Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 10, No. 2, April, 2026

BIoT-DApp: A Prototype for Real Time Traceability in Agricultural Supply Chains

Sajid Safeer [1*], Victoria Lemieux [2], Chang Lu [3], Cataldo Pulvento [4]

[1] *International School of Advanced Studies, University of Camerino, 62032 Camerino, Italy.*

[2] *School of Information, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada.*

[3] *School of Business, University of Fraser Valley, Abbotsford, BC V2S 7M8, Canada.*

[4] *Department of Soil, Food and Plant Sciences, University of Bari Aldo Moro, 70126 Bari, Italy.*

Abstract

Agricultural supply chains frequently experience inefficiencies, including a lack of transparency, post-harvest losses, and inequitable compensation for stakeholders. This study aims to develop and evaluate a Blockchain-IoT decentralized application (BIoT-DApp) that enhances traceability, efficiency, and resilience in agri-food supply chains. Utilizing a lab-based prototyping methodology, the system integrates Ethereum smart contracts with IoT sensors to automate workflows from cultivation to retail, employing a hybrid architecture that stores raw sensor data off-chain while anchoring cryptographic hashes on-chain. The methods involve designing role-specific smart contracts, managing batch life cycles across six stages, and conducting real-time environmental monitoring through IoT data processed by Raspberry Pi, with deployment and testing performed on the Sepolia testnet. The findings demonstrate automated quality control, reduced storage costs through optimized on-chain practices, and seamless product ownership transfers validated by four role-based MetaMask accounts representing a farmer, wholesaler, retailer, and end-user. Core functions were executed successfully, with gas costs ranging from 30,000 (data logging) to 112,300 (batch initialization), confirming both cost efficiency and scalability. The novelty of this work lies in bridging blockchain theory and practice by providing a modular, adaptable prototype capable of supporting perishable agricultural supply chains globally. This offers policymakers and agri-tech developers actionable insights for decentralized solutions in resource-constrained environments.

Keywords:

Agriculture;
Blockchain;
Internet of Things;
Decentralized Application;
Supply Chain.

Article History:

Received:	10	June	2025
Revised:	07	March	2026
Accepted:	12	March	2026
Published:	01	April	2026

1- Introduction

Agricultural supply chains are fundamental to global food security, rural livelihoods, and economic sustainability. Despite their critical importance, many agri-food systems continue to face persistent challenges, including fragmented coordination among stakeholders, limited transparency, post-harvest losses, and unequal value distribution [1]. These issues are particularly pronounced in supply chains for perishable commodities, where time sensitivity and quality degradation compound operational risks and widen existing inefficiencies. Alongside these structural vulnerabilities, climate change, market volatility, and escalating consumer expectations regarding food safety and traceability have further intensified pressure on conventional supply chain structures. Collectively, these pressures underscore a growing imperative for digital solutions capable of enhancing transparency, efficiency, and resilience across agricultural supply chains [2].

Within this broader agricultural context, the Italian tomato sector offers a highly instructive case study, owing to its considerable economic scale and structural complexity. Tomatoes rank among Italy's most important horticultural

* **CONTACT**: sajid.safeer@unicam.it

DOI: https://doi.org/10.28991/ESJ-2026-010-02-02

commodities, contributing substantially to rural employment and export revenues. According to the USDA, the Italian processed tomato industry was valued at approximately USD 5.4 billion in 2022, accounting for 14.8% of global production and 56.5% of European output, with a production volume of 5.5 million metric tons placing Italy third among global tomato processors, after the United States and China [3]. Despite this strong performance, the sector is beset by recurring inefficiencies, including production losses, fragmented value chains, price volatility, and mounting exposure to climate-related shocks. The European Commission estimates that weak supply chain management practices can reduce productivity by up to 30%, significantly elevating costs for producers [4]. These challenges erode farm incomes, weaken traceability, and undermine consumer confidence in product quality and safety, underscoring the need for more robust and integrated supply chain solutions.

Recent advances in digital technologies, including sensor networks, cloud-based platforms, and decision support systems, offer promising pathways for addressing these long-standing challenges. However, their adoption within tomato supply chains remains uneven and largely fragmented. Where digital tools have been implemented, they are frequently deployed in isolation, with farm-level monitoring systems disconnected from logistics, processing, and market operations. This absence of integration restricts end-to-end, data-driven supply chain management and diminishes the effectiveness of traceability mechanisms. For highly perishable crops such as tomatoes, these limitations translate into higher post-harvest losses, delayed settlements, and weaker enforcement of quality standards, increasing the urgency for comprehensive, integrated digital architectures [5].

Blockchain technology has emerged as a particularly promising approach to overcoming these limitations, enabling immutable, verifiable records and automated transactions through smart contracts. Since its introduction in 2008, blockchain has evolved well beyond cryptocurrency applications to underpin provenance tracking, traceability, and supply chain coordination [6, 7]. When integrated with the IoT, blockchain systems can directly link sensor-verified events such as harvesting conditions, storage temperatures, or transportation timelines to tamper-resistant digital records. This convergence enhances transparency and enables automated processes, including condition-based payments and compliance verification [8]. While several pilot studies have demonstrated the technical feasibility of blockchain-IoT integration, concerns persist regarding deployment complexity, cost efficiency, and accessibility, particularly for smallholder farmers [9].

The existing literature on blockchain applications in agriculture broadly addresses three core themes: traceability and food safety, sustainability and climate resilience, and economic inclusion. With respect to traceability, blockchain's immutable ledger has been widely recognized for strengthening provenance and reducing food fraud. Initiatives such as IBM Food Trust have demonstrated that shared, auditable data records can improve transparency and reduce waste across agri-food systems [10]. For perishable products like tomatoes, timely and reliable provenance information is essential to maintaining quality and minimizing losses [11]. Smart contracts further extend these capabilities by enabling rule-based automation, such as the verification of quality criteria prior to payment execution. However, many existing solutions focus primarily on data recording and visualization, with limited integration of real-time sensor data and minimal participation by small-scale producers [12].

The integration of IoT technologies considerably expands these capabilities by enabling continuous, sensor-based monitoring of parameters such as soil moisture, temperature, and geolocation. When immutably recorded on blockchain platforms, such data support condition-based automation and adaptive responses to environmental variability. For instance, tracking fertilizer inputs and irrigation practices can enhance compliance with EU food safety and sustainability standards while reducing contamination risks [13]. IoT-enabled irrigation control can also improve water use efficiency by dynamically adjusting schedules based on soil moisture thresholds, an increasingly vital feature for climate-resilient tomato production [14]. Although participatory governance models that involve farmers in data validation can further strengthen trust and system legitimacy, few studies have delivered reproducible architectures that operationalize these approaches within fragmented, perishable crop supply chains [15].

Beyond traceability and sustainability, blockchain technology also holds significant potential to reshape economic relationships within agricultural supply chains. By enabling decentralized marketplaces and direct buyer-seller interactions, blockchain systems can reduce dependence on intermediaries and promote fairer value distribution. Empirical studies indicate that such mechanisms can increase farmer revenues and reduce exposure to exploitative brokerage practices [16, 17]. Smart contracts triggered by IoT verified delivery events can substantially reduce settlement times, thereby improving liquidity for producers [18]. Combined with data-driven analytics, blockchain-based systems can further support more efficient resource allocation, with pilot studies reporting reductions in input waste and improvements in operational efficiency. Nevertheless, most existing research addresses these dimensions in isolation, without delivering integrated and scalable solutions suited to real-world agricultural contexts [19].

A limited number of recent studies have proposed frameworks that combine traceability, automated payments, and sensor data management within unified architectures [20]. While these contributions represent meaningful progress, they often lack empirical validation for perishable crops, fail to address the cost and data integrity trade-offs associated with hybrid on-chain and off-chain designs, and rarely account for the adoption barriers faced by smallholder farmers. These gaps are particularly acute in the Italian tomato sector, where climate variability and delayed payments remain persistent challenges, yet no tailored BIoT-DApp has been empirically tested to address both technical and socio-economic constraints simultaneously [21]. Furthermore, broader adoption of BIoT solutions continues to be constrained by technical complexity, implementation costs, and gaps in digital literacy among agricultural stakeholders.

In response to these challenges, this study develops and evaluates a practical BIoT-DApp designed specifically for the Italian tomato supply chain. The proposed system integrates IoT sensor data with a hybrid on-chain and off-chain blockchain architecture and smart contracts to enable real-time monitoring, traceability, and automated settlements. The prototype is empirically evaluated using the Sepolia testnet to assess key performance metrics, including transaction cost, latency, and system throughput. Beyond technical validation, the study examines operational constraints and socio-economic barriers encompassing affordability and digital skill requirements and discusses strategies to enhance accessibility for smallholder farmers. By situating the proposed solution within the existing literature and relevant policy context, this research bridges technical feasibility with practical adoption considerations, contributing to the transparency, resilience, and sustainability of one of Europe's most significant agricultural supply chains.

1-1- Paper Organization

Building on the challenges and research gaps identified in the preceding discussion, this paper is structured to guide the reader systematically from architectural design through empirical validation to broader implications for agricultural policy and practice. Section 2 details the four-layer architecture of the proposed BIoT-DApp, elaborating on its smart contract logic and the hybrid on-chain/off-chain data workflow that underpins real-time monitoring and automated settlements. Section 3 presents empirical validation of the system, reporting performance outcomes including transaction cost, latency, and throughput derived from deployment on the Sepolia testnet. Section 4 critically evaluates the system's performance against three interconnected dimensions: scalability challenges, climate resilience outcomes, and the socio-economic impacts for smallholder farmers, thereby contextualizing technical findings within the practical realities of the Italian tomato supply chain. Finally, Section 5 synthesizes the study's contributions, proposes policy frameworks to support broader blockchain adoption in agriculture, and identifies directions for future research in decentralized agri-tech systems.

