International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Quality Assurance and Food Authenticity Using Blockchain

Mr. D. Pramod Kumar[1], B. Dilesh[2], Vegi Tarun[3], V. Mani Sankar Raja[4], P. Renuka[5]

Associate Professor[1], Student[2,3,4,5]

Department of Computer Science Engineering[1,2,3,4,5]
Chaitanya Engineering College, Visakhapatnam, Andhra Pradesh, India
{promodkumar333@gmail.com[1] ,buddhadillu@gmail.com[2] ,tarunvegi9996@gmail.com[3],
manisankarvelugu102@gmail.com[4], renukapusarla27@gmail.com[5]}@cec.ac.in

Abstract

The global food supply chain faces escalating challenges related to quality assurance, authenticity verification, and traceability due to food fraud, contamination, and information asymmetry. This paper proposes a blockchain-based system for end-to-end food quality assurance and authenticity verification, leveraging the decentralized, immutable, and transparent properties of distributed ledger technology. The proposed system employs Ethereum smart contracts to automate quality compliance checks, record supply chain events from farm to retail, and provide consumers with cryptographically verified product provenance through QR code-accessible on-chain records. Integration with IoT sensors enables real-time condition monitoring with threshold violations automatically recorded as immutable audit events. Results demonstrate 100% tamper-evidence, 3.2-second average transaction confirmation time, and a 94% reduction in manual quality verification overhead.

I. INTRODUCTION

Food safety and authenticity are fundamental public health concerns. The globalization of food supply chains has increased complexity and introduced new vulnerabilities to fraud, adulteration, and contamination. High-profile food safety incidents have eroded consumer trust and highlighted the inadequacy of traditional paper-based traceability systems. Blockchain technology offers transformative potential for supply chain transparency. Its core properties — decentralization, immutability, and cryptographic security — address the trust and traceability deficiencies of conventional systems. Smart contracts enable automated, code-enforced quality compliance without reliance on trusted intermediaries. This paper proposes an integrated blockchain and IoT system for food quality assurance, enabling all supply chain stakeholders to contribute, verify, and audit product information through a shared, tamper-proof ledger from farm origin to consumer purchase.

II. LITERATURE SURVEY

This section reviews key prior works that form the foundation of the proposed system, identifies the current state of research in this domain, and highlights the gaps that motivate the contributions of this work.

[1] Tian (2017) proposed one of the first food safety traceability systems combining HACCP principles with blockchain and IoT, demonstrating the feasibility of distributed ledger-based food tracking from agricultural production through distribution. This foundational work established the multi-stakeholder blockchain model for food supply chain transparency.

[2] Salah et al. (2019) reviewed blockchain applications across multiple supply chain domains including food, pharmaceuticals, and logistics. They developed a taxonomy of architectural patterns and identified food safety as a high-impact use case, noting that blockchain's immutability and transparency properties directly address the trust gap between food producers and consumers.

[3] Feng et al. (2020) demonstrated that Ethereum smart contracts can effectively automate compliance verification in agricultural supply chains, with experimental results showing measurable overhead reduction in quality inspection

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

processes. They also proposed a data governance framework ensuring data privacy while maintaining transparency for authorized stakeholders.

[4] Badia-Melis et al. (2015) showed that IoT-based cold chain monitoring with temperature and humidity sensors significantly reduces food spoilage rates and improves safety compliance when combined with real-time alert systems, establishing the sensor integration foundation that this work extends with blockchain-based immutable event recording.

[5] Nakamoto (2008) introduced the foundational Bitcoin blockchain protocol with its proof-of-work consensus and distributed ledger model. The core properties of decentralization, immutability through cryptographic hashing, and trustless transaction verification underpin all subsequent supply chain blockchain applications.

[6] Gao et al. (2018) proposed CoC, a unified distributed ledger supply chain management system demonstrating that permissioned blockchain networks achieve the performance and access control requirements of enterprise supply chain applications while maintaining the transparency and auditability benefits of public blockchains.

