The figures depict the tested blockchain system. The tested architecture contains one environment, one member, and three smart contract script codes (finance, storage environment, and traceability).

Figure 6 depicts the dashboard system, which allows the user to learn about the testing environment. Among these data, we can find information about network participants like the membership name, the membership ID, the primary contact e-mail, and the joining date.

The KALEIDO organization issues a membership certificate with a common name and a serial code. In this situation, the dashboard shows a list of applications directly connecting to our blockchain system, like our smart contract script codes.

Figure 7 shows the current condition of the blockchain system, such as the creation date, block height, network participants, release version, and protocol utilized (Geth/Poa). We established two nodes (node 1 and node 2) for testing, and one of them is inserted by smart contract codes. There is also the node that has the name system monitor. These nodes have all got started.

The details of the block node injected with the smart contract script are shown in Figure 8. Among these details, we can find the node name (node 1), the node ID, the node size (small), the owner, the status (started or paused), and the AWS region because we have used the Amazon Web Services, along with the date it was formed and the date it was last modified.



Dashboard

Overview

Environments
1

Memberships
1

Apps
3

Home Region
AWS: us-east-2

Environments (1)

Env1
Geth / Poa
✓ Live

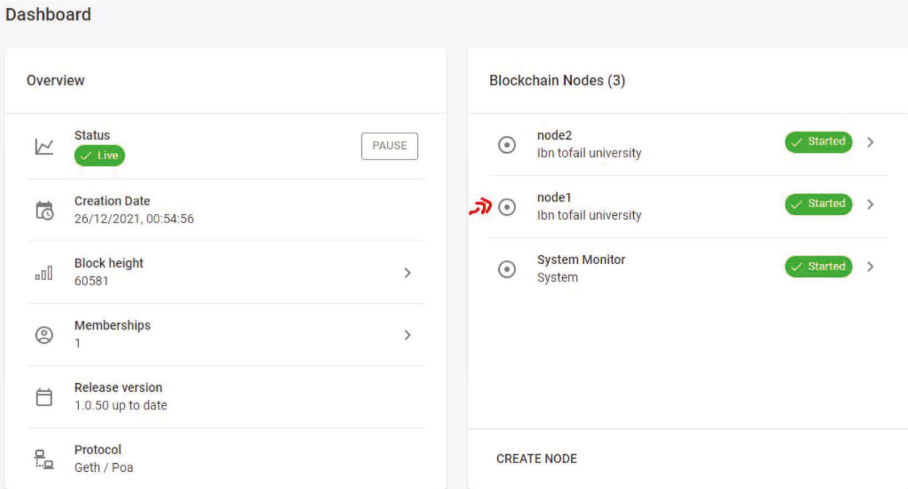
Apps (3)

financing
Ibn tofail university

storageEnv
Ibn tofail university

traceability
Ibn tofail university

Figure 6: Characteristics of the testing environment.



Dashboard

Overview

Status
✓ Live

Creation Date
26/12/2021, 00:54:56

Block height
60581

Memberships
1

Release version
1.0.50 up to date

Protocol
Geth / Poa

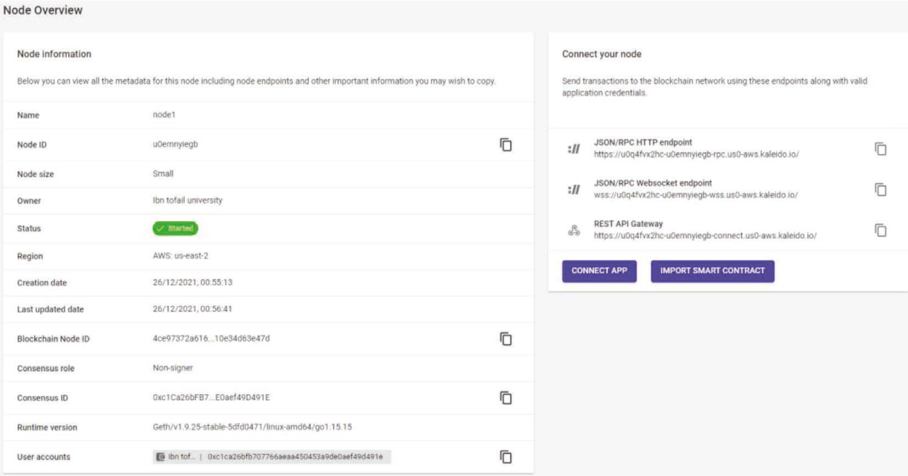
Blockchain Nodes (3)

node2
Ibn tofail university
✓ Started

node1
Ibn tofail university
✓ Started

System Monitor
System
✓ Started

Figure 7: Implementing the smart contracts on node 1.