2- Material and Methods

This study adopted a laboratory-based prototyping approach [22] to design, develop, and evaluate a BIoT-DApp tailored for the Italian tomato supply chain. The methodology centered on simulating the integration of IoT sensing devices with blockchain technology, including the development of smart contracts to manage and automate key supply chain processes. A detailed overview of the system architecture, technological components, and implementation workflow is illustrated in Figure 1.

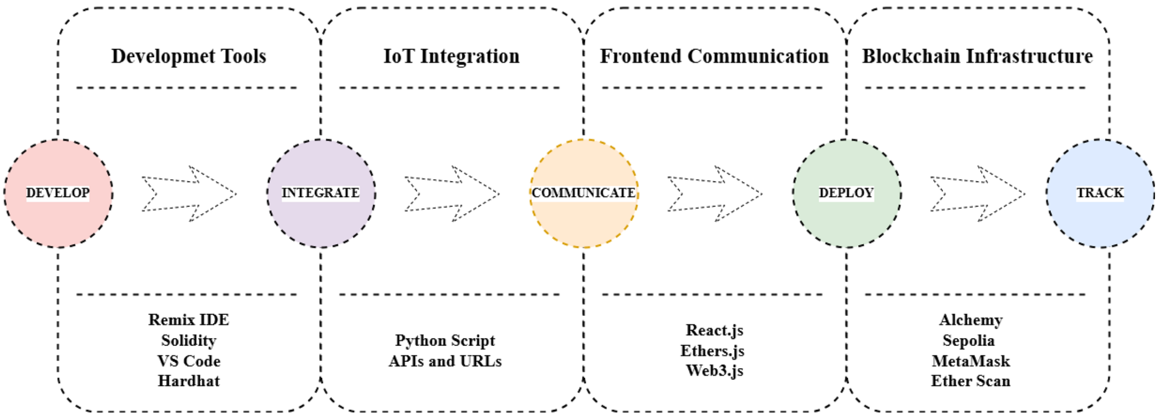

Figure 1. Architectural overview of BIoT-DApp prototype

The proposed BIoT-DApp is organized into four interdependent layers, each fulfilling a distinct functional role within the overall system. The IoT Layer is responsible for sensing and collecting environmental data including temperature, humidity and soil moisture using sensors connected to a Raspberry Pi microcontroller. The Middleware Layer consists of a Python based service, also running on the Raspberry Pi, which processes and summarizes incoming sensor data and facilitates interactions between on-chain and off-chain storage components. The Blockchain Layer employs Ethereum smart contracts written in Solidity to manage the immutability, traceability and security of supply chain records. Finally, the Application Layer provides a web-based frontend developed with React.js and integrated with Ethers.js and Web3.js, enabling stakeholders to interact directly with the blockchain in an accessible and transparent manner. Together, these four layers constitute a coherent end-to-end architecture capable of supporting real time monitoring, automated transactions and verifiable data provenance across the tomato supply chain.

2-1- Development Tools and Integration

The BIoT-DApp was developed using a carefully selected set of tools and frameworks designed to ensure security, scalability, and usability. Each layer encompassing smart contract development, IoT integration, frontend interface, and blockchain infrastructure was configured to meet the specific functional requirements of agricultural supply chains. Together, these components formed a coherent and interoperable system capable of delivering end-to-end functionality under realistic operating conditions.

2-1-1- Smart Contract Development

Ethereum-based smart contracts were authored and debugged using Remix IDE in conjunction with Solidity, which provided real-time syntax validation and streamlined deployment features, consistent with approaches demonstrated in prior studies [23]. Building on this foundation, Hardhat was employed to facilitate local testing and subsequent deployment to the Sepolia testnet. Its advanced debugging capabilities proved essential for verifying contract logic and confirming that the smart contracts operated securely and efficiently under realistic network conditions [24].

2-1-2- IoT Integration

IoT integration was achieved through a Python script developed in Visual Studio Code, which aggregated sensor readings and generated SHA-256 hashes to ensure data integrity before transmission. These hashed data summaries were subsequently synchronized with the blockchain via the *WeatherRecord.sol* smart contract, establishing a verifiable and tamper resistant link between off-chain raw data and on-chain records. To optimize scalability and reduce on-chain storage costs, raw sensor datasets were offloaded to a local server, allowing the system to maintain decentralized storage efficiencies while preserving transparency and auditability across the supply chain [25].

2-1-3- Frontend Development

The BIoT-DApp user interface was developed using React.js, whose component-based architecture enabled the construction of dynamic and responsive dashboards for real time supply chain monitoring [26]. To ensure seamless and reliable blockchain interaction, both Ethers.js and Web3.js were integrated into the frontend layer. Ethers.js was primarily utilized for its lightweight and efficient API, while Web3.js complemented this by supporting broader transaction management tasks. Together, these libraries provided a robust and secure communication layer between the user interface and the underlying blockchain infrastructure [27].

2-1-4- Blockchain Infrastructure

The broader blockchain infrastructure was built on three interconnected tools selected for their reliability and suitability within the Ethereum ecosystem. Alchemy provided high reliability node access to the Sepolia testnet, ensuring stable and consistent connectivity throughout the deployment process [28]. The Sepolia testnet itself served as the primary validation environment, enabling performance testing under conditions closely resembling those of the Ethereum mainnet, without incurring the associated financial costs [29]. To further enhance transparency and operational oversight, Etherscan was employed to monitor and verify on-chain transactions, while MetaMask facilitated secure wallet interactions for different stakeholder roles, enabling seamless execution of role-based operations within the BIoT-DApp ecosystem [30].

2-2- Work Flow and Data Flow

The BIoT-DApp workflow integrates IoT-based data collection, blockchain anchoring, and role-specific transactions to ensure transparency and traceability at every stage of the supply chain. As formalized in the pseudo-code presented in Table 1, the process begins with sensor data capture and off-chain storage on a local server. Secure cryptographic anchoring of data summaries on-chain is then performed through dedicated smart contracts, creating an immutable audit trail linked to physical supply chain events. Stakeholder interactions are subsequently executed via role-based transactions encompassing activities by farmer, wholesaler, retailer, and end-user, while the frontend interface dynamically updates batch and environmental information to support real-time monitoring and stakeholder engagement. The complete algorithmic flow, from system initialization through to the generation of transaction logs and automated state transitions, is detailed in Table 1.

Table 1. Pseudo-code for BIoT-DApp work and data flow

Step	Description
Initialize	Deploy smart contracts and connect IoT sensors
IoT Data Capture	Sensors → Raspberry Pi → Hashing → Store raw data off-chain (local server)
Blockchain Anchoring	Submit hash to *WeatherRecord.sol*; update batch state in *TomatoSupplyChain.sol*
Role Based Transactions	Farmer: Cultivate → Harvest → Price → Ship Wholesaler: Buy → Receive → Price → Ship Retailer: Buy → Receive → Price → Ship End-user: Buy → Confirm Receipt
Frontend Update	Display batch details, weather data and MetaMask connectivity
Outputs	Immutable on-chain records, automated state transitions and role specific transaction logs

3- Results

3-1- BIoT-Dapp Development and Integration

This subsection presents the technical implementation of the BIoT-DApp, encompassing smart contract design, blockchain-IoT integration, and frontend architecture. The development process followed a modular strategy in which each component was built, tested, and integrated incrementally to ensure functional coherence across the full system.

3-1-1- Smart Contract Selection and Functionality Enhancements

The foundational component of the BIoT-DApp is the *TomatoSupplyChain.sol* smart contract, initially proposed by [31] and extended in this study with additional functionality tailored to the Italian tomato supply chain. This contract governs supply chain logic by implementing a finite state machine (FSM) that enforces sequential progression of tomato batches through six distinct stages: Cultivated → Harvested → For Sale → Sold → Shipped → Received. By restricting state transitions to predefined workflows, the contract ensures compliance with supply chain protocols and prevents unauthorized modifications to batch records (see Figure 2).

