Wireless Personal Communications
https://doi.org/10.1007/s11277-023-10230-9

A Novel Blockchain and Internet of Things-Based Food Traceability System for Smart Cities

Ashish Kumar Tripathi[1] · **K. Akul Krishnan**[1] · **Avinash Chandra Pandey**[2]

Accepted: 15 February 2023

Abstract

Rapid urbanization has recently caused serious problems for cities all around the world. Smart cities have drawn much interest from researchers in the present research paradigm to manage the expanding urban population. Frameworks for smart cities are planned and implemented using platforms based on blockchain and the Internet of Things (BIOT). Smart cities may use the BIoT platform to provide improved transportation, food traceability, and healthcare services. Food safety is one of the sectors where less research has been done than the others. The importance of food safety is now more widely recognized, making it essential to improve the traceability and transparency of the food supply chain. In this paper, a novel BIOT-based layered framework using EOSIO has been proposed for effective food traceability. The proposed system first identifies the suitable traceability units to provide better transparency and traceability and then defines and implements a layered architecture using Ethereum and EOSIO blockchain platforms. The performance of the proposed EOSIO-based model is evaluated using the practicality of the consensus algorithm, block production rate, throughput, and block confirmation time. The proposed traceability system attains a block production rate of 0.5 s and a block confirmation time of 1 s, which is much lower than the Ethereum-based traceability system. Hence, from the experimental evidence, the superiority of the proposed EOSIO-based food traceability can be observed.

Keywords Smart city · Blockchain · Internet of things · Food traceability · Ethereum

✉ Avinash Chandra Pandey
avish.nsit@gmail.com

Ashish Kumar Tripathi
mail2ashish07@gmail.com

K. Akul Krishnan
akusresukri@gmail.com

[1] Department of Computer Science, MNIT, Jaipur, Rajasthan, India

[2] Discipline of Computer Science, PDPM-IIITDM, Jabalpur, Madhya Pradesh, India

Published online: 06 March 2023

1 Introduction

In recent years, the global urban population has experienced rapid growth, rising from 751 million in 1950 to 4.2 billion in 2018 [1]. According to the United Nations department of economic and social affairs (DESA), the urban population covers around 55% of the world's population, which is projected to reach 68% by 2050 [2]. The majority of cities around the globe are facing severe challenges in coping with rapid urbanization. Moreover, the impact of the COVID-19 pandemic has further aggravated the issue. To overcome these challenges, it is necessary to make cities sustainable through better utilization of emerging technologies such as IoT, cloud computing, and others. Modern technologies provide efficient decision-making tools to enhance the efficiency of planning, monitoring, and controlling resource consumption. Cities that use the technologies mentioned above are known as "Smart Cities." Smart cities not only improve the quality of life but also optimally utilize resources, safeguarding the environment [3]. In smart cities, IoT platforms are used to monitor and manage city infrastructure such as traffic flow, parking, water distribution, waste disposal and treatment, air quality, etc., leading to improved public welfare, economy, government services, and urban planning. The smart cities market was valued at USD 739.78 billion in 2020 and is expected to reach USD 2036.10 billion by 2026 [4]. Well managed smart cities offer significant opportunities for economic development and greater access to essential services, including health care and education.

A smart city ecosystem comprises smart healthcare, smart banking, and smart transportation, which are interconnected, instrumented, intelligent, and sustainable [5]. Such an increase in living standards has elevated people's concerns about the quality, hygiene, and source of food. Nowadays, food and hygiene sources are a major concern for people around the globe. WHO also presented statistics that food from unhygienic sources can adversely affect the health of millions of people. As per statistics presented by WHO, an estimated 600 million people, almost 1 in 10 in the world, fall ill every year by eating contaminated food [6]. In the past, many food scandals has happened all over the world. There was a horse meat scandal in parts of Europe in 2013, where lamb meat was replaced by horse meat [7]. Some other notable food-related scandals and hazards are the Salmonella outbreak caused by papayas [8], Escherichia coli caused by eating baby spinach [9], and Norovirus outbreak linked to raw oysters from British Columbia [10]. The WHO states that USD 110 billion is lost each year as a result of the production costs and medical expenses resulting from unsafe food [6]. Hence, an efficient food traceability system can not only enhance food safety but also reduce the financial loss incurred in an outbreak. It justifies the current research direction toward food traceability in smart cities and the corporate interest in the field. To guarantee food safety, it is necessary to build a complete system that can trace and track every component in food production. In literature, a few efficient food traceability systems have been presented.

Alfian et al. introduced IoT-based perishable food traceability systems that employ RFID for shipment tracking [11]. A similar IoT-based system has been presented by González-Amarillo et al. [12] to trace the crop traceability. The literature illustrates how the utilization of IoT devices such as RFID and QR codes has improved the quality of data flow in the supply chain. However, IoT-based food traceability systems are riddled with the issue of data integrity, trust, authenticity, and security. Moreover, with IoT devices having limited computing power and storage capacity, tracing food sources and batches becomes time-consuming. To address the same, Michelin et al. [13] presented an efficient approach based on blockchain. A blockchain is a distributed immutable ledger that stores data as continuous chains of blocks in a decentralized network. Blocks are interlinked with the cryptographic hash value. Blockchain provides seamless authentication, privacy, security, transparency, and trustworthiness. Besides, it also reduces the computational overhead of resource-constrained IoT devices. Furthermore, for leveraging smart contracts and decentralized applications, blockchain provides more autonomy to the smart traceability system. Several blockchain smart traceability systems have been presented in the literature. The majority of proposed solutions are just conceptual. Casino et al. [14], Hasan et al. [15], and Shahid et al. [16] among many others, presented a traceability system based on the Ethereum blockchain platform [17]. However, the Ethereum blockchain platform is unsuitable for a food traceability solution as it employs proof-of-work as the consensus mechanism [18]. Therefore, to implement a food traceability system, a novel platform based on the hyperledger sawtooth has been used [19]. The versatility of Hyperledger allows to use different consensus mechanisms, thus discrediting the main disadvantage of the Ethereum blockchain. Hyperledger's versatility enables the use of various consensus mechanisms, thus negating the main drawbacks of the Ethereum blockchain. Employing hyperledger sawtooth enhances the food traceability system. However, it still lacks some key functionalities, such as the mechanism for role-based permission management and determining the payee of a transaction cost. Therefore, this paper proposes an EOSIO-based food traceability system to address the same. The main objective of the proposed food traceability system is to identify the raw material used in a product, as shown in Fig. 1. It can be observed from the figure that raw materials used in Burger are wheat, tomato, and chicken. The following contributions are made in this work.

- A novel food-traceability model has been introduced for smart cities using blockchain and IoT.

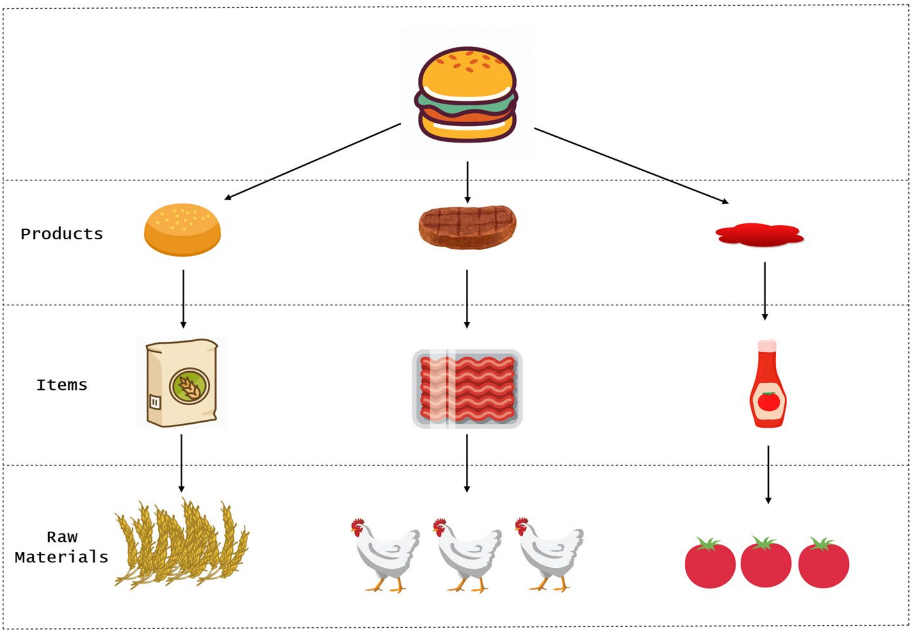


Fig. 1 Traceability of raw materials

- The proposed system leverages the strengths of the EOSIO platform, providing an enhanced food traceability solution with an in-built authentication and authorization mechanism and optimal resource utilization.
- The performance of the proposed system is validated in terms of resource utilization, CPU, and net bandwidth (NET).
- The proposed EOSIO-based system is also compared with the Ethereum-based traceability system in terms of consensus algorithm, throughput, scalability, block confirmation time, latency, and block production rate.

The rest of the paper is organized as follows. Section 2 expands upon the technologies used. Section 3 discusses the related works and existing implementations. Section 4 introduces the proposed layered architecture for food traceability in the smart city. Section 5 illustrates the implementations of the proposed layered architecture along with the evaluation parameters. Section 6 summarises the work and describes the limitations of the proposed solution. Besides, it also highlights the future direction of the research.

