Volume 2, 2026, Pages 162-173 *Journal of Smart and Sustainable Farming*
| Regular Article | Open Access

Blockchain Enabled Traceability Framework for Sustainable Agricultural Supply Chain Verification

Elena Mikhailovna
Plant and Environmental Sciences, University of Copenhagen, 1871 Frederiksberg C, Denmark.
elena8566@hotmail.com

Article Info
Journal of Smart and Sustainable Farming
https://www.ansispublications.com/jssf/jssf.html

© The Author(s), 2026.

https://doi.org/10.64026/JSSF/2026014

Received 28 March 2026
Revised from 16 May 2026
Accepted 02 July 2026
Available online 30 July 2026
Published by Ansis Publications.

Corresponding author(s):
Elena Mikhailovna, Plant and Environmental Sciences, University of Copenhagen, 1871 Frederiksberg C, Denmark.
Email: elena8566@hotmail.com

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/ licenses/by/4.0/).

Abstract - The growing demand from buyers, regulators and consumers for evidence that food is safe, fairly produced and environmentally responsible, and the evidence is contained in separate, self-reported and easily changeable records. This paper presents a traceability system based on the blockchain that can be used for verifying agricultural supply chains and the sustainability claims of them. The framework incorporates IoT and tag-based information collection, off-chain storage of information, hash-based digests of the information (SHA-256), anchored with a Merkle-root on the ledger, a permissioned ledger with smart contracts for lot registration, custody transfer, cold-chain compliance and sustainability scoring, and a QR-based consumer verification process. A prototype implementation of the data layer was tested and measured, while the ordering and commit pipeline was tested using discrete-event simulation. Verifying a Merkle proof for 100,000 records used 17.5 µs, and a proof of only 544 bytes was used, while hash-chain verification found and fixed all 1200 records in a test of tampering and found 5 altered records 23 % of the time in a test of random audits of 5 % of records. The ledger size was decreased by 289 times by using the digests rather than raw sensor data. The simulated four node network could process as many as 572 transactions per second, while throughput dropped when there were more than approximately 32 nodes in the ordering cluster. A cold-chain contract triggered an alert 50 min after the first exceedance, and ledger-verified data showed that self-reported sustainability scores were 9.5 points higher than that of the farm with the highest of the misreporting farms and promoted 37 of 120 lots to a higher score. The framework details how blockchain technology can shift the focus of agricultural sustainability claims from assertion to verifiable evidence.

Keywords - Blockchain, Traceability, Agricultural Supply Chain, Sustainability Verification, Smart Contracts, Merkle Tree.

I. INTRODUCTION

Agricultural supply chains connect smallholder farms, aggregators, processors, cold stores, distributors and retailers, sometimes on a transnational basis and through dozens of transactions. Traceability—to track the movement of a product and its characteristics from production through processing and distribution—is a fundamental food safety, recall management, quality assurance and market access requirement [1]. Its concepts for bulk goods like grain were developed in the field of lot identification and in the fields of keeping records and exchanging information between supply-chain partners [2] and in early digital systems, tagging with RFID and centralised databases [3] were used. These systems are able to track the movement of the product, but they suffer from two structural limitations: the database is managed by one party who can change or remove information, and there is no universal and verifiable "truth" that the partners agree on.

Blockchain technology provides an alternative structure. A Blockchain is a list of blocks that is replicated across the participants, and each block on the list is linked to each other by cryptographic hashes, and new blocks are added to the list by following a consensus protocol, so that the list/chain cannot be altered without consensus between the participants in the network [4]. The idea of permissioned platforms like Hyperledger Fabric, which introduce identity–based membership, private channels, and an execute–order–validate transaction model that better supports business consortia compared to

open, energy–consuming cryptocurrency designs, is also a consideration. Reviews of the application of blockchain to agriculture and food describe a rapidly growing set of projects and initiatives [6], synthesise applications, challenges and research directions for value-chain management [7], and discuss the challenges for traceability analysis [8]. There are several practical implementations integrating blockchain with IoT sensing [9] and with standards like EPCIS for food safety traceability [10].

In the realm of sustainability, there's no room for complacency. Reports of carbon footprint, water use, pesticide stewardship, fair labour practices, and cold chain integrity are in growing use on procurement and labelling but are for the most part self-reported and expensive to audit. Studies have explored the potential of blockchain to become an enabler, while also assessing the barriers to its adoption [11,12]; however, the quantitative impact of better verification on the sustainability results has not been extensively studied. Meanwhile, blockchain has proven scalability problems, storage needs, privacy and latency issues [13] which means that a blockchain that stores raw sensor streams on-chain, or uses a consensus algorithm that isn't scalable will fail in practice. Furthermore, a ledger does not ensure data is accurate when it is added, it only ensures data cannot be changed after they are added, called the oracle problem.

This paper aims to fill these gaps. Its contributions are: (1) a layered, permissioned-blockchain framework for agricultural traceability and sustainability verification that keeps bulk data off-chain and anchors them on-chain by hash and Merkle root; (2) smart-contract logic for custody, cold-chain compliance and a composite sustainability score computed from ledger-verified indicators; (3) a QR-based consumer verification procedure that requires no trust in the retailer; (4) an evaluation combining measured cryptographic operations, tamper-detection experiments, and a discrete-event simulation of throughput, latency and scalability; and (5) a quantification of how ledger verification changes sustainability scores relative to self-reporting.

II. BACKGROUND AND POSITIONING

Table 1 shows that the existing approaches for traceability can be compared based on the integrity guarantees they offer and cost placed on it. Paper records and centralised databases are inexpensive and easy to use, but are vulnerable to change by the record holder, and can only be verified by sampling audits. RFID- or barcode-based systems provide better identification and automation of data capture, but still rely on centralised data control. While public blockchains do not require a trusted operator they suffer from low throughput, gas fees that fluctuate with use, public access to the blockchain data, and the energy intensive consensus. Public blockchains do not need a trusted operator, but they are not well suited to supply-chain data for commercial applications due to low throughput, variable transaction fees, data is public, and the blockchain's energy intensive consensus [13]. Permissioned blockchains are able to provide tamper evidence and shared control, as well as known participants, confidentiality and efficient consensus, which is why the proposed framework is based on a permissioned platform with Byzantine-fault-tolerant ordering [5, 15].

Table 1. Qualitative Comparison of Traceability Architectures

Property	Paper / central database	RFID + central database	Public blockchain	Proposed permissioned framework
Tamper evidence	None	None	Strong	Strong
Single point of control	Yes	Yes	No	No (consortium)
Throughput	High	High	Low	Moderate to high
Data confidentiality	Operator-controlled	Operator-controlled	Public by default	Channel-based
Cost per record	Low	Low	Variable, fee-based	Low, shared
Verification by consumer	Not possible	Limited	Possible	Possible (QR + proof)

The framework is tailored to the unique challenges of agriculture: there are many small producers with little connectivity and digital skills; products are perishable and require temperature and time to be considered safe; the aggregation and processing of products result in a merging and splitting of product lots; and the sustainability claims rely on measurements outside the ledger.