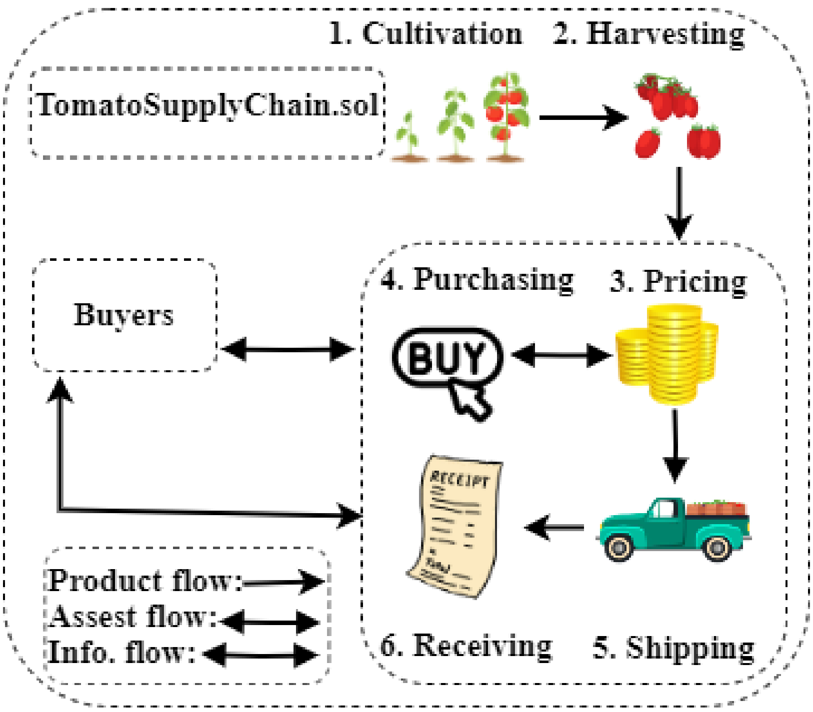

Figure 2. Overview of smart contract-based state transitions

- **Batch Cultivation**

The *cultivateBatch()* function initializes a new batch at the point of cultivation. It stores essential information, including the farmer's ID, farm name, and cumulative batch details; assigns a unique product ID; and records the batch state as "Cultivated". A corresponding event is emitted to allow all participants to track this initial step on the blockchain.

```
Function cultivateBatch(
        uint _batchID,
        address _originFarmerID,
        string memory _originFarmName,
        string memory _originFarmInformation
        ) public onlyFarmer onlyWhenActive {
        Batch storage batch = batches[_batchID];
        batch.batchID = _batchID;
        batch.originFarmerID = _originFarmerID;
        batch.ownerID = _originFarmerID;
        batch.productID = globalItemID + globalBatchID;
        batch.itemID = globalItemID;
        batch.batchState = State.Cultivated;
        batch.farmInfo = FarmInfo({
        farmName: _originFarmName,
        farmInformation: _originFarmInformation});
        globalItemID += 1;
emit Cultivated(_batchID);}
```

- **Batch Harvesting**

The subsequent *harvestBatch()* function lets farmers update the status of a tomato batch as soon as it has been harvested. This function confirms that the batch has already been cultivated before changing its state to “Harvested”. The function also takes into account product notes, like facts about the harvest, and initiates an event to inform contributors that the batch has moved to the subsequent stage in the supply chain.

```
function harvestBatch(uint _batchID, string memory _productNotes) public onlyFarmer
onlyWhenActive {
        Batch storage batch = batches[_batchID];
        require(batch.batchState == State.Cultivated, "Batch is not cultivated");
        batch.productNotes = _productNotes;
        batch.batchState = State.Harvested;
emit Harvested(_batchID);}
```

- **Batch Pricing**

As soon as it is harvested, the batch is readily available for sale through the *setBatchPrice()* function, which allows only the current batch owner to fix the price and update the batch status to “For Sale”. This function certifies that only the lawful owner can outline the sale price, and it can only be achieved once the batch touches specific stages like “Harvested”. By limiting this act to the owner, the contract avoids unlawful users from modifying critical details like pricing. Once the price is effectively set, the batch is marked as for sale, and this adjustment is announced by means of an event, confirming transparency and notifying related stakeholders of the batch’s availability for purchase.

```
function setBatchPrice(uint _batchID, uint _price) public onlyWhenActive {
        Batch storage batch = batches[_batchID];
        require(
        batch.batchState == State.Harvested ||batch.batchState == State.Received,"Batch is not
ready for sale");
        require(msg.sender == batch.ownerID, "Only owner can set the price");
        batch.productPrice = _price;
        batch.batchState = State.ForSale;
emit ForSale(_batchID);}
```

- **Batch Purchasing**

The *buyBatch()* function was designed to help secure transactions by authenticating payments and transferring batch ownership to the purchaser. When a buyer starts buying, the function first verifies that the batch is either available for sale or not and checks that the buyer has sent an adequate amount to cover the set price. If the buyer sends more than the required amount, the function automatically reimburses the extra. However, if the amount is inadequate, the function will not execute. As soon as the actual payment is completed, the ownership of the batch is transmitted from the seller to the buyer and the batch standing is updated to “Sold”. In addition, this function also guarantees that the payment is securely moved to the seller and logs the transaction using an event, providing transparency and keeping a precise record of the sale.

```
function buyBatch(uint _batchID) public payable onlyWhenActive
paidEnough(batches[_batchID].productPrice) checkValue(_batchID) {
        Batch storage batch = batches[_batchID];
        require(batch.batchState == State.ForSale, "Batch is not for sale");
        address previousOwner = batch.ownerID;
        uint price = batch.productPrice;
        batch.ownerID = msg.sender;
        batch.batchState = State.Sold;
        payable(previousOwner).transfer(price);
emit Sold(_batchID);}
```

- **Batch Shipment**

After a sale, the seller is liable for shipping the batch. The *shipBatch()* function permits the seller to mark the batch as shipped, allowing actual tracking of its status. This function makes sure that only the seller, rather than the new owner, can initiate the shipment, maintaining flawless roles within the supply chain. As soon as the shipment is initiated by the seller, the batch's status is updated to "Shipped," and an event is emitted to inform all contributors that the batch is en-route.

```
function shipBatch(uint _batchID) public onlyWhenActive {
        Batch storage batch = batches[_batchID];
        require(batch.batchState == State.Sold, "Batch is not sold");
        address seller = tx.origin;
        require(seller != batch.ownerID, "Current owner cannot ship; only seller can ship");
        batch.batchState = State.Shipped;
emit Shipped(_batchID);}
```

- **Batch Receipt**

Upon the batch’s arrival, the *receiveBatch()* function is utilized by the new owner or buyer to endorse that they have received the batch. Only the current lawful owner is permitted to execute this action, which automatically updates the batch's status to “Received”. This final step confirms full traceability and accountability all through the entire supply chain, from cultivation to delivery, closing the transaction loop.

```
function receiveBatch(uint _batchID) public onlyWhenActive {
        Batch storage batch = batches[_batchID];
        require(batch.batchState == State.Shipped, "Batch is not shipped");
        require(msg.sender == batch.ownerID, "Only the current owner can receive the batch");
        batch.batchState = State.Received;
emit Received(_batchID);}
```

3-1-2- Customization and Task Definitions

To ensure structured and secure participation within the tomato supply chain, stakeholder roles and permissions are defined through a set of auxiliary contracts that operate in conjunction with *TomatoSupplyChain.sol*. As illustrated in Figure 3, these supplementary contracts, *FarmerTask.sol*, *WholesalerTask.sol*, *RetailerTask.sol* and *EndUserTask.sol* empower participants to register, execute transactions via MetaMask, and fulfill their respective responsibilities within the supply chain. Each contract is specifically designed for a distinct role, ensuring that farmers, wholesalers, retailers, and end-users engage with the system in a controlled and auditable manner.

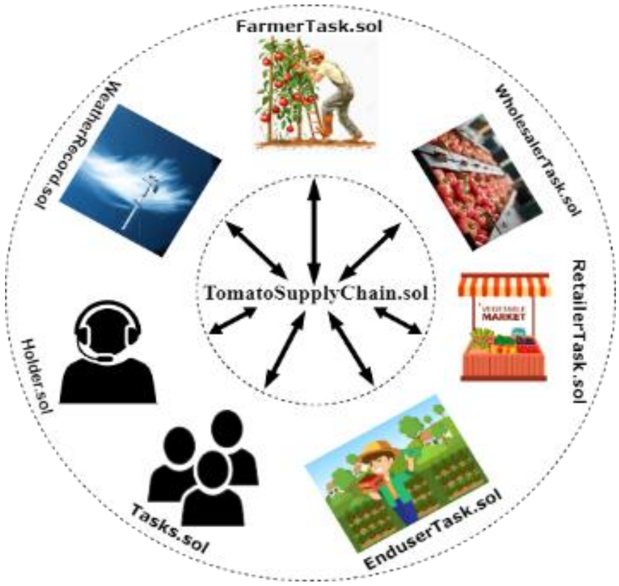

Figure 3. Smart contract architecture and task assignments

In addition to these role-specific contracts, the *Tasks.sol* library is developed to effectively manage task assignments by allowing specific addresses to gain or lose access to tasks. Key functions within this library include add, which grants access to an account after confirming it is not a zero address and does not already have access; remove, which revokes access while ensuring the account currently holds it; and has, which checks if an account is assigned to a task. This library is crucial for implementing task-based access control within the BIoT-DApp, facilitating efficient task management and user permissions.

```
function add(Task storage task, address account) internal {
        require(account != address(0), "Account is the zero address");
        require(!has(task, account), "Account already has this task");
        task.assignee[account] = true;}
function remove(Task storage task, address account) internal {
        require(account != address(0), "Account is the zero address");
        require(has(task, account), "Account does not have this task");
        task.assignee[account] = false;}
function has(Task storage task, address account) internal view returns (bool) {
        require(account != address(0), "Account is the zero address");
        return task.assignee[account];}
```

Complementing the task library, the Holder.sol contract provides a streamlined ownership mechanism by designating a single address referred to as the keeper to oversee contract operations. Upon deployment, the deploying account is automatically recognized as the original keeper and an event is emitted to record this assignment. The keeper may subsequently transfer ownership to a new address or relinquish it entirely, leaving the contract without an active keeper. This mechanism enhances transparency and accountability by maintaining a clear and auditable chain of custody. TomatoSupplyChain.sol inherits from all of these auxiliary contracts, thereby consolidating role management, task assignment and ownership control into a unified governance framework that streamlines the entire supply chain workflow.

3-1-3- Blockchain-IoT Integration Logic

In our system, environmental data such as temperature, soil moisture, and humidity are continuously collected by the deployed IoT sensors. This raw data is initially processed by a Raspberry-Pi, which computes approximate values for relevant environmental metrics. To address the high cost and limited scalability of storing large volumes of data directly on the blockchain, the raw sensor data is stored off-chain on secure local servers (see Figure 4).