Node Overview

Node information

Below you can view all the metadata for this node including node endpoints and other important information you may wish to copy.

Name	node1
Node ID	u0emnyiegb
Node size	Small
Owner	Ibn tofail university
Status	✓ Started
Region	AWS: us-east-2
Creation date	26/12/2021, 00:55:13
Last updated date	26/12/2021, 00:56:41
Blockchain Node ID	4ce97372a616...10e34d63e47d
Consensus role	Non-signer
Consensus ID	0xc1Ca26bFB7...E0aef49D491E
Runtime version	Geth/v1.9.25-stable-5dfd0471/linux-amd64/go1.15.15
User accounts	Ibn tof... | 0xc1ca26bfb707766aeaa450453a9de0aef49d491e

Connect your node

Send transactions to the blockchain network using these endpoints along with valid application credentials.

JSON/RPC HTTP endpoint
https://u0q4fvx2hc-u0emnyiegb-rpc.us0-aws.kaleido.io/

JSON/RPC Websocket endpoint
wss://u0q4fvx2hc-u0emnyiegb-wss.us0-aws.kaleido.io/

REST API Gateway
https://u0q4fvx2hc-u0emnyiegb-connect.us0-aws.kaleido.io/

Figure 8: The metadata of node 1.

The details include

(i) The blockchain node ID.

(ii) The consensus role (non-signer).

(iii) The consensus ID.

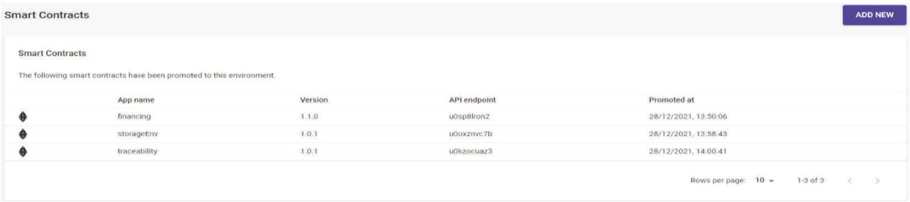
(iv) The runtime version.

(v) The user accounts.

In general, the figure represents all of the metadata for node 1.

Figure 9 shows all of the smart contracts added to the blockchain. Finance, storage environment, and traceability are the three aspects of smart contracts. Solidity is the programming language used to create them. The Solidity programming language exists only to facilitate the creation of smart contracts on the Ethereum blockchain.

The smart contract algorithm is depicted in Figure 10. The blockchain system now includes these algorithms. As you can see, this figure shows some functions in the storage environment Solidity file. For instance, consider the