III. PROPOSED FRAMEWORK

Requirements and Threat Model

The framework must satisfy five requirements: integrity (any alteration of a record is detectable and attributable), shared control (no participant can unilaterally edit history), scalability (the ledger must remain manageable for millions of lots and sensor readings), timeliness (compliance violations must be recorded and signalled within seconds), and verifiability for outsiders (regulators and consumers can check a claim without trusting the party that makes it). The threat model assumes that individual participants may be dishonest, for example by editing past records, misreporting indicators or forging identities, that up to f of n ordering nodes may behave arbitrarily with $n \geq 3f + 1$, and that network messages may

be delayed but not forged because all parties are authenticated by certificates. The framework does not assume that sensors and humans always report truthfully, which is addressed in Section 3.10.

Layered Architecture

The architecture (**Fig. 1**) has four layers. The data-capture layer collects temperature, humidity and location from IoT loggers, identifies lots with RFID or QR labels, and records farm operations, laboratory certificates and weighbridge data. The data-integrity layer stores bulk data off-chain, in a database or a content-addressed store such as IPFS [16], and computes SHA-256 digests and Merkle roots. The blockchain layer maintains the permissioned ledger and executes smart contracts under a membership service that issues identities to farms, aggregators, processors, distributors, retailers and regulators. The application layer exposes producer apps with low-bandwidth operation, dashboards, a regulator audit interface and the consumer portal.

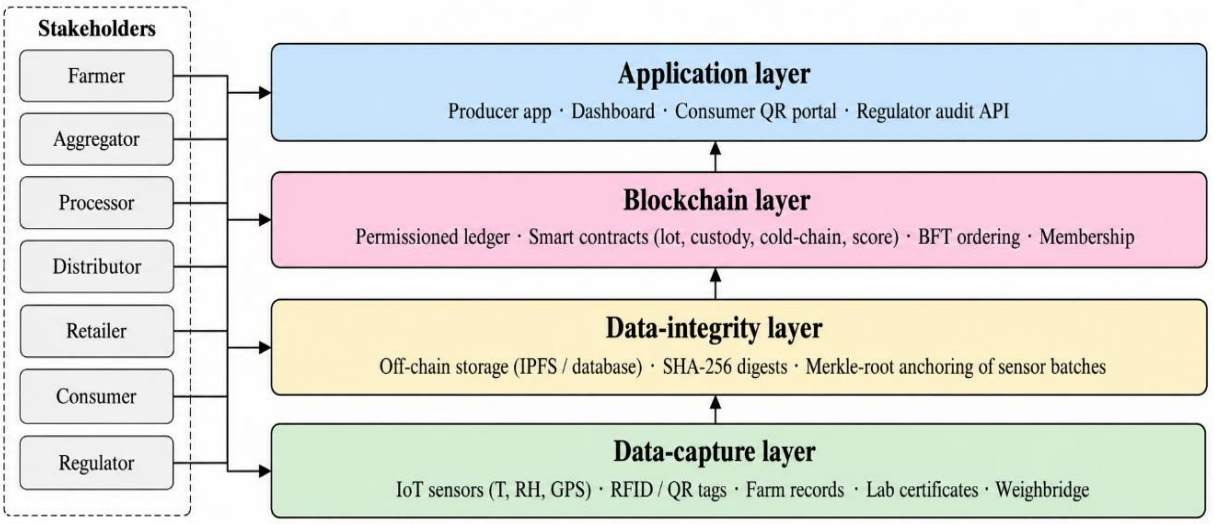


Fig 1. Layered Architecture of The Proposed Blockchain-Enabled Traceability Framework.

Data Model and Hash Chaining

Each supply-chain event, such as harvest, packing, storage, shipment or receipt, is recorded as a transaction that names the lot, the acting organisation, the event type, quantity, timestamp, a pointer to off-chain evidence and the Merkle root of any associated sensor batch. Records belonging to a lot form a hash chain (**Fig. 2**): each record commits to the digest of its predecessor, so the digest of record i is

$$h_i = H(h_{i-1} \| m_i) \tag{1}$$

where H is SHA-256, $\|$ denotes concatenation, m_i is the canonical encoding of the record and h_0 is a fixed genesis value. Altering any earlier record changes its digest and therefore every later digest, which makes tampering detectable by recomputation. At the network level, records are additionally grouped into blocks that are chained in the same way, so that the order of all events is agreed by the consortium. Lot splitting and merging, which are frequent during aggregation and processing, are represented by transactions that reference several parent lots and create several child lots, so that the provenance forms a directed acyclic graph that can be traversed in both directions for trace-back and trace-forward.

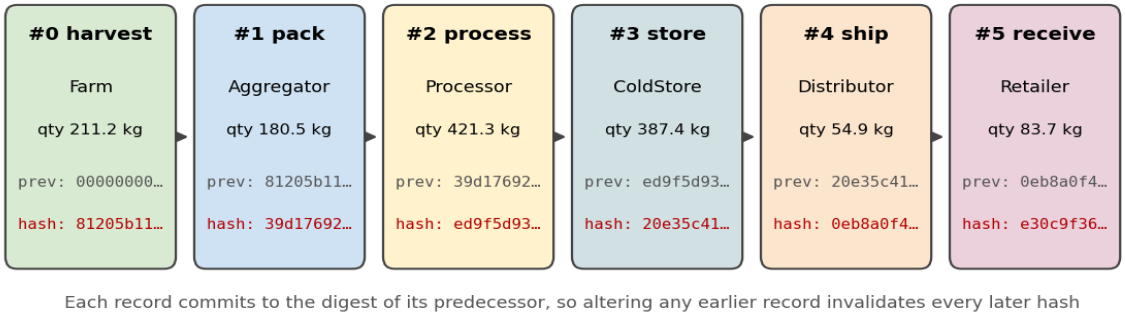


Fig 2. Hash-Chained Custody Records of One Lot (Real SHA-256 Values, Truncated). Each Record Stores the Digest of Its Predecessor.

Off-Chain Storage and Merkle Anchoring

A cold-chain logger produces hundreds of readings per shipment, and storing them on-chain would waste ledger space and expose commercially sensitive data. The framework therefore stores readings off-chain and anchors them by a Merkle tree [14]. The readings of a batch are hashed as leaves, and each internal node is the hash of its two children,

$$M_{l,j} = H\left(M_{l-1,2j} \| M_{l-1,2j+1}\right) \tag{2}$$

and only the root is written to the ledger (**Fig. 3).** Later, any single reading can be proved to belong to the anchored batch by revealing the reading and the sibling hashes along its path to the root, a proof of $\lceil \log_2 n \rceil$ hashes:

$$|\pi| = \lceil log_2 n \rceil, \quad size = 32 \lceil log_2 n \rceil \; bytes \tag{3}$$

This lets an auditor or a consumer verify a single reading without downloading the whole stream, and lets a producer prove compliance for specific time windows while keeping the rest private.