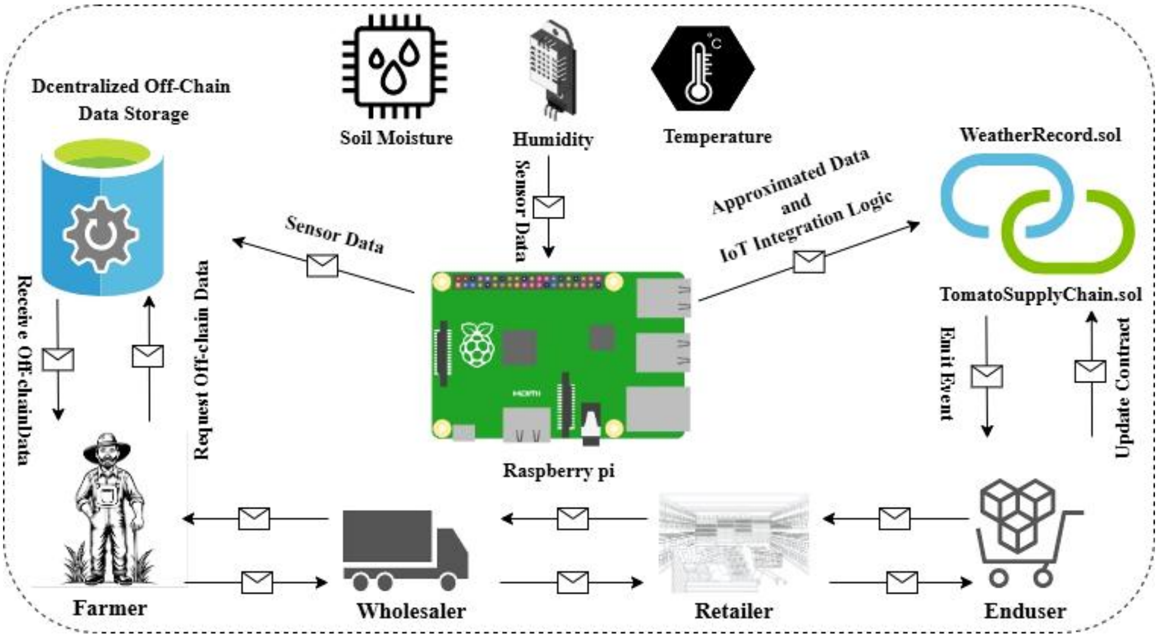

Figure 4. **BIoT-DApp system architecture**

To ensure the integrity and traceability of this off-chain data, each dataset is linked to a unique cryptographic hash. Only the hash and the processed data are stored on-chain, maintaining a lightweight and efficient blockchain footprint while preserving transparency and verifiability. These on-chain records are managed through the *WeatherRecord.sol* smart contract. This contract not only stores the approximated data and their associated hashes but also provides a verification mechanism through the *validateWeatherData()* function. This enables the BIoT-DApp frontend to cross-check on-chain records against the corresponding data stored on local servers. As a result, the system achieves a cost-effective and reliable hybrid architecture where essential information is kept on-chain for transparency, and comprehensive records are accessible off-chain by authorized users when needed.

To automate this workflow, a Python script running on the Raspberry-Pi transmits the approximated sensor data to the blockchain via APIs. Upon receiving this data, *WeatherRecord.sol* logs the information, which is then linked to the *TomatoSupplyChain.sol* contract. This integration allows real-time environmental updates to be reflected directly within the BIoT-DApp frontend. By executing *WeatherRecord.sol* in tandem with other task-oriented contracts, the system enhances its overall functionality. It ensures that stakeholders across the supply chain have direct access to accurate, timely environmental data, enabling informed decision-making, improving transparency, and promoting efficient crop management throughout the lifecycle of the agricultural product.

3-1-4- BIoT-DApp Frontend: Integrated Design and Blockchain Functionality

The BIoT-DApp frontend was developed using React.js, a framework selected for its component-based architecture, which enabled the creation of a modular, responsive, and maintainable user interface (UI). This design approach ensured an intuitive and adaptable experience for stakeholders while supporting seamless integration with blockchain infrastructure. As shown in Figure 5, the UI is structured to balance functionality and clarity. The layout features dedicated sections for real-time data visualization and user interactions, including the “Latest Weather Updates” panel, which displays IoT-derived environmental metrics such as temperature, humidity, and soil moisture. This data empowers stakeholders to make informed decisions based on live on-farm conditions. Adjacent to this, the “MetaMask Connection” section shows the connected wallet address, ensuring secure authentication and transparent blockchain transactions.

Beneath these panels, a series of action-specific input boxes allows users to interact with key supply chain stages. These inputs are tailored for tasks such as submitting batch IDs, entering farm details, setting product pricing, adding notes, and recording transactions, ensuring simplicity for users across roles. On the bottom right, the “Batch Details” component provides a real-time summary of critical batch attributes, including lifecycle status, origin, and pricing. This transparency fosters accountability by enabling stakeholders to verify product histories at a glance.

To enable secure and efficient blockchain interactions, Web3.js and Ethers.js libraries were integrated into the React frontend. These tools facilitated direct communication with the Ethereum network, allowing users to execute transactions such as recording batches, updating prices, and retrieving blockchain data like ownership details and interacting with smart contracts governing supply chain logic. React's Context API further streamlined state management, ensuring real-time synchronization of dynamic data such as batch states, weather conditions, and transaction statuses across all UI components. This architecture not only enhanced responsiveness but also maintained consistency during high-frequency updates, creating a cohesive experience where IoT data, blockchain transparency, and user-centric design converge.

```
    import requests
from datetime import datetime
import time
from web3 import Web3
import json
w3 = Web3(Web3.HTTPProvider('http://127.0.0.1:8545'))
if not w3.is_connected():
    print("Failed to connect to the local blockchain.")
    exit(1)
try:
    with open('WeatherRecord.json') as f:
    contract_info = json.load(f)
    abi = contract_info['abi']
except FileNotFoundError:
    print("Error: WeatherRecord.json file not found. Please check the file path.")
    exit(1)
contract_address = '0x5FbDB2315678afecb367f032d93F6
42f64180aa3'
contract = w3.eth.contract(address=contract_address, abi=abi)
private_key = '0xac0974bec39a17e36ba4a6b4d238ff944b
acb478cbed5efcae784d7bf4f2ff80'
account = w3.eth.account.from_key(private_key)
nonce = w3.eth.get_transaction_count(account.address)
while True:
    weather_data = requests.get('YOUR_WEATHER_API_URL').json()
    temperature = int(weather_data['main']['temp'])
    humidity = int(weather_data['humidity']['main'])
    soil_moisture = int(weather_data['soil']['moisture'])
    tx = contract.functions.updateWeather(temperature, humidity,
soil_moisture).buildTransaction({
    'chainId': 5,
    'gas': 3000000,
    'gasPrice': w3.toWei('50', 'gwei'),
    'nonce': nonce,})
    signed_tx = w3.eth.account.sign_transaction(tx, private_key)
    tx_hash = w3.eth.send_raw_transaction(signed_tx.rawTransaction)
    print(f'Transaction sent: {tx_hash.hex()}')
    time.sleep(3600)
```

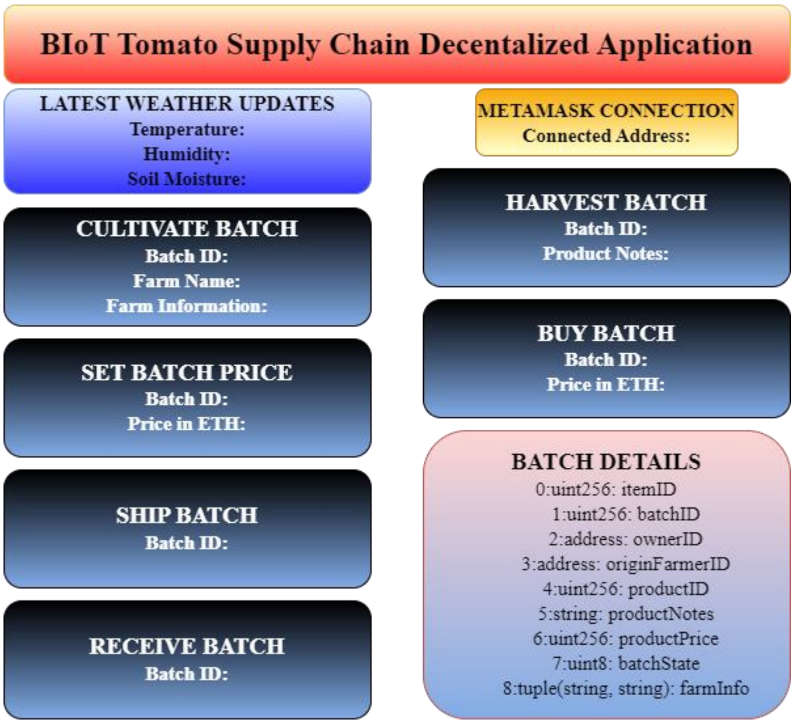

Figure 5. User interface of BIoT-DApp

3-2- Testing and Performance Evaluation

This subsection evaluates the system's robustness through controlled testing, public testnet deployments, and end-to-end transaction validation, with a focus on gas efficiency, scalability, and security.