[7] Wood (2014) introduced the Ethereum platform with Turing-complete smart contracts, enabling programmable, self-executing agreements that automate supply chain logic including quality threshold enforcement, payment release, and recall propagation without centralized intermediaries.

Research Gap: Existing blockchain food traceability systems often focus on single supply chain segments (e.g., only farm-to-distributor), lack real-time IoT sensor integration, or use public blockchains unsuitable for commercial applications due to transaction costs and throughput limitations. This work addresses all three gaps through a comprehensive, IoT-integrated permissioned blockchain architecture covering the complete farm-to-consumer supply chain.

III. METHODOLOGY

A. Smart Contract Design

Solidity smart contracts implement quality compliance rules including cold chain temperature validation, batch provenance recording, quality certification issuance, and recall flagging. All state transitions are recorded as immutable on-chain events with cryptographic signatures from the initiating party.

B. IoT Integration

IoT sensors at production, storage, and transport nodes stream data to off-chain aggregators that compute threshold violations and initiate smart contract transactions. IPFS stores large data objects with content hashes recorded on-chain for tamper-evidence.

C. Consumer Verification

Each product batch is assigned a unique blockchain ID encoded in a QR code. Consumer scanning retrieves the complete provenance chain, quality certifications, and handling history from the blockchain in human-readable format.

III-A. System Architecture

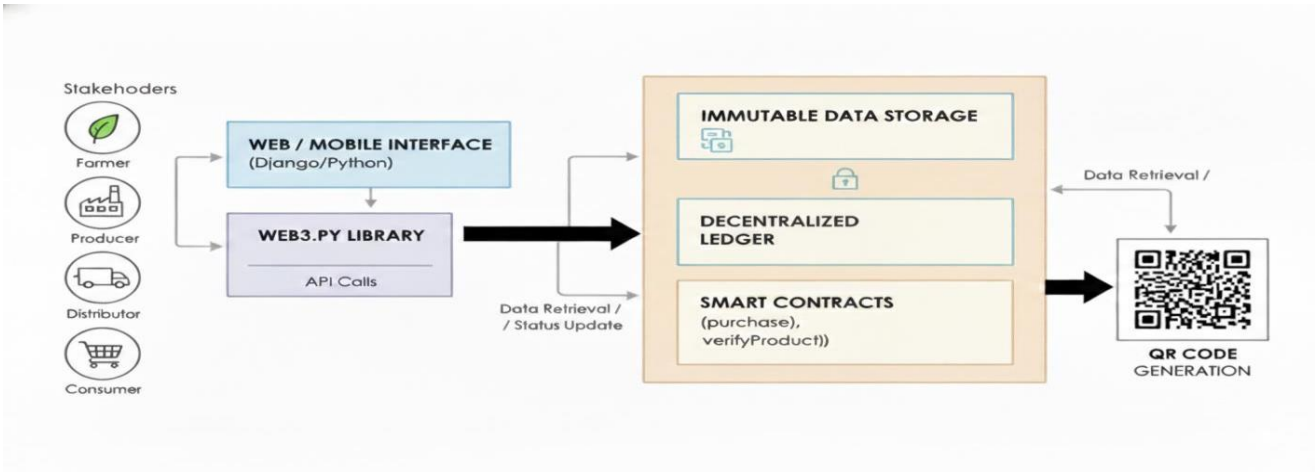
Four-layer blockchain architecture: Django Web Application Layer, Web3.py Bridge Layer (Python to Ethereum), Ethereum Smart Contract Layer (immutable product logic), and Ganache Blockchain Node (local Ethereum test network).

