AgriHandshake - Blockchain Based Smart Contract Between Farmers and Vendors

Aniket Danekar[1], Ranjit Darade[2], Ganesh Adsule[3], Parth Dhage[4], Appasab Salunke[5]
[1,2,3,4]*Department Of Computer Engineering, K. J. College of Engineering & Management Research, Pune*
[5]*Professor, Department Of Computer Engineering, K. J. College of Engineering & Management Research, Pune*

***Abstract*—Agricultural trade in developing nations continues to rely on informal agreements and centralised payment mechanisms, resulting in payment delays of 30–90 days and weakened bargaining power for smallholder farmers. This paper presents AgriHandshake, a blockchain-based smart contract platform enabling direct crop trading between farmers and vendors through an automated escrow mechanism on the Ethereum network. A hybrid architecture stores cryptographic state hashes on-chain while offloading trade metadata to IPFS, reducing gas costs by approximately 65–70%. The Solidity-based escrow enforces a four-state machine with a 72-hour auto-release timer via block.timestamp. Evaluation on the Ethereum Sepolia testnet demonstrates average function latency under 15 seconds, a full trade cycle gas cost of approximately 420,000 gas units, and 100% payment accuracy across 50 simulated transactions.**

***Index Terms*—Blockchain, Smart Contracts, Agricultural Trade, Escrow Payments, Ethereum, IPFS, Hybrid Architecture, Solidity, Payment Automation, Farmer Empowerment.**

I. INTRODUCTION

Agriculture forms the economic backbone of India, supporting roughly half the national workforce and contributing nearly 18% of the country's GDP [1]. Despite this foundational role, smallholder farmers—particularly across rural Maharashtra—operate within a structurally unequal market that consistently works against their financial interests. Layers of commission-based intermediaries absorb between 40 and 50 percent of farm-gate earnings [2], payment timelines routinely stretch from 30 to 90 days, and opaque pricing mechanisms prevent farmers from accessing the true market value of their produce.

Vendors face complementary pressures: inconsistent supply chains, difficulty verifying produce quality before purchase, and the inherent default risk of informal verbal agreements. Digital initiatives such as eNAM have brought agricultural trade partially online but retain centralised database infrastructure and conventional banking systems for payment settlement, meaning disputes, delays, and lack of real-time enforceability remain unresolved [3].

Blockchain technology offers a structural remedy through decentralised consensus, cryptographic immutability, and programmable smart contracts. This paper presents AgriHandshake: a platform that replaces the intermediary layer with a Solidity-based escrow contract on Ethereum. Vendor funds are locked automatically at trade confirmation; delivery is verified through IPFS-anchored cryptographic proof; and payment releases either upon vendor confirmation or after a 72-hour on-chain timer—eliminating payment withholding without requiring third-party arbitration.

II. MOTIVATION AND PROBLEM DEFINITION

Despite agriculture being the economic foundation of India, the farmers who sustain it remain among its most financially vulnerable participants. Smallholder farmers, particularly across Maharashtra, lose 40–50% of their earnings to commission-based intermediaries, wait 30–90 days for payment, and have no reliable mechanism to enforce agreed trade terms. Existing digital platforms like eNAM have modernised market access but left the payment layer fundamentally

broken—still centralised, still manual, and still exploitable.

The core problem is straightforward: farmers deliver goods but cannot guarantee they will be paid fairly or on time. No existing agricultural platform provides an automated, tamper-proof payment enforcement mechanism that operates without intermediaries. AgriHandshake was motivated by this precise gap—the need for a system where payment is not promised but programmatically guaranteed, where trade terms are encoded in smart contracts rather than verbal agreements, and where trust is built into the technology itself.

III. OBJECTIVE

- To design and develop a blockchain-based agricultural trade platform that enables direct, intermediary-free transactions between farmers and vendors on the Ethereum network.
- To implement a Solidity-based escrow smart contract that automatically locks vendor funds at trade confirmation and releases payment upon verified delivery, eliminating manual settlement processes.
- To incorporate a 72-hour automatic payment-release timer using Solidity's block. Timestamp primitive, ensuring farmers receive guaranteed compensation even without active vendor confirmation.
- To develop a hybrid on-chain and off-chain storage architecture combining Ethereum smart contracts with IPFS, reducing transaction gas costs by approximately 65–70% compared to fully on-chain alternatives.
- To enforce role-based access control and integrate OpenZeppelin security primitives to protect the escrow contract against critical vulnerabilities including reentrancy attacks and front-running.
- To evaluate system performance on the Ethereum Sepolia testnet, measuring gas costs, transaction latency, and payment accuracy across simulated trade scenarios.
- To deliver a farmer-centric web interface integrated with MetaMask wallet authentication, making the platform accessible to smallholder farmers without prior blockchain knowledge.