3-2-1- Local and Public Test-Net Testing

After completing both back-end and front-end development, the smart contract was rigorously tested within a local blockchain environment using Hardhat, as illustrated in Figure 6. Successful deployment generated a unique contract address *(0x5FbDB2315678afecb367f032d93F642f64180aa3)*, serving as verifiable proof of correct compilation and execution. This outcome confirmed that the smart contract was fully functional and aligned with its intended business logic, while also demonstrating its readiness for secure migration to the Sepolia testnet, where further optimization and performance evaluation could be carried out under realistic network conditions.

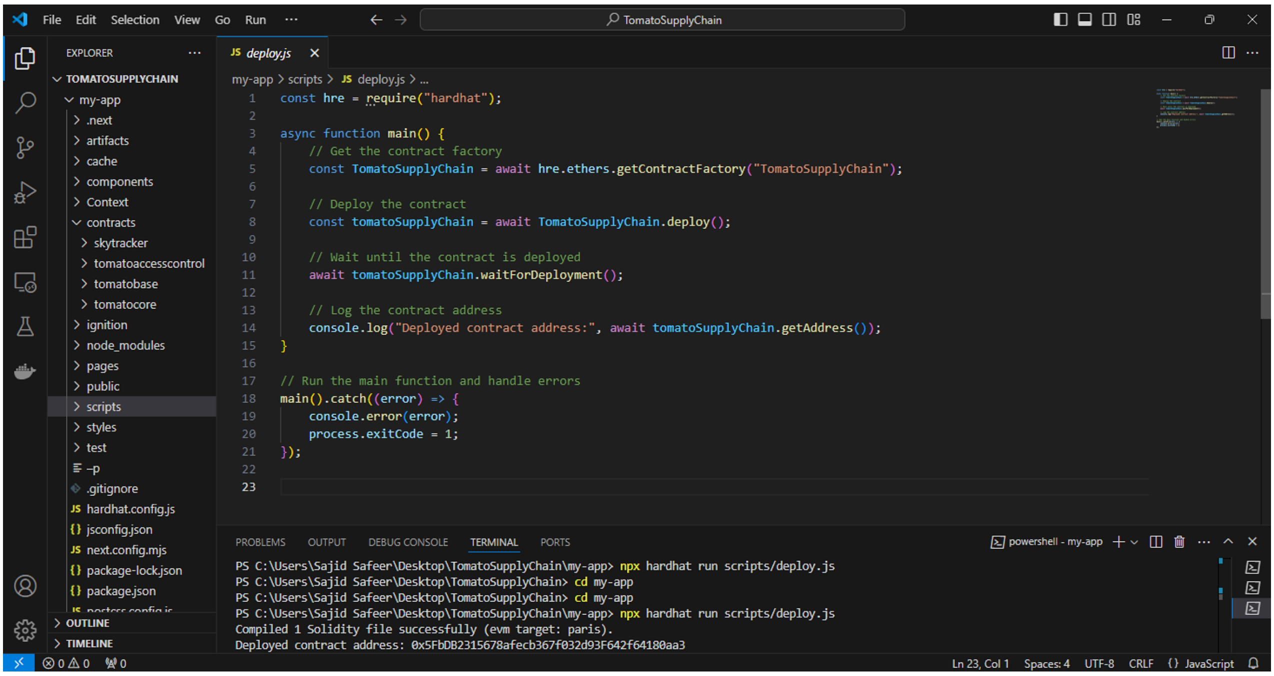

Figure 6. Smart contract local testing and deployment

Following successful local testing, the contract underwent extensive evaluation on the Ethereum-based Sepolia testnet, which was chosen because it provides a stable, widely supported, and resource-efficient testing environment that closely mirrors Ethereum mainnet conditions without incurring financial costs. The decision to use Sepolia was also influenced by its compatibility with major development tools and infrastructure providers, ensuring reproducibility and reliability. While alternative platforms such as Layer 2 solutions such as Polygon, Arbitrum, and Hyperledger were considered for their lower gas costs and higher throughput, Sepolia was prioritized at this stage to preserve interoperability with Ethereum's mainnet environment, thereby ensuring that results such as gas consumption and confirmation times remain directly transferable to real-world deployment scenarios. Future iterations of this work will evaluate Layer 2 integrations to address cost efficiency and scalability more explicitly, as such solutions hold clear potential for production-ready systems involving frequent transactions.

On Sepolia, a range of tests were performed, including unit tests to validate individual functions and integration tests to evaluate the overall performance within the BIoT-DApp ecosystem. This rigorous approach ensured both functionality and security prior to potential mainnet deployment. Importantly, scalability was also examined: the architecture was designed to accommodate multiple farms and thousands of batches by implementing a hybrid on-chain/off-chain design. Core transaction records such as ownership transfers, payments, and shipment states are immutably logged on-chain, while bulk sensor data and metadata are processed and stored off-chain, with only hashed summaries committed to the blockchain. This approach reduces gas consumption and storage overhead, enabling the system to handle large-scale operations without compromising transparency or verifiability.

Alchemy's infrastructure played a vital role in this stage, providing a stable and scalable connection to Sepolia. By using Alchemy's endpoint *URL (https://eth-sepolia.g.alchemy.com/v2/9riSEL3JygItI2OX8WoOw6q3C0-HExAf),* seamless blockchain interactions were achieved, enabling efficient contract deployment, continuous testing, and real-time monitoring. Alchemy's support enhanced the development process with actionable analytics and streamlined debugging, as illustrated in the comprehensive request log shown in Figure 7.

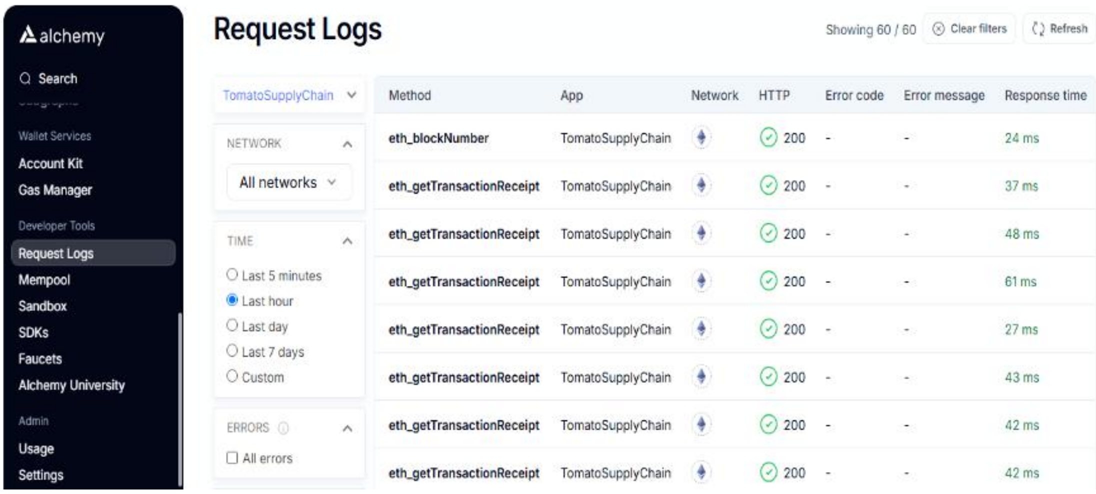

Figure 7. Alchemy-Sepolia integration for real time analytics

3-2-2- Smart Contract Verification and Etherscan Analytics

Post-deployment contract verification on Etherscan enabled a high level of public transparency, ensuring that all stakeholders could independently audit the system's operations. The finalized Sepolia contract address *(0x2f48c6f0310779dcb7b8906aa0432c7b417f20dB)* served as a permanent reference point, allowing verification of all supply chain transactions recorded during the experiments (see Figure 8). Through this process, key performance metrics were analyzed, including gas consumption per function call, which reflected the computational complexity of different operations, as well as block confirmation times under varying network loads, which provided insights into transaction efficiency and network responsiveness. In addition, wallet balance histories were monitored across role-specific accounts, offering a clear picture of how automated settlements and payments unfolded in practice. Together, these indicators not only validated compliance with the predefined business logic embedded in the smart contracts but also highlighted areas where further optimization could improve cost efficiency and overall system performance. This combination of verifiability, performance assessment, and accountability underscores the potential of blockchain-based solutions to deliver both technical robustness and stakeholder trust in agricultural supply chain management.

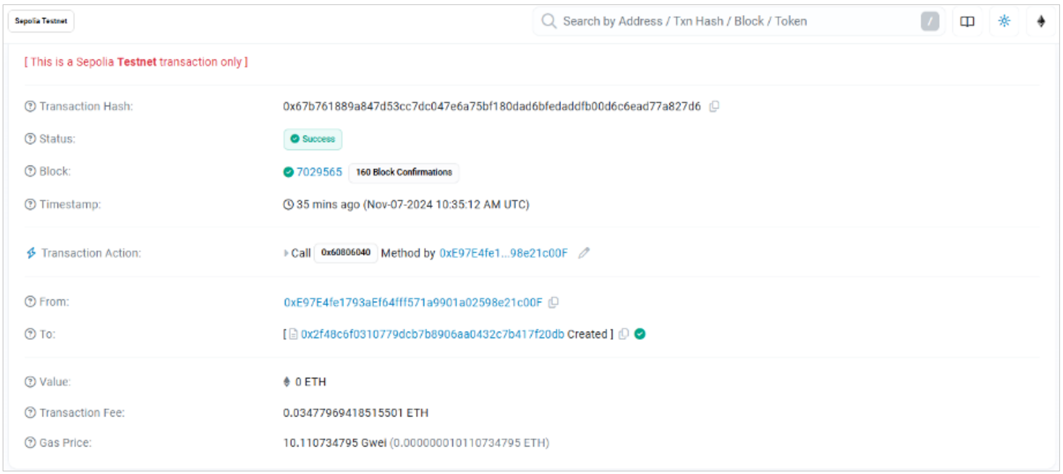

Figure 8. Etherscan transaction tracking on Sepolia test-net

3-2-3- Gas Cost Analysis

A detailed gas cost analysis of the core smart contract functions within the BIoT-DApp is presented in Table 2. Each function was evaluated on the Sepolia testnet to determine its gas consumption and the corresponding ETH cost. The most gas-intensive function, *cultivateBatch(),* consumed 112,300 gas (approximately 0.0012 ETH), primarily due to its role in initializing batch data. To optimize performance, struct packing was applied by combining smaller data types into compact storage slots, reducing the number of storage operations. Additionally, event emissions were minimized by including only essential parameters such as *batchID,* and commonly used modifiers like *onlyWhenActive* were reused to avoid bytecode duplication. The *buyBatch()* function, responsible for buyer transactions, used 89,450 gas (0.0009 ETH). It adopts the checks, effects, and interactions pattern to prevent reentrancy attacks and incorporates automatic refund logic, which helps secure user funds while keeping operations efficient.

