Ensuring End-to-End Traceability and Sustainability in the FSC: A Modular Web3 Architecture Integrating Blockchain, IoT, and Machine Learning

Addou Kamal, Mohammed Yassine El Ghoumari
Faculty of Sciences Ben M'sik, Hassan II University, Morocco

***Abstract*—Traceability in food supply chains is crucial for ensuring safety, enabling effective quality control, and maintaining consumer trust. However, traditional paper-based or digital tracking systems often prove too slow and opaque during food safety incidents or investigations into fraud. To address these limitations, this paper presents a modular Web3 architecture that integrates Ethereum blockchain smart contracts, Internet of Things (IoT) sensors, and machine learning (ML) to achieve end-to-end traceability and sustainability in agrifood supply chains, and to support auditable, partially automated decision-making. The system design separates concerns into layers: an on-chain layer of Ethereum smart contracts for tamper-proof event logging and automated business logic, and an off-chain layer for secure storage of detailed sensor data and documents, linked by cryptographic hashes to ensure data provenance. Low-cost IoT sensors are deployed from farm to distributor, continuously monitoring environmental conditions (temperature, humidity, geolocation) and uploading signed, time-stamped summaries to the blockchain. In addition, ML models perform predictive quality control by estimating expected conditions, detecting anomalies, and scoring the conformity of product batches, which enables smart contracts to automatically trigger state transitions (acceptance or dispute escrow of shipments) based on real-time data. Using Ethereum smart contracts, a prototype that manages the life cycle of a specific food product was implemented, and two cases (conformant vs non-conformant shipments) were studied to demonstrate how cryptographically verifiable data and events make decisions transparent and trustworthy.**

***Keywords*—Food supply chain; traceability; blockchain; web3; smart contracts; IoT; machine learning; data integrity**

I. Introduction

Ensuring end-to-end traceability of food products has become a strategic priority in modern supply chains. Over the past two decades, high-profile food safety incidents and fraud scandals have exposed significant weaknesses in existing tracking systems. For instance, an E. coli outbreak in spinach in 2006 took nearly two weeks to trace back to its source, and Europe's 2013 horse-meat scandal saw mislabeled meat adulterate more than 1,000 tons of food [22]. Such delays and opacity in tracking contaminated or fraudulent products can lead to prolonged public health risks, economic losses, and loss of consumer confidence [19].

Nevertheless, implementing robust traceability in the food supply chain (FSC) [11] faces multiple challenges due to the complexity of food supply networks, which involve numerous stages such as production, processing, transport, and distribution, and a wide range of actors operating across different locations. Traditional tracking systems (based on paper records or isolated databases) [8], [9] often cannot cope with this complexity. These legacy systems tend to be unreliable, opaque, and slow to react during crises.

At the same time, recent advances in digital technologies have opened new possibilities for real-time traceability. Sensors and IoT devices allow continuous monitoring of products and conditions throughout their life cycle. In parallel, blockchain technology has emerged as a promising tool to enhance trust and data integrity in multi-party processes. By providing a decentralized ledger [1], [21], blockchain can ensure that records of food origin and handling are immutable and verifiable by all FSC stakeholders.

Related works and commercial projects highlight the potential of blockchain [6] to improve transparency and traceability. For example, Walmart adopted a blockchain-based food traceability system (IBM Food Trust) and reduced the time required to trace a package of mangoes from 7 days to 2.2 seconds [2], demonstrating much faster recall capability. In addition, IoT sensors and radio-frequency identification (RFID) tags are used to automatically collect large amounts of data on product location and condition (temperature, humidity, etc.) at each stage of the FSC.

A number of prototypes and systems have been developed that combine blockchain and IoT for food traceability, covering products from fresh produce and meat to dairy and halal foods. These implementations show that hybrid blockchain–IoT solutions can address many limitations of traditional systems and increase the resilience of agri-food supply chains [5], [10], [4], [3]. They enable recalls by instantly identifying affected lots and enhance consumer trust by allowing end-to-end verification of a product's history.

Despite this progress, significant limitations remain. Most blockchain traceability projects to date use permissioned (private) blockchains or closed consortia, which, while improving accountability among known parties, do not fully leverage the openness and interoperability of public networks. Moreover, current solutions often focus on data logging and traceability but do not aim to automate decision-making or financial transactions based on that data [21]. Integrating smart contracts that execute business logic (such as payment settlements or quality compliance checks) in response to sensor data is a relatively unexplored frontier. In addition, the use of machine learning (ML) for predictive analytics in this context has been limited. Advanced analytics could predict quality issues or detect subtle anomalies, turning raw sensor streams into actionable insights, but few existing food blockchain systems incorporate an AI-

driven decision support layer [25], [20]. Therefore, there is a need for a new architecture that brings together blockchain (for data integrity and decentralized trust), IoT (for real-time data acquisition), and ML (for intelligent automation) into one unified framework.

In this work, a novel modular Web3 architecture is presented, this architecture integrates Ethereum blockchain smart contracts, IoT instrumentation, and ML analytics to deliver end-to-end traceability and automated quality control in food supply chains. The approach is designed to be open and extensible, using the public Ethereum network to ensure transparency across organizations while carefully addressing performance and privacy concerns. The architecture introduces a clear separation of concerns: the blockchain layer handles data provenance and event orchestration (with minimal data on-chain), the off-chain layer handles data storage and processing, and the IoT layer interfaces with the physical world.

The remainder of this paper is organized as follows: 1) related work on blockchain-based food traceability, IoT sensing, and machine learning for food quality prediction is reviewed; 2) the proposed modular Web3 architecture and its main components, including the blockchain layer, IoT layer, hybrid data management, and ML analytics, are described; 3) the prototype implementation and experimental results on conformant and non-conformant lot scenarios are presented; 4) the advantages, limitations, and adoption challenges of the approach are discussed; and 5) the paper is concluded with an outline of directions for future work.

II. Related Work

Research and industry initiatives have increasingly explored the use of blockchain technology within supply chain management. The early implementations often relied on permissioned blockchain platforms and focused on improving traceability within a known set of stakeholders. Feng Tian (2017), for example, proposed a food traceability system that tightly integrated the Hazard Analysis and Critical Control Points (HACCP) food safety methodology with a blockchain-backed database (BigchainDB) and IoT sensing [15]. This system recorded critical control point data (temperature at certain processing steps) on a distributed ledger, leveraging BigchainDB's high transaction throughput while preserving data integrity. Tian's work demonstrated that even a private or hybrid blockchain could enhance trust in food safety management by providing an immutable log of monitoring data.

Another notable effort is the MultiChain WSC (Wine Supply Chain) project [17], which implemented end-to-end wine traceability using a private blockchain (the MultiChain framework). In this system, five main actors (grape grower, winemaker, wholesaler, bottler, distributor) participated, but only a subset were authorized as blockchain "miners" to validate transactions. This selective validation mechanism illustrates a typical consortium blockchain approach, where read/write permissions are restricted according to governance rules. MultiChain WSC used structured data streams on-chain to log each batch of wine and its transformations, achieving fine-grained traceability for specific products.