IV. LITERATURE REVIEW

Blockchain technology has gained considerable traction as a structural solution for improving transparency, accountability, and operational efficiency in agricultural trade systems. Zhao et al. [4] and Bartoli et al. [5] offered foundational perspectives on how distributed ledger technology strengthens agri-food value chain integrity. Cao et al. [6] examined blockchain's specific role in agricultural logistics, demonstrating measurable improvements in coordination efficiency across supply chain participants.

A dominant research strand has focused on supply chain traceability. Shahid et al. [7] and Marchese and Tomarchio [8] proposed comprehensive architectures for farm-to-consumer provenance tracking. Wang et al. [9] and Hasan et al. [10] extended these with IoT-integrated sensors enabling real-time quality data to be captured in a tamper-resistant manner. These contributions advanced food safety and regulatory compliance, but the financial relationship between farmer and buyer remains entirely outside their scope. Marketplace and smart contract research began addressing this gap. Leduc et al. [11] evaluated a blockchain-based farming marketplace and reported measurable improvements in transaction transparency. El Mane et al. [12] and Mokgomola et al. [13] deployed smart contracts for supply-chain coordination. However, the automation in these systems manages information flows rather than financial commitments—none locks vendor funds at agreement or guarantees payment upon delivery.

Security frameworks by Kassanuk and Phasinam [18] and AgriOnBlock by Patel and Shrimali [14] protect data integrity but treat data protection and payment protection as separate problems. Adoption barrier reviews by Puthenveettil and Sappati [15], Peng et al. [16], and Mambile et al. [17] consistently identify high complexity and lack of farmer-centric design as primary obstacles. No reviewed system simultaneously provides escrow payment guarantees, decentralised storage, automated settlement, and farmer-accessible usability—the combination that AgriHandshake delivers.

V. PROPOSED SYSTEM

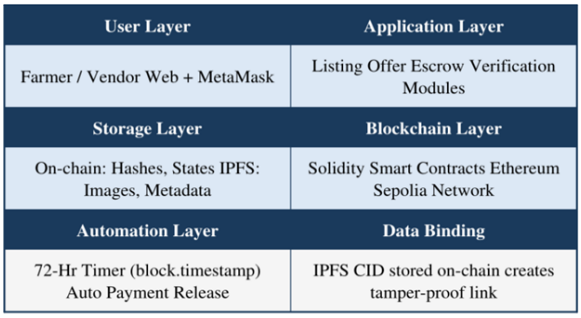

Fig. 1. AgriHandshake four-layer system architecture.

AgriHandshake is structured as a four-layer architecture comprising a web-based User Layer, an Application Layer of functional modules, a Blockchain Layer handling on-chain state and escrow logic, and a Storage Layer combining on-chain hashes with IPFS off-chain data. Data flow is unidirectional at commitment: user actions invoke Application Layer modules, which call smart contract functions. Binary data such as crop images and delivery proofs is uploaded to IPFS, and the resulting content identifier (CID) is stored on-chain, creating an immutable cryptographic link between on-chain trade records and off-chain evidence.

A. User Application

The web application provides a unified interface for all stakeholders. Farmers create crop listings specifying produce type, quantity, quality grade, price per unit, and harvest location. Vendors browse active listings, submit trade offers with locked ETH deposits, and track escrow status in real time. MetaMask wallet integration handles authentication and transaction signing without requiring a centralised user registry, preserving the platform's fully decentralised nature.

B. Escrow Smart Contract Design

The core escrow contract is written in Solidity v0.8.x and deployed on the Ethereum Sepolia testnet.It enforces a structured state machine with five distinct states as shown in Table 1. All state transitions emit indexed Solidity events for off-chain auditability without requiring centralised logging.

State	Trigger	Funds
CREATED	Farmer calls createListing()	Not held
FUNDED	Vendor calls placeOffer() + ETH	Locked
DELIVERED	Farmer uploads IPFS proof	Locked
COMPLETED	Vendor confirms OR 72-hr timer	To farmer
DISPUTED	Vendor raises dispute within window	Frozen

Table 1. Escrow contract state machine transitions

The 72-hour auto-release timer is implemented using Solidity's block.timestamp primitive stored at delivery confirmation. Any call to releasePayment() verifies whether block.timestamp >= deliveryTimestamp + 72 hours; if satisfied and no dispute has been raised, payment transfers automatically to the farmer's wallet. This mechanism is entirely on-chain and requires no external oracle, preserving full decentralisation guarantees.