2 Technologies Used

2.1 Blockchain

Blockchain is a distributed ledger based on a decentralized peer-to-peer (P2P) network that is both immutable and transparent in nature [20]. In blockchain, distributed ledger is arranged as chains of blocks, where each block is linked to the previous block by its cryptographic hash [21]. Each block has a block number and a time-stamp. Inside each block, multiple transactions are arranged as a Merkle tree. Each transaction is cryptographically signed using an asymmetric digital key. The consensus on maintaining the distributed ledger is defined by a governance mechanism referred to as the consensus algorithm. Blockchain technology was first presented in a white paper by Satoshi Nakamoto in 2008 [22]. The first programmable public permissionless blockchains blockchain was introduced by Vitalik Buterin in 2013 [23]. The public permissionless blockchains are categorized into Bitcoin and Ethereum blockchain networks. This categorization is based on read-write access to the blockchain and on participation in the consensus of a blockchain network. Public blockchains are open to all, i.e., anyone can view the ledger. while, in a private blockchain, only select members can view the ledger. A permissionless blockchain allows anyone to perform transactions and participate in the consensus, and a permission-based blockchain allows only select members to do the same. Permissioned blockchain lies under a single organization. There is another blockchain technology named “Consortium Blockchain,” which is similar to the permissioned blockchain, except it is controlled by a collection of organizations rather than a single organization [24]. Hence, in Consortium blockchain, a specific predefined set of members is only allowed to make transactions and perform consensus. To encode the blockchain data efficiently and securely, a Merkle Tree is used. Besides, consensus algorithms are also used to achieve synchronization and reach an agreement between the thousands of nodes on a blockchain network. The following sections discuss Merkle trees and the consensus algorithm.

2.2 Merkle Tree

A Merkle tree is used to enhance the immutability of the blockchain [25]. A Merkle tree adds up all of the transactions in a block and provides a digital fingerprint of the complete set of operations, allowing the user to check whether the block contains any transactions. A Merkle tree is a binary tree in which the leaf is each node's hash, i.e., the block's hash in a transaction, while each non-leaf node is the hash of a combination of its corresponding child nodes [26]. Further, the root of the Merkel tree is used to verify the data on a Merkle tree. A Merkle root is a straightforward mathematical method for validating the data on a Merkle tree. Merkle roots are used in cryptocurrency to ensure that data blocks sent between peers on a peer-to-peer network are complete, undamaged, and undamaged.

2.3 Consensus Algorithm

The consensus algorithm defines how to maintain the distributed ledger [27]. In a blockchain, no centralized authority verifies transactions or blocks. So, to reach an agreement on any single block, we require a consensus algorithm that is accepted across the entire blockchain. In the literature, various types of consensus algorithms, as discussed below, are presented and implemented [28].

1. Proof-of-Work: Proof-of-Work (PoW) is a consensus algorithm that enforces the peers in the blockchain network to solve a hard mathematical problem [29]. PoW is used in Bitcoin as well as Ethereum. In PoW, peers that compete with each other to solve the problem and mine the block is termed, miners. They receive a reward according to the resource consumed. Blockchain validators in a PoW system must continuously pass data from a block header through a cryptographic hash function. Every time the input data is passed through the cryptographic hash function, validators add an arbitrary number called a nonce. To decide what data is added to the next block in a blockchain, PoW requires a lot of electricity and processing power. The PoW technique requires specialized computers known as ASICs to solve complicated mathematical problems. This leads to high financial costs and ecological damage [30].
2. Proof-of-Stake: Proof-of-Stake (PoS) consensus algorithm was developed as an alternative to PoW in 2011 [31]. Although the objectives of PoS and PoW are comparable, there are several significant distinctions between them, especially when validating new blocks on the blockchain network. The Proof of Stake (PoS) consensus algorithm validates blocks based on stakes held by network participants. In PoS, each block's validator is selected randomly from the stakeholders depending on the available computing power. Although each PoS system may execute the algorithm differently, the blockchain is generally safeguarded by a pseudo-random election process that considers a node's allocation. The allocation evaluates the party's commitment to ensuring the network. To boost the network's scalability and avoid excessive electricity waste, the Ethereum blockchain, the world's largest blockchain network in developer activity, has migrated from the PoW algorithm to the PoS algorithm [31].

2.3.1 Delegated-Proof-of-Stake

An improved PoS method that tries to address the flaws of both PoW and PoS consensus algorithms is called Delegated-Proof-of-Stake [32]. The DPoS consensus algorithm is a democratic mechanism where a few nodes have delegated the responsibility for generating and validating new blocks [33]. This is done by a voting mechanism where each party's voting power is proportional to the token staked in the system. Since the reputation of the delegated nodes influences the voting, misbehavior will lead to being expelled and replaced. DPoS is more scalable than PoS and PoW and can provide more transactions per second. The EOSIO blockchain employs a DPoS consensus mechanism.

2.3.2 Smart Contracts

Smart contracts are programs generally deployed on the blockchain and run when some specific criteria are met [34]. Smart contracts are used to implement the business logic on a blockchain network. A smart contract has a unique address and is visible to all nodes in the blockchain network once it is deployed. Utilizing this address will activate blockchain smart contract features [35]. Ethereum requires the use of the Solidity programming language to write smart contracts [36]. Once compiled, the byte code of the smart contract is generated and runs on the Ethereum Virtual Machine. EOSIO supports multiple programming languages to write smart contracts, including C++, Rust, Go, etc. They are compiled into Web Assembly (WASM), which runs on the EOSIO blockchain network [37].

2.4 Ethereum Blockchain

Ethereum is an open-source, public blockchain network. In the Ethereum environment, everyone agrees on the state of a Ethereum virtual machine (EVM) [38]. Every node in the blockchain network maintains a copy of the EVM state. When a transaction request is made, all the nodes validate and carry out the request, which subsequently updates the EVM states of all the nodes. Ether (ETH) is the default token/cryptocurrency used on the Ethereum main network. Ether also makes it easier to validate transactions and incentivizes nodes to contribute computational resources to the network [39]. An additional Ether for a transaction is given to achieve the same. The incentive paid corresponds to the time required to complete the computation. Ethereum supports smart contracts. Anyone with enough Ether can deploy smart contracts to the Ethereum main network because it is a public permissionless network. Ethereum supports Solidity and Vyper programming languages for writing smart contracts. The interactions made with the smart contract are final and cannot be reversed. Ethereum employs proof-of-work as the consensus algorithm. PoW has very high and unsustainable energy consumption.

2.5 EOSIO Blockchain

EOSIO is an open-source blockchain platform. It was created in 2018 and is maintained by Block One [40]. EOSIO blockchain networks supports high performant decentralised applications. C++ is used to write EOSIO smart contracts [41]. NodeOs, cleOs, keosd, EOSIO.CDT and system contracts make up the core layer of the EOSIO blockchain. In an

EOSIO blockchain, smart contracts can be updated to customize resource allocation and governance rules. A WebAssemby virtual machine (WASM) serves the functionality of the smart contract in EOSIO [42]. The core components of EOSIO are explained as follows.

- *Nodeos (Node + EOSIO):* NodeOs is the node daemon in EOSIO that enables a user to run a single-node blockchain network with a local development platform and API endpoints. The nodeOs configuration file can be tweaked to allow or disable EOSIO features.
- *Cleos (CLI + EOSIO):* Cleos is the command-line tool used to interact with nodeOs and is used to deploy EOSIO smart contracts, invoke actions, and interact with keosd.
- *Keosd:* Keosd is the key management daemon in EOSIO. It is used to store private keys securely and sign transactions and digital messages.
- *EOSIO.CDT:* EOSIO.CDT is a tool that supports building smart contracts. It helps to generate optimized WebAssemby (WASM) files with library support that can be deployed in the EOSIO blockchain.

The EOSIO platform emloys DPoS as the consensus algorithm. DPoS allows to select block producers through a continuous approval voting system, provided that one holds sufficient tokens [43]. When tokens are staked for CPU and NET, system resources are assigned proportionally to the percentage of tokens staked for the same resource at the same moment, regardless of market fluctuations. Establishing the smart system contract allows system resources to be dynamically allotted to EOSIO accounts. The following is a list of system resources in an EOSIO blockchain.

- *RAM:* In the EOSIO blockchain network, RAM is one of the core system resources. Blockchain accounts, permissions, smart contracts, and other data are kept in RAM for quick access. It is a long-term storage solution that must be purchased. As RAM is a restricted persistent resource, the EOSIO staking mechanism does not apply to it.
- *CPU:* It is processing time that is measured in microseconds (μs). The staking mechanism of EOSIO applies to the CPU, as it is a transient system resource.
- *Network (NET):* It is the network bandwidth, measured in bytes. An EOSIO account is used to deploy actions to the blockchain. The staking mechanism of EOSIO applies to NET, as it is a transient system resource.