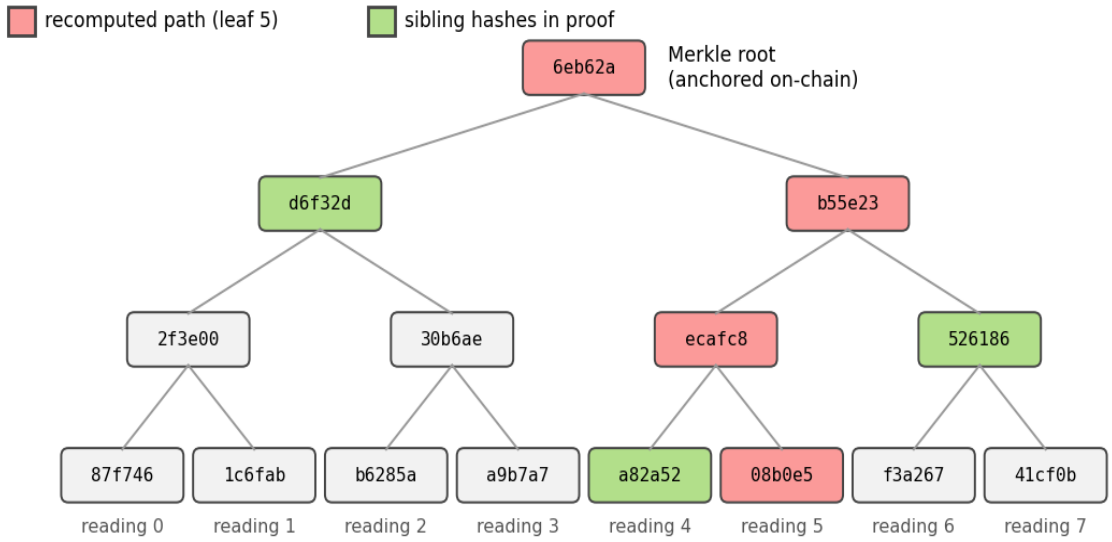


Fig 3. Merkle Tree Over Eight Sensor Readings (Real Hash Values, Truncated). Pink Nodes are Recomputed and Green Nodes are the Sibling Hashes Supplied in The Proof for Reading 5.

Smart Contracts

The business rules are written in smart contracts (chaincode) that are installed on a channel that is used by the consortium. The main functions are summarised in **Table 2.** Access control is based on the caller's identity: for instance, only the current owner can transfer a lot and only an approved lab can add a certificate. Contract functions are deterministic and will check that all arguments are valid as every endorsing peer should return the same result.

Table 2. Main Smart-Contract Functions

Function	Caller	Effect on ledger
registerLot	Farm	Creates the lot with crop, origin, harvest date and evidence hash
transferCustody	Current custodian	Moves custody to the next actor; appends a hash-linked record
mergeLots / splitLot	Aggregator, processor	Links parent and child lots; conserves total quantity
anchorSensorBatch	Logger gateway	Stores the Merkle root of a sensor batch and checks the cold-chain rule
attachCertificate	Accredited laboratory	Binds a certificate digest (residue, organic, quality) to a lot
computeScore	Any member (read-only)	Calculates the sustainability score from verified indicators
verifyLot	Anyone with read access	Returns the event trail and anchored roots for external recomputation

Transaction Flow and Consensus

Transactions follow the execute–order–validate model of permissioned platforms [5] (**Fig. 4**). A client sends a proposal to the endorsing peers of the organisations named in the endorsement policy, which simulate the chaincode and return signed read/write sets. The client submits the endorsed transaction to the ordering service, which establishes a total order, cuts blocks when a batch of B transactions is full or a timeout T_o expires, and delivers them to all committing peers. Each peer validates the endorsement policy and checks for read–write conflicts before appending the block to its ledger. For ordering, a Byzantine-fault-tolerant protocol in the style of PBFT [15] tolerates f faulty nodes out of $n \geq 3f + 1$ and exchanges, per block, the pre-prepare, prepare and commit messages, whose number grows quadratically with the number of nodes:

$$M(n) = (n-1) + (n-1)^2 + n(n-1) \approx 2n^2 \tag{4}$$

The end-to-end latency of a transaction is the sum of endorsement, batching, ordering and validation delays,

$$L = L_{end} + min\left(\frac{B}{\lambda}, T_o\right) + L_{ord}(n) + a + bB \tag{5}$$

where λ is the arrival rate, $L_{ord}(n)$ the ordering time per block and $a + bB$ the validation and commit time of a block of B transactions (fixed cost a plus a per-transaction cost b). Because ordering and validation are pipelined, the sustainable throughput is limited by the slower stage:

$$\lambda_{max} = min\left(\frac{B}{L_{ord}(n)}, \frac{B}{a+bB}\right) \tag{6}$$

Equation (6) exposes the central trade-off: large batches raise throughput by amortising the fixed cost a but lengthen latency at low load because of the batching delay in Eq. (5), and large ordering clusters improve fault tolerance but reduce throughput through $L_{ord}(n)$.

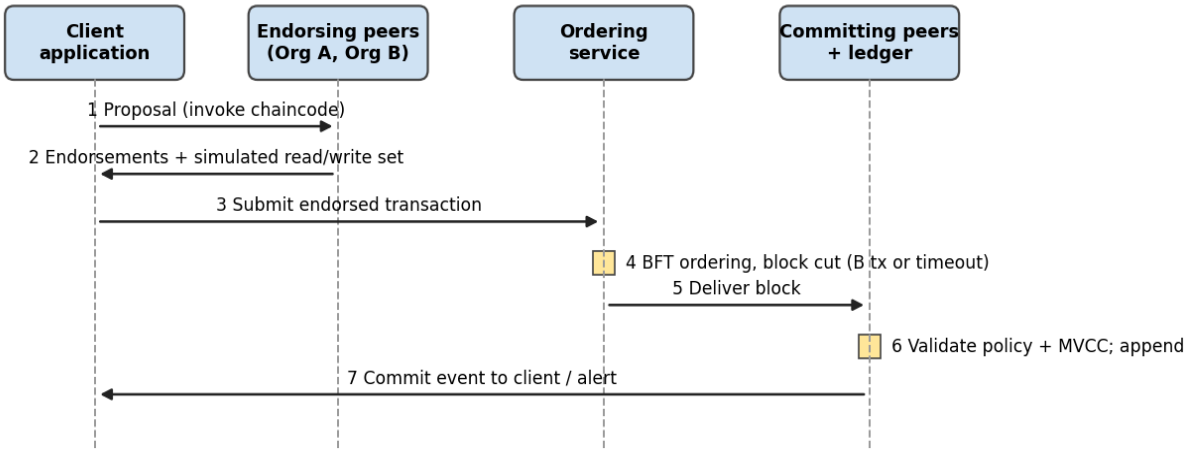


Fig 4. Execute–Order–Validate Transaction Flow Used by The Framework.

Cold-Chain Compliance Contract

Temperature abuse is the largest controllable cause of quality loss and food-safety risk in perishable produce, so compliance rules are executed on-chain when a sensor batch is anchored. To avoid false alarms from brief door openings while still catching sustained abuse, the contract accumulates the thermal exposure above the allowed maximum T_{max} over the sampling interval Δt,

$$E(t_k) = \sum_{j=1}^{k} max\left(0, T_j - T_{max}\right)\Delta t \tag{7}$$

and emits an alert event when E exceeds a threshold E_{th}. The same record then lowers the lot's cold-chain compliance indicator, which feeds into the sustainability score. Sensors sign their readings with device keys so that the gateway cannot silently modify them.