Functions such as *harvestBatch()* and *setBatchPrice()* showed moderate gas usage of 52,150 and 48,700 gas, respectively, achieved by limiting storage writes and implementing logic-level validation through *require* statements and access control modifiers. Shipment-related actions like *shipBatch()* and *receiveBatch()* consumed 68,200 and 43,800 gas, reflecting their lightweight state transitions and role-based execution constraints that enforce accountability. Finally, *updateWeather()*, which integrates off-chain weather data, incurred 30,000 gas (0.0003 ETH). This function anchors external sensor inputs using hashes, ensuring data integrity while avoiding direct on-chain storage of raw environmental data, and demonstrates how hybrid on-chain/off-chain design choices can balance cost efficiency with reliability.

Table 2. Gas cost analysis of core functionalities

Function	Gas Used	ETH Cost (Sepolia)
cultivateBatch()	112,300	0.0012
harvestBatch()	52,150	0.0005
setBatchPrice()	48,700	0.0005
buyBatch()	89,450	0.0009
shipBatch()	68,200	0.0007
receiveBatch()	43,800	0.0004
updateWeather()	30,000	0.0003

3-2-4- Role-Based Transaction Flow Validation

To validate the BIoT-DApp's real-world functionality, role-based transaction testing was carried out using four distinct MetaMask accounts on the Sepolia testnet. Each account corresponded to a key supply chain participant farmer, wholesaler, retailer and end-user with their respective wallet addresses listed in Table 3. The testing workflow began with the farmer, who cultivated, harvested, and priced a tomato batch. The wholesaler then purchased the batch, received shipment, and updated pricing before transferring it to the retailer. The retailer repeated these operations, preparing the batch for sale to the end-user, who completed the final purchase and confirmed receipt. During this process, specific scenarios such as automated price recalculation during farmer-to-wholesaler transfers, inventory updates triggering consumer-facing interface changes, and failed delivery cases activating insurance smart contracts were tested. These interactions demonstrated that the BIoT-DApp's smart contracts performed as intended, while MetaMask integration ensured seamless execution of role-specific operations. Overall, the tests confirmed that the BIoT-DApp provides secure, transparent, and resilient end-to-end interactions across the agricultural supply chain.

Table 3. Wallet addresses of the participants

Role	Wallet Address
Farmer	0xB564226D6588d5dB3c350aE1335f4f0Edc4540eA
Wholesaler	0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
Retailer	0xaB318C395Ca271C3d383F0B487E9c3d3B08f63C9
End-user	0x4e134042fde5AD3904fB47ad9Bf0F02675507579

In addition, a total of fifty end-to-end transactions were executed under varied conditions, including successful transfers, automated pricing, and simulated failures. No critical bottlenecks were observed, although slight latency was recorded during bursts of shipment-related transactions. These outcomes suggest that the prototype can scale efficiently in small-to-medium scenarios but will require optimization for deployments involving multiple farms and thousands of batches. User feedback from test participants highlighted that while the MetaMask interface and front-end design were intuitive for technically experienced users, onboarding non-technical farmers will require structured training and digital literacy support. This observation underscores the importance of governance and consortium structures, where cooperatives, regulators, and supply chain stakeholders could facilitate both access control and farmer onboarding. Furthermore, although this phase did not include a detailed cost-benefit analysis, preliminary evidence indicates that operational efficiencies such as reduced post-harvest loss, faster settlements, and decreased reliance on intermediaries may offset transaction costs in practice. A formal economic feasibility study therefore represents an important direction for future research and deployment.

4- Discussion

The development and empirical evaluation of the BIoT-DApp prototype presented in this study offer a timely and context-specific contribution to the growing body of literature on digital agriculture, with particular relevance to the persistent inefficiencies characterizing the Italian tomato supply chain. Previous research has widely documented the challenges of limited traceability, fragmented data exchange, and opaque transactional processes, all of which erode stakeholder trust and disproportionately disadvantage smallholder farmers operating within fragmented value chains. Building directly on these identified gaps, the findings of this study demonstrate that blockchain's inherent immutability, when combined with IoT-enabled real-time environmental monitoring, can deliver transparent and tamper-proof records across the entire supply chain lifecycle. While earlier studies [32-34] emphasized the theoretical potential of decentralized ledgers to enhance scalability and security in agri-food systems, the present work advances this discourse by moving beyond conceptual propositions to implement and empirically validate a functioning prototype tailored to the specific operational requirements of a perishable crop within a fragmented national agricultural context.

The integration of IoT sensors for monitoring environmental parameters, including soil moisture, humidity, and temperature, not only strengthens precision farming practices but simultaneously supports the principles of climate-smart agriculture. Although prior studies proposed hybrid blockchain-IoT architectures as a means of overcoming Ethereum's inherent data storage limitations, such proposals generally remained conceptual and lacked empirical validation under realistic operating conditions [35]. The present study bridges this gap by demonstrating, under testnet conditions, that anchoring hashed summaries of raw sensor data on-chain while retaining full datasets off-chain achieves both cost efficiency and verifiable data integrity. This is evidenced concretely by the updateWeather() function, which required only 30,000 gas (0.0003 ETH), illustrating that hybrid design choices can substantially reduce operational costs relative to approaches that commit full sensor data streams directly on-chain. These empirical findings not only confirm the practical feasibility of strategies outlined in [36] but also extend them by quantifying the trade-offs between scalability, transparency, and cost, thereby offering actionable and reproducible insights into blockchain-IoT integration for agricultural systems.

The economic implications of these findings further underscore the relevance and potential of this approach. Previous research [37, 38] has argued that blockchain technology can empower smallholder farmers through fairer pricing mechanisms, reduced dependence on intermediaries, and improved market access. The role-based transaction testing conducted across farmer, wholesaler, retailer and end-user accounts in this study empirically validates these claims by demonstrating the operational feasibility of automated pricing and settlement within a real supply chain workflow. Core functions such as *harvestBatch()* and *setBatchPrice()* executed at moderate gas costs (52,150 and 48,700 gas, respectively), while shipment and delivery operations (*shipBatch()* and *receiveBatch()*) proved equally efficient. These outcomes collectively illustrate how blockchain automation can streamline supply chain functions without imposing prohibitive costs, reinforcing the economic benefits theorized by [39]. Furthermore, by simulating failed delivery scenarios that activated insurance contract conditions, this study provides rare empirical evidence for blockchain's potential to support innovative risk-sharing mechanisms, a dimension that has been widely theorized in the literature but seldom tested in practice.

Notwithstanding these promising results, a critical comparison with existing work reveals important limitations that must be acknowledged. As emphasized by [40, 41], digital infrastructure deficits, limited digital literacy, and high transaction fees remain the most significant structural barriers to widespread blockchain adoption in agricultural contexts. The findings of this study confirm these challenges: while system performance on the Sepolia testnet was satisfactory with average block confirmation times remaining within practical operational limits, hence deployment on the Ethereum mainnet would likely be financially unsustainable for smallholder farmers due to substantially higher gas fees. This empirical evidence lends concrete support to earlier warnings [42] that public blockchain networks may not be viable for agricultural applications unless targeted cost mitigation strategies are actively pursued. In this respect, the present study departs from more optimistic prior assessments, which emphasized blockchain's economic empowerment potential without sufficiently addressing the financial and technical constraints inherent in real-world implementation [43].

Addressing these limitations requires considering alternative architectures. Our findings align with the proposals of [44, 45], which advocate Layer 2 solutions such as Polygon or Arbitrum to reduce transaction costs and improve throughput or enterprise frameworks like Hyperledger Fabric for semi-permissioned networks. By demonstrating that hybrid designs and role-based access controls can mitigate some scalability issues, this study provides empirical support for these proposals while also highlighting the urgent need for broader cost reduction. Similarly, the suggestion to expand sensor coverage into warehouses and transportation hubs resonates with the vision of [46], and our prototype indicates that such an expansion could be achieved within a modular framework without compromising system performance.

Beyond technical aspects, the governance dimension is equally critical. This study supports the argument of [47] that consortium-based blockchain networks, underpinned by government investment in digital literacy and infrastructure, are vital for ensuring inclusive adoption. Moreover, our findings open pathways for integrating financial institutions as blockchain nodes, consistent with [48]. The system's ability to anchor verified sensor data on-chain could provide the basis for transparent and fraud-resistant credit assessment, thereby offering farmers better access to financial services. Unlike prior studies, which have largely theorized about such integrations, our prototype demonstrates how they could be operationalized in practice through its modular smart contract architecture.