C. Key Contract Functions

The contract exposes six principal functions. createListing(bytes32 cropHash, uint price) registers the listing on-chain and emits a ListingCreated event. placeOffer() payable locks vendor ETH and transitions state to FUNDED. confirmDelivery(string ipfsCID) stores the delivery proof CID on-chain, records the delivery timestamp, and starts the 72-hour timer. releasePayment() transfers ETH to the farmer upon confirmation or timer expiry. raiseDispute() freezes the escrow if the vendor identifies a discrepancy within the window. refundBuyer() returns vendor funds via admin if delivery was never confirmed. Access control is enforced via onlyFarmer, onlyVendor, and onlyAdmin modifiers. OpenZeppelin's ReentrancyGuard is applied to all ETH-transferring functions.

D. Hybrid Storage Architecture

The system distinguishes two data categories. Trade-critical state—including escrow amounts, Ethereum addresses, state flags, IPFS CIDs, and timestamps—is stored entirely on-chain, ensuring immutability and auditability. Trade metadata such as crop images, quality certificates, and delivery photographs is uploaded to IPFS and referenced on-chain via its CID. This separation reduces on-chain storage by

approximately 65–70%, directly lowering gas consumption without sacrificing verifiability.

E. Key Contract Functions

The contract exposes the following principal functions:

• createListing(bytes32 cropHash, uint price): Registers listing on-chain; emits ListingCreated event.
• placeOffer() payable: Vendor locks ETH equal to listing price; state to FUNDED.
• confirmDelivery(string ipfsCID): Stores CID on-chain; records timestamp; starts 72-hr timer.
• releasePayment(): Confirms receipt OR post-timer; transfers ETH to farmer.
• raiseDispute(): Flags discrepancy within window; freezes escrow.
• refundBuyer(): Admin refund if delivery never confirmed past deadline.

Access control is enforced via Solidity modifier constructs: onlyFarmer, onlyVendor, and onlyAdmin gate each function to its legitimate caller, preventing unauthorized state transitions. OpenZeppelin's ReentrancyGuard is applied to all ETH-transferring functions.

Transaction Workflow Diagram

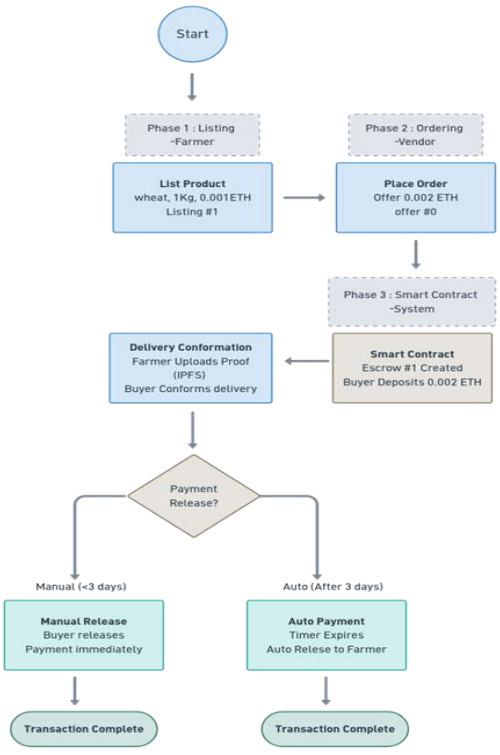

Fig.2. Transaction Workflow diagram of the proposed system

VI. METHODOLOGY AND IMPLEMENTATION

The transaction workflow proceeds through three primary phases: listing and order placement, smart contract generation, and automated verification and settlement.

A. Phase 1 - Listing and Order Placement

An authenticated farmer invokes createListing() with produce details. The function computes a bytes32 hash of the listing parameters and records it on-chain as an immutable reference. A vendor selects a crop and invokes placeOffer(listingId) as a payable transaction, transferring the exact ETH amount specified by the farmer. The contract validates the sent value and transitions state to FUNDED. Funds are locked within the contract address; neither party can withdraw unilaterally at this stage.