Sustainability Scoring and Carbon Footprint

Sustainability claims are converted into a composite score computed by a contract from indicators whose underlying data are anchored on the ledger. Five indicators are used in this study: carbon intensity (kg CO_2e per kg), water use (L per kg),

a pesticide-use index, cold-chain compliance (% of time in range) and food miles. Each indicator x_k is normalised between fixed bounds and oriented so that higher is better, and the score is the weighted sum.

$$S = 100 \sum_{k=1}^{K} w_k z_k, \quad z_k = \frac{x_k - x_{k,min}}{x_{k,max} - x_{k,min}}, \quad \sum_k w_k = 1 \tag{8}$$

with weights 0.25 (carbon), 0.20 (water), 0.20 (pesticide), 0.20 (cold chain) and 0.15 (food miles); for indicators where lower values are better, z_k is replaced by $1 - z_k$. Lots are graded A ($S \geq 70$), B ($55 \leq S < 70$) or C ($S < 55$). The carbon footprint of a lot accumulates along the chain from stage-specific energy use and emission factors and from freight, where m is the mass in tonnes, d_j the distance of leg j and EF_j the emission factor of the transport mode:

$$CF = \sum_s E_s \cdot EF_s + m \sum_j d_j \cdot EF_j \tag{9}$$

Weights and thresholds are governance parameters agreed by the consortium and stored on the ledger, so that changes to the scoring rules are themselves transparent.

Consumer and Regulator Verification

Each retail unit carries a QR label that encodes the lot identifier and a prefix of the lot's anchored digest **(Fig. 5a)**. A consumer who scans it reaches a verification page (**Fig. 5b**) that retrieves the lot's events from any ledger node, recomputes the hash chain, checks Merkle proofs of the sensor batches against the anchored roots and displays the custody trail, cold-chain compliance and sustainability grade only if every check succeeds (**Fig. 5c**). Because the checks are recomputed from the ledger rather than reported by the retailer, the consumer does not need to trust the page provider; a regulator can run the same procedure over a whole lot family for recalls.

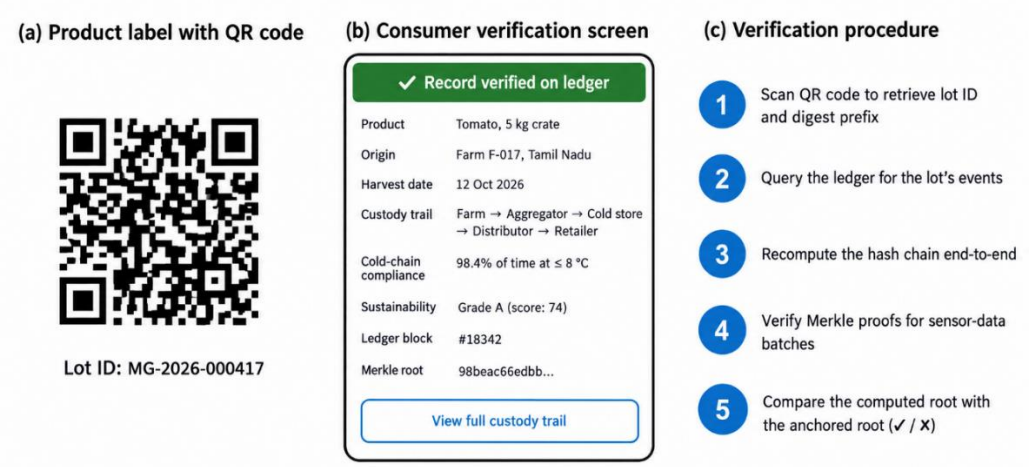


Fig 5. Consumer Verification: (a) Real, Decodable QR Label for A Demonstration Lot, (B) Illustrative Verification Screen and (C) Verification Procedure.

Security Analysis and The Oracle Problem

If the history gets altered, this is detected through hash recomputation and if the history rewrite consistently, it would need the cooperation of more than f ordering nodes. Membership is based on certificates and certificate-based membership prevents identity forgery. Replay is prevented by transaction identifiers and nonces. Channel separation and private data collection ensure confidentiality of prices and customer lists, except for digests, which are available to non-parties. The weakest link is the line between the physical world and the ledger: a dishonest farmer can credit a fake pesticide application, or a logger can be put in a cooler for the test period. This is addressed in the framework, where the readings by a sensor are signed with attestation of the device, certificates issued by third-party laboratories are bound to lots and physical audits of the sensors are conducted at random, and their results are recorded on the ledger, while participants who have data that is not confirmed suffer reputational and contractual penalties. These measures are designed to minimize, but not remove, the possibility of an error on input.

IV. EVALUATION METHODOLOGY

Implementation and Experimental Design

Four experiments were conducted. In the first, the data-integrity layer (SHA-256 hashing, hash chaining, Merkle trees, proofs) was implemented in Python 3 with the standard hashlib library and run on a general-purpose processor; times were measured with a high-resolution timer and are representative of an application server rather than of a constrained gateway.

The second experiment tested tamper detection on 200 lots of 12 chained records each (2,400 records), in which k randomly chosen records (k = 1 to 50) were modified in 200 independent trials per k, and detection and localisation of the first altered record were recorded. The probability that a sampling audit of s records out of N detects at least one of k altered records is hypergeometric, with $C(a, b)$ the number of combinations of b items from a,

$$P_{detect} = 1 - \frac{C(N-k,s)}{C(N,s)} \approx 1 - \left(1 - \frac{s}{N}\right)^k \tag{10}$$

and was estimated by Monte Carlo simulation with 4000 trials for audits of 1, 5 and 10 % of the records. The third experiment compared on-chain storage for two designs. Storing raw readings at r = 864 readings per 72-h shipment, each of b_r = 96 bytes, requires C_{raw} per lot, whereas anchoring needs only a 32-byte digest plus b_m = 256 bytes of metadata:

$$C_{raw} = N(rb_r + b_m), \quad C_{anchor} = N(b_h + b_m) \tag{11}$$

The fourth experiment evaluated the ordering and commit pipeline by discrete-event simulation of Poisson transaction arrivals over 60 s. Endorsement times were log-normal with a median of 25 ms; blocks were cut at B transactions or T_o = 1 s; ordering time followed 10 ms + 0.05 ms × $M(n)$; and validation time was 15 ms + 1.6 ms per transaction. These service times are assumptions of the same order as those reported in benchmarking studies of private blockchains, not measurements of a particular deployment, and the results should be read as showing trends and trade-offs rather than absolute capacities. A synthetic 72-h cold-chain shipment (5-min readings, setpoint 4 °C, five door-opening transients and one refrigeration failure) was used to exercise the compliance contract with T_{max} = 8 °C and E_{th} = 60 °C·min. A synthetic population of 30 farms and 120 lots was used to compare scores computed from ledger-verified indicators with scores computed from self-reported indicators, in which 30 % of the farms understated carbon, water and pesticide indicators by 10–35 % and overstated cold-chain compliance by 3–10 points.