Smart Contracts

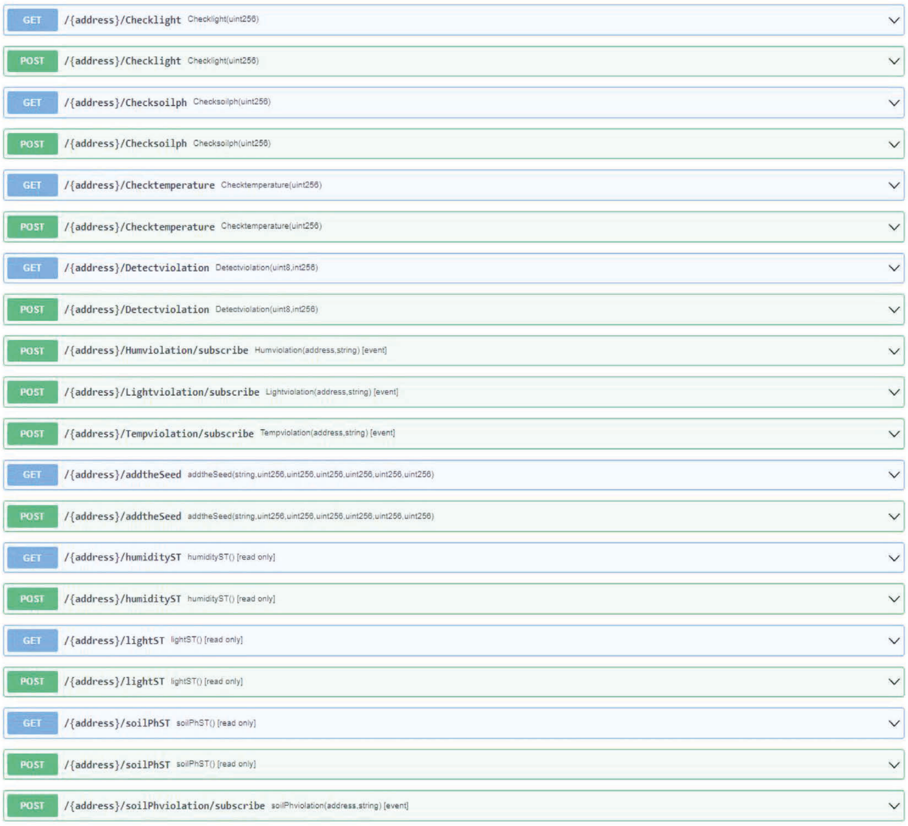
Smart Contracts

The following smart contracts have been promoted to this environment.

	App name	Version	API endpoint	Promoted at
◆	financing	1.1.0	u0sp8lron2	28/12/2021, 13:50:06
◆	storageEnv	1.0.1	u0oxznvc7b	28/12/2021, 13:58:43
◆	traceability	1.0.1	u0kzocuaz3	28/12/2021, 14:00:41

Rows per page: 10 ▾ 1-3 of 3 ‹ ›

Figure 9: The blockchain includes smart contracts.



```
GET  /{address}/Checklight  Checklight(uint256)
POST /{address}/Checklight  Checklight(uint256)
GET  /{address}/Checksoilph  Checksoilph(uint256)
POST /{address}/Checksoilph  Checksoilph(uint256)
GET  /{address}/Checktemperature  Checktemperature(uint256)
POST /{address}/Checktemperature  Checktemperature(uint256)
GET  /{address}/Detectviolation  Detectviolation(uint8,int256)
POST /{address}/Detectviolation  Detectviolation(uint8,int256)
POST /{address}/Humviolation/subscribe  Humviolation(address,string) [event]
POST /{address}/Lightviolation/subscribe  Lightviolation(address,string) [event]
POST /{address}/Tempviolation/subscribe  Tempviolation(address,string) [event]
GET  /{address}/addtheSeed  addtheSeed(string,uint256,uint256,uint256,uint256,uint256,uint256)
POST /{address}/addtheSeed  addtheSeed(string,uint256,uint256,uint256,uint256,uint256,uint256)
GET  /{address}/humidityST  humidityST() [read only]
POST /{address}/humidityST  humidityST() [read only]
GET  /{address}/lightST  lightST() [read only]
POST /{address}/lightST  lightST() [read only]
GET  /{address}/soilPhST  soilPhST() [read only]
POST /{address}/soilPhST  soilPhST() [read only]
POST /{address}/soilPhviolation/subscribe  soilPhviolation(address,string) [event]
```

Figure 10: Some of the functionalities of the smart contract.

programming method that evaluates light exposure, soil pH, temperature, and humidity. There are other functions, such as the one that identifies environmental violations.

In addition, the script code displays the current state of all these factors and the variety of additional features. This section is concerned with creating the best possible environment for seeds and products at farms and warehouses.

The quantity of resources required to run these script codes defines their complexity. Time and memory constraints are given special attention. These processes should take the same time on a particular system regardless of the input size. Another resource that gets used is the number of arithmetic operations.

These graphs (see Figure 11) illustrate that injecting smart contract code has a minor impact on the resources. The memory used by the smart contracts was just about 150 megabytes. In addition, the smart contracts only utilized roughly 40 megabytes of disk space. These numbers boost efficiency speed and have no negative impact on the system, computer nodes, or RAM.

Figure 12 depicts several successful block node transactions. These findings indicate that smart contracts might get used on a sensor dataset in the future (temperature, light, humidity, and soil pH).

The system is divided into three major categories. The first one is distribution and traceability, which encompasses all aspects of tracking and tracing a product from start to finish. The second category is finance, which involves all aspects of financial transactions. Finally, storage is the third category, which covers all violation management and regulates the environmental data of the farm and the warehouse.