The commercial sector has also produced blockchain-based traceability platforms. One pioneering example is AgriDigital [12], an Australian agri-tech startup founded in 2015 that built a comprehensive platform for managing grain supply chain transactions using blockchain and cloud infrastructure. AgriDigital enables farmers, grain elevators, brokers, and end-buyers to trade and settle agricultural commodities with transparent, verifiable tracking of each delivery. In December 2016, AgriDigital conducted the world's first-ever sale of wheat via blockchain, executing a full end-to-end transaction between a farmer, a storage silo, and a buyer without traditional bank intermediaries. This milestone demonstrated blockchain's capability to handle not only traceability but also financial settlement in supply chain operations.

Other agricultural supply chains have adopted blockchain–IoT solutions. Ferrández-Pastor et al. (2022) [18] developed an industrial hemp traceability model that combines IoT sensors and blockchain to cope with the regulatory complexity of hemp production. Their system deploys sensors in the field (monitoring temperature, humidity, GPS location, etc.) and feeds this data to a permissioned blockchain (Hyperledger Fabric, as part of IBM Food Trust). Only authorized actors (farmers, processors, distributors) can submit or validate transactions. A user interface allows stakeholders to input or view data at each stage, while the blockchain backbone (Hyperledger) guarantees data immutability and security. The authors focused on designing the system under Hyperledger Fabric's model, demonstrating how a private blockchain can be tailored for a specific agri-food context to enhance transparency and data integrity in real time.

Large multinational companies have also actively explored blockchain for food safety and provenance. Walmart's pork and mango pilots with IBM in 2017–2018, which evolved into the IBM Food Trust network, are landmark case studies. Using a distributed ledger (built on Hyperledger Fabric), Walmart was able to trace sliced mangoes back to their farm of origin in just 2.2 seconds [2], compared to nearly a week using traditional methods. Following these projects, Walmart mandated suppliers of certain products to join its blockchain system, indicating strong industry validation of the technology's value by improving supply chain visibility. At the same time, other retailers and food companies (such as Carrefour with its blockchain food labels, and Alibaba's projects for food imports) have reported improvements in consumer trust and efficiency by using distributed ledgers to share traceability data among producers, inspectors, and consumers.

Additionally, academic studies have examined the role of blockchain in supply chains, such as the systematic literature review by Casino et al. (2022) [5], which surveyed dozens of blockchain-based traceability implementations and concluded that many projects remain in prototype stages, often using permissioned blockchains with limited scope. On the other hand, Montecchi et al. (2019) [10] reviewed practices, challenges, and opportunities in blockchain and supply chain management, concluding that while blockchain can enhance transparency and tracking, organizations face technical integration challenges and must address issues like data privacy and interoperability. Similarly, Kumar et al. (2020) [4] asked whether blockchain is a "silver bullet" for supply chain management, identifying technical challenges such as scalability and the need for better integration with Internet-of-Things

sensors and enterprise systems. These analyses suggest that even though the potential of blockchain is widely recognized, comprehensive frameworks that integrate blockchain with IoT and analytics are still emerging [23].

Many research papers address machine learning and analytics to predict food safety risks or to verify product information. For instance, Thota et al. (2020) [13] applied multi-source deep learning for food package verification (detecting mismatches between labels and contents), and Nogales et al. (2020) [14] used neural networks to predict the likelihood of food safety alerts from historical data. Kollia et al. (2021) [16] proposed an "intelligent food supply chain" employing AI to optimize various processes. While these studies did not involve blockchain, they highlight the growing role of predictive algorithms in managing quality and safety [7]. In the context of IoT-based cold chain monitoring, researchers have also tried combining sensor networks with ML [24]. For example, one experiment deployed IoT sensors (temperature, humidity, gas) and federated learning algorithms to classify the freshness of street-vended foods. These efforts show that ML can extract valuable insights (such as freshness or spoilage indicators) from sensor data. However, integrating such ML-driven insights into an auditable, trustworthy framework (like a blockchain-backed system) remains nascent. Most blockchain traceability systems still rely on predefined rules or human inspection for decision-making, rather than automated predictive analytics.

Prior work has laid important groundwork by demonstrating blockchain-based traceability and IoT-based monitoring in food supply chains, and by exploring AI techniques for food quality prediction. Yet, a gap remains in combining these elements into a unified architecture. This work differentiates itself by using a public blockchain (Ethereum) for openness, by implementing a smart-contract-controlled state machine that links evidence to financial outcomes, and by integrating an ML layer for predictive quality control. To our knowledge, this is one of the first end-to-end systems that not only records supply chain data on a blockchain but also automates operational decisions (such as payment or rejection of goods) based on real-time IoT data analytics. Additionally, our design emphasizes modularity and reusability, aiming to serve as a reference model for similar traceability challenges.

III. Methodology: A Modular Web3 Traceability Architecture

A. Architecture Overview and Design Principles

The proposed approach follows a modular architecture that segregates functionalities into distinct layers, ensuring scalability, maintainability, and a clear separation of concerns. At a high level, the system consists of (1) an IoT sensing layer deployed in the physical supply chain, (2) an off-chain data management layer for processing and storing detailed records, (3) an on-chain blockchain layer (Ethereum smart contracts) that anchors critical events and enforces business logic, and an analytics layer employing machine learning for predictive analysis and decision support. Fig. 1 illustrates these components and their interactions in the context of a typical farm-to-client supply chain segment.

The use of the Ethereum public blockchain as the backbone for trust and data integrity is a fundamental design choice. By

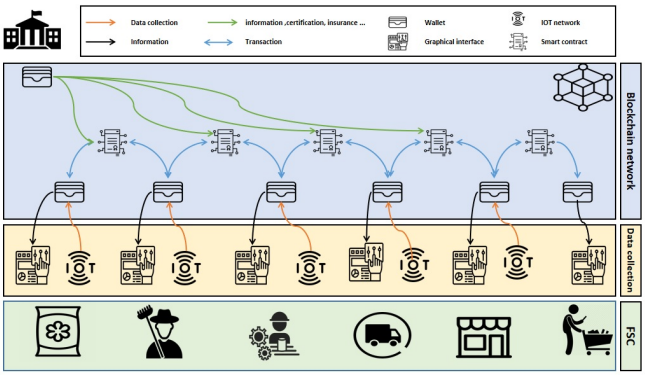

Fig. 1. End-to-end blockchain–IoT architecture for food supply traceability.

leveraging Ethereum, the designed system inherits a decentralized and permissionless security model: no single party controls the ledger, and tampering with recorded events is computationally infeasible. This choice aligns with our goal of openness (any stakeholder or regulator can verify records) and longevity (public blockchains tend to be highly fault tolerant and continuously maintained by a global community).