Performance Metrics

The following metrics were used: construction, proof-generation and verification time and proof size for the Merkle tree; detection probability, localisation accuracy and false-positive rate for tampering; ledger size for storage; committed throughput, mean and 95th-percentile latency for the pipeline; alert delay for the cold-chain contract; and the difference in score and grade between verified and self-reported data for sustainability.

V. RESULTS AND DISCUSSION

Cost of Hashing and Merkle Verification

The results in **Fig. 6** indicate that the construction of a merkle tree was linear with the number of records, ranging from 0.17 ms for 100 records to 181 ms for 100,000 records (≈ 1.8 µs per record). Checking a proof is much less expensive than constructing the tree: For 100 records, it took 11.3 µs and for 100,000 records, it took 17.5 µs, growing with the log of the number of records, as expected by Eq. (3), with the size of the proof increasing only from 224 to 544 bytes. A proof for a single reading in a batch of 100,000 fits in a single network packet and a proof is verified in microseconds, allowing for verification to occur on a phone. Only 1% of the 500 negative tests were successful when the proof was altered by changing one item in a leaf. The recomputation of the hash chains of all 200 lots (2,400 records) was 15 ms. The hash constraint is thus not a practical concern for the framework; consensus and storage are the practical constraints discussed below, and cryptographic costs are negligible.

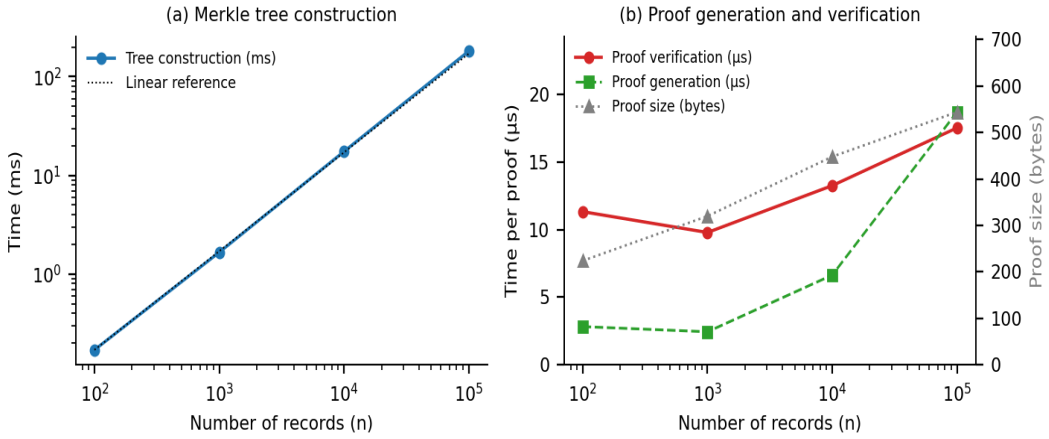


Fig 6. Measured Cost of Merkle Operations: (A) Tree Construction Time and (B) Proof Generation and Verification Time and Proof Size, Versus Number of Records.

Tamper Detection and Storage

All 1200 trials were performed for all numbers (from 1 to 50) of altered records and in all cases the first altered record of the lot was correctly detected and localised (detection and localisation rate 100 %); no alarm was raised for the unmodified chains (false positive rate 0 %). In contrast, few alterations (5 altered records) were found for the 1 % sampling audit (4.5 % probability of detecting the alterations), the 5 % sampling audit (22.9 % probability of detecting the alterations) and the 10 % sampling audit (41.6 % probability of detecting the alterations) (**Fig. 7a**), as predicted in Eq. (10). When audits were conducted at 1 % and fifty records were altered, the audits discovered at least one only 39 % of the time. What this will render is intuitive but important: audits provide probabilistic assurance at a price that increases with the level of coverage, hash chains provide full and automatic detection at a negligible price, and they can be performed on a continuous basis by all participants. The limitation of this guarantee is that it relates to integrity after recording, not to integrity at source; if a false entry occurs at source, it is only recorded in the mitigations provided in Section 3.10.

Fig. 7b shows that the storage was an order of magnitude different between the two designs. 83.2 kB per 72-h shipment is needed for each reading while anchoring only requires 288 bytes, 289x smaller. The raw design would consume some 83.2 GB per lot per cycle, versus 288 MB for anchoring, giving a decisive advantage to commodity machines in a cooperative or small processor to run a node. Not very much integrity is traded for the off-chain data as they are still verifiable with Merkle proofs.

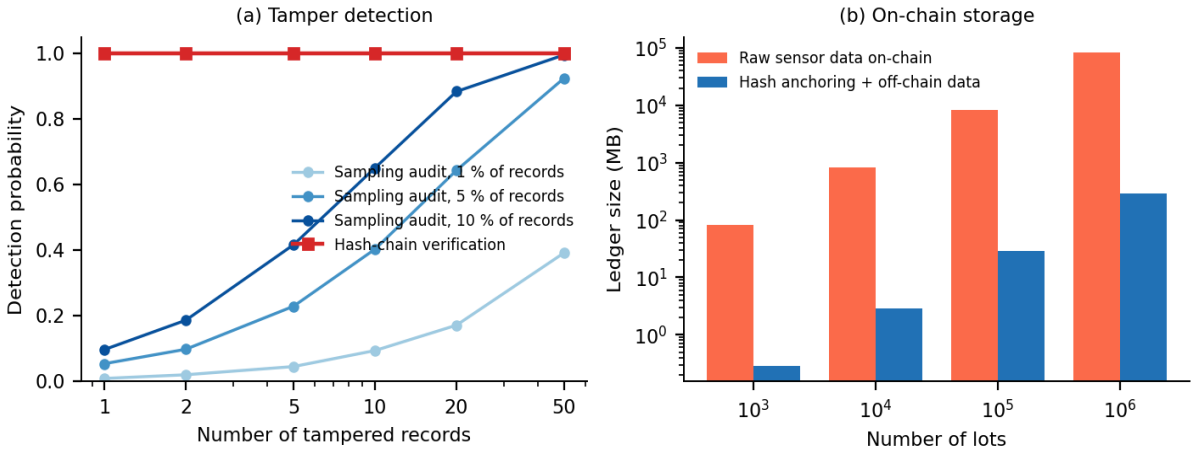


Fig 7. (a) Probability of Detecting Tampered Records by Hash-Chain Verification and by Sampling Audits (200 Trials Per Point for Chains, 4000 For Audits) and (b) Ledger Size for Raw On-Chain Storage Versus Hash Anchoring.