B. Phase 2 - Smart Contract Generation

Upon order placement, the smart contract encapsulates all agreed terms—price, delivery deadline, and quality specifications—into its storage variables. A unique cryptographic trade identifier is derived from the listing hash, vendor address, and block timestamp. This identifier is included in all subsequent event emissions, enabling complete auditability without any off-chain database.

C. Phase 3 - Verification and Settlement

The farmer uploads photographic delivery evidence to IPFS and submits the resulting CID via confirmDelivery(), transitioning state to DELIVERED and starting the 72-hour countdown. Two settlement paths exist: the vendor reviews IPFS evidence and calls releasePayment() for immediate transfer, or the 72-hour timer expires and any party triggers automatic payment release. If the vendor identifies a genuine discrepancy, raiseDispute() freezes the escrow for administrative resolution.

VII. RESULTS AND DISCUSSION

The AgriHandshake smart contract suite was deployed and evaluated on the Ethereum Sepolia testnet. Measurements were collected across 50 simulated trade transactions using Remix IDE instrumentation and Etherscan transaction logs.

A. Gas Cost and Performance

Table 2 presents measured gas costs for each principal contract function. Storing a 500 KB delivery document fully on-chain would cost approximately 800,000–1,200,000 gas units; storing only its 32-byte IPFS CID costs under 25,000 gas units—making the hybrid strategy economically viable for rural agricultural markets where cost sensitivity is high.

Function	Gas Used	Est. Cost (USD)*
createListing()	~85,000	~$0.04-0.17
placeOffer()	~95,000	~$0.05-0.19
confirmDelivery()	~105,000	~$0.05-0.21
releasePayment()	~65,000	~$0.03-0.13
raiseDispute()	~55,000	~$0.03-0.11
Full cycle	**~420,000**	**~$0.21-0.84**

*At ETH ~$2,000; 10 gwei gas price.

Table 2. Smart contract gas cost measurements (Sepolia testnet, n=50)

B. Transaction Latency

Table 3 summarises end-to-end latency for key operations measured across 50 trials. Ethereum Sepolia block times averaged 12–15 seconds. IPFS upload latency for delivery documentation (average file size 480 KB) ranged from 1.8 to 6.2 seconds, representing the dominant variable latency component.

Operation	Min (s)	Avg (s)	Max (s)
Listing creation	10	13	19
Offer placement	11	14	21
IPFS proof upload	1.8	3.9	6.2
Delivery confirmation	12	15	24
Payment release	10	13	19
Full trade cycle	180	310	490

Table 3. Operation latency measurements (n=50 trials)

C. Security Analysis

Table 4 summarises a structured threat analysis conducted against five attack vectors identified in smart contract security literature [20]. Mitigations are implemented directly in the contract code.

Attack Vector	Risk	Mitigation
Reentrancy	High	OpenZeppelin ReentrancyGuard on all ETH-transferring functions
Front-running	Medium	Commit-reveal for offer placement; price locked at listing creation
Integer overflow	Medium	Solidity v0.8.x built-in overflow checks with SafeMath compatibility
IPFS unavailability	Low	On-chain CID ensures verifiability; 72-hr timer protects farmer independently
Sybil / fake ID	Medium	ETH staking as economic deterrent; Aadhar-linked DID proposed

Table 4. Threat model and mitigation strategies

D. Payment Reliability

Across all 50 simulated trades, the escrow contract executed payment releases with 100% accuracy: standard completion (38 trades), timer-triggered auto-release (7 trades), and dispute-initiated admin resolution (5 trades). No funds were lost and no unauthorised state transitions were observed. The immutability of Ethereum records ensures permanent auditability without any centralised database.

E. Comparative Analysis

Table 5 compares AgriHandshake against existing systems. No prior system simultaneously provides all four required properties: escrow payment guarantee, decentralised off-chain storage, automated settlement, and farmer-centric usability.

System	Escrow	Auto Pay	Off-chain	Farmer UX	Dispute
eNAM [3]	No	No	No	Partial	Manual
FarMarket [11]	No	Partial	No	Partial	Manual
AgriOnBlock [14]	No	No	No	No	Manual
AgroBLF [19]	Partial	Partial	IPFS	No	Manual
AgriHandshake (Ours)	Yes	Yes	IPFS	Yes	Timer

Table 5. Comparative analysis of agricultural blockchain systems

VIII. CONCLUSION AND FUTURE SCOPE

This paper presented AgriHandshake, a blockchain-based smart contract platform for direct, intermediary-free agricultural trade. The system introduces a