3 Related Works

Food traceability has always been considered a complex process. Throughout the food supply chain, a wide range of data collection nodes are present, each providing different types of data. Data in a food supply chain ranges from product life and transport information to environmental data. Establishing a good traceability system for such a problem has always been challenging. Therefore, traceability units are first identified before establishing a food traceability system. The food traceability system determines components before the model's development. An efficient system should cover production, manufacturing, trade, and logistics. Moreover, reliable and accurate data must be recorded at each stage for better and more efficient traceability. Employing RFID in logistics and supply chains provides a systematic improvement to the food traceability systems. Dabbene et al. [44] employed RFID in monitoring critical steps of food distribution, improving visibility in the supply chain. A similar approach was proposed by Farooq et al. [45] to introduce a cost-effective solution for improving consumer health. The utilization of IoT technology for recording environmental data, product, and shipment details has improved food traceability systems. Moreover, IoT can be regarded as a critical component in digitizing the supply chain. Abdel-Basset et al. [46] introduced an IoT-based system that was able to establish a dynamic real-time business model with minimal human error. Recently, QR codes and IoT-based blockchain frameworks have also gained the attention of consumers to acquire traceability details [47, 48]. Alfian et al. [11] introduced a shipment and transport tracking system using RFID and smart sensors to monitor and control environmental variables. IoT devices have systematically improved food traceability systems due to their limited computing power and storage capacity. However, managing the trust, authenticity, integrity, and security in the IoT-based system is a challenging task [49]. Therefore, blockchain has been used to mitigate these limitations and improve the foundation of food supply chain management and traceability. Recently, the integration of Blockchain and IoT (BIoT)-based models has significantly received the attention of researchers [30]. Several theoretical frameworks for a BIoT-enabled food traceability system have been introduced in the literature [14, 50, 51]. However, the majority of the studies are limited to theoretical frameworks.

Recently, Walmart and IBM have jointly introduced a commercial BIoT model for food traceability [52]. Besides, some other BIoT-based models were also presented in the literature. Casino et al. [14] introduced an Ethereum-based BIoT model for traceability in the supply chain. However, the traceability units defined in the proposed model are batches. Hence, the model can only provide limited traceability in food safety. Hasan et al. [15] also introduced an Ethereum-based BIoT model for shipment tracking. Furthermore, Salah et al. [53] presented a traceability system based on the Ethereum platform. In the above model, Ethereum smart contracts are leveraged to track shipments, which sometimes fail to guarantee the authentication or authorization of the product. Therefore, Shahid et al. [16] presented an effective solution for traceability that ensures product authenticity. Food traceability solutions based on Ethereum are impractical for food traceability because Ethereum uses the PoW consensus mechanism, which is not suited for food safety. To mitigate the same, Tsang et al. [54] presented a model integrated with a consensus mechanism developed for supply chain management (SCM). BIoT models are also a trending research topic in other domains. Multiple papers have been published using the BIoT model in the medical field. Uddin et al.[55] presented a model for remote patient monitoring using the BIoT framework. In the literature, many BIoT-based healthcare frameworks have been presented. Liu et al. [56] introduced a smart BIoT traceability system for drugs. Besides, a secure Blockchain-enabled transportation system has also been presented. Although the proposed model addresses the issue of practical consensus algorithms, their practical use is impossible until a product is developed and published.

Moreover, the Hyperledger Sawtooth platform has also been used to implement a food traceability system [19, 57]. The sawtooth-based food traceability model attained better scalability and higher throughput. Although proven to be a better solution, Sawtooth lacks a permission management module, while other techniques cannot distinguish between the business logic and security modules. Considering the aforementioned limitations, an improved EOSIO-based layered architecture is designed and implemented in this paper. The proposed layered architecture decreases the security gaps and attack vectors. Further, to enhance the strengths of EOSIO, the novel BIoT-based food traceability model for the smart city has been implemented. Some of the significant enhancements to the EOSIO-based food traceability system include permission management modules, controls for payer scheduling of transactions, and a token-free blockchain. The Table 1 provides a compact

Table 1 Comparison of proposed system with existing literature

Research Paper	Authentication & authorization	Smart Contract	Practical Consensus Mechanism	Scalability	Implementation
Casio et al. [14]		✓			Ethereum
Baralla et al. [19]	✓	✓	✓	✓	Sawtooth
Hasan et al. [15]		✓			Ethereum
Caro et al. [57]	✓	✓	✓	✓	Sawtooth
Salah et al. [53]		✓			Ethereum
Shahid et al. [16]	✓	✓			Ethereum
Proposed model	✓✓	✓	✓	✓	EOSIO

summary of the related works and the proposed EOSIO-based layered architecture. Check marks in the table indicate that a model supports a particular transaction protocol. It can be observed from the table that there are double-check marks for authorization and authentication transaction protocols for the proposed, which means the proposed EOSIO model enhances both authorization and authentication. On the contrary, other models enhance either authorization or authentication.

4 Proposed Model

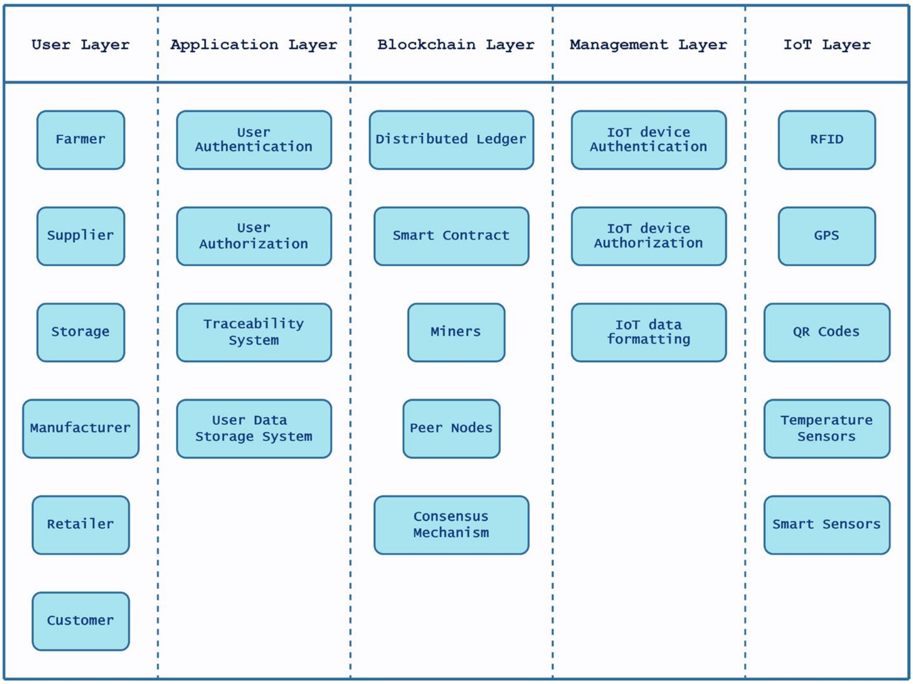
Integrating Blockchain and IoT (BIoT) has systematically enhanced food traceability systems by improving control over the supply chain. The BIoT traceability system offers better real-time dynamics and ensures significant improvements in auditability, accountability, immutability, efficiency, process quality, high availability, security, and fault-tolerance. In this paper, a BIoT-based layered architecture has been introduced for food traceability systems. The following section details the proposed layered architecture.

4.1 Proposed Layered Architecture for Food Traceability

Figure 2 presents the layered architecture of the proposed model. It consists of 5 layers, namely perception/IoT layer, management layer, blockchain layer, service layer, and user layer. All layers in the proposed architecture are logically connected. The functionality of each layer has been discussed in the following subsections.

4.1.1 IoT/ Perception Layer

The perception layer is the lowest layer that encapsulates different IoT devices. Each of these devices is responsible for forwarding data to the management layer. The IoT devices include temperature, pressure, pH, and GPS sensors along with identification tools such as RFID, QR code, etc., for monitoring environmental variables. At this layer, data is constantly generated and forwarded to the next layer. The next layer then adds the received data to the blockchain at fixed intervals or in response to events.



User Layer	Application Layer	Blockchain Layer	Management Layer	IoT Layer
Farmer	User Authentication	Distributed Ledger	IoT device Authentication	RFID
Supplier	User Authorization	Smart Contract	IoT device Authorization	GPS
Storage	Traceability System	Miners	IoT data formatting	QR Codes
Manufacturer	User Data Storage System	Peer Nodes		Temperature Sensors
Retailer		Consensus Mechanism		Smart Sensors
Customer				

Fig. 2 Layered architecture of the proposed model

4.1.2 Management Layer

The management layer acts as a bridge between the IoT and the blockchain layer. The functionality of this layer is to manage IoT devices. Besides, this layer is also responsible for the addition or removal (temporary/ permanent) of the IoT device, along with device authentication and authorization. Moreover, forwarding data to corresponding smart contract actions is another task of this layer.

4.1.3 Blockchain Layer

This layer consists of the distributed ledger, the consensus mechanism, smart contracts, miner nodes, and peer nodes. The distributed ledger in the blockchain is arranged as chains of blocks, where each block is linked to the previous block in the chain by its cryptographic hash. The consensus algorithm defines how to reach an agreement on any single block and thus maintain the distributed ledger. Smart contracts facilitate, verify, enforce a control sequence, and implement logic using code. Section 3 explains blockchain and its components in detail.

4.1.4 Application Layer

The primary function of the application (service layer) is to process the authorized user layer requests. In this layer, either a block is added to the blockchain network or data is retrieved. Smart contracts are used only to implement this layer, and user permissions are managed in the implementation.

