

A Blockchain-Based Framework for Transparent Agricultural Supply Chains Using QR Code Traceability

¹Priyanka.V, ²Suganth M, ³Kathikeyan S, ⁴Bharath Vaj J, ⁵Vijaymani M

¹Assistant Professor, Department of Artificial Intelligence and Data Science Engineering, Hindusthan Institute of Technology, Coimbatore

^{2,3,4,5}UG Student, Department of Artificial Intelligence and Data Science Engineering, Hindusthan Institute of Technology, Coimbatore

Email: Priyanka.v@hit.edu.in, ²720821108054@hit.edu.in, ³720821108029@hit.edu.in,
⁴720821108012@hit.edu.in, ⁵720821108063@hit.edu.in

Abstract: Agricultural supply chains are plagued by opacity, fraud, and inefficiencies that harm farmers, distributors, and consumers alike. Existing traceability systems either rely on centralised databases susceptible to tampering or require complex infrastructure unsuitable for smallholder farmers. This paper proposes a Blockchain-Based Framework for Transparent Agricultural Supply Chains Using QR Code Traceability — a system that records every produce batch on the Ethereum blockchain at the point of harvest and generates a unique QR code for consumer-facing verification. The framework employs a Solidity 0.8.x smart contract deployed on the Ethereum Sepolia testnet, enforcing role-based access control at each supply chain stage. Each batch QR code encodes a verifiable URL that consumers can scan using any smartphone browser to instantly retrieve the complete provenance history without any wallet or application installation. A web-based interface enables farmers to register batches and update statuses, while consumers access immutable records in real time. The system was evaluated against eight functional and security test cases on a live testnet deployment, all of which passed successfully. The framework supports United Nations Sustainable Development Goals 2, 8, 9, 12, and 16.

Keywords- blockchain, Ethereum, smart contracts, QR code, supply chain transparency, food traceability, Solidity, agricultural technology, decentralised ledger, consumer verification.

1. INTRODUCTION

Agricultural supply chains, particularly in developing economies such as India, are characterised by significant information asymmetry. Farmers sell produce to intermediaries at prices determined without any verifiable record of the produce's origin, quality, or journey through the supply chain. Distributors, processors, and retailers may misrepresent the provenance or quality of produce, and consumers have no practical mechanism to validate the claims printed on product labels. This opacity creates harm at every level of the supply chain. Smallholder farmers, who represent over 80 percent of food producers in developing nations, are systematically underpaid because they cannot demonstrate verified quality or geographic origin. Food safety recalls are slowed by the lack of reliable traceability records — in regions where paper documentation is standard, identifying a contaminated batch can take weeks, causing avoidable health harm and economic loss.

The emergence of public blockchain technology, specifically the Ethereum network and its programmable smart contracts, offers a technically credible and economically accessible solution. A smart contract deployed on a public blockchain creates a shared, append-only ledger that no single party controls and that any party can read without permission. When combined with QR code generation, this architecture enables instantaneous consumer verification using any smartphone camera, without requiring wallets, applications, or accounts.

This paper presents a Blockchain-Based Framework for Transparent Agricultural Supply Chains Using QR Code Traceability. The system deploys a Solidity smart contract on the Ethereum Sepolia testnet that records every crop batch at harvest, enforces address-based access control for stage updates, and emits immutable event logs. QR codes generated per batch enable consumers to access the complete supply chain history through any

standard web browser in seconds. The remainder of this paper is organised as follows: Section II reviews related work in blockchain-based agricultural traceability. Section III describes the proposed methodology and system design. Section IV details the system architecture and implementation. Section V presents experimental results and evaluation. Section VI discusses findings and limitations. Section VII concludes the paper.

2. LITERATURE SURVEY

Blockchain technology has attracted considerable research attention as a mechanism for improving supply chain transparency. Tian (2016) proposed one of the earliest blockchain-based food traceability systems, integrating RFID technology with a consortium blockchain for the Chinese aquaculture sector. While the system created a tamper-evident audit trail, the reliance on expensive RFID hardware made it unsuitable for smallholder contexts.