Throughput and Latency

Fig. 8 and **Table 3** illustrate the throughput and latency of the simulated four node ordering cluster for four batch sizes. Throughput followed the offered load until the pipeline saturated, at 323 tx/s for B = 10, 526 tx/s for B = 50, 572 tx/s for B = 100 and 595 tx/s for B = 250, in line with Eq. (6). The per-block cost increases as the number of transactions in the queue increases, such that in the case of 400 tx/s, a B of 10 would have resulted in a 95th percentile latency of 11.2 s. However, lightly loaded networks are penalized when sending the same amount of data in many requests: at 20 tx/s, the 95th-percentile latency was 1.1 s with B = 250 compared to 637 ms with B = 10, dominated by the 1-s block timeout. Intermediate batches showed the best combination, where batch size B = 50, with 200 tx/s, 95th percentile latency was 394ms, and the throughput was 526 tx/s. A consortium running a million lots per year with a dozen transactions per lot produces only a few tens of transactions per second (several hundred at peak seasons) so their capacity is sufficient for the numbers of transactions a regional supply chain might require, and sensor data are relatively high volume, but are batched and anchored and not sent individually. Therefore, the block timeout and batch size are a tuning choice, rather than a limit.

Table 3. Simulated Throughput and 95th-Percentile Latency for Different Batch Sizes (Four Ordering Nodes)

Batch size	Max throughput (tx/s)	p95 at 50 tx/s	p95 at 200 tx/s	p95 at 500 tx/s	Mean latency at 100 tx/s
10	323	300 ms	136 ms	26.4 s	119 ms
50	526	1.0 s	394 ms	260 ms	376 ms
100	572	1.1 s	708 ms	415 ms	688 ms
250	595	1.1 s	1.3 s	937 ms	731 ms

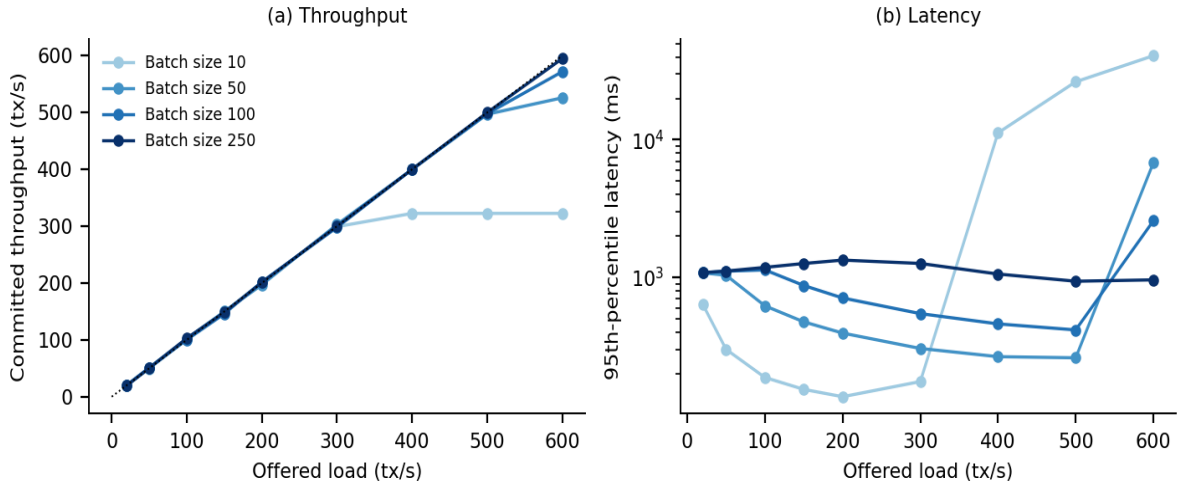


Fig 8. Simulated Pipeline Performance Versus Offered Load For Four Batch Sizes: (A) Committed Throughput and (B) 95th-Percentile Latency (Logarithmic Scale).

Scalability With the Number of Ordering Nodes

It is predicted that for 4 nodes, 24 messages are needed per block to be able to tolerate more Byzantine nodes, and distribute trust over more organisations, while 32 nodes require 1,984 messages per block and 64 nodes require 8,064 messages per block (see Eq. (4)). With ordering being the slower stage, the saturation throughput of the simulation results (**Fig. 9**) were similar up to a node count of 32, but reduced to 408 tx/s at 49 nodes and 242 tx/s at 64 nodes. The latency at a constant load of 300 tx/s was found to increase from 538 ms to 640 ms for the 4 nodes to the 32 nodes cluster, and the 64 nodes cluster failed to keep up with the load (13.2 s). These results support a small (4-16) well-chosen ordering node set that can still tolerate 1-5 faulty nodes, over a large open committee; and they reinforce the well-known wisdom that a supply-chain consortium should employ crash/fault/Byzantine ordering, but not use 16-17 other nodes as orders.

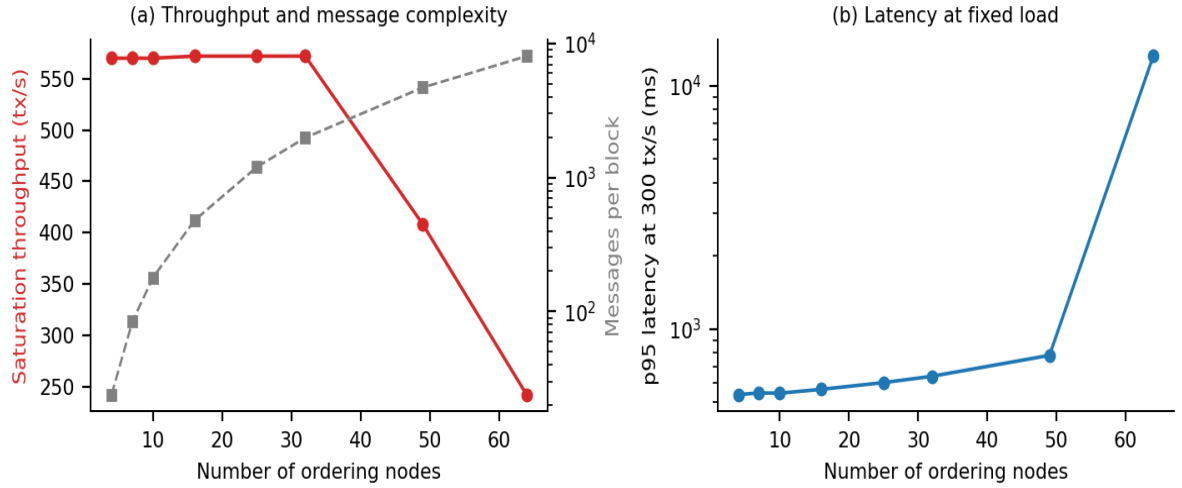


Fig 9. Scalability with The Number of Ordering Nodes at Batch Size 100: (A) Saturation Throughput and BFT Messages Per Block and (B) 95th-Percentile Latency At 300 tx/s.

Real-Time Cold-Chain Monitoring

Fig. 10 illustrates the synthetic 72 h shipment comprising 864 sensor reading. At normal operating conditions, the average temperature was 4.1 °C without any breaches of the 8 °C limit and five doors opened that caused temporary increases of around 2 °C that did not exceed the limit, but did not add exposure either, and so no alert was triggered. The temperature increased to 13.1 °C due to refrigeration failure, which started at 40 h. The temperature reached the limit at 40.6 h and the Cumulative Exposure reached the limit at 60 °C·min at 50 min later at 41.4 h, triggering an alert on the contract and marking the Lot as non-compliant. This total exposure was 789 °C·min for 245 min above the range and 94.3 % of the readings were within the range. The 50 min alert delay time is a by design feature of the cumulative rule that grants immunity against nuisance alerts and is a compromise for the average delay of 385 ms from sensor batch to committed ledger entry, with a 95th percentile of 633 ms at 100 tx/s in the simulator. The advantage of the alert to the driver or to the

destination is that they can act on it, for example by rerouting, inspecting and diverting the product, and the product is still salvageable, and the compliance record is not subsequently altered to cover up the event. The anchoring one Merkle root every 30 minutes translated to 144 roots being written, instead of 864 readings.