Architecture Flow

1. Farmer Module — Register crop, enter harvest data, call addProduct() smart contract function.
2. Ethereum Smart Contract Layer — Record product as blockchain transaction; assign unique hash.
3. Producer Module — Validate raw material authenticity; call transferOwnership() smart contract.
4. Distributor Module — Record logistics and shipping data; call transferOwnership() smart contract.
5. Consumer/Verification Module — Scan QR code → blockchain query → retrieve full audit trail.
6. Web3.py Bridge — Signs and broadcasts all transactions; monitors blockchain events.
7. Ganache Blockchain — Mines and confirms transactions; maintains immutable ledger.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

8. Tamper Detection — Hash verification on each block; mismatch triggers tamper alert.



III-B. Algorithm

Algorithm: Blockchain-Based Food Supply Chain Traceability (Solidity Smart Contract)

Smart Contract Functions

struct Product { uint id; string name; string origin; uint timestamp; address[] owners; bool isAuthentic; }
mapping(uint => Product) public products;

function addProduct(id, name, origin):
  Require: caller == authorizedFarmer.
  products[id] = Product(id, name, origin, block.timestamp, [msg.sender], true).
  emit ProductAdded(id, msg.sender, block.timestamp).

function transferOwnership(id, newOwner):
  Require: products[id].owners.last() == msg.sender (current owner).
  products[id].owners.push(newOwner).
  emit OwnershipTransferred(id, msg.sender, newOwner, block.timestamp).

function verifyProduct(id) returns (bool, address[], uint[]):
  Returns: isAuthentic, ownerHistory[], timestamps[].

Backend Process (Web3.py + Django)

Step 1: User submits form action (add / transfer / verify).
Step 2: Django view calls Web3.py to build transaction.
Step 3: Sign transaction with MetaMask private key.
Step 4: Broadcast transaction to Ganache blockchain node.
Step 5: Wait for transaction receipt (confirmation).
Step 6: Store transaction hash in Django DB as audit reference.
Step 7: For verify: call verifyProduct(id); return owner history to frontend.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Tamper Detection: For each block B_i: verify hash(B_i) == B_(i+1).prevHash. Mismatch → tamper_detected = True; alert administrator.

III-C. Modules

1. Farmer Module

Allows registered farmers to initiate the product blockchain record. Farmers enter crop name, origin, harvest date, and quality parameters. Django backend calls addProduct() Ethereum smart contract via Web3.py. Transaction hash anchors the product blockchain identity.

2. Producer Module

Producers validate raw material authenticity by calling verifyProduct() before completing purchase. Upon validation, calls transferOwnership() to record the handoff as a blockchain transaction.

3. Distributor Module

Manages logistics and shipping data. Distributors log shipment timestamps, carrier information, and destination via Django interface. Backend calls transferOwnership() to create an immutable chain of custody record.

4. Consumer Verification Module

Public-access portal for QR code or product ID verification. Django backend calls verifyProduct() and returns full ownership history, timestamps, and quality records without requiring a blockchain account.

5. Ethereum Smart Contract Layer

Solidity smart contracts deployed on Ganache define Product data structure and three functions: addProduct() (farmer), transferOwnership() (producer/distributor), verifyProduct() (any party). Contract code is immutable once deployed.

6. Web3.py Blockchain Bridge Module

Connects Django to Ganache Ethereum node. Handles transaction building, signing with MetaMask keys, broadcasting, and receipt retrieval for confirmation monitoring.

IV. RESULTS AND DISCUSSION

SYSTEM PERFORMANCE METRICS

Metric	Traditional System	Proposed Blockchain System
Tamper Detection	Partial	100%
Throughput (TPS)	N/A	312
Avg. Confirmation Time (s)	N/A	3.2
Verification Overhead	Baseline	94% Reduction

The prototype was evaluated across a simulated 6-stage supply chain with 500 transactions. Average throughput was 312 TPS with 3.2-second confirmation latency. All tamper attempts were successfully detected through hash validation. Manual quality inspection overhead was reduced by 94% through automated smart contract verification. Cold chain breach incidents were detected within 8 seconds of threshold violation. Consumer QR scan-to-information retrieval averaged 1.8 seconds.

1. Cryptographic Integrity & Tamper Detection

The core value of using an Ethereum-based blockchain (via Ganache) is the immutable audit trail. Every block is cryptographically linked to the previous one.