4.1.5 User Layer

This layer consists of all stakeholders in the food supply chain. The stakeholders in the traceability system include farmers, suppliers, manufacturers, storage warehouses, retailers, consumers, and administrators. Each stakeholder tries to provide better service to the consumer. All stakeholders add data to the blockchain in line with their role. They are authenticated and authorized by the service layer based on their role. The administrator deploys the smart contract on the blockchain and is the smart contract owner. Furthermore, the administrator performs the addition or removal of users and IoT devices.

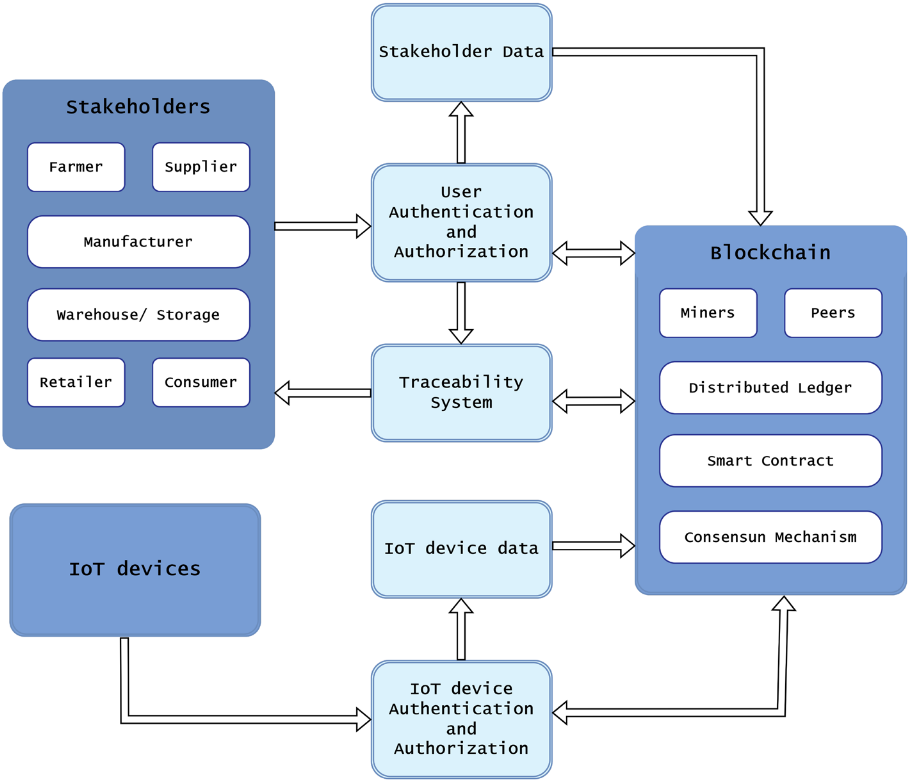


Fig. 3 Architecture of proposed model

4.2 Key Components of the Architecture

Figure 3 represents the architecture of the proposed model with control flow between the key components. The major components of the proposed model are discussed below.

4.2.1 Traceability System

The first and most important stage in the food traceability system is to identify and define the traceability units. The traceability unit determines the extent to which the traceability system is detailed. Figure 3 depicts the components of the traceability units. The traceability units provide extensive and low-level information on the product's raw material. Data at each stage of the supply chain is stored in the blockchain as traceability units. Each traceability unit contains information that serves as a primary key. The primary key is mapped to various additional data structures to enhance traceability. Other data, such as block timestamps is not used in tracing but is used to calculate product life, shelf life, and other metrics. The data flow model represented in the Fig. 4 demonstrates the workflow of the traceability system. The proposed traceability system first acquires details such as product

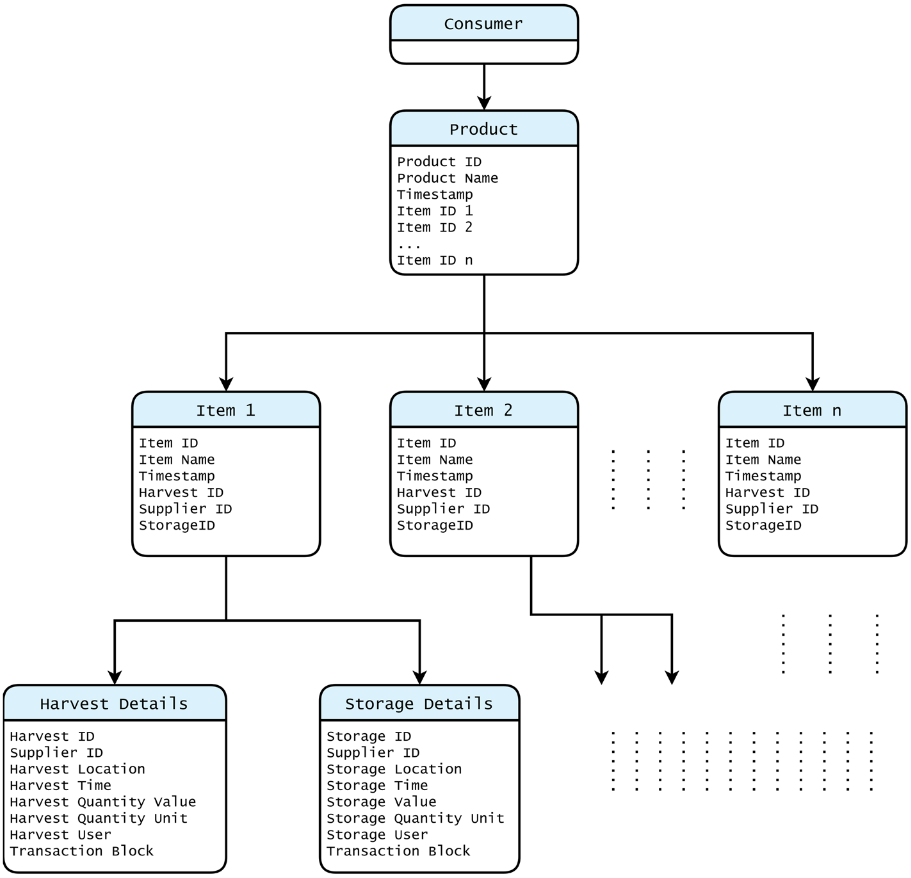


Fig. 4 Data flow of the proposed model

ID, product name, and items involved in making the product. After that, each item is linked to its source to get the details of the raw materials used. Both Ethereum and the proposed EOSIO use the same traceability units mentioned above.

4.2.2 User Identity Management

The primary goal of this study is to improve user identity management. For the same, actions and transactions are authorized with permissions associated with the account. Permissions are generally stored in an authority table along with the thresholds. The threshold is used to determine whether or not an action is approved. The proposed EOSIO supports hierarchical permissions in which any permission can be created and added at any time to the authorization table. The authority of the newly added permission will be equal to the authority of its parent. Account public key, account name, time wait, and permission level determine the authorization of accounts in EOSIO while Ethereum updates the smart contracts. The EOSIO permission management is a separate module from the business logic implementation. Hence, the complexity is reduced, and better layer separation is achieved.

4.2.3 IoT Device Identity Management

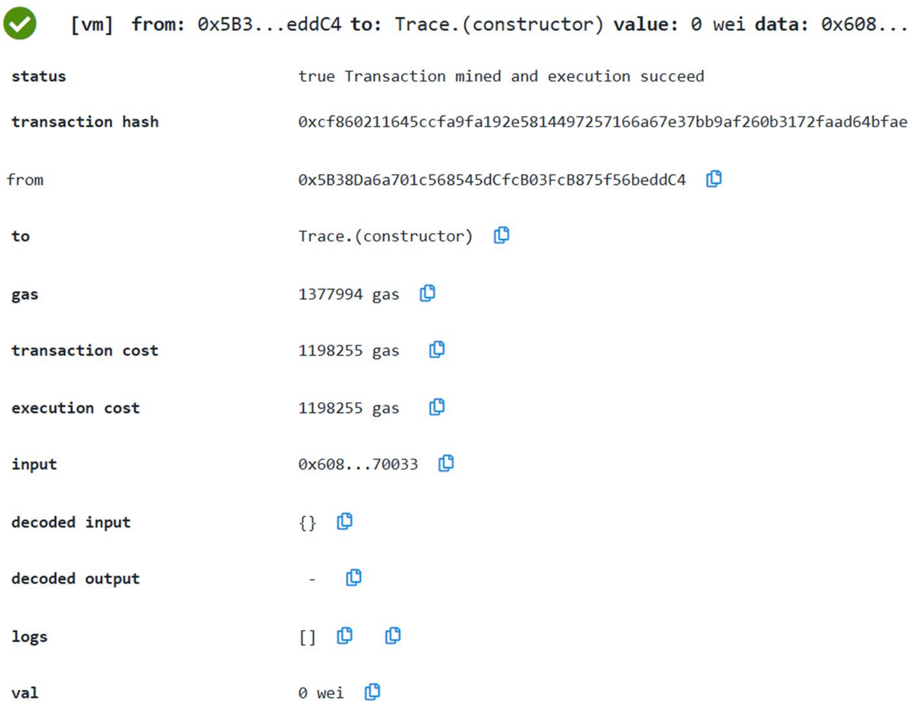
IoT device management involves verifying the devices' identity and facilitating functions to be triggered by the corresponding devices. Verifying the identity of IoT devices is achieved in the same way as user authentication. Smart contract plays an essential role in maintaining IoT device identity. The administrator adds the addresses of IoT devices to the blockchain with a corresponding modifier to provide the necessary function. When these devices call the smart contract, they are first authenticated with the addresses, and after that, the usable function is provided.