To accommodate Ethereum's constraints (notably transaction costs and throughput limits), a hybrid on-chain/off-chain model is designed in such a way that only essential metadata and cryptographic commitments are stored on-chain, while bulk sensor readings and documents reside off-chain in a secure repository. In this way, the integrity and non-repudiation of all data (via on-chain hashes) are ensured without incurring excessive gas costs or bloating the blockchain. The on-chain records act as an immutable "ledger of hashes" that attest to the existence, timestamp, and origin of each piece of evidence, whereas the off-chain storage holds the full content accessible to authorized parties.

Another key principle is modularity in the smart contract design. Two complementary smart contracts are implemented, each handling a different aspect of the system's logic. The first is the Data Collection Contract (DCC), which is responsible for registering IoT devices and anchoring the sensor data they produce. The second is the Commercial Relationship Contract (CRC), which encapsulates the business logic of a supply agreement (for example, the sale of a batch of produce) as a finite state machine with transitions based on data-driven conditions. By splitting functionality this way, the separation of concerns is achieved: the DCC focuses on data provenance (ensuring that all sensor inputs are recorded and tied to their source), while the CRC focuses on transactional logic between parties (ensuring that payment or acceptance of goods is contingent on the satisfaction of specified criteria).

This modular approach not only improves clarity but also enhances security, since each contract is simpler and can be audited or updated independently if needed.

Fig. 2 and 3 emphasize the modular decomposition, highlighting that each functional unit (data collection, commercial logic, analytics, etc.) can be developed and evolved somewhat independently. This modularity also facilitates reuse: for instance, the Data Collection Contract could be reused across different types of supply chains or integrated with additional

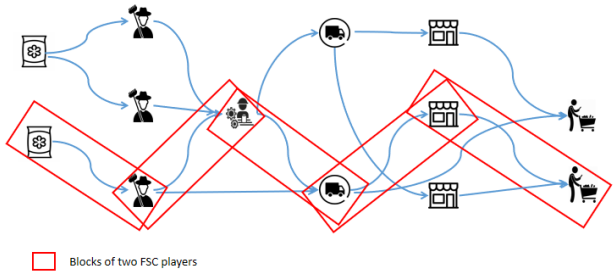

Fig. 2. Modular bi-lateral chain architecture.

sensor types with minimal changes to the blockchain layer.

B. Data and Business Logic Ethereum Smart Contracts Layer

At the core of the traceability system is an Ethereum-based smart contract layer that provides a trustworthy, automated ledger of events. Two families of smart contracts are deployed to orchestrate the lifecycle of a shipment or lot:

1) Data Collection Contract (DCC): This contract serves as a tamper-proof registry for sensor data related to a particular lot (or a group of lots). When an IoT sensor node collects data (temperature readings over a certain time window), these readings are first aggregated and signed off-chain. The DCC allows authorized entities (the farmer or a designated data aggregator) to record a signed summary of the sensor data on the blockchain. Specifically, the DCC records (see Fig. 4).

- The unique identity of each sensor or device that will report data (bound to an owner or role, such as "Farm #7 – Sensor A"). Only registered device IDs can submit data, enforcing source authenticity.
- Calibration windows for sensors, to ensure that data is only accepted when the device is within its valid calibration period (this helps maintain data quality).
- Data collection windows or batch identifiers, defining time intervals or phases of the supply chain (farm storage, transport leg 1, processing facility) during which data is collected for a specific lot.
- The hashed summary of sensor data for each window, along with a digital signature from the data aggregator. In practice, the raw sensor readings (which may be numerous) are kept off-chain, but a cryptographic digest (a Merkle root or SHA-256 hash of all readings in that batch) is computed as a compact representation. This digest, along with metadata ("Lot #123, Window 1, Sensor A stats"), is stored in the DCC. The contract thereby anchors an immutable proof that a set of readings was observed, without revealing the raw data on-chain.
- Any anomaly flags or statistics associated with that window's data. For instance, the off-chain aggregator includes in the summary the minimum, maximum, and mean of the temperature during transport, as well as a boolean flag indicating if any reading exceeded a critical threshold. These summary statistics are recorded to provide a quick on-chain insight into data quality, and they are signed to prevent tampering.

The DCC essentially functions as a public audit trail of sensor inputs. Each submission to the DCC emits a blockchain event (for transparency) and updates the state to reflect that "for Lot X, during time window Y, sensor Z reported data with hash H at time T." Because all entries are timestamped and cryptographically linked to the raw data, any future verification can confirm if the data has been altered or if a sensor reported plausibly. The DCC by itself does not make judgments about the data; it simply secures it and makes it available for the business logic to reference.

2) Commercial Relationship Contract (CRC): This contract encodes the business transaction between supply chain parties (for example, a farmer selling a batch of produce to a processor). The CRC is implemented as a contractual state machine that progresses through defined states based on events and conditions (see Fig. 5):

- States include OPEN (create an order to buy a lot), IN TRANSIT (goods have been dispatched and are en route), RECEIVED (goods arrived at destination, pending acceptance), VERIFIED and SETTLED (buyer accepts the goods, triggering settlement), and DISPUTED (buyer raises a dispute, triggering an escrow or arbitration process). There may also be an AUTOSETTLED sub-state to indicate that the contract self-executed settlement because all conditions were met without intervention.
- The CRC holds the payment escrow for the transaction (the buyer's payment is locked in the contract upon creation). It defines rules for releasing or refunding that payment depending on the state transitions. This ensures that financial exchanges are tightly coupled to traceability evidence.
- Crucially, transitions between states can be made conditional on data anchored by the DCC or other evidence. For example, the CRC might have a function markReceived() that the buyer (or an automated agent) can call to move from Delivered to Accepted, but only if certain conditions are satisfied: passRate $\geq$ threshold AND docsComplete $=$ true. These conditions can be evaluated by checking the on-chain data references.
  - passRate $\geq$ threshold is required; passRate is defined as the ratio of correct readings by IoT sensors during a specific window for a specific lot. The correctness of the readings is determined by pretrained machine learning models. The data hash for the transport window is present in the DCC, and no anomaly flag was raised (or the ML conformity score is above a threshold).
  - `docsComplete` could require that the hash of a quality certificate or regulatory document has been submitted to the blockchain (perhaps via another function of the CRC or a document registry contract).

- If conditions are met, the contract automatically executes the acceptance: marking the state as Accepted and releasing payment to the seller (minus any escrow fees).

- If there is a discrepancy – for example, the seller disputes the rejection – either party can invoke `openDispute()`, which moves the contract to the DISPUTED state. In Dispute, the funds might remain locked or go into a separate escrow, and an arbitrator or predefined oracle could be involved to resolve the issue.