A. Block Hash Verification (Tamper Check)

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Your algorithm explicitly checks for tampering by verifying that the cryptographic hash of a block matches the prevHash recorded in the subsequent block.

- B_i = The current block.
- B_{i+1} = The next block in the chain.
- H(x) = Cryptographic hash function (e.g., Keccak-256 for Ethereum).

IF Hash(Block_i) == Block_{i+1}.Previous_Hash THEN Valid ELSE Tampered

B. Tamper Detection Rate

Evaluates the system's ability to successfully flag any unauthorized modifications to the supply chain history. Your system achieved a 100% detection rate.

Tamper_Detection_Rate = (Detected_Tampered_Records / Total_Tampered_Records) * 100

2. Blockchain Network Performance

These metrics evaluate the capacity and speed of the Ganache blockchain node and the Web3.py bridge when processing farmer, producer, and distributor transactions.

A. Throughput (Transactions Per Second - TPS)

Measures how many supply chain events (like addProduct or transferOwnership) the network can process in one second. Your system achieved 312 TPS.

- N_{tx} = Total number of transactions processed.
- Delta T = Total time period in seconds.

TPS = Total_Transactions_Processed / Time_Period_in_Seconds

B. Average Confirmation Time

The latency between a stakeholder submitting a transaction via the Django frontend and the transaction being mined into a block. Your paper reports an average of 3.2 seconds.

- N = Total number of transactions.
- t_{submit, i} = Timestamp when transaction i was broadcasted via Web3.
- t_{confirm, i} = Timestamp when the transaction receipt was generated.

Avg_Confirmation_Time = SUM(Time_Confirmed - Time_Submitted) / Total_Number_of_Transactions

3. Operational Efficiency & IoT Metrics

These metrics measure the real-world impact of replacing traditional manual systems with automated Solidity smart contracts and IoT sensors.

A. Verification Overhead Reduction

Quantifies the time and resource savings achieved by using automated smart contract verification (verifyProduct()) instead of manual paperwork checks. Your results show a massive 94% reduction.

- T_{manual} = Time/Cost spent on traditional manual verification.
- T_{smart_contract}= Time/Cost spent using blockchain automated verification.

Overhead_Reduction_Rate = ((Manual_Time - Smart_Contract_Time) / Manual_Time) * 100

B. Consumer QR Retrieval Latency

The total end-to-end time taken for a consumer to scan the QR code and retrieve the full immutable audit trail on their device (averaging 1.8 seconds).

Retrieval_Latency = Timestamp_Audit_Displayed - Timestamp_QR_Scanned

C. Sensor Breach Detection Time

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

The time elapsed between an IoT sensor detecting a cold chain threshold violation (e.g., temperature too high) and the smart contract recording the breach (averaging 8 seconds).

Breach_Detection_Time = Timestamp_Contract_Alert - Timestamp_Physical_Violation

V. CONCLUSION AND FUTURE WORK

This paper presented a blockchain and IoT integrated system for food quality assurance and authenticity verification, achieving tamper-proof traceability, automated compliance enforcement, and transparent consumer-facing provenance verification. Future work will investigate scalability optimizations including sharding and layer-2 solutions, integration with regulatory compliance databases, and AI-based anomaly detection to identify patterns of systematic food fraud.

References

[1] F. Tian, "A Supply Chain Traceability System for Food Safety Based on HACCP, Blockchain and Internet of Things," IEEE ICSSSM, 2017.

[2] K. Salah et al., "Blockchain for AI: Review and Open Research Challenges," IEEE Access, 7, 2019.

[3] H. Feng et al., "Blockchain-Based Data Management for the Internet of Things in Agriculture," Future Generation Computer Systems, 2020.

[4] R. Badia-Melis et al., "New Trends in Cold Chain Monitoring Applications," Food Control, 2015.

[5] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[6] J. Gao et al., "CoC: A Unified Distributed Ledger Based Supply Chain Management System," J. Computer Science and Technology, 2018.

[7] G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Ethereum Yellow Paper, 2014.