After employing IoT devices to insert product information into the system, the associated authority can verify the real-time temperature, humidity, soil pH, and light exposure conditions. The contract self-checks data to keep

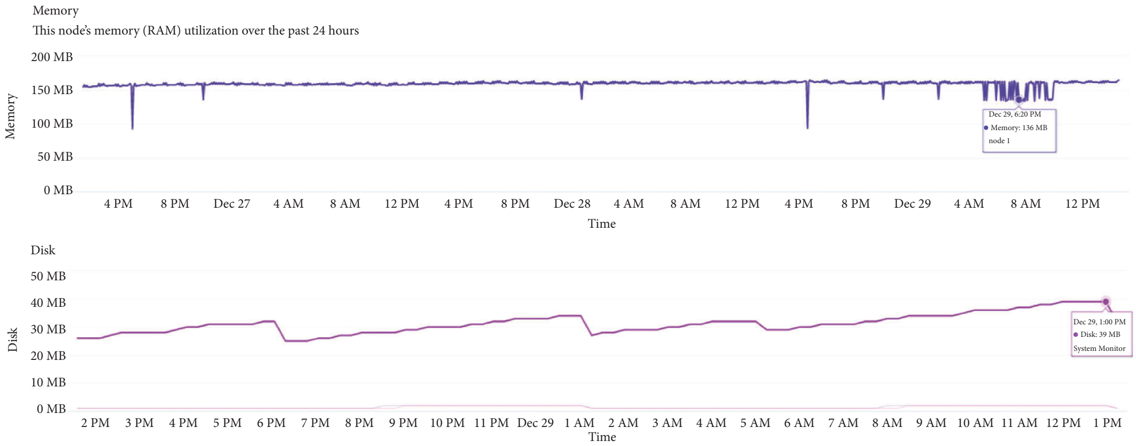
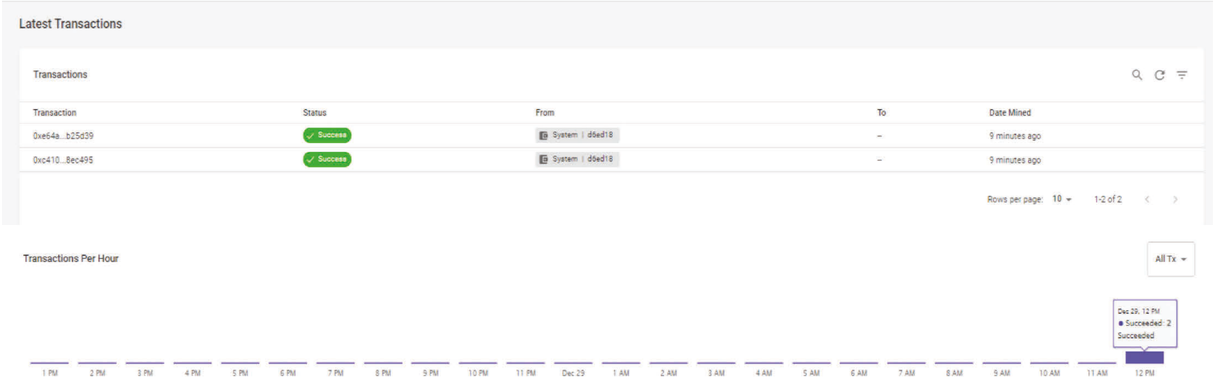


Figure 11: The minor impact of the smart contract script on the machine resources.


Latest Transactions

Transactions

Transaction	Status	From	To	Date Mined
0xe64a...b25d39	✓ Success	System | d6ed18	–	9 minutes ago
0xc410...8ec495	✓ Success	System | d6ed18	–	9 minutes ago

Rows per page: 10 1-2 of 2

Transactions Per Hour

Figure 12: Several successful block node transactions.

track of status and violations. The system displays temperature, humidity, soil pH, and light exposure values and checks whether or not they are close to ideal conditions.

The suggested architecture also maintains confidence between entities. Before acquiring a product, it guarantees that the consumer is adequately informed about the reputation of other sides (the seller). Furthermore, the early warning feature is added to reduce the usage of chemicals and establish the safety of agricultural produce.