5 Implementation and Experimental Analysis

A two-fold implementation and experimental analysis have been conducted to explore the relevant blockchain for food traceability. First, the proposed model is implemented using Ethereum and then enhanced using EOSIO. The detailed implementation steps of the Ethereum and the proposed EOSIO have been presented in Sect. 5.1 and 5.2. Further, the proposed EOSIO has been experimentally validated in sect. 5.3 in terms of execution cost, consensus algorithm, authentication, authorization, scalability, block-confirmation time, block production rate, and throughput.

5.1 Implementation Using Ethereum

This section discusses the implementation of the food traceability model using the Ethereum blockchain. In addition, it also detail how smart contracts are used to set up traceability systems and user management modules. Ethereum is a public, permissionless blockchain network. It utilizes the solidity programming language for developing smart contracts. The smart contracts are utilized to facilitate the food traceability system and the interactions between stakeholders. The smart contract is deployed and tested in two environments: Remix virtual machine [58], and Ropsten test network [59]. Remix is an online Ethereum platform that enables writing, compiling, deploying, running, and

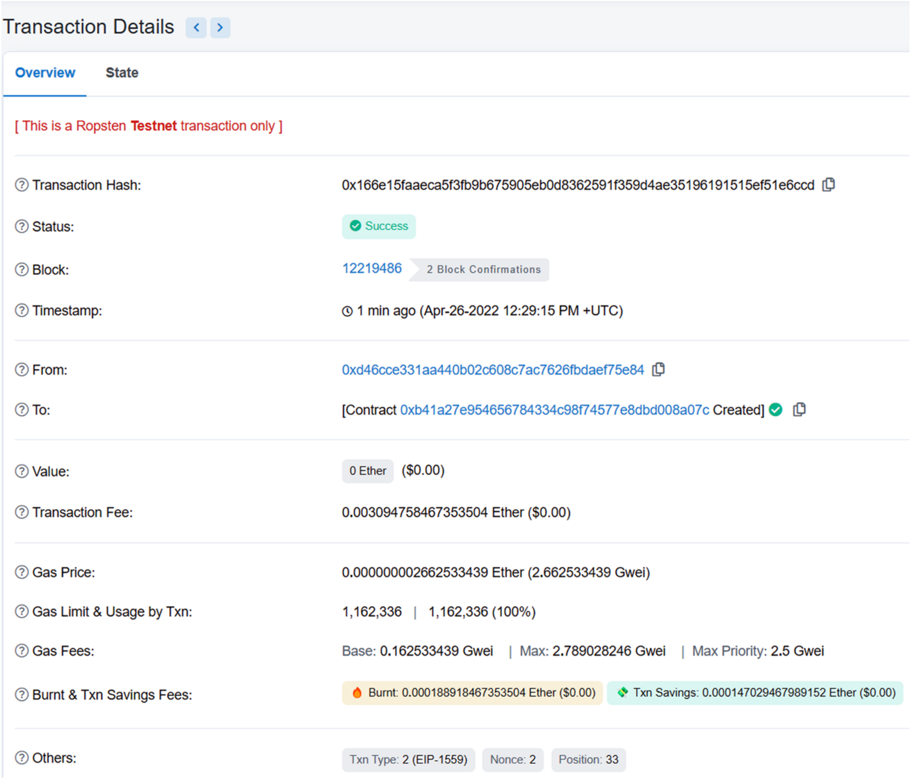


```
✓  [vm]  from: 0x5B3...eddC4 to: Trace.(constructor) value: 0 wei data: 0x608...

status               true Transaction mined and execution succeed

transaction hash     0xcf860211645ccfa9fa192e5814497257166a67e37bb9af260b3172faad64bfae

from                 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

to                   Trace.(constructor)

gas                  1377994 gas

transaction cost     1198255 gas

execution cost       1198255 gas

input                0x608...70033

decoded input        {}

decoded output        -

logs                 []

val                  0 wei
```

Fig. 5 Transaction details of contract creation in remix virtual machine

interacting with smart contracts. The Remix platform is used to deploy smart contracts on the Ethereum main network, different test networks, local machines, and a virtual machine (VM). VM is helpful in the initial phase of smart contract development. The Remix VM offers a variety of test accounts with synthetic Ethers, allowing users to test gas usage and optimize smart contracts without worrying about connectivity or mining. In the Ropsten test network, the working scenario of the food traceability system is simulated by deploying the smart contracts. The Ropsten test network is similar to the Ethereum main network in the currently available open-source test networks. Hence, the metrics obtained from the Ropsten networks are considered for evaluating the food traceability system using the Ethereum blockchain. The contract creation details using the Remix VM and Ropsten test network are depicted in Figs. 5 and 6 respectively. Further, the key attributes used in the Remix VM and Ropsten test network transaction information are listed below.

- Status: Indicates the success or failure of transaction.
- Block: Indicates the block number of transaction.
- Timestamp: Provides the time of addition of block in the blockchain
- From: The address of the user invoking the smart contract.
- To: Address of the smart contract that uniquely identifies the contract in the blockchain network.
- Gas Limit / Usage: the amount of gas allocated for the transaction to execute.
- Transaction Fee: the amount of work performed to add a block in the blockchain.



[This is a Ropsten **Testnet** transaction only]

Transaction Hash:	0x166e15faaeca5f3fb9b675905eb0d8362591f359d4ae35196191515ef51e6ccd
Status:	✓ Success
Block:	12219486 2 Block Confirmations
Timestamp:	1 min ago (Apr-26-2022 12:29:15 PM +UTC)
From:	0xd46cce331aa440b02c608c7ac7626fbdaef75e84
To:	[Contract 0xb41a27e954656784334c98f74577e8dbd008a07c Created] ✓
Value:	0 Ether ($0.00)
Transaction Fee:	0.003094758467353504 Ether ($0.00)
Gas Price:	0.000000002662533439 Ether (2.662533439 Gwei)
Gas Limit & Usage by Txn:	1,162,336 | 1,162,336 (100%)
Gas Fees:	Base: 0.162533439 Gwei | Max: 2.789028246 Gwei | Max Priority: 2.5 Gwei
Burnt & Txn Savings Fees:	Burnt: 0.000188918467353504 Ether ($0.00) Txn Savings: 0.000147029467989152 Ether ($0.00)
Others:	Txn Type: 2 (EIP-1559) Nonce: 2 Position: 33

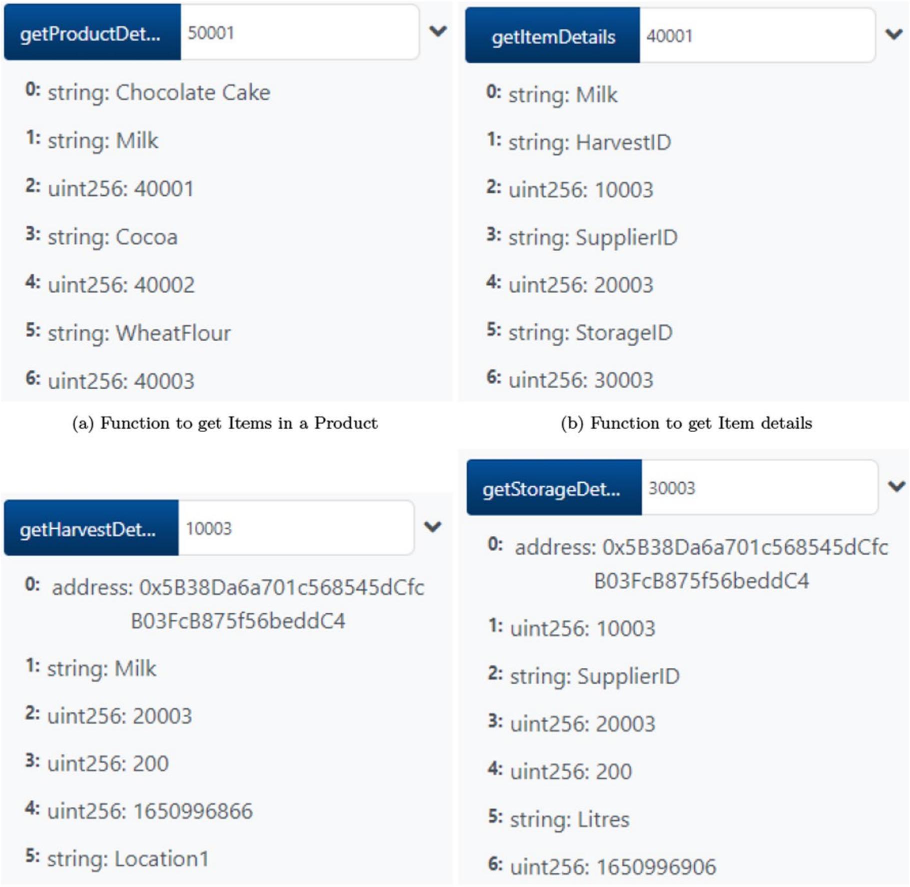
Fig. 6 Transaction details of contract creation in Ropsten test network

Table 2 Ethereum transaction cost

Transaction	Block Confirmation Time (in Seconds)	Gas Used (in Ether)	Gas Price (in Gwei)	Transaction Cost (in Ether)	Value (USD)
Contract creation	58	2.4E+06	3.17	0.0076	21.96
adduser	40	9.6E+04	3.39	0.00032	0.94
addharvest	33	4.3E+05	3.55	0.00153	4.42
addstorage	27	3.7E+06	3.02	0.00113	3.28
additem	23	2.7E+05	2.73	0.00075	2.18
addproduct	30	2.4E+06	2.64	0.00063	1.84

- Value: amount that is transferred in the transaction. This is only applicable if the smart contract functions are payable.