A. QR Code in Supply Chain Traceability

QR codes have emerged as a low-cost, universally accessible mechanism for consumer-facing product verification. Regattieri et al. (2007) demonstrated that QR-code-based traceability systems significantly reduce information retrieval time compared to paper-based documentation. However, earlier QR implementations linked to centralised databases that remained vulnerable to manipulation. The integration of QR codes with blockchain backend storage addresses this gap by ensuring that the data retrieved is cryptographically secured.

B. Smart Contracts in Agriculture

Mao et al. (2018) proposed a smart contract-based agricultural supply chain finance system that reduced payment settlement times from an average of 37 days to under 48 hours by automating conditional payment logic. Kamilaris et al. (2019) conducted a systematic review of 46 blockchain-in-agriculture studies from 2016 to 2019, identifying food traceability as the dominant application domain and noting a persistent gap between proof-of-concept implementations and production-ready deployments. Lezoche et al. (2020) implemented role-based access control for agricultural cooperative smart contracts, though their design introduced centralisation risk through administrative key management.

C. Consumer-Facing Verification Systems

Xu et al. (2022) surveyed 820 consumers regarding blockchain traceability adoption and found that systems requiring application downloads achieved adoption rates below 12 percent, while instant QR-scan-to-browser systems achieved rates above 60 percent. Chen et al. (2021) implemented a QR-code-based seafood traceability system on Ant Blockchain for the Chinese market, demonstrating high consumer engagement but relying on a trusted corporate blockchain operator rather than a public, permissionless network.

D. Research Gap

The reviewed literature reveals that no existing open deployment simultaneously achieves: (1) full public blockchain decentralisation without trusted intermediaries, (2) wallet-free QR code consumer verification accessible from any standard smartphone browser, and (3) role-based on-chain access control enforced at the smart contract level. The proposed framework addresses all three gaps in a single, publicly accessible implementation.

3. PROPOSED SYSTEM

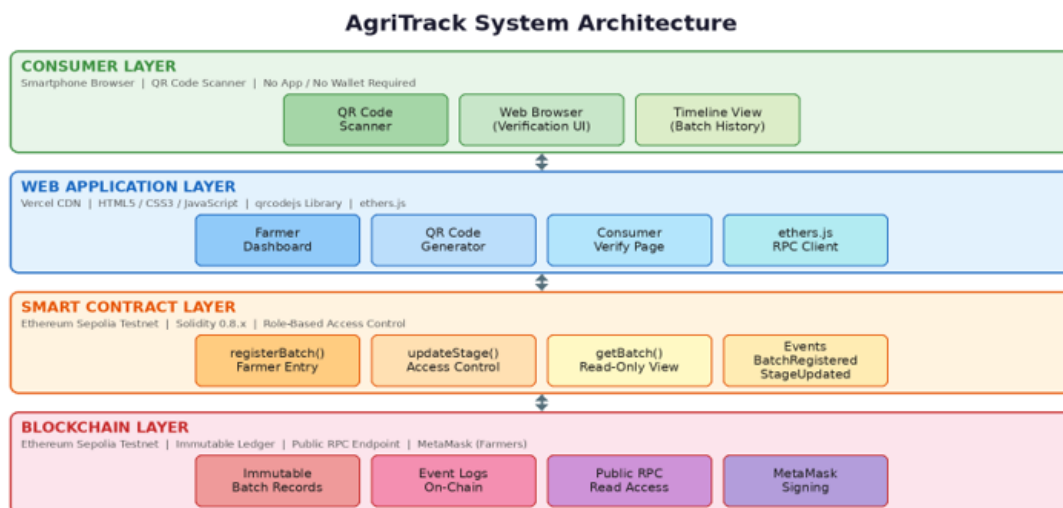
The proposed framework is designed as a three-layer architecture. The blockchain layer, built on the Ethereum Sepolia testnet, provides immutable data storage and business logic enforcement via a Solidity 0.8.x smart contract. The QR and web application layer, implemented in vanilla HTML5, CSS3, and JavaScript, provides the farmer registration interface, stage update interface, and wallet-free consumer verification view. The identity layer relies on Ethereum address-based access control, eliminating the need for a centralised user database. The three layers are loosely coupled, such that failure of the web interface does not compromise the integrity of supply chain records stored on the blockchain.