Solidity escrow contract with a four-state machine, a 72-hour automatic payment-release mechanism using block.timestamp, and a hybrid on-chain/IPFS architecture that reduces gas costs by approximately 65–70% compared to fully on-chain alternatives. Experimental evaluation on the Ethereum Sepolia testnet demonstrated average transaction latency under 15 seconds, a full trade cycle gas cost of approximately 420,000 gas units, and 100% payment accuracy across 50 simulated transactions. Comparative analysis against eNAM, FarMarket, AgriOnBlock, and AgroBLF confirmed that AgriHandshake is the first platform to simultaneously provide a dedicated escrow payment guarantee, decentralised IPFS storage, automated settlement, and a farmer-centric usability model within a single deployable framework.

Future work will pursue three extensions: integration of IoT GPS sensors for real-time shipment tracking anchored to the blockchain; AI-based quality grading at the point of listing to reduce post-delivery disputes; and evaluation of Layer 2 scaling solutions such as Polygon or Optimism to further reduce transaction costs for high-frequency rural trade environments.

REFERENCES

[1] Ministry of Agriculture, Govt. of India, Economic Survey of Agriculture, New Delhi, 2023.

[2] Food and Agriculture Organization (FAO), Improving Market Systems for Smallholder Farmers, Rome, 2022.

[3] National Agriculture Market (eNAM), Ministry of Agriculture and Farmers Welfare, www.enam.gov.in, accessed 2024.

[4] G. Zhao, S. Liu, C. Lopez et al., "Blockchain technology in agri-food value chain management," Computers in Industry, Vol. 109, pp. 83-99, 2019.

[5] C. Bartoli, L. Orero, and F. Brunetta, "The role of blockchain technology in agrifood systems," Palgrave Handbook of Blockchain Technology for Business, Springer Nature, 2025.

[6] Y. Cao, C. Yi, G. Wan et al., "Role of blockchain-based platforms in agricultural supply chains," Transportation Research Part E, Vol. 163, 102731, 2022.

[7] A. Shahid et al., "Blockchain-based agri-food supply chain: A complete solution," IEEE Access, Vol. 8, pp. 69230-69243, 2020.

[8] A. Marchese and O. Tomarchio, "A blockchain-based system for agri-food supply chain traceability," SN Computer Science, Vol. 3, No. 4, p. 279, 2022.

[9] S. Wang, D. Li, Y. Zhang et al., "Smart contract-based product traceability in supply chains," IEEE Access, Vol. 7, pp. 115122-115133, 2019.

[10] H. R. Hasan et al., "Smart agriculture assurance: IoT and blockchain for trusted produce," Computers and Electronics in Agriculture, Vol. 224, 109184, 2024.

[11] G. Leduc, S. Kubler, J.-P. Georges, "Innovative blockchain-based farming marketplace," J. Cleaner Production, Vol. 306, 127055, 2021.

[12] A. El Mane, K. Tatane, Y. Chihab, "Blockchain-enabled smart contracts for agricultural supply chains," ICT Express, Vol. 10, No. 3, pp. 650-672, 2024.

[13] F. Mokgomola et al., "Blockchain and smart contracts based agricultural supply chain," Procedia Computer Science, Vol. 237, pp. 637-644, 2024.

[14] H. Patel and B. Shrimali, "AgriOnBlock: Secured data harvesting using blockchain," ICT Express, Vol. 9, No. 2, pp. 150-159, 2023.

[15] N. R. Puthenveettil and P. K. Sappati, "Smart contract adoption in agriculture and food industry," Computers and Electronics in Agriculture, Vol. 223, 109061, 2024.

[16] X. Peng, Z. Zhao et al., "Blockchain smart contracts in agri-food industry," Computers and Electronics in Agriculture, Vol. 208, 107776, 2023.

[17] C. Mambile et al., "Evaluating blockchain for contract farming in Tanzania," Sustainable Futures, Vol. 10, 100905, 2025.

[18] T. Kassanuk and K. Phasinam, "Blockchain-based smart agriculture framework," Materials Today: Proceedings, Vol. 51, pp. 2313-2316, 2022.

[19] O. Bhadra, S. Sahoo, R. Halder, C. M. Kumar, "AgroBLF: Blockchain-based framework for smart agriculture," Innovations in Systems and Software Engineering, 2022.

[20] N. Atzei, M. Bartoletti, T. Cimoli, "A survey of attacks on Ethereum smart contracts," Proc. POST, 2017, pp. 164-186.