Though bitcoin popularized blockchain transactions, putting it in a real-world use case still has various difficulties and many goals to achieve. Nevertheless, blockchain is still in its early phases and contains serious flaws. Also, the applications are a considerable challenge to optimize the solutions and the entire food supply chain.

We all agree that the procedures are automated and dependable after these analyses. Smart contracts periodically self-check data, keep track of all the details, and help save data to the blockchain. When it comes to the blockchain, the date, time, and product identifier allow all participants to track products and assure that they have access to information about the source of the products. The integrity is not threatened because the only participants authorized were those who could execute operations. Furthermore, using smart contracts eliminates the need for an intermediary party, and no one can alter the data inside the blockchain after it is verified.

So, it will be impossible to change the data if someone does something wrong or by mistake. Also, data collection relies on IoT devices; if they get broken, data collection is impossible. Furthermore, IoT devices are sometimes subject to security vulnerabilities.

Possibly, some participants cannot comprehend the system or blockchain technology. They may have limited knowledge or a lack of understanding of some functionalities. Perhaps we cannot persuade some participants to give information. But, in general, the system offers a radical change in all aspects of an existing business.

The comparison system (see Table 1) provides all the details we have discussed. It is conceivable that the proof-of-work consensus mechanism is the algorithm that operates when the system employs the Ethereum platform. However, we have found that many studies neglect to include this point.

4. Performance Evaluation

Throughput is probably the most misunderstood performance testing principle that beginner testers occasionally have trouble mastering. In general, "throughput" refers to

Table 1: System comparison with existing methods.

Ref	IoT app	IoT devices	Consensus algo.	Security analysis techniques	Energy and data size tests	Platform	Environment control smart contracts	Trace and distribution smart contract	Financing smart contract	Other smart contracts and notes
[12]	Yes	Temperature sensor, water level sensor, oxygen sensor, pH sensor, and water pump	PBFT	Not mentioned	Yes	Hyperledger Fabric	No	Basic	No	Records history of all transactions and device management
[13]	Yes	Temperature sensor, light sensor, and humidity sensor	Not mentioned	Not mentioned	Yes	Ethereum	Yes	Yes	Basic	Storage smart contracts
[14]	No	No IoT app mentioned	PoW	Not mentioned	Yes	Ethereum	No	Yes	Yes	No other smart contracts
[15]	No	No IoT app mentioned	PoW	Yes	Yes	Ethereum	No	No	No	The agent orchestrates offloading services
[16]	Yes	Sensors that manage water, air, sunlight, and soil quality escarpments	Not mentioned	Yes	Basic	Hyperledger Fabric	No	Yes	Yes	Storage of planting and farm information
[17]	Yes	Sensors, cameras, GPS locator, and 4G communication	Not mentioned	Not mentioned	Basic	Ethereum	Yes	Yes	Yes	Use case of soybean supply chains
[20]	Not mentioned	No equipment mentioned	Not mentioned	Yes	No	Ethereum	No	Yes	Yes	Insurance claim
[21]	Yes	Cameras, smart sensors, RFID, and GPS	Their consensus mechanism (OASCs' consensus network)	Yes	Yes	It looks like an Ethereum sys.	No	Yes	Basic	Automated workflows of OASCs
Our survey	Yes	Temperature sensor, humidity sensor, light exposure sensor, and soil pH sensor	PoW (in future PoS)	Yes	Yes	Ethereum (in future Ethereum 2.0)	Yes	Yes	Basic	No other smart contracts

the volume of transactions generated throughout the test. It may also be described as the minimum amount of capacity that an application or website can handle.

Additionally, it is typical to set a throughput goal for the application before beginning a performance test run so that it can handle a certain number of requests per hour.

The suggested model, for instance, uses multiple blockchains. User information blockchain, agri-product information blockchain, and transaction information blockchain are the three blockchains. Let us assume that, regardless of the volume or quantity of the data, it always takes one second to save a new set of three records on each block of the blockchain that we have.

A performance report for this scenario would reveal that the throughput is three records per moment. For more explanation, the user information blockchain can save the first recording data, the agri-product information blockchain can keep the second recording data, and the transaction information blockchain can save the third recording data. This thinking is our maximum throughput, but the goal now is to increase it by updating the programming level of smart contracts by incorporating new methods and functions that reduce execution time and work to prevent bottlenecks in the blockchain network.