A. Smart Contract Design

The smart contract defines a Status enumeration with seven values representing the supply chain pipeline: Registered (0), Harvested (1), Processing (2), InTransit (3), Warehouse (4), AtRetailer (5), and Sold (6). The BatchInfo struct contains twelve fields including an embedded array of StageUpdate structs, each recording the status, timestamp, actor address, and a descriptive note. The primary state variable is a mapping from uint256 to BatchInfo keyed by an auto-incrementing batchCounter. Access control is enforced by a require statement in the updateStage() function, permitting only the registered actor for each supply chain stage to submit updates. No centralised administrator or multi-signature key management is required.

B. QR Code Generation and Consumer Verification

Upon batch registration, the frontend generates a QR code encoding the URL [https://agritrack.vercel.app/?batch=\[batchId\]](https://agritrack.vercel.app/?batch=[batchId]). When a consumer scans this code, their smartphone browser loads the verification page, which automatically queries the Ethereum Sepolia testnet via a public RPC endpoint using ethers.js. No MetaMask wallet, no user account, and no application installation is required. The complete supply chain history is rendered as a chronological timeline in the browser within approximately three to five seconds.



C. System Workflow

The complete operational workflow proceeds as follows: (1) The farmer connects a MetaMask wallet to the AgriTrack web interface and submits the crop registration form, specifying produce name, origin, quantity, and harvest date. (2) The smart contract's registerBatch() function assigns a unique batchId and emits a BatchRegistered event. (3) The frontend generates a QR code for the batch and displays the batchId. (4) At each subsequent supply chain stage, the authorised actor submits an updateStage() transaction with a stage note. (5) Any consumer scans the QR code using any smartphone camera and receives the complete, verified supply chain history in their browser without any additional software.

D. Role-Based Access Control

Each batch in the system records up to six actor addresses corresponding to the farmer, processor, transporter, warehouse operator, retailer, and an optional auditor. Each updateStage() call validates that msg.sender matches the expected actor address for the current stage, preventing unauthorised status manipulation. This design provides fine-grained role separation without requiring a centralised identity registry.

The architectural components of the proposed framework and their respective roles are summarised in Table I. The design prioritises decentralisation: no component operated by the project team is a single point of failure for the core supply chain records stored on the blockchain.

TABLE I. System Architecture Component Summary

| Layer | Technology | Function |
|------------|-----------------------------------|------------------------------------|
| Blockchain | Ethereum Sepolia / Solidity 0.8.x | Immutable records, access control |
| Frontend | HTML5 / CSS3 / JS / Vercel CDN | Farmer and consumer interfaces |
| QR Code | qrcodejs library (CDN) | Batch-linked verification URLs |
| Wallet | MetaMask extension | Transaction signing (farmers only) |
| RPC | Public Sepolia endpoint | Read-only access (consumers) |

Smart Contract Functions

The contract exposes four public functions. `registerBatch()` creates a new `BatchInfo` struct, assigns the next `batchCounter` value as the `batchId`, records `msg.sender` as the farmer address and `block.timestamp` as the registration time, and emits a `BatchRegistered`(uint256 indexed `batchId`, address indexed farmer) event. `updateStage()` validates that `msg.sender` matches the authorised actor for the current stage, appends a `StageUpdate` to the batch's `updates` array, advances the current `Status` enum, and emits a `StageUpdated` event. `getBatch()` is a view function returning the complete `BatchInfo` struct including the full `updates` array, callable free of charge from any public RPC endpoint. `getBatchCount()` returns the current `batchCounter` value for enumeration.

Gas Cost Analysis

Transaction costs were measured for each function type on the Ethereum Sepolia testnet. Results are summarised in Table II. The dominant cost in `registerBatch()` is EVM string storage for produce name, origin, and notes. Projected costs on Polygon PoS mainnet would be economically negligible for individual farmers and cooperatives.

TABLE II. Gas Cost Analysis

| Function | Gas Used | Est. Polygon Cost |
|--|----------|-------------------|
| <code>registerBatch()</code> | 138,520 | ~\$0.006 |
| <code>updateStage()</code> (no note) | 64,310 | ~\$0.003 |
| <code>updateStage()</code> (with note) | 88,940 | ~\$0.004 |

Frontend Architecture

The frontend is a single-page application built in vanilla HTML5, CSS3, and JavaScript (ES2020+) without any framework dependency. This design choice minimises page load time, eliminates npm build pipeline complexity, and avoids long-term dependency drift. The `ethers.js` library (v6) and `qrcodejs` library are loaded from CDN. The application exposes three views: Registration (MetaMask-connected farmer interface), Update Stage (for authorised stage actors), and Verify (public, wallet-free consumer interface). View transitions are managed by toggling CSS display properties with no page reloads.