In an academic context, this work contributes to bridging the persistent gap between theory and practice. While [49] emphasized the need for empirical evaluations to move beyond conceptual frameworks, this study provides a tested model that integrates smart contracts, hybrid storage, and role-based workflows. Its modular design is consistent with the adaptability stressed in [50], and the demonstrated flexibility suggests that it could be reconfigured to support other perishable commodities beyond tomatoes. Furthermore, our results echo the calls of [51–53] for comparative evaluations, socio-economic impact studies, and interdisciplinary collaboration. By generating measurable testnet outcomes and identifying areas requiring further investigation, such as scalability across diverse crops and geographies, this research establishes a foundation for future work.

Taken together, these findings validate and extend previous research while also revealing areas of divergence that highlight important directions for further study. By demonstrating how a blockchain-IoT system can function under real-world testnet conditions, the BIoT-DApp narrows the gap between conceptual promise and operational reality. It shows that while blockchain holds substantial potential for enhancing transparency, sustainability, and equity in agriculture, its success will ultimately depend on addressing cost barriers, scaling hybrid architectures, and building institutional and infrastructural support systems. In this way, the present study provides both practical and academic contributions, advancing the literature while offering actionable insights for policymakers, technologists, and practitioners in the agri-food domain.

5- Conclusion

In conclusion, the BIoT-DApp developed in this study shows that integrating blockchain with IoT can provide a practical, transparent, and scalable solution to the persistent inefficiencies in agricultural supply chains, particularly in perishable and fragmented sectors such as the Italian tomato industry. By merging blockchain's immutability and smart contract functionality with IoT's real-time monitoring capabilities, the system enhances traceability, accountability, and process automation across multiple stakeholder levels, effectively addressing critical issues like delayed payments, quality verification, and compliance with sustainability standards. Unlike many previous works that remain conceptual or limited to partial traceability pilots, this research offers empirical validation through the design, deployment, and evaluation of a functional prototype on the Sepolia testnet, yielding measurable insights into latency, transaction costs, and throughput under realistic conditions. Furthermore, the study emphasizes the socio-technical aspects of adoption by examining barriers such as digital skills, usability, and cost structures that impact smallholder inclusion, while proposing mitigation strategies aimed at enhancing accessibility and farmer participation. Consequently, the findings extend the existing literature beyond technological promise to evidence-based application, illustrating how integrated BIoT architectures can be tailored to real-world agricultural contexts. The broader implications of this work are significant, informing not only future agricultural technology design but also policy frameworks and academic discussions surrounding digital transformation, rural inclusion, and climate-smart food systems, thereby establishing the groundwork for more resilient, efficient, and equitable agri-food networks worldwide.

6- Declarations

6-1- Author Contributions

Conceptualization, S.S. and C.P.; methodology, S.S.; software, S.S.; validation, S.S., V.L., and C.L.; formal analysis, S.S.; investigation, S.S.; resources, C.P. and V.L.; data curation, C.P.; writing—original draft preparation, S.S.; writing—review and editing, S.S. and C.L.; visualization, S.S.; supervision, C.P. and V.L.; project administration, C.P. and V.L.; funding acquisition, C.P. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6-3- Funding

This document presents the findings of research conducted at Blockchain@UBC, University of British Columbia, Canada, with support from the International School of Advanced Studies, University of Camerino, Italy, as part of the Doctorate Program in Blockchain and Distributed Ledger Technology for Agriculture and Agri-Food (Ministerial Decree 226/2021).

6-4- Institutional Review Board Statement

Not applicable.

6-5- Informed Consent Statement

Not applicable.

6-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Movilla-Pateiro, L., Mahou-Lago, X. M., Doval, M. I., & Simal-Gandara, J. (2021). Toward a sustainable metric and indicators for the goal of sustainability in agricultural and food production. Critical Reviews in Food Science and Nutrition, 61(7), 1108–1129. doi:10.1080/10408398.2020.1754161.

[2] Romeo, E., Virglerova, Z., & Giampaola, V. (2025). Cultural and organizational challenges of adopting technology, traceability and blockchain in the Italian agri-food supply chain: a qualitative study. British Food Journal, 1–22, 1–22. doi:10.1108/BFJ-04-2025-0555.

[3] de las Mercedes Capobianco-Uriarte, M., Aparicio, J., de Pablo-Valenciano, J., & del Pilar Casado-Belmonte, M. (2021). The European tomato market. An approach by export competitiveness maps. PLoS ONE, 16(5 May), 250867. doi:10.1371/journal.pone.0250867.

[4] Xue, L., Cao, Z., Scherhaufer, S., Östergren, K., Cheng, S., & Liu, G. (2021). Mapping the EU tomato supply chain from farm to fork for greenhouse gas emission mitigation strategies. Journal of Industrial Ecology, 25(2), 377–389. doi:10.1111/jiec.13080.

[5] Dayioğlu, M. A., & Türker, U. (2021). Digital transformation for sustainable future-agriculture 4.0: A review. Tarim Bilimleri Dergisi, 27(4), 373–399. doi:10.15832/ankutbd.986431.

[6] Sunny, J., Undralla, N., & Madhusudanan Pillai, V. (2020). Supply chain transparency through blockchain-based traceability: An overview with demonstration. Computers and Industrial Engineering, 150, 106895. doi:10.1016/j.cie.2020.106895.

[7] Gkogkos, G., Lourenço, P., Pechlivani, E. M., Encarnação, L., Votis, K., Giakoumoglou, N., da Silva, J. M., & Tzovaras, D. (2023). Distributed Ledger Technologies for Food Sustainability indexing. Smart Agricultural Technology, 5, 100312. doi:10.1016/j.atech.2023.100312.

[8] Chen, H. Y., Sharma, K., Sharma, C., & Sharma, S. (2023). Integrating explainable artificial intelligence and blockchain to smart agriculture: Research prospects for decision making and improved security. Smart Agricultural Technology, 6, 100350. doi:10.1016/j.atech.2023.100350.

[9] Choruma, D. J., Dirwai, T. L., Mutenje, M. J., Mustafa, M., Chimonyo, V. G. P., Jacobs-Mata, I., & Mabhaudhi, T. (2024). Digitalisation in agriculture: A scoping review of technologies in practice, challenges, and opportunities for smallholder farmers in sub-saharan africa. Journal of Agriculture and Food Research, 18. doi:10.1016/j.jafr.2024.101286.

[10] Mukherjee, A. A., Singh, R. K., Mishra, R., & Bag, S. (2022). Application of blockchain technology for sustainability development in agricultural supply chain: justification framework. Operations Management Research, 15(1–2), 46–61. doi:10.1007/s12063-021-00180-5.

[11] Alobid, M., Abujudeh, S., & Szűcs, I. (2022). The Role of Blockchain in Revolutionizing the Agricultural Sector. Sustainability (Switzerland), 14(7), 4313. doi:10.3390/su14074313.

[12] Shao, D., & Marwa, N. (2024). Blockchain-enabled smart contracts for enhancing seed certification transparency: A design science approach. Smart Agricultural Technology, 9, 100651. doi:10.1016/j.atech.2024.100651.

[13] Pal, T., Bommisetti, V., Suvarnakanti, P., Dudala, S. S., Asha, S., & Jaiswal, V. (2025). The Revolutionary Role of Blockchain Technology in the Agri-Food Sector Focusing on the Food Supply Chain. Food Reviews International, 41(3), 770–801. doi:10.1080/87559129.2024.2418464.

[14] Buyssens, H., & Viaene, S. (2024). Design principles for a blockchain-based multi-sided platform for the sustainable trade of water: An affordance approach. Journal of Cleaner Production, 471, 143212. doi:10.1016/j.jclepro.2024.143212.

[15] Toader, D. C., Rădulescu, C. M., & Toader, C. (2024). Investigating the Adoption of Blockchain Technology in Agri-Food Supply Chains: Analysis of an Extended UTAUT Model. Agriculture (Switzerland), 14(4), 614. doi:10.3390/agriculture14040614.

[16] Akhtaruzzaman Khan, M., Emran Hossain, M., Shahaab, A., & Khan, I. (2022). ShrimpChain: A blockchain-based transparent and traceable framework to enhance the export potentiality of Bangladeshi shrimp. Smart Agricultural Technology, 2, 100041. doi:10.1016/j.atech.2022.100041.

[17] Miller, T., Cao, S., Foth, M., Boyen, X., & Powell, W. (2023). An asset-backed decentralised finance instrument for food supply chains – A case study from the livestock export industry. Computers in Industry, 147, 103863. doi:10.1016/j.compind.2023.103863.

[18] Hasan, H., AlHadhrami, E., AlDhaheri, A., Salah, K., & Jayaraman, R. (2019). Smart contract-based approach for efficient shipment management. Computers and Industrial Engineering, 136, 149–159. doi:10.1016/j.cie.2019.07.022.

[19] Dogo, E. M., Salami, A. F., Nwulu, N. I., & Aigbavboa, C. O. (2019). Blockchain and Internet of Things-Based Technologies for Intelligent Water Management System. Artificial Intelligence in IoT, 129–150. doi:10.1007/978-3-030-04110-6_7.

[20] Viriyasitavat, W., Anuphaptrirong, T., & Hoonsopon, D. (2019). When blockchain meets Internet of Things: Characteristics, challenges, and business opportunities. Journal of industrial information integration, 15, 21-28. doi:10.1016/j.jii.2019.05.002.