The first issue users have represented in transaction latency is the delay between initiating a transaction or payment and receiving confirmation that it is valid. You would anticipate a flattening of the throughput once all users have logged in, begun to work, and sent requests. In our case, however, we have not added cryptocurrency-based financial transactions yet, and the project's goal is to simultaneously receive data from sensors and send it to the blockchain network. Hence, the automated system of smart contracts helps data arrive in any case and in good time for analyses.

With minimal competition, throughput varies by simply adjusting the load while latency remains constant. The reason is that whatever comes in comes out directly since there is a reasonable minimum cost to complete a transaction, and the queuing time is zero at low congestion.

Every newly formed block has a timestamp. The blockchain system uses timestamps to count the number of blocks added and produced during specific time intervals, such as per hour. The transaction latency metric can be calculated by looking up each transaction's timing and comparing the time it got completed to the time it was approved and saved. This measure can also reveal the speed at which consensus methods got used. The outcomes of these tests provide evaluations of the blockchain system's functionality and scalability.

A 3.5 GHz CPU device with 32 GB RAM and a 3 TB hard drive will execute the model (Ethereum blockchain network—8 nodes) with a margin of 100–120 seconds for latency and a margin of 235–260 transactions per second (TPS) for throughput. Data must wait to get transferred when the local network's capacity gets exceeded by traffic. Naturally, there will be more delay as a result. The latency will increase as the network becomes more crowded and we have many activities simultaneously. However, solid wireless signals make it possible for data to be transported from the source to the destination within a short period and decrease latency. Adding more controllers or validators who confirm blockchain transactions may boost performance.

The analyzers must measure a large-scale distributed system to identify obstacles or blockages and predict expected behavior under pressure.

4.1. Security Analysis. Now we will go on to the security analysis part, where we will list all of the model's components and discuss how they can provide security services and how they operate together to secure our data and deliver accurate information to all network members.

Smart contracts, apps, and the blockchain environment are vulnerable to attacks if the model gets weakly built. Let us start with the blockchain-based system, a public ledger of data kept on all nodes, with all participants receiving the same version of information and updates in real time as data changes. Furthermore, users authenticate using public-key cryptography, and validators check the data for accuracy before storing them in blocks; thus, all data transactions ought to have acceptance from all nodes in the network. Finally, once transactions get preserved in blocks, no one or entity may edit or alter the block's content.

The level of security in the blockchain scheme is intriguing, but let us now discuss smart contract security, which is more than simply a software program that gets run automatically. It is a part of the system. In our example, it can receive, transmit, and store data about the environment, such as temperature, humidity, light, and soil pH. It can also track down the merchandise and determine which company owns it. We must not forget that smart contracts specify the extent of an agreement between business parties by defining criteria that function as a trigger event in the contractual terms. The code gets performed, and the results get displayed on all of the network's nodes. Smart contracts get written conditionally, using if-then expressions, and thus neglect many of the problems caused by regular agreements, such as fraud. The use of blockchain to establish these types of arrangements eliminates the need for an intermediary. As a result, the total expenditures of the company get reduced. To prevent losses, professionals make timely code optimizations, conduct frequent code audits, and monitor the aberrant behavior of implemented deals.

Unless you install protection, IoT devices might be the weakest link in terms of security. For example, the IoT confronts fake cell tower assaults and threats such as man-in-the-middle attacks and SMS attempts such as phishing.

Blockchain technology has been a remarkable example of providing safety in transactions and information transfer. It offers a unique data structure in addition to integrated security measures. Consensus, decentralization, and cryptography—which ensure the trust of transactions—are the foundations of the blockchain.

However, poor technical implementation has led to several security vulnerabilities with blockchain. Although blockchain has significant security flaws, cyber security experts with special analytical and technical skills may take some steps to minimize these problems and implement blockchain in the safest possible way. Although the suggested approach may not stop these attempts, it makes it difficult for hackers to carry them out.

A 51% assault happens when one person or group of hackers gather more than half of the hash rate and takes over the entire system, which may be fatal for blockchains that handle financial transactions. In this case, hackers can alter transactions and stop the verification process. In sequence, the entire model will not add the transactions to the blocks. The attackers can even undo already finished transactions, which leads to double-spending. Thankfully, by increasing the hash rate and reducing the usage of proof-of-work techniques, we can avoid 51% assaults. The results conclude that this particular threat does not benefit the attacker itself because the main goal of the suggested agricultural model is to organize farm environmental data, distribute it to all network participants, and provide a layer of security and confidentiality to these data.