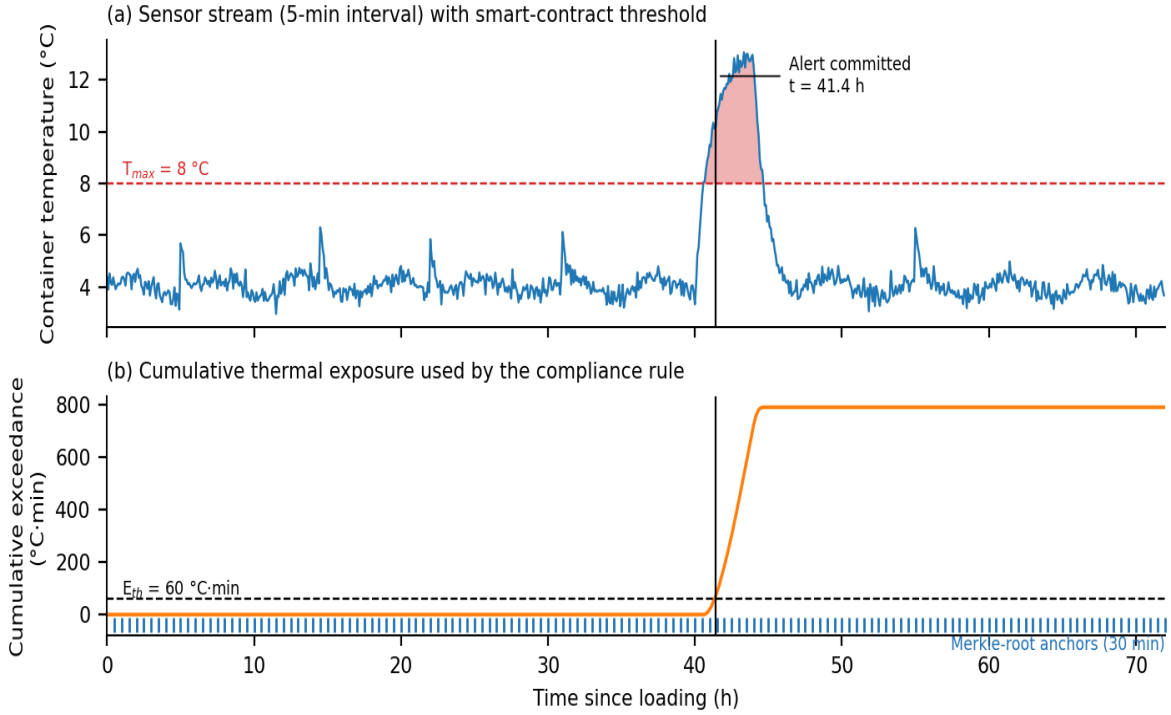


Fig 10. Synthetic 72-H Cold-Chain Shipment: (A) Temperature Trace with the 8 °C Limit and The Alert Time and (B) Cumulative Thermal Exposure Used by The Compliance Contract; Blue Ticks Mark Merkle-Root Anchors.

Verified Versus Self-Reported Sustainability

The central test of verification is whether it changes outcomes. Of the 30 farms, 12 (48 lots) misreported indicators in the simulated population. Across all lots, self-reported scores averaged 67.8 against 64.0 for ledger-verified data, an average inflation of 3.8 points, but the inflation was concentrated: lots from misreporting farms were overstated by 9.5 points on average (**Fig. 11a**). The consequence for grading is larger than the average suggests (**Table 4**): the number of lots graded A rose from 23 under verified data to 53 under self-reporting, and 37 of 120 lots (31 %) received a better grade than the evidence supports. The rank correlation between the two score sets was only 0.80 (Spearman), so self-reporting does not even preserve the ordering of producers. For honest farms this is unfair, since they compete against inflated claims, and for buyers it is costly, since green premiums are paid on unverified claims. Ledger verification removes the incentive to misreport because the claim is checked against anchored evidence, and it rewards the honest producer with a credible grade.

Table 4. Sustainability Grades of The Lots Under Ledger-Verified and Self-Reported Data

Grade	Ledger-verified	Self-reported	Change
A ($S \geq 70$)	23	53	+30
B (55–70)	81	58	-23
C ($S < 55$)	16	9	-7

Fig. 11b shows the carbon footprint accumulated along three illustrative chains computed with Eq. (9). Production dominated all of them (85 % of the local chain, 76 % of the regional chain and 66 % of the export chain), while the share of transport rose from 2 % for the local chain to 19 % for the export chain, which totalled 0.83 kg CO_2e per kg against 0.64 for the local chain. Two lessons follow. First, a food-miles indicator alone is a poor proxy for footprint, since most emissions arise on the farm, which is why the score combines several indicators. Second, meaningful footprint figures need verified input data from every stage, which is exactly what a shared ledger provides: the on-farm figure depends on fertiliser and energy records, and the transport figure on routes and modes recorded by the carriers.

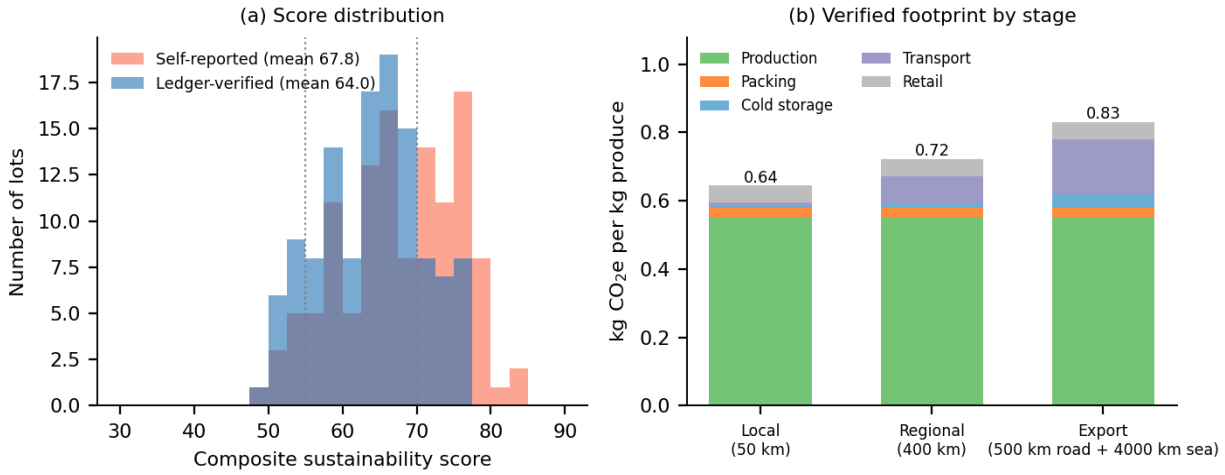


Fig 11. (a) Distribution of Composite Sustainability Scores from Self-Reported and Ledger-Verified Indicators (Dotted Lines: Grade Boundaries) and (B) Verified Carbon Footprint by Stage for Three Illustrative Supply Chains.