The CRC thereby couples cryptographic evidence with financial logic. It ensures that objective data (temperature log, presence of certifications) directly influences the outcome (payment or penalty), rather than relying solely on after-the-fact human judgment. Of course, if automatic criteria are not clear-cut, the contract can still allow human intervention (for example, an inspector can examine off-chain data and then call an approve() function). But importantly, the blockchain provides an immutable record of all such events (deliveries, acceptances, rejections, disputes), with timestamps and responsible identities.

Both the DCC and CRC are written in Solidity (for Ethereum) and designed to be general templates that can be reused for different commodities or supply chain relationships. They incorporate role-based access control: only the designated "producer" role can submit certain data, only the "buyer" can call accept/reject, etc. Identities are managed via Ethereum addresses (public keys).

To preserve privacy on a public blockchain, pseudonymous identifiers are used to avoid any personal or sensitive data on-chain. For instance, lots and users are referred to by IDs or hashes rather than names, and even location data is not posted in plaintext on-chain. The on-chain events contain only minimal information, such as a lot ID, an event type, a hash or root of the relevant data, a pointer to off-chain storage, a timestamp, and the signer's identity (address). This minimal disclosure ensures that while anyone can see that "Lot #ABC was rejected at time X for reason Y (hashed) by party Z," they cannot derive the sensitive details (like exact temperature values or the names of the parties) without proper authorization to view off-chain content.

C. Hybrid On-Chain/Off-Chain Data Management

One of the novel aspects of the architecture is the hybrid data management strategy, which balances transparency with efficiency and privacy. The blockchain (on-chain) ledger acts as a public, tamper-proof journal of essential events and evidence hashes, while the off-chain storage serves as a secure document repository for full data records. The interface between off-chain and on-chain components is carefully designed to ensure they remain cryptographically linked.

1) On-chain: As described, the blockchain stores event metadata and hashes. Key on-chain data includes:

- Lot identifiers and key lifecycle events (creation, shipment dispatch, receipt, etc.), each recorded as an event in the CRC. This provides a time-ordered trail of what happened to each lot, immutable and timestamped.
- Pseudonymized participant identifiers (representing, for example, the farmer, logistics provider, processor). These allow the system to show which role took which action without exposing personal or sensitive information.
- Hashes of sensor data summaries (from DCC) and hashes of document bundles (submitted to CRC or a document contract). These are the integrity anchors: by storing a cryptographic hash of a data item on Ethereum, any tampering with the off-chain data can be detected (the hash would mismatch). Storing Merkle roots when dealing with many data points allows efficient aggregation. For example, if a sensor reports 1000 readings in a journey, instead of storing 1000 hashes, a single Merkle root representing all readings is stored. Similarly, a batch of multiple documents (certificates, bills of lading, inspection reports) can be represented by one root hash.
- Minimal descriptive tags (non-sensitive) such as data type indicators ("temp_humidity_summary"), version numbers, or references to standards. These help in interpreting the hashes. For instance, a hash might be tagged as "calibrationCertHash" vs. "transportDataHash" to indicate what evidence it corresponds to.

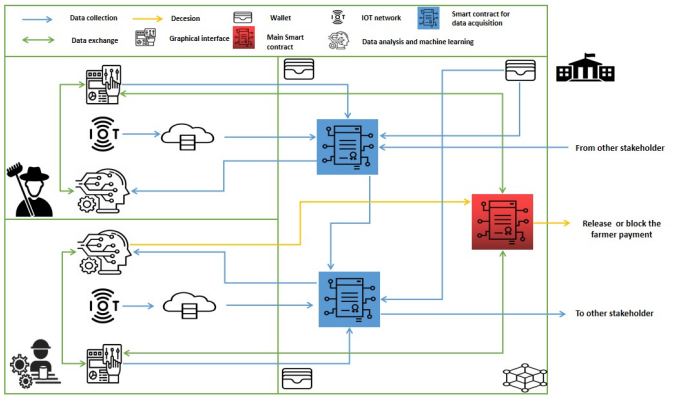

Fig. 3. Detailed IoT–Blockchain–ML module design.

The on-chain content is kept lean to adhere to the principle of parsimony: nothing extraneous (no large data, no personal info, no redundant records) is stored. This minimizes blockchain storage costs and avoids performance issues. Because each on-chain entry is immutable and time-stamped, an instant public audit of the "skeleton" of the supply chain events is guaranteed. At any time, an auditor or stakeholder can query the blockchain for a given lot ID and retrieve the sequence of events (Lot 123: created by X at time t0, shipped at t1, sensor data hash H1 at t2, received by Y at t3, anomaly flag raised, dispute opened at t4, etc.). This timeline is trustlessly verifiable and cannot be altered retroactively. If a dispute arises or a recall is needed, this immediate single source of truth is invaluable.

2) Off-Chain: The off-chain layer is where the detailed data resides. The off-chain storage is implemented as a secure cloud

database (in the prototype, a combination of a cloud file storage for raw files and a database for structured sensor logs). In a production deployment, the off-chain repository contains:

a) Sensor data logs: All the raw reading data collected by IoT devices, stored both in raw form and in processed aggregates. For example, actual temperature readings are stored every minute during transit, as well as any computed aggregates (such as daily averages) used to form the anchored summary. Keeping raw data is important for forensic analysis or re-processing with improved algorithms.

b) Operational and regulatory documents: Calibration certificates for sensors, safety certificates, quality inspection reports, photos of the cargo, shipping manifests, delivery receipts, etc. These are typically PDF or image files. Each such document is hashed, and the hash is placed on-chain via the DCC when the document becomes available. The actual file is stored off-chain and can be fetched for viewing by those with permission.

c) Quality control logs: Records of any automated or manual quality checks performed, including which parameters or thresholds were applied and what results were obtained. For instance, if the ML module PassRate ratio of the lot is 92% at delivery, this score and the reasoning (which sensor or parameter influenced it) can be recorded. Storing these off-chain allows later auditing of why a decision was made.

d) Index and versioning: The repository indexes data by lot, time, and source, enabling quick lookup ("retrieve all data for Lot #123"). If a document is updated or a data entry corrected (which might happen if, say, an initial upload was wrong but later amended with proper authorization), versioning preserves past versions for audit without compromising integrity. The hash on-chain can either reference the latest version, or each version can be hashed and chained.

Off-chain storage is effectively the source of truth for content, while the on-chain ledger serves as the source of proof of integrity. To tightly bind them, every item stored off-chain gets a cryptographic fingerprint (hash) that is placed on-chain. This one-to-one linkage (or one-to-many in the case of Merkle trees) means that if anyone were to alter or fake an off-chain record, the discrepancy would be immediately detectable by recomputing the hash and comparing it to the blockchain entry. In the prototype, this verification is automated, since an auditor module can fetch a document or data log, hash it, and confirm the hash matches the blockchain, thereby validating authenticity and that the data has not been changed since submission.