The Ethereum-based food traceability method uses the amount of gas allocated to measure the completed work. This approach provides a priority fee higher than the



```
getProductDet...   50001
0: string: Chocolate Cake
1: string: Milk
2: uint256: 40001
3: string: Cocoa
4: uint256: 40002
5: string: WheatFlour
6: uint256: 40003
```

(a) Function to get Items in a Product

```
getItemDetails   40001
0: string: Milk
1: string: HarvestID
2: uint256: 10003
3: string: SupplierID
4: uint256: 20003
5: string: StorageID
6: uint256: 30003
```

(b) Function to get Item details

```
getHarvestDet...   10003
0: address: 0x5B38Da6a701c568545dCfc
            B03FcB875f56beddC4
1: string: Milk
2: uint256: 20003
3: uint256: 200
4: uint256: 1650996866
5: string: Location1
```

(c) Function to get harvest details

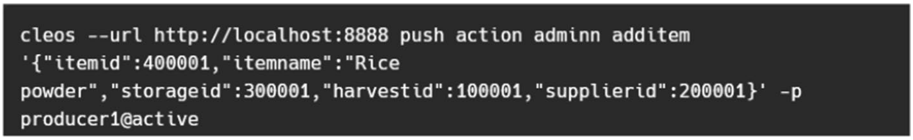
```
getStorageDet...   30003
0: address: 0x5B38Da6a701c568545dCfc
            B03FcB875f56beddC4
1: uint256: 10003
2: string: SupplierID
3: uint256: 20003
4: uint256: 200
5: string: Litres
6: uint256: 1650996906
```

(d) Function to get storage details

Fig. 7 Working of the traceability system

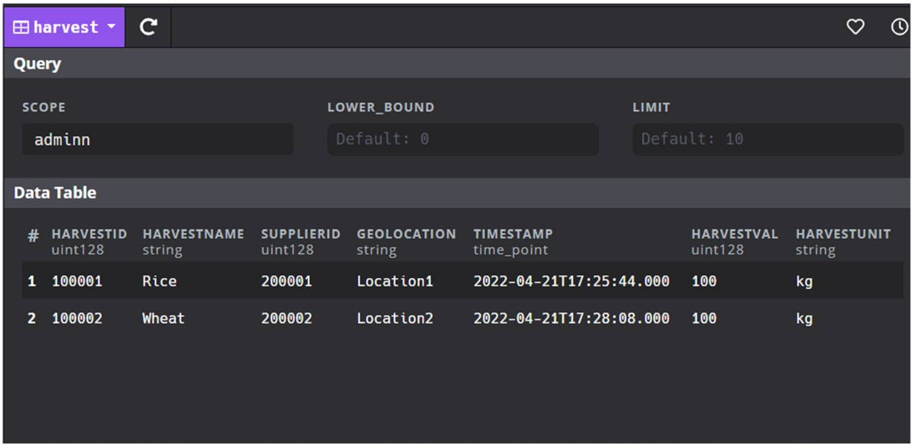
base price to prioritize and execute essential transactions. The same can be observed in Fig. 6. Moreover, the Table 2 provides the summary of the gas usage, block confirmation time, and transaction cost for the functions in the food traceability system. The values mentioned in the table are obtained using the Ropsten test network. The transaction fee is calculated by multiplying the sum of the gas price and priority fees with the gas usage. As the gas price is continually changing, repeating the same transaction may result in different transaction fees. The transaction cost in Table 2 is the weighted average of the values obtained in different runs.

The flow of the proposed Ethereum-based food traceability system has been presented in Fig. 7. Traceability functions are similar to those described in Fig. 4. First, the getProductDetails() function is used to get the product ID, product name, and item IDs of any particular item, and then to get the traceability details of any specific item, the getItemDetails() function is used. In the subsequent step, Harvest ID and Storage ID are used to obtain the raw materials used in the product.



```
cleos --url http://localhost:8888 push action adminn additem
'{"itemid":400001,"itemname":"Rice
powder","storageid":300001,"harvestid":100001,"supplierid":200001}' -p
producer1@active
```

Fig. 8 EOSIO input command for item data entry



SCOPE: adminn
LOWER_BOUND: Default: 0
LIMIT: Default: 10

#	HARVESTID uint128	HARVESTNAME string	SUPPLIERID uint128	GEOLOCATION string	TIMESTAMP time_point	HARVESTVAL uint128	HARVESTUNIT string
1	100001	Rice	200001	Location1	2022-04-21T17:25:44.000	100	kg
2	100002	Wheat	200002	Location2	2022-04-21T17:28:08.000	100	kg

Fig. 9 EOSIO harvest data multi-index table

5.2 Proposed EOSIO Based Food Traceability Model

It has been observed from the literature that the EOSIO-based models show better efficacy than the Ethereum-based models. Therefore, EOSIO blockchain-based traceability [60] has been presented in this paper to enhance food traceability. The details of the EOSIO blockchain have been presented in Sect. 2.5. In the EOSIO blockchain, business logic is implemented on the blockchain by utilizing smart contracts. The proposed model utilizes EOSIO libraries defined in C++ to implement the food traceability system. A system with the following specifications has been used to implement the proposed food traceability model.

- *CPU: Intel i7*
- *Memory: 16 GB*
- *Disk: 128 GB NVMe SSD*

For the proposed model, NVMe SSDs are recommended for higher disk read/write speeds. To implement the proposed approach, EOS Studio [61] and Docker [62] have been used, in which Docker provides an environment to run the EOSIO and EOS Studio provides the development environment. Moreover, with the help of EOS Studio and Docker, smart contracts can be built and deployed on local nodes, test networks, and the EOSIO main network. The local node supports both single-node and multiple node deployments and is also helpful in the initial smart contract development stage. Additionally, the EOS Studio are used as a contract inspector and network manager, allowing users to interact with deployed contracts and fetch the resource utilization metrics. EOSIO uses Delegated-Proof-of-Stake (DPoS) as the consensus mechanism. In DPoS, validators/producers are responsible for producing a block and are delegated by different stakeholders.

Furthermore, Fig. 8 depicts the commands used in EOSIO to deploy and test smart contract actions. Cleos is the command-line tool that runs on every EOSIO node and is used to interface with nodeOs. The NodeOs is set up to handle smart contracts, validate transactions, generate blocks containing valid transactions, and confirm blocks before they are recorded on the blockchain. The –url indicates the http/https URL where nodeOs is running, while the push subcommand is used to push transactions to the blockchain. The smart contract actions in the adminn account are addharvest, addstorage, and additem. Further, producer1@active denotes that the active user has the needed permission key. The EOSIO blockchain provides a distinct layer of authentication and authorization for users. It can be claimed to be more secure because authentication and authorization are not required to be implemented individually. Moreover, Fig. 9 displays the multi-index table for the actions of the deployed data. Moreover, Fig. 9 displays the multi-index table for the actions of the deployed data. The multi-index table of the proposed EOSIO blockchain is similar to Ethereum's multi-index table, which is used at a later stage for providing traceability.

5.3 Experimental Analysis

This section presents the performance evaluation of the proposed layered architecture for food traceability. The following performance parameters have been considered for food traceability.

- Variation in gas usage with increasing string length.
- Transaction fee of the implementation
- Gas usage of smart contract functions.
- Block confirmation time.
- Resource consumption.