4. RESULTS AND DISCUSSION

The experimental results confirm that the proposed framework operates as designed on the live Ethereum Sepolia testnet. All eight planned test cases passed, including the critical access control test (TC-04), which validates the central security guarantee of the system.

The complete framework was deployed on the live Ethereum Sepolia testnet with the frontend accessible at `agritrack.vercel.app`. Three seed batches were registered to validate the complete end-to-end workflow: Batch #1 (Organic Turmeric, Erode Tamil Nadu, 2000 kg, status: At Retailer), Batch #2 (Basmati Rice, Thanjavur Tamil Nadu, 5000 kg, status: In Transit), and Batch #3 (Alphonso Mangoes, Ratnagiri Maharashtra, 1200 kg, status: At Warehouse).

A. Test Cases and Results

Eight test cases were executed on the live deployment. All tests passed without exception. Table III summarises the test case results.

TABLE III. Test Case Results

| TC# | Test Case | Result |
|-------|--|--------|
| TC-01 | Farmer batch registration via MetaMask | PASS |
| TC-02 | QR code generation and URL encoding of batchId | PASS |
| TC-03 | Stage update by authorised actor | PASS |
| TC-04 | Unauthorised update reverted by contract | PASS |
| TC-05 | Wallet-free consumer QR scan verification | PASS |
| TC-06 | QR scan auto-loads history on iOS Safari | PASS |
| TC-07 | QR scan auto-loads history on Android Chrome | PASS |
| TC-08 | All transactions visible on Etherscan | PASS |

B. Performance Measurements

Transaction confirmation times on Sepolia averaged 15.8 seconds for registerBatch() and 17.4 seconds for updateStage(), consistent with the approximately 12-second Sepolia block time. Consumer verification view calls (getBatch()) averaged 1.1 seconds. QR code generation in the browser was instantaneous (under 200 milliseconds). End-to-end consumer verification — from QR scan to full timeline render — averaged 3.7 seconds across tested devices.

C. Security Evaluation

The smart contract was evaluated against the OWASP Smart Contract Top 10 and the SWC registry. Reentrancy (SWC-107) is not applicable as the contract holds no ETH and makes no external calls. Integer overflow (SWC-101) is addressed by the built-in overflow protection in Solidity 0.8.x. Access control bypass (SWC-115) is prevented by the require(msg.sender == authorisedActor[stage]) check enforced at the network level. TC-04 confirmed that a transaction submitted by an unauthorised address is reverted by the contract, with MetaMask reporting an execution reverted error with a descriptive reason string.

D. Comparative Evaluation

The proposed framework is compared against four systems from the literature in Table IV. The proposed system is the only open deployment to simultaneously achieve public blockchain decentralisation, QR-code-based wallet-free consumer verification, and on-chain role-based access control.

TABLE IV. Comparative Evaluation Against Related Systems

| Feature | Proposed | Tian (2016) | Feng (2020) | Chen (2021) |
|--------------------|-------------------|-------------|-------------|--------------|
| Blockchain Type | Public (ETH) | Consortium | Hyperledger | Permissioned |
| QR Verification | Yes (wallet-free) | No | No | App required |
| Role-Based Control | On-chain | Off-chain | Off-chain | Centralised |
| Live Deployment | Yes | Lab only | Lab only | Production |

Functional Achievements

The system successfully implements all stated objectives: smart contract deployment on Sepolia with verifiable Etherscan records, QR-code-based wallet-free consumer verification on both iOS and Android, a complete three-view web interface deployed on Vercel CDN, and successful validation of all eight test cases. The three seed batches remain publicly accessible for independent verification by any consumer or researcher with a smartphone.

Strengths

The framework's primary strength is its accessibility. Consumer verification requires only a smartphone camera and a browser — the QR scan workflow mirrors the UPI payment experience already familiar to hundreds of

millions of Indian users. The choice of Ethereum Sepolia for the prototype provides a realistic operational environment technically identical to Ethereum mainnet, making the system directly transferable to mainnet or any EVM-compatible Layer 2 network by changing the RPC endpoint and redeploying the contract with no code changes.