In this way, an equilibrium among transparency, performance, and privacy is achieved. The blockchain provides public auditability and immutability (satisfying transparency and trust), while the heavy lifting of data storage and retrieval is done off-chain (ensuring operational efficiency and scalability). Sensitive details remain protected behind access controls off-chain (addressing privacy), yet the integrity chain (hash linking) means even those who cannot see the data can trust that it hasn't been maliciously modified. This approach follows the emerging best practice in enterprise blockchain use: keep the ledger lean (for integrity and coordination) and manage data off-ledger in a controlled environment, using the ledger as a fingerprint register.

D. IoT Sensing and Data Ingestion Layer

The IoT layer is the interface to the physical world, capturing real-time data about the food products and their environment. A network of low-cost IoT sensor nodes is implemented at strategic points in the supply chain, from the farm through transportation and into storage at the processing facility. The choice of hardware and sensors prioritized affordability, ease of deployment, and sufficient accuracy for food safety parameters.

Each IoT node in the prototype consists of an ESP8266-class microcontroller (a cheap Wi-Fi-enabled microcontroller) connected to environmental sensors (for temperature and humidity), and optionally a GPS module for location data. The ESP8266 was chosen for its low cost (on the order of $5) and built-in wireless connectivity, which makes it feasible to deploy many units across a supply chain without significant expense. A DHT22 humidity/temperature sensor and BMP280 barometric sensors for capturing climate conditions are used, covering the typical range and accuracy needed for produce monitoring (temperature accuracy ±0.5°C, humidity ±2% RH). In addition, nodes can interface with other sensors like light or gas (CO_2/O_2) if needed, and the architecture can accommodate any sensor that provides digital readings. Each device has a unique ID that is registered in the blockchain (via the DCC) along with the identity of its owner or manager. This binding ensures accountability – data from a sensor can be traced to the party responsible for that segment of the chain. It also enables role-based permissions: for instance, only the owner's private key (or an authorized delegate) can sign the data from that sensor, preventing spoofing.

1) Data capture and aggregation: The IoT nodes continuously collect measurements (one reading per minute). Rather than transmit every raw reading to the blockchain (which would be infeasible due to volume and cost), an edge aggregation strategy is employed, so the device (or a local gateway or cloud service) aggregates readings over a defined time window – for example, computing hourly or per-transport-trip summaries. In this implementation, the sensors send their raw readings via MQTT to a lightweight serverless processing pipeline (implemented with ThingSpeak server). This pipeline is event-driven, so when a new batch of data is available or a shipment status changes, it triggers data consolidation.

During aggregation, relevant summary statistics are computed, such as minimum, maximum, average temperature in the window, cumulative time out of safe range, and standard deviation, etc. Also, initial anomaly checks are performed, for instance, flagging if any reading exceeds a threshold. Simple rules or z-score-based outlier detection can be applied at this stage. The result is a compact summary of the window's data, including any flags (anomaly_flag = true if an out-of-range incident occurred). Then, a structured data object containing these stats, the timeframe, the device ID, and the lot ID is created.

2) Digital signing: The aggregated data summary is then digitally signed by the responsible party's private key (or by a device key). In the prototype, the aggregation service signs the summary using an Ethereum account associated with the data provider. This signature is critical for security – it proves that the data came from a legitimate source (and not an attacker injecting false data). Since the public key is known

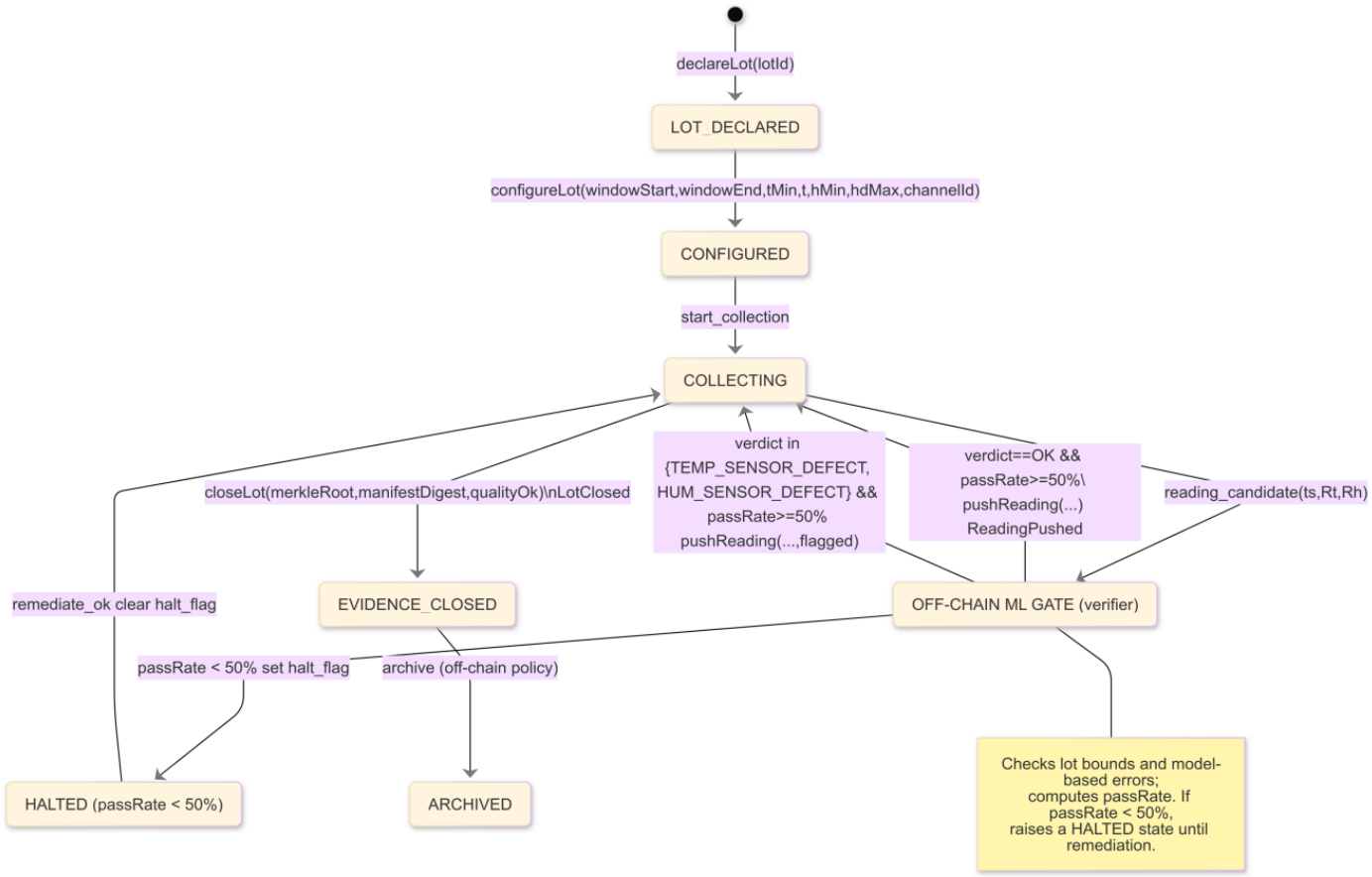

Fig. 4. DCC contract state diagram.

on-chain (the device was registered in DCC), others can verify the signature. The signed summary is then hashed (or its Merkle root computed if it's complex data) and prepared for blockchain submission.