Fig. 10 String Length vs Gas Usage

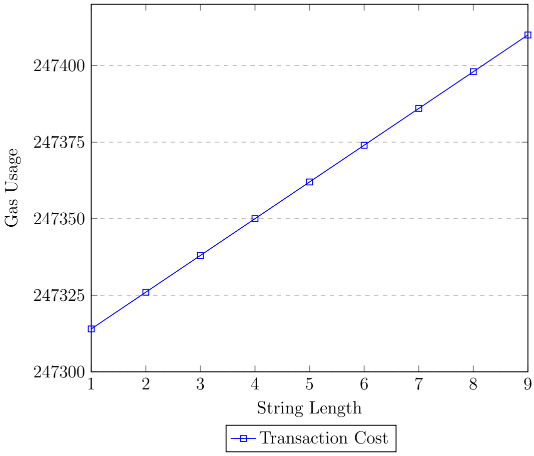


Table 3 Comparison of Ethereum and EOSIO

	Ethereum	EOSIO
Permission Management	X	✓
Language Used	Solidity	C++
Block Production rate	20 s	0.5 s
Scalability	Low	Very High
Consensus Algorithm	Proof-of-Work	Delegated-Proof-of-Stake
Throughput	Low	High
Block confirmation time	56 s	1 s

Figure 10 depicts the relationship between gas usage and input string length in Ethereum. The plotted graph highlights that gas usage and input length are directly proportional. Hence, the use of strings in the blockchain should be limited. Further, costs for performing various transactions in Ethereum have been tabulated in Table 2. The transaction costs can be reduced if the private implementation of Ethereum is employed. Besides, table also highlights the block production time in Ethereum. The weighted average of all test results produces a 56-second block production time for Ethereum. The latency, throughput, and block production rate are affected by such a large block production time. Therefore, an improved EOSIO-based on DPoS implementation has been used in the proposed traceability system to enhance the latency, block production rate, scalability, and throughput. The improved EOSIO-based on DPoS implementation achieved block confirmation time in the range of 1 s. Other performance parameters such as throughput and block production rate have also improved. Moreover, the proposed EOSIO-based implementation is compared with the Ethereum-based implementation in Table 3 to validate the efficacy of the proposed approach. It can be observed from the table that the proposed traceability system outperforms the Ethereum-based food traceability system in terms of permission management, block production rate, throughput, and block confirmation rate. Additionally, in the EOSIO blockchain, the payer can be delegated for action in the smart contract. Such a

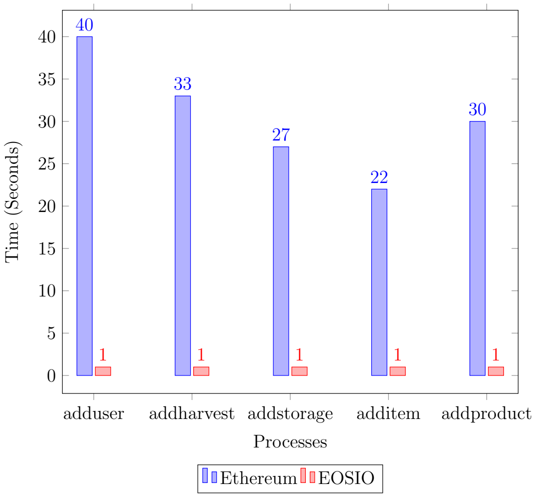
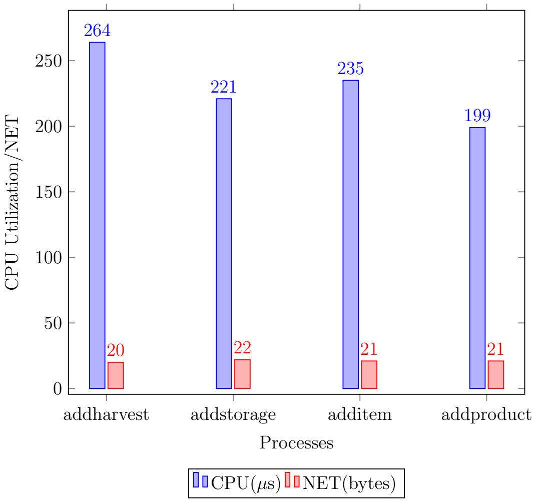


Fig. 11 Block confirmation time

Fig. 12 EOSIO resource utilisation



configuration control allows the transactions to be independent of the monetary concept and further improves the practicality of EOSIO-based implementation. Additionally, the EOSIO permission management further improves the efficacy of the proposed approach.

Furthermore, the block confirmation time of the Ethereum and EOSIO-based traceability systems is compared in Fig. 11. It can be envisioned from the figure that the proposed EOSIO-based traceability system performs better than the Ethereum-based implementation. Besides, resource utilization of the proposed food traceability system has also been depicted in Fig. 12. It can be pertinent from the above analysis that the proposed EOSIO-based food traceability system is much superior to the Ethereum-based system.

6 Conclusion and Future Works

This paper introduced a Blockchain-IoT-based model using EOSIO to achieve a secure, transparent, immutable, and scalable food traceability system in smart cities, with inherent authentication and authorization features. The overall functionality of the proposed food traceability system can be detailed in three phases. The first phase identifies suitable traceability units to provide better transparency and traceability. After that, a layered architecture of the food traceability system is presented in the second phase to improve security. Further, layered architecture is implemented in the third phase using two blockchain platforms, Ethereum and EOSIO. Both implementations are extensively studied to identify the better-suited blockchain for food traceability systems. It has been identified from the analysis that the EOSIO-based traceability system is more suitable than Ethereum. The strengths of EOSIO allow for better implementation of the proposed model. With the role-based permission management in EOSIO, a more secure and practical food traceability system for smart cities has been developed. The framework implemented is a valuable road map for identifying the best-suited blockchain for food traceability, with many enterprise-ready blockchains available today. The implementation was also deployed in test networks to simulate the real-world scenario. As a future improvement, the implementation may be deployed in a real-world scenario and assessed on how well it can handle the complex food supply chain process.

Author Contributions All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content, and (c) approval of the final version.

Funding There has been no significant financial support for this work.

Data Availability Data used in this research are publicly available for research purpose.

Declarations

Conflict of interest Authors have no conflicts of interest associated with this publication.

Ethical Approval Authors declare that this manuscript is original, has not been published before, and is not currently being considered for publication elsewhere.

Consent for Publication We (authors) consent to publish the above research article in this Journal.

References

1. Profiroiu, C. M., Bodislav, D. A., Burlacu, S., & Rădulescu, C. V. (2020). Challenges of sustainable urban development in the context of population growth. *European Journal of Sustainable Development, 9*(3), 51.
2. UN-DESA. (2018). World urbanization prospects: The 2018 revision (st/esa/ser.a/420). United nations. https://desapublications.un.org/publications/2018-revision-world-urbanization-prospects.
3. Dameri, R. P., & Rosenthal-Sabroux, C. (2014). Smart city and value creation. In *Smart city* (pp. 1–12). Springer.
4. M. Intelligence. (2022 - 2027). Smart cities market - growth, trends, covid-19 impact, and forecasts. https://www.mordorintelligence.com/industry-reports/smart-cities-market.
5. Riic, O., Jukic, T., Ridjic, G., Mangafic, J., Buvsatlic, S., & Karamehic, J. (2022). Implementation of blockchain technologies in smart cities, opportunities and challenges. In *Blockchain technologies for sustainability* (pp. 71–89).
6. WHO. (2020). Food safety. https://www.who.int/news-room/fact-sheets/detail/food-safety.
7. Wikipedia. (2016). 2013 horse meat scandal. https://en.wikipedia.org/wiki/2013_horse_meat_scandal.
8. Food, U., Administration, D. (2020). Outbreak investigation of salmonella uganda: Fresh papayas (June 2019).
9. CDC. (2022). E. coli outbreak linked to baby spinach. https://www.cdc.gov/ecoli/2021/o157h7-11-21/index.html.
10. CDC. (2022). Norovirus outbreak linked to raw oysters from british columbia. https://www.cdc.gov/norovirus/outbreaks/index.html.
11. Alfian, G., Syafrudin, M., Farooq, U., Ma'arif, M. R., Syaekhoni, M. A., Fitriyani, N. L., Lee, J., & Rhee, J. (2020). Improving efficiency of rfid-based traceability system for perishable food by utilizing iot sensors and machine learning model. *Food Control, 110*, 107016.
12. González-Amarillo, C. A., Corrales-Muñoz, J. C., Mendoza-Moreno, M. Á., Hussein, A. F., Arunkumar, N., Ramirez-González, G., et al. (2018). An iot-based traceability system for greenhouse seedling crops. *IEEE Access, 6*, 67528–67535.
13. Michelin, R. A., Dorri, A., Steger, M., Lunardi, R. C., Kanhere, S. S., Jurdak, R., & Zorzo, A. F. (2018). Speedychain: A framework for decoupling data from blockchain for smart cities. In *Proceedings of the 15th EAI international conference on mobile and ubiquitous systems: Computing, networking and services* (pp. 145–154).
14. Casino, F., Kanakaris, V., Dasaklis, T. K., Moschuris, S., Stachtiaris, S., Pagoni, M., & Rachaniotis, N. P. (2021). Blockchain-based food supply chain traceability: A case study in the dairy sector. *International Journal of Production Research, 59*(19), 5758–5770.