Phishing attacks are seriously impacting blockchain networks. The purpose of a phishing attack is for the hacker to obtain the user's login information. To the owner of the wallet key, they may send emails that appear to be honest. The user must enter their login information into a false hyperlink that is connected. A user's passwords and other sensitive information may be compromised, which may cause harm to both the individual and the blockchain network. If blockchain network members get an e-mail asking for login information, they must confirm with the partner (other members, for example). Additionally, installing malicious link detection software might aid in defending against this threat. We can point out that connecting to the blockchain through public Wi-Fi could be risky.

Routing attacks, for example, are attacks when a hacker uses the anonymity of a user account to intercept data as they get sent to Internet service providers. Data transfer and activities continue as usual in the case of a routing attack, so the participants are typically oblivious to the threat. However, establishing a secure routing protocol will solve this issue. The users of the network must use strong passwords and update them often. In practice, some farms are pretty likely to adopt this architecture, and there is a critical need to notify every blockchain participant of the risks related to data security.

Users connect with the blockchain using electronic devices like computers and smartphones, which are the endpoints of the blockchain network. Hackers can target devices and monitor user activity to obtain the user's key. The participants should not save blockchain keys on computers as text files by network users. Identifying dangers and securing the device include installing antivirus software.

Hackers could consider creating plenty of phony network nodes (Sybil attack). They can gain majority consensus and stop the chain's transactions using this technique. Massive Sybil strikes implied a 51% attack. The suggested approach utilizes suitable consensus methods and keeps track of the actions of other nodes.

Smart contract vulnerabilities can be risky because they can lead to security flaws in the source code. In this case, the code is straightforward, only analyzes information, detects crop growth conditions, and assesses the state of the product. Our smart contract source code will continually get updated. For instance, we may utilize the Oyente Analysis Tool to find vulnerabilities in contracts built on Ethereum. This technique evaluates the bytecode of the Ethereum blockchain.

The goal of analyzing and measuring connectivity, monitoring sensors, and making intelligent modifications and enhancements is to maximize price and quality. IoT devices generally use Linux operating system, typically the customized version. Data are protected when they flow from the local machine to the other scheme stages in IoT security (blockchain and smart contract layer). IoT devices frequently have a mobile application and are used to connect to devices via the Internet. When monitoring the sensors wirelessly, use a VPN to connect to them. Man-in-the-middle attacks will never be possible with a VPN solution. Block unneeded network ports, for example, the Telnet port should be deactivated to prevent further Telnet protocol attacks. By allowing traceable proof of ownership of products and by integrating unique tags (e.g., RFID, NFC, and QR codes) to create smart tags, we can add a few supply chain logistics to ensure traceability and protect the information exchanged through blockchain.

5. Conclusion and Future Work

In conclusion, blockchain and IoT technologies can aid in developing a secure, transparent, open, and innovative ecological agriculture system that involves all participants.

This work aims to provide a possible technique to build practical blockchain-based applications and change the agriculture industry, even though the evolution of blockchain and agriculture research studies is still in its infancy. This model is considered a prototype for reducing financial loss and agricultural pollution. The system defines the three primary entities in the agriculture domain: data, process, and stakeholders. Adding a cryptocurrency process for the interaction between the entities and registering/tracking the seller's land will be a big step for this blockchain system model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G. S. Sajja, K. P. Rane, K. Phasinam, T. Kassanuk, E. Okoronkwo, and P. Prabhue, "Towards applicability of Blockchain in Agriculture sector," *Materials Today Proceedings*, vol. 5, 2021.

[2] J. Lin, Z. Shen, A. Zhang, and Y. Chai, "Blockchain and IoT Based Food Traceability for Smart Agriculture," in

Proceedings of the 3rd International Conference on Crowd Science and Engineering, Singapore, 2018.

[3] W. Lin, X. Huang, V. Wang et al., "Blockchain technology in current agricultural systems: from techniques to applications," *IEEE Access*, vol. 8, pp. 143920–143937, 2020.

[4] H. Xiong, T. Dalhaus, P. Wang, and J. Huang, "Blockchain technology for agriculture: applications and rationale," *IEEE Access*, vol. 3, pp. 1–7, 2020.