3) Anchoring data on-chain: The final step is to send a transaction to the Data Collection Contract on Ethereum. The transaction includes the device ID, the time window, the computed hash (and any metadata like anomaly flags), and the signature. When this transaction is mined into the blockchain, the DCC emits an event logging the submission. At this point, the sensor data summary is irrevocably anchored on-chain, with a timestamp and signer identity. The actual summary data and raw readings remain off-chain (they have been stored in the cloud DB), but anyone with access can later retrieve them and recompute the hash to verify integrity.

4) Hardware reliability and calibration: Given the critical role of sensors, measures to ensure data quality from IoT devices were implemented. Before deployment, each sensor was calibrated, and its calibration certificate's hash was stored (so that later on, one can verify that the device was certified at a certain accuracy). The system also supports periodic calibration events (a sensor might need recalibration every 6 months), which would be recorded. If a sensor's calibration expires, the DCC can reject new data from it until re-certified. Also, basic fault tolerance is handled: if a sensor fails or sends obviously erroneous data, the anomaly flags will catch it, and that data can be marked as invalid. Redundant sensors can be deployed (two per shipment) to provide backup data for critical parameters.

The IoT layer instrumentalizes the supply chain with continuous data. By deploying inexpensive sensors, opaque segments (like the inside of a truck or a storage facility) are turned into sources of real-time information. The implemented data pipeline – from sensor to cloud aggregator to blockchain – was able to operate in near-real time. In the prototype, data from sensors was posted to the blockchain within a few seconds to a minute of being recorded (depending on blockchain transaction times). This is sufficient for many use cases, as quality issues like temperature abuse typically happen over hours. However, the system could be tuned for faster response if sub-second reaction was needed for certain applications.

E. Machine Learning Analytics for Predictive Quality Control

A distinguishing feature of the system is the integration of a machine learning (ML) layer that works in tandem with the IoT and blockchain components to provide predictive analytics and automated decision support. The goal of this ML layer is to enhance the raw sensor data with intelligence – detecting subtle patterns, predicting outcomes, and assessing conformity to

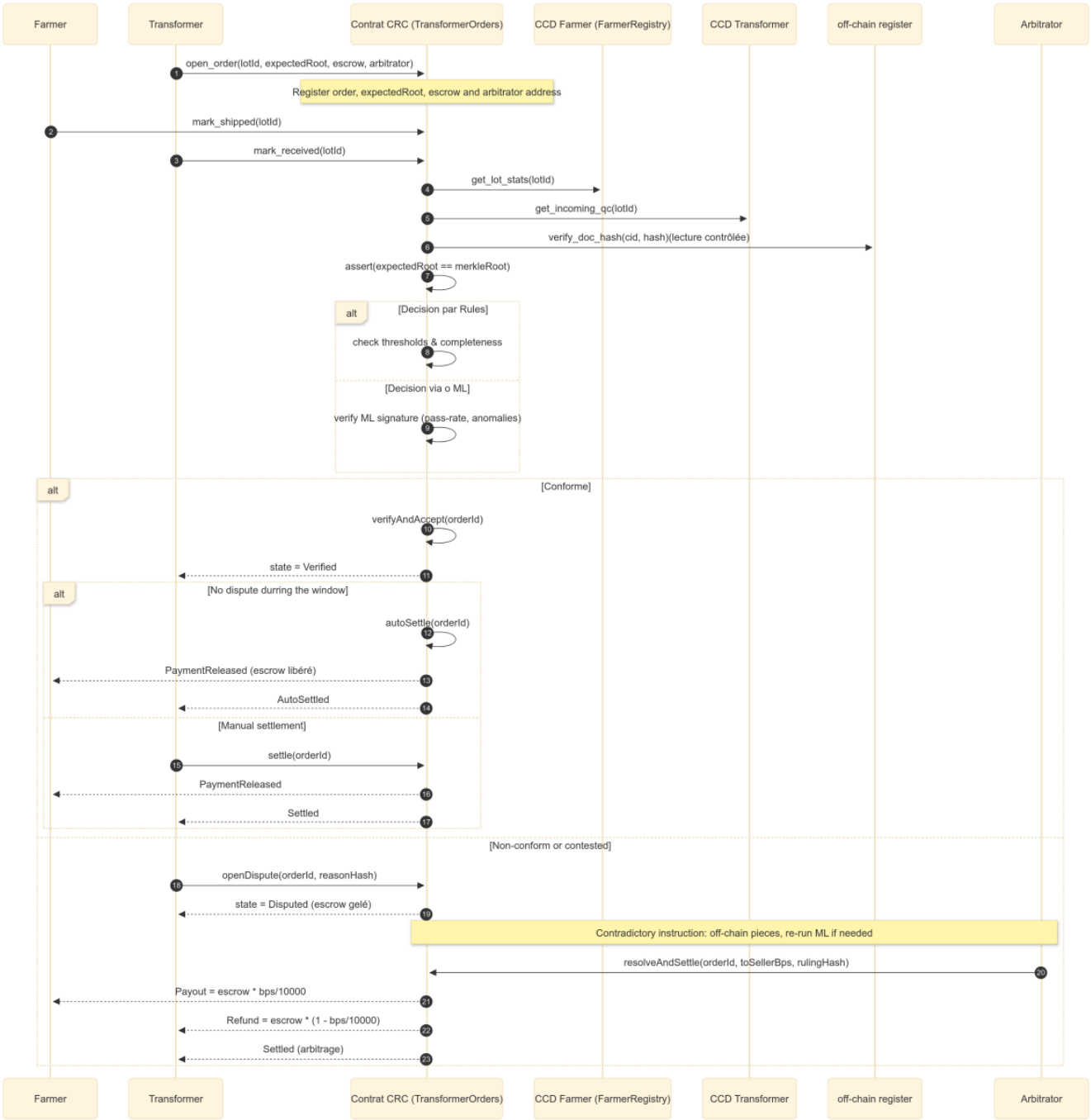

Fig. 5. CRC contract state diagram.

quality standards – which can then inform the smart contracts (CRC) on-chain.