[21] Pulighe, G., Di Fonzo, A., Gaito, M., Giuca, S., Lupia, F., Bonati, G., & De Leo, S. (2024). Climate change impact on yield and income of Italian agriculture system: a scoping review. Agricultural and Food Economics, 12(1), 23. doi:10.1186/s40100-024-00317-7.

[22] Witt, J., Schoop, M., & Knaus, K. (2025). Opportunities and Challenges of Blockchain Technology for Negotiation Support Systems. Group Decision and Negotiation, 34(2), 373–403. doi:10.1007/s10726-024-09916-7.

[23] Amir Latif, R. M., Hussain, K., Jhanjhi, N. Z., Nayyar, A., & Rizwan, O. (2022). A remix IDE: smart contract-based framework for the healthcare sector by using Blockchain technology. Multimedia Tools and Applications, 81(19), 26609–26632. doi:10.1007/s11042-020-10087-1.

[24] Yang, X., Yu, Y., Shirowzhan, S., Sepasgozer, S., & Li, H. (2020). Automated PPE-Tool pair check system for construction safety using smart IoT. Journal of Building Engineering, 32, 101721. doi:10.1016/j.jobe.2020.101721.

[25] Gaikwad, S. Y. (2024). Secure Data Transmission in the Wireless Sensor Network with Blockchain Cryptography Network. Journal of Sensors, IoT & Health Sciences, 2(2), 41–55. doi:10.69996/jsihs.2024009.

[26] Nabil, D. H., Rahman, M. H., Chowdhury, A. H., & Menezes, B. C. (2023). Managing supply chain performance using a real time Microsoft Power BI dashboard by action design research (ADR) method. Cogent Engineering, 10(2), 2257924. doi:10.1080/23311916.2023.2257924.

[27] Razzaq, A. (2024). A Web3 secure platform for assessments and educational resources based on blockchain. Computer Applications in Engineering Education, 32(1), 22677. doi:10.1002/cae.22677.

[28] Kumar, V., Budhiraja, I., Jabbari, A., Garg, D., Singh, D., & Mengani, N. (2025). Efficient blockchain interoperability design for cross-chain transactions in future internet-of-value. Peer-to-Peer Networking and Applications, 18(3), 1–21. doi:10.1007/s12083-025-01941-w.

[29] Silva, E. J. L. A. da, de Araújo, C. S., da Silva Junior, J. M., Gomes, R. J. R., Carvalho, L. R., Rodriguez, L. C., & Pinheiro, C. L. (2024). Development of cryptogames with Unity on an Ethereum Blockchain Test Network: Case Study and Challenges. Journal on Interactive Systems, 15(1), 549–560. doi:10.5753/jis.2024.4188.

[30] Thakur, K. S., Ahuja, R., & Singh, R. (2024). IoT-GChain: Internet of Things-Assisted Secure and Tractable Grain Supply Chain Framework Leveraging Blockchain. Electronics (Switzerland), 13(18), 3740. doi:10.3390/electronics13183740.

[31] Safeer, S., & Pulvento, C. (2024). Blockchain-Backed Sustainable Management of Italian Tomato Processing Industry. Agriculture (Switzerland), 14(7), 1120. doi:10.3390/agriculture14071120.

[32] Safeer, S., Mastro, G. De, & Pulvento, C. (2024). IoT based climate smart agriculture succeeded by blockchain database—A bibliometric analysis. Frontiers in Sustainable Food Systems, 8, 1406871. doi:10.3389/fsufs.2024.1406871.

[33] Jahanbin, P., Wingreen, S. C., Sharma, R., Ijadi, B., & Reis, M. M. (2023). Enabling affordances of blockchain in agri-food supply chains: A value-driver framework using Q-methodology. International Journal of Innovation Studies, 7(4), 307–325. doi:10.1016/j.ijis.2023.08.001.

[34] Marchesi, L., Mannaro, K., & Porcu, R. (2021). Automatic Generation of Blockchain Agri-food Traceability Systems. Proceedings - 2021 IEEE/ACM 4th International Workshop on Emerging Trends in Software Engineering for Blockchain, WETSEB 2021, 10, 41–48. doi:10.1109/WETSEB52558.2021.00013.

[35] T., M., Makkithaya, K., & V.G., N. (2023). A trusted IoT data sharing and secure oracle-based access for agricultural production risk management. Computers and Electronics in Agriculture, 204, 107544. doi:10.1016/j.compag.2022.107544.

[36] Fernández-Iglesias, M. J., Delgado von Eitzen, C., & Anido-Rifón, L. (2024). Efficient Traceability Systems with Smart Contracts: Balancing On-Chain and Off-Chain Data Storage for Enhanced Scalability and Privacy. Applied Sciences (Switzerland), 14(23), 11078. doi:10.3390/app142311078.

[37] Bhatia, M. S., Chaudhuri, A., Kayikci, Y., & Treiblmaier, H. (2024). Implementation of blockchain-enabled supply chain finance solutions in the agricultural commodity supply chain: a transaction cost economics perspective. Production Planning and Control, 35(12), 1353–1367. doi:10.1080/09537287.2023.2180685.

[38] Behnke, K., & Janssen, M. F. W. H. A. (2020). Boundary conditions for traceability in food supply chains using blockchain technology. International Journal of Information Management, 52, 101969. doi:10.1016/j.ijinfomgt.2019.05.025.

[39] Safeer, S., Gallo, P., & Pulvento, C. (2025). Agri-farming with computer vision, IoT and blockchain towards climate smart cultivation. Internet of Things (The Netherlands), 34. doi:10.1016/j.iot.2025.101749.

[40] Rejeb, A., Keogh, J. G., Zailani, S., Treiblmaier, H., & Rejeb, K. (2020). Blockchain Technology in the Food Industry: A Review of Potentials, Challenges and Future Research Directions. Logistics, 4(4), 27. doi:10.3390/logistics4040027.

[41] Puthenveettil, N. R., & Sappati, P. K. (2024). A review of smart contract adoption in agriculture and food industry. Computers and Electronics in Agriculture, 223, 109061. doi:10.1016/j.compag.2024.109061.

[42] Doan, T. V., Psaras, Y., Ott, J., & Bajpai, V. (2022). Toward Decentralized Cloud Storage With IPFS: Opportunities, Challenges, and Future Considerations. IEEE Internet Computing, 26(6), 7–15. doi:10.1109/MIC.2022.3209804.

[43] Chen, Z. (2024). Design, development, and deployment of decentralized applications. Applied and Computational Engineering, 48(1), 46–52. doi:10.54254/2755-2721/48/20241132.

[44] Bhat, A., Nor, R. M., Amiruzzaman, M., Islam, M. R., & Quadir, M. (2025). An Evaluation of EVM-Compatible Blockchain Platforms for Trade Finance. Journal of Advanced Research Design, 127(1), 150–172. doi:10.37934/ard.127.1.150172.

[45] Kharche, A., Badholia, S., & Upadhyay, R. K. (2024). Implementation of blockchain technology in integrated IoT networks for constructing scalable ITS systems in India. Blockchain: Research and Applications, 5(2), 100188. doi:10.1016/j.bcra.2024.100188.

[46] Vilas-Boas, J. L., Rodrigues, J. J. P. C., & Alberti, A. M. (2023). Convergence of Distributed Ledger Technologies with Digital Twins, IoT, and AI for fresh food logistics: Challenges and opportunities. Journal of Industrial Information Integration, 31, 100393. doi:10.1016/j.jii.2022.100393.

[47] Lin, Y. P., Petway, J. R., Anthony, J., Mukhtar, H., Liao, S. W., Chou, C. F., & Ho, Y. F. (2017). Blockchain: The evolutionary next step for ICT e-agriculture. Environments, 4(3), 1–13. doi:10.3390/environments4030050.

[48] Mishra, R., Kshetri, N., & Jha, S. K. (2025). Leveraging Blockchain Technology for Making Secure IoT Networks. In Blockchain Technology for Cyber Defense, Cybersecurity, and Countermeasures: Techniques, Solutions, and Applications. CRC Press, Florida, United States. doi:10.1201/9781003449515-2.

[49] Chacko, N. M., Narendra, V. G., Balachandra, M., & Rathinam, S. (2023). Exploring IoT-Blockchain Integration in Agriculture: An Experimental Study. IEEE Access, 11, 130439–130450. doi:10.1109/ACCESS.2023.3334726.

[50] Daraghmi, E. Y., Jayousi, S., Daraghmi, Y. A., Daraghma, R. S. M., & Fouchal, H. (2024). Smart Contracts for Managing the Agricultural Supply Chain: A Practical Case Study. IEEE Access, 12, 125462–125479. doi:10.1109/ACCESS.2024.3439412.

[51] Ge, H., Gray, R., & Nolan, J. (2015). Agricultural supply chain optimization and complexity: A comparison of analytic vs simulated solutions and policies. International Journal of Production Economics, 159, 208–220. doi:10.1016/j.ijpe.2014.09.023.

[52] Lee, N. M., Varshney, L. R., Michelson, H. C., Goldsmith, P., & Davis, A. (2022). Digital trust substitution technologies to support smallholder livelihoods in Sub-Saharan Africa. Global Food Security, 32, 100604. doi:10.1016/j.gfs.2021.100604.

[53] Tasic, I., & Cano, M. D. (2024). An orchestrated IoT-based blockchain system to foster innovation in agritech. IET Collaborative Intelligent Manufacturing, 6(2), 12109. doi:10.1049/cim2.12109.