15. Hasan, H., AlHadhrami, E., AlDhaheri, A., Salah, K., & Jayaraman, R. (2019). Smart contract-based approach for efficient shipment management. *Computers & Industrial Engineering, 136*, 149–159.
16. Shahid, A., Almogren, A., Javaid, N., Al-Zahrani, F. A., Zuair, M., & Alam, M. (2020). Blockchain-based agri-food supply chain: A complete solution. *IEEE Access, 8*, 69230–69243.
17. Ethereum. (2022). Welcome to ethereum. https://ethereum.org/en/.
18. Varavallo, G., Caragnano, G., Bertone, F., Vernetti-Prot, L., & Terzo, O. (2022). Traceability platform based on green blockchain: An application case study in dairy supply chain. *Sustainability, 14*(6), 3321.
19. Baralla, G., Pinna, A., & Corrias, G. (2019). Ensure traceability in european food supply chain by using a blockchain system. In *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)* (pp. 40–47). IEEE.
20. Sun, R.-T., Garimella, A., Han, W., Chang, H.-L., & Shaw, M. J. (2020). Transformation of the transaction cost and the agency cost in an organization and the applicability of blockchain–A case study of peer-to-peer insurance. *Frontiers in Blockchain, 3*, 24.
21. Hepp, T., Wortner, P., Schönhals, A., & Gipp, B. (2018). Securing physical assets on the blockchain: Linking a novel object identification concept with distributed ledgers. In *Proceedings of the 1st workshop on cryptocurrencies and blockchains for distributed systems* (pp. 60–65).
22. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system, Decentralized Business Review 21260.
23. Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *White paper, 3*(37).
24. Smith, S. S. (2020). Blockchain, smart contracts and financial audit implications. *IUP Journal of Accounting Research & Audit Practices, 19*(1), 7–17.
25. Amin, M. R., Zuhairi, M. F., & Saadat, M. N. (2020). Enhanced blockchain transaction: A case of food supply chain management. *Journal of Engineering and Applied Sciences, 15*(1), 99–106.
26. Nam Nguyen, H., AnhTran, H., Fowler, S., & Souihi, S. (2021). A survey of blockchain technologies applied to software-defined networking: Research challenges and solutions. *IET Wireless Sensor Systems, 11*(6), 233–247.
27. Viriyasitavat, W., & Hoonsopon, D. (2019). Blockchain characteristics and consensus in modern business processes. *Journal of Industrial Information Integration, 13*, 32–39.
28. Bamakan, S. M. H., Motavali, A., & Bondarti, A. B. (2020). A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications, 154*, 113385.
29. Sriman, B., Ganesh Kumar, S., & Shamili, P. (2021). Blockchain technology: Consensus protocol proof of work and proof of stake. In *Intelligent computing and applications* (pp. 395–406). Springer.
30. Majeed, U., Khan, L. U., Yaqoob, I., Kazmi, S. A., Salah, K., & Hong, C. S. (2021). Blockchain for iot-based smart cities: Recent advances, requirements, and future challenges. *Journal of Network and Computer Applications, 181*, 103007.
31. Ge, L., Wang, J., & Zhang, G. (2022). Survey of consensus algorithms for proof of stake in blockchain, Security and Communication Networks 2022.
32. Sun, Y., Yan, B., Yao, Y., & Yu, J. (2021). Dt-dpos: A delegated proof of stake consensus algorithm with dynamic trust. *Procedia Computer Science, 187*, 371–376.
33. Wang, B., Li, Z., & Li, H. (2020). Hybrid consensus algorithm based on modified proof-of-probability and dpos. *Future Internet, 12*(8), 122.
34. Sayeed, S., Marco-Gisbert, H., & Caira, T. (2020). Smart contract: Attacks and protections. *IEEE Access, 8*, 24416–24427.
35. Zheng, Z., Xie, S., Dai, H.-N., Chen, W., Chen, X., Weng, J., & Imran, M. (2020). An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems, 105*, 475–491.
36. Pierro, G. A., Tonelli, R., & Marchesi, M. (2020). An organized repository of ethereum smart contracts' source codes and metrics. *Future Internet, 12*(11), 197.
37. Hilbig, A., Lehmann, D., & Pradel, M. (2021). An empirical study of real-world webassembly binaries: Security, languages, use cases. In *Proceedings of the web conference 2021* (pp. 2696–2708).
38. Logu, K., Devi, T., Deepa, N., Gayathri, N., & Rakesh kumar, S. (2022). A real-time monitoring tool for analyzing ethereum digital currency in global business transaction. In *Blockchain security in cloud computing* (pp. 167–188). Springer.
39. Queralta, J. P. & Westerlund, T. (2021). Blockchain for mobile edge computing: Consensus mechanisms and scalability. In *Mobile edge computing* (pp. 333–357). Springer.

40. Janssen, M., Weerakkody, V., Ismagilova, E., Sivarajah, U., & Irani, Z. (2020). A framework for analysing blockchain technology adoption: Integrating institutional, market and technical factors. *International Journal of Information Management, 50*, 302–309.
41. Oliva, G. A., Hassan, A. E., & Jiang, Z. M. J. (2020). An exploratory study of smart contracts in the ethereum blockchain platform. *Empirical Software Engineering, 25*(3), 1864–1904.
42. Huang, Y., Jiang, B., & Chan, W. K. (2020). Eosfuzzer: Fuzzing eosio smart contracts for vulnerability detection. In *12th Asia-Pacific Symposium on Internetware* (pp. 99–109).
43. Kaur, S., Chaturvedi, S., Sharma, A., & Kar, J. (2021). A research survey on applications of consensus protocols in blockchain, Security and Communication Networks 2021.
44. Dabbene, F., Gay, P., & Tortia, C. (2016). Radio-frequency identification usage in food traceability. In *Advances in food traceability techniques and technologies* (pp. 67–89). Elsevier.
45. Farooq, U., Tao, W., Alfian, G., Kang, Y.-S., & Rhee, J. (2016). Epedigree traceability system for the agricultural food supply chain to ensure consumer health. *Sustainability, 8*(9), 839.
46. Abdel-Basset, M., Manogaran, G., & Mohamed, M. (2018). Internet of things (iot) and its impact on supply chain: A framework for building smart, secure and efficient systems. *Future Generation Computer Systems, 86*, 614–628.
47. Kim, Y. G., & Woo, E. (2016). Consumer acceptance of a quick response (qr) code for the food traceability system: Application of an extended technology acceptance model (tam). *Food Research International, 85*, 266–272.
48. Verdouw, C., Robbemond, R. M., Verwaart, T., Wolfert, J., & Beulens, A. J. (2018). A reference architecture for iot-based logistic information systems in agri-food supply chains. *Enterprise Information Systems, 12*(7), 755–779.
49. Khan, M. A., & Salah, K. (2018). Iot security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems, 82*, 395–411.
50. Yu, Y., Li, Y., Tian, J., & Liu, J. (2018). Blockchain-based solutions to security and privacy issues in the internet of things. *IEEE Wireless Communications, 25*(6), 12–18.
51. Vangala, A., Das, A. K., Kumar, N., & Alazab, M. (2020). Smart secure sensing for iot-based agriculture: Blockchain perspective. *IEEE Sensors Journal, 21*(16), 17591–17607.
52. Kamath, R. (2018). Food traceability on blockchain: Walmart's pork and mango pilots with ibm. *The Journal of the British Blockchain Association, 1*(1), 3712.
53. Salah, K., Nizamuddin, N., Jayaraman, R., & Omar, M. (2019). Blockchain-based soybean traceability in agricultural supply chain. *IEEE Access, 7*, 73295–73305.
54. Tsang, Y. P., Choy, K. L., Wu, C. H., Ho, G. T. S., & Lam, H. Y. (2019). Blockchain-driven iot for food traceability with an integrated consensus mechanism. *IEEE Access, 7*, 129000–129017.
55. Uddin, M. A., Stranieri, A., Gondal, I., & Balasubramanian, V. (2019). A decentralized patient agent controlled blockchain for remote patient monitoring. In *2019 International conference on wireless and mobile computing, networking and communications* (WiMob) (pp. 1–8). IEEE.
56. Liu, X., Barenji, A. V., Li, Z., Montreuil, B., & Huang, G. Q. (2021). Blockchain-based smart tracking and tracing platform for drug supply chain. *Computers & Industrial Engineering, 161*, 107669.
57. Caro, M. P., Ali, M. S., Vecchio, M., & Giaffreda, R. (2018). Blockchain-based traceability in agri-food supply chain management: A practical implementation. In *2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany)* (pp. 1–4) IEEE.
58. Remix. (2022). Deploy & run transactions in the blockchain. https://remix-project.org/.
59. Wu, S. X., Wu, Z., Chen, S., Li, G., & Zhang, S. (2021). Community detection in blockchain social networks. *Journal of Communications and Information Networks, 6*(1), 59–71.
60. EOSIO. (2022). Eosio fast, flexible, and forward-driven. https://eos.io/.
61. Studio, E. (2022). Eos studio a graphic ide to expedite your dapp development. https://www.eosstudio.io/.
62. Docker. (2022). Docker developers love docker. businesses trust it. https://www.docker.com/.

Ashish Kumar Tripathi (Member, IEEE) received his M.Tech. and Ph.D. degrees from the Department of Computer Science and Engineering, Delhi Technological University, Delhi, India, in 2013 and 2019, respectively. He is currently working as an Assistant Professor with the Department of Computer Science and Engineering, Malaviya National Institute of Technology (MNIT), Jaipur, India. His research interests include big data analytics, social media analytics, the Internet of Things, and video and image data processing. Dr. Tripathi is an active reviewer for several journals of repute.

K. Akul Krishnan received his B.Tech in Computer Science and Engineering from the Cochin University of Science and Technology in 2018. After that, he worked in the IT industry for about two years as a secops engineer. He is currently pursuing his M.Tech in Computer Engineering and Information Security from the Malaviya National Institute of Technology Jaipur. His current research interests include blockchain technology and deep learning.

Avinash Chandra Pandey (Member, IEEE) is currently working as an Assistant Professor in the Discipline of Computer Science and Engineering of the PDPM Indian Institute of Information Technology Design and Manufacturing, Jabalpur. He has more than eight years of teaching and research experience. He has guided many M.Tech. dissertations and B.Tech projects. He has published more than 20 journal and conference papers in text mining, data mining, and soft computing. He has been an active member of many organizing committees for various conferences and workshops. His research areas include data analytics, text mining, and soft computing.