1) Roles of ML in the architecture: ML models were incorporated for three main tasks: (1) sensor data estimation and anomaly detection, (2) PassRate ratio conformity, and (3) decision automation. These correspond to increasingly advanced analyses:

a) Estimation and anomaly detection: Using historical data and contextual features, predict expected sensor readings and flag deviations. For example, given the time of day, ambient weather, and previous readings, predict what the temperature in a truck should be – if the actual reading diverges significantly, it might indicate a sensor fault. An ML model can learn typical patterns and provide a more robust anomaly flag than a static threshold. Anomaly flags computed by ML complement the simple threshold checks in the IoT layer.

b) PassRate ratio conformity: This is a predictive quality ratio for the produced lot, indicating the likelihood that it meets quality/safety criteria at the end of the chain. The ML model considers the entire sensor history (and potentially external factors like weather or transport duration) to output a ratio (0 to 100%) or classification (conformant vs non-conformant). For example, if a product was kept consistently in ideal conditions, the ratio will be high, and if there were temperature excursions or delays, the ratio drops. This essentially automates a pre-inspection, so it predicts if the lot will pass quality control or if it's at risk.

c) Decision automation: The outputs of the above models feed into a decision logic (some of which is encoded on-chain). For instance, if the conformity ratio is above a threshold and all required documents are present, the contract can auto-accept the delivery and trigger payment (this is termed AutoSettlement). If the score is low or anomalies were detected, the contract might automatically trigger a hold or require manual review (possibly entering a Dispute state). Thus, ML enables a shift from reactive to proactive management – catching issues early and handling many routine cases without manual intervention.

2) Model development and OFF-CHAIN ML GATE: A dataset containing time-series of temperature and humidity from a specific location (Casablanca, Morocco) was used to train several regression models, including:

- M1: predicts temperature based on the time of year and a given humidity,
- M2: predicts humidity based on the time of year and a given temperature,
- M3/M4: predict temperature and humidity at a given time of year.

These models are trained offline on historical data and are used at run time as part of an OFF-CHAIN ML GATE that transforms raw sensor readings into a compact conformity indicator (PassRate) and associated anomaly flags. The operation of this gate is illustrated conceptually in Fig. 6. For each lot and each monitoring window (for example, a transport leg or storage phase), the corresponding temperature and humidity readings are grouped into an ordered time series and preprocessed to handle missing values and obvious sensor glitches.

At run time, models M_1–M_4 are applied to these time series to produce expected values $\hat{x}_t$ for each observed reading x_t. For each time step t in the monitoring window, the deviation $|x_t - \hat{x}_t|$ is computed and compared to a tolerance ϵ, and the measured value is also checked against regulatory or contractual bounds $[x_{\min}, x_{\max}]$ appropriate for the product. A reading is classified as conformant ("pass") if both conditions are satisfied (small prediction error and within the allowed range), and as non-conformant ("fail") otherwise. After all N readings in the window have been evaluated, the PassRate is computed as:

$$\text{PassRate} = \frac{N_{\text{pass}}}{N},$$

where N_{pass} is the number of pass readings.

In addition to PassRate, summary statistics (minimum, maximum, mean) and a boolean anomaly flag indicating whether any severe violation occurred are produced. These outputs, together with the hash of the underlying raw data batch, form the OFF-CHAIN ML GATE summary for that window. The summary is stored off-chain, and its cryptographic hash, PassRate value, and anomaly flag are anchored on-chain via the Data Collection Contract (DCC) so that the Commercial Relationship Contract (CRC) can reference them at decision time.

When a lot reaches a decision point (for example, at delivery), the CRC reads the latest available summary for the relevant window. Automatic acceptance is authorized if PassRate $\geq \tau$ for a predefined threshold τ and if no anomalies or missing documents are indicated. In this case, the CRC transitions to an accepting state and releases payment. If the threshold condition is not satisfied or an anomaly is present, the contract remains in a hold or disputed state, and the detailed off-chain records can be inspected by the parties or by an external auditor. In the prototype, the threshold is set to 0.8 (80%).

The following table summarizes the best results obtained after training the models using multiple ML algorithms; see Table I:

TABLE I. PERFORMANCE COMPARISON OF MODELS FOR DIFFERENT CASES

Case	Target	Model	MAE	RMSE	R^2
M1	temperature	MLP_small	1.6217	2.0908	0.8462
M2	humidity	SVR_RBF	6.5615	9.3837	0.6789
M3	temperature	GradientBoosting	1.8683	2.4381	0.7909
M4	humidity	SVR_RBF	7.4696	10.8154	0.5734

The verification process and ML outcomes are summarized in Fig. 6.

Instead of only collecting and displaying data, the system triggers autonomous decisions based on these ML outputs. The threshold and model parameters can be adjusted according to risk tolerance. During development, care was taken to ensure that the contract does not blindly follow ML outputs; for example, if a required document hash is missing, the contract