There are some limitations to the study. They are ordering and commit results based on the assumption of service times and one simulated topology – these results should be verified on a deployed platform with real endorsement policies, network latency, chaincode complexity. The score inflation is illustrative because the data for the cold chain and sustainability is synthetic, and the misreporting model is an assumption. The framework is integrity proof, not truthful proof and relies on the quality of sensors, attestations at the point of entry. There is no assessment on governance, legal liability and compatibility with national regulations. Future research will include using the framework with a producer cooperative and a cold-chain operator, introducing signed IoT devices and laboratory attestations, assessing the impact of score verification on behavior, and assessing the impact of score verification on price premium.

VI. CONCLUSION

Bulk data may be kept off-chain, anchored by hash and Merkle root, and executed in smart contracts, allowing claims to be made and verified in agriculture's supply-chain. The experiments demonstrated that the cryptographic operations were inexpensive, verifying the Merkle proofs took about 18 µs per 100,000 records, the hash-chain verified the entire transaction while being proven invalid for the smallest of changes, and the anchoring mechanism reduced ledger storage by a factor of 289. The simulated ordering pipeline achieved hundreds of transactions per second, suitable for regional ordering pipelines, and demonstrated that small groups of ordering nodes are better than large groups. The cold-chain contract experienced an equipment failure that could not be edited later and verified indicators showed that 31 % of lots had inflated sustainability grades. The main remaining risk is the integrity of data at the point of entry, which must be addressed by device attestation, third-party certification and audit. The next steps are a field pilot with real producers, real sensors and a deployed ledger, and an assessment of governance, cost and the effect of verified sustainability data on market outcomes.

CRediT Author Statement

The author reviewed the results and approved the final version of the manuscript.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding was received for conducting this research.

Competing Interests

There are no competing interests.

References

[1]. F. Anastasiadis, I. Apostolidou, and N. Tsolakis, "Challenges and opportunities of supply chain traceability: insights from emergent agri-food sector," Supply Chain Management: An International Journal, vol. 30, no. 1, pp. 106–126, Dec. 2024, doi: 10.1108/scm-05-2024-0322.
[2]. S. Pang et al., "A survey on evaluation of blockchain-based agricultural traceability," Computers and Electronics in Agriculture, vol. 227, p. 109548, Dec. 2024, doi: 10.1016/j.compag.2024.109548.
[3]. T. Bosona and G. Gebresenbet, "The Role of Blockchain Technology in Promoting Traceability Systems in Agri-Food Production and Supply Chains," Sensors, vol. 23, no. 11, p. 5342, Jun. 2023, doi: 10.3390/s23115342.

[4]. M. Reitano, M. S. Segovia, and R. M. Nayga, "A systematic review on the impact of Artificial Intelligence in the agri-food supply chain," Food Policy, vol. 137, p. 102983, Nov. 2025, doi: 10.1016/j.foodpol.2025.102983.
[5]. P. Sharma, R. Jindal, and M. D. Borah, "Blockchain-based distributed application for multimedia system using Hyperledger Fabric," Multimedia Tools and Applications, vol. 83, no. 1, pp. 2473–2499, May 2023, doi: 10.1007/s11042-023-15690-6.
[6]. A. Kamilaris, A. Fonts, and F. X. Prenafeta-Boldú, "The rise of blockchain technology in agriculture and food supply chains," Trends in Food Science & Technology, vol. 91, pp. 640–652, Sep. 2019, doi: 10.1016/j.tifs.2019.07.034.
[7]. E. Yontar, "Critical success factor analysis of blockchain technology in agri-food supply chain management: A circular economy perspective," Journal of Environmental Management, vol. 330, p. 117173, Mar. 2023, doi: 10.1016/j.jenvman.2022.117173.
[8]. A. Singh, A. Gutub, A. Nayyar, and M. K. Khan, "Redefining food safety traceability system through blockchain: findings, challenges and open issues," Multimedia Tools and Applications, vol. 82, no. 14, pp. 21243–21277, Oct. 2022, doi: 10.1007/s11042-022-14006-4.
[9]. M. Aggarwal, P. Rani, P. Rani, and P. Sharma, "Revolutionizing agri-food supply chain management with blockchain-based traceability and navigation integration," Cluster Computing, vol. 27, no. 9, pp. 12919–12942, Jun. 2024, doi: 10.1007/s10586-024-04609-x.
[10]. S. Chatzinikolaou, G. Vassiliou, M. Gianniou, M. Vassalos, and N. Papadakis, "Integrating Veterinary Public Health Data into EPCIS-Based Digital Traceability for Dairy Supply Chains," Foods, vol. 15, no. 9, p. 1566, May 2026, doi: 10.3390/foods15091566.
[11]. B. Neto et al., "Supply chain traceability by recording EPCIS data on Hyperledger Fabric," Procedia Computer Science, vol. 278, pp. 709–716, 2026, doi: 10.1016/j.procs.2026.03.042.
[12]. S. Kravenkit and C. So-In, "Blockchain-Based Traceability System for Product Recall," IEEE Access, vol. 10, pp. 95132–95150, 2022, doi: 10.1109/access.2022.3204750.
[13]. L. Preto, S. Eisa, O. Remédios, D. R. Matos, and M. L. Pardal, "Dependable Food Traceability Data Through Blockchain and Database Integration," 2025 20th European Dependable Computing Conference Companion Proceedings (EDCC-C), pp. 163–167, Apr. 2025, doi: 10.1109/edcc-c66476.2025.00050.
[14]. N. Etemadi, P. Van Gelder, and F. Strozzi, "An ISM Modeling of Barriers for Blockchain/Distributed Ledger Technology Adoption in Supply Chains towards Cybersecurity," Sustainability, vol. 13, no. 9, p. 4672, Apr. 2021, doi: 10.3390/su13094672.
[15]. H. Chunawala, M. Ihsan, R. V. S. Praveen, N. S. Boob, H. P. Thethi, and A. Badhoutiya, "Agriculture Supply Chain Management System Using Blockchain," Sustainable Agriculture Production Using Blockchain Technology, pp. 15–26, Jan. 2026, doi: 10.1002/9781394248711.ch2.
[16]. A. Qadir, L. Guangming, M. Arshad, Z. Huiqin, and W. Haiyan, "Sustainable Agriculture Development in Rural Regions: The Combination of Green Innovation, Green Supply Chains, and Farmer Education," Sustainable Development, vol. 34, no. 4, pp. 4835–4851, Dec. 2025, doi: 10.1002/sd.70591.

Publisher's note: The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations. The content is solely the responsibility of the authors and does not necessarily reflect the views of the publisher.

ISSN: 3104-4654