[5] P. Dutta, T. M. Choi, S. Somani, and R. Butala, "Blockchain technology in supply chain operations: Applications, challenges and research opportunities," *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, Article ID 102067, 2020.

[6] G. Leduc, S. Kubler, and J. P. Georges, "Innovative Blockchain-based farming marketplace and Smart contract performance evaluation," *Journal of Cleaner Production*, vol. 306, pp. 1–15, 2021.

[7] P. Bottoni, N. Gessa, G. Massa, R. Pareschi, H. Selim, and E. Arcuri, "Intelligent smart contracts for innovative supply chain management," *Frontiers in Blockchain*, vol. 3, no. 52, 2020.

[8] K. Dey and U. Shekhawat, "Blockchain for sustainable e-agriculture: literature review," *Architecture for data management, and implications, Journal of Cleaner Production*, vol. 316, pp. 1–17, 2021.

[9] L. Hang, I. Ullah, and D. Kim, "A secure fish farm platform based on Blockchain for agriculture data integrity," *Computers and Electronics in Agriculture*, vol. 170, pp. 1–15, 2020.

[10] T. H. Pranto, A. A. Noman, A. Mahmud, and A. K. M. Haque, "Blockchain and Smart contract for IoT enabled smart agriculture," 2021, https://arxiv.org/abs/2104.00632.

[11] X. Huang, D. Ye, and L. Shu, "Securing parked vehicle assisted fog computing with Blockchain and optimal Smart contract design," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 426–441, 2020.

[12] L. Wang, L. Xu, S. Liu et al., "Smart contract-based agricultural food supply chain traceability," *IEEE Access*, vol. 9, pp. 9296–9307, 2021.

[13] K. Salah, N. Nizamuddin, and M. Omar, "Blockchain-based soybean traceability in agricultural supply chain," *IEEE Access*, vol. 7, pp. 73295–73305, 2019.

[14] I. Widi Widayat and M. Köppen, *Blockchain Simulation Environment on Multi-Image Encryption for Smart Farming Application*, Springer International Publishing, Berlin, Germany, 2021.

[15] C. M. Balaceanu, I. Marcu, and G. Suciu, "Telemetry System for Smart Agriculture," *Business Information Systems Workshops*, Springer, Berlin, germany, 2019.

[16] H. Patel and B. Shrimali, "AgriOnBlock: Secured data harvesting for agriculture sector using Blockchain technology," *ICT Express*, vol. 10, 2021.

[17] S. Hu, S. Huang, J. Huang, and J. Su, "Blockchain and edge computing technology enabling organic agricultural supply chain: a framework solution to trust crisis," *Computers & Industrial Engineering*, vol. 153, pp. 1–12, 2021.

[18] H. N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of Things: a survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[19] M. A. Shu and L. Shu, "The performance evaluation of blockchain-based security and privacy systems for the internet of Things: a tutorial," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17236–17260, 2021.

[20] K. Peng, M. Li, C. Wang, S. Wan, and K.-K. R. Choo, "Security challenges and Opportunities for smart contracts in internet of Things: a survey," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12004–12020, 2021.

[21] L. Hang, B. Kim, K. Kim, and D. Kim, "A permissioned blockchain-based clinical trial service platform to improve trial data transparency," *BioMed Research International*, vol. 2021, no. 22, 2021.

[22] L. Hang, B. Kim, and D. Kim, "A transaction traffic control approach based on fuzzy logic to improve hyperledger fabric performance," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 2032165, 19 pages, 2022.

[23] R. Kumar, P. Kumar, A. Aljuhani, A. K. M. Najmul Islam, A. Jolfaei, and S. Garg, "Deep learning and smart contract-assisted secure data sharing for IoT-based intelligent agriculture," *IEEE Intelligent Systems*, vol. 99, 2022.

[24] J. DafniRose and J. DafniRose, "Blockchain-enabled smart agricultural knowledge discovery system using edge computing," *Procedia Computer Science*, vol. 202, pp. 73–82, 2022.

[25] A. E. Mane, M. Chihab, O. Bencharef, and C. Younes, "Architectural scheme of a multi-Blockchain in the Agricultural field," *E3S Web of Conferences*, vol. 297, no. 4, 2021.