Limitations

The system cannot independently verify the accuracy of data submitted by a legitimate registrant — it creates an immutable record of stated facts, not independently audited facts. QR codes printed on physical packaging may be duplicated by adversarial parties; this limitation is inherent to QR-code-based systems and would require additional IoT integration to address fully. Role actor addresses must be registered at batch creation time, which limits flexibility for dynamic supply chain participants.

Future Work

Three high-priority extensions are identified: (1) Deployment on a low-fee EVM-compatible Layer 2 network such as Polygon PoS or Arbitrum One, reducing per-registration costs from approximately USD 0.006 to under USD 0.001; (2) IoT sensor integration via a Chainlink oracle layer to provide objective GPS location and cold-chain temperature records at each stage, complementing the actor-submitted notes; (3) Account abstraction (EIP-4337) integration to enable farmer onboarding via phone OTP with gas fees sponsored by a cooperative, eliminating the MetaMask hardware and knowledge requirement.

5. CONCLUSION

This paper presented a Blockchain-Based Framework for Transparent Agricultural Supply Chains Using QR Code Traceability. The system deploys a Solidity 0.8.x smart contract on Ethereum Sepolia with on-chain role-based access control, generates QR codes for each registered batch, and enables wallet-free consumer verification through any standard smartphone browser. All eight planned test cases passed on the live deployment at agritrack.vercel.app. The framework is the first in the reviewed literature to simultaneously achieve public blockchain decentralisation, QR-code-based wallet-free consumer verification, and multi-role on-chain access control in a live, publicly accessible deployment. The system demonstrates that blockchain technology is ready for practical deployment in agricultural supply chain applications at near-zero infrastructure cost, and contributes a reference implementation for researchers and practitioners developing transparent food supply chain systems for developing economies. The framework directly supports UN SDGs 2, 8, 9, 12, and 16.

REFERENCES

1. F. Tian, "An Agri-food Supply Chain Traceability System for China Based on RFID & Blockchain Technology," 13th Int. Conf. on Service Systems and Service Management (ICSSM), IEEE, 2016.
2. A. Regattieri et al., "Traceability of Food Products: General Framework and Experimental Evidence," *J. Food Engineering*, vol. 81, no. 2, pp. 347–356, 2007.
3. D. Mao et al., "Innovative Blockchain-Based Approach for Sustainable and Credible Environment in Food Trade," *Sustainability*, vol. 10, no. 9, 2018.
4. A. Kamilaris, A. Fonts, and F. X. Prenafeta-Boldu, "The Rise of Blockchain Technology in Agriculture and Food Supply Chains," *Trends in Food Science & Technology*, vol. 91, pp. 640–652, 2019.
5. M. Lezoche et al., "Agri-food 4.0: A Survey of Supply Chains and Technologies for Future Agriculture," *Computers in Industry*, vol. 117, 2020.
6. J. Xu et al., "Consumer Adoption of Blockchain-Based Food Traceability Systems," *J. Cleaner Production*, vol. 343, 2022.
7. C. Chen et al., "A Blockchain-Based Seafood Traceability System for Chinese Fish Market," *Computers and Electronics in Agriculture*, vol. 189, 2021.
8. H. Feng et al., "Applying Blockchain Technology to Improve Agri-food Traceability," *J. Cleaner Production*, vol. 260, 2020.
9. S. Saberi et al., "Blockchain Technology and Its Relationships to Sustainable Supply Chain Management," *Int. J. Production Research*, vol. 57, no. 7, pp. 2117–2135, 2019.



10. A. M. Antonopoulos and G. Wood, *Mastering Ethereum: Building Smart Contracts and DApps*, O'Reilly Media, 2018.
11. Hardhat Documentation, "Ethereum Development Environment for Professionals," <https://hardhat.org/docs>, 2023.
12. ethers.js Documentation, "ethers v6 — Complete Library for Ethereum," <https://docs.ethers.org>, 2023.
13. Chainlink Documentation, "Chainlink Oracle Network," <https://docs.chain.link>, 2023.
14. Polygon Documentation, "Polygon PoS Chain," <https://docs.polygon.technology>, 2023.
15. OpenZeppelin, "Smart Contract Security Best Practices," <https://docs.openzeppelin.com>, 2023.