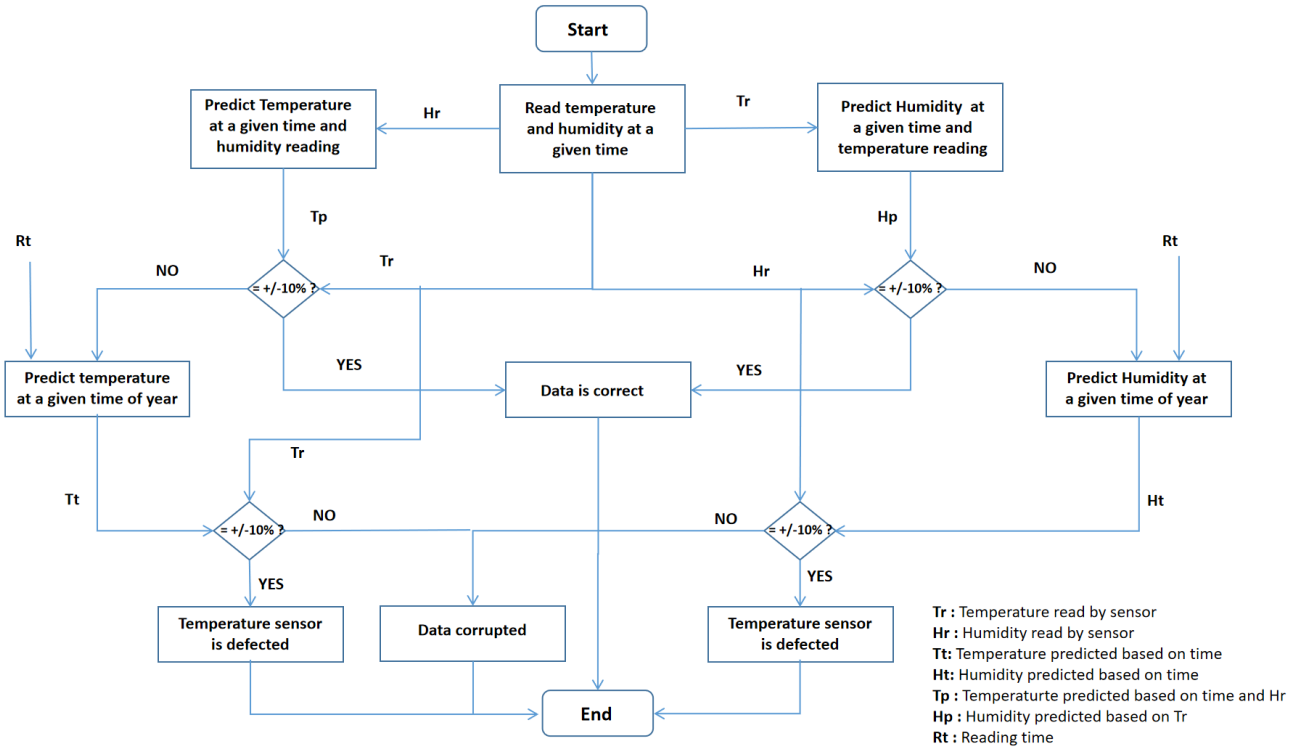

Fig. 6. Verification process and ML outcomes.

does not auto-settle even if the PassRate is high. Thus, the ML layer provides predictive quality control that works hand-in-hand with IoT data collection and blockchain enforcement, augmenting but not overriding fundamental checks.

IV. Experimental Results and Prototype Validation

To validate the proposed architecture, a full end-to-end prototype was developed and tested under realistic conditions. The system integrates IoT hardware, a cloud-based back end, and Ethereum smart contracts deployed on a local test network, enabling the simulation of a complete food supply chain transaction. Scenario-based experiments were subsequently conducted to evaluate traceability, automation, and the integrity of the system's outputs.

1) Prototype setup: A supply chain scenario with two primary actors – a producer (farmer) and a buyer (processor) – engaged in a transaction of a food product (a lot of fresh vegetables) has been configured. The IoT network consisted of four ESP8266-based sensor nodes; two of them were installed at the farm (monitoring storage conditions pre-shipment) and two in the delivery truck (monitoring conditions during transit). Each node recorded temperature and humidity every minute and was connected via Wi-Fi to an internet gateway in order to transmit data to an IoT cloud service (ThingSpeak was used as the cloud server) in near real-time.

On the blockchain side, the smart contracts have been deployed on the Ganache local blockchain. Ganache is an Ethereum blockchain simulator. The deployed smart contracts were the Data Collection Contract (DCC) and Commercial Relationship Contract (CRC). The farmer's account was set as the data provider (authorized to post sensor data and documents), and the buyer's account was set with the right to accept or dispute. A payment amount (simulated in test Ether) was escrowed in the CRC to mimic the financial transactions.

Also, a simple web page for visualization and manual interactions in order to display its incoming sensor readings in real-time has been developed to show the state of the smart contracts, and allow the user (playing the role of buyer, farmer, or arbitrator) to manually trigger certain events (like markShipped() or raise a dispute) if needed. The pages polled the Ethereum network for new events and also fetched data from off-chain storage when requested (for instance, to display the full temperature log of a trip when an arbitrator clicks a particular button).

2) Traceability and data integrity: First, a verification process was conducted to ensure that the system provided continuous traceability of the lot's condition and movements. As the shipment left the farm, an "Expedition" event was recorded on-chain (through the CRC, marking that the goods were shipped). The sensor nodes in the truck immediately began sending data, at the same time, the cloud aggregator created its first summary after 60 minutes of transit and invoked the DCC, resulting in a DataAnchored event on Ethereum for "Lot-TEST-123-Window 01" with the hash of that data. As the truck arrived at the processor, another "markReceived()" event was logged by the CRC (triggered by the buyer's app scanning a QR code on delivery, in the simulation). The second window of sensor data (covering the latter half of transit and unloading)

was then summarized and anchored on-chain. Finally, the buyer performing a quick inspection and then either accepting or rejecting the lot depending on the scenario was simulated (see Fig. 5). During this process, end-to-end traceability was demonstrated, every key step in the lot's journey was recorded immutably on the blockchain (with timestamps and responsible identities), and every data artifact (sensor readings summary, certificate, etc.) was linked via hashes, so that any independent auditor or arbitrator could reconstruct the entire chain of custody by following the blockchain trail and pulling the evidence from off-chain storage. A verification of the data item's hash took milliseconds, and any alteration of data (a try to modify a sensor summary file by a small amount) led to a hash mismatch, which immediately indicated tampering.

3) Machine learning outcomes: The ML models were integrated into the pipeline after each reading was pulled. The real sensor data from the farm storage and the truck were fed into the models. In the normal scenario, the temperature remained within 4–8°C and humidity within 70–90% (suitable for the concerned products), which the ML model correctly recognized as normal. It output a conformity ratio of 0.95 at the end of transit, indicating high confidence that the lot is good. In the anomalous scenario (detailed next), a period where the temperature rose to 15°C for 1 hour was introduced. The ML anomaly detection flagged this deviation (the predicted temp vs actual diverged significantly), and the conformity score dropped to 0.4 (indicating likely non-conformity). These scores were packaged into the final data summary and anchored. The CRC read the score and, according to the preset threshold (0.8), automatically took different actions in the two scenarios:

- In the normal case, after receiving the delivery and the final data hash, the CRC saw score=0.95, no anomalies, all documents present, and immediately executed the AutoSettlement transition, as a result, an on-chain event AutoSettled was emitted, indicating that the contract released payment to the farmer. The buyer's interface showed "auto-accepted. PaymentReleased". The entire process required no manual decision, and both parties could verify why by checking the blockchain, one sees the data hash and can retrieve the underlying report, which shows all conditions green. This showcases the power of smart contract automation informed by ML.
- In the anomaly case, when the delivery was markReceived(), the CRC noted score=0.4 or `anomalyFlag`=true. The buyer could open a DisputeOpened event, which froze the payment and alerted the farmer that a problem had been detected, and both parties could open the off-chain sensor log (via the hash) to see the temperature spike – which confirmed a breach of contract terms (temperature out of range). In a real setting, they might then formally reject or negotiate a solution. The main point is that the system proactively caught the issue and preserved evidence. Because of the cryptographically linked data, the farmer could not contest the truth of the temperature records (they were signed by the devices and anchored). And therefore, a dispute resolution is greatly simplified; it revolves around interpreting the data (which is trusted), rather than arguing about facts.

4) Scenario analysis: Two contrasted scenarios as described in Fig. 7, were effectively tested:

a) Conformant lot scenario: All conditions were kept within agreed ranges, and all required documents (sensor calibration cert, farm origination cert, etc.) were provided. As a result, the system produced a verifiable trace for each step, the ML score was high, and the smart contract auto-settled. The final outcome was a successfully completed transaction with both parties confident in the result. Notably, the time from delivery to payment release was near-instant (just the time to mine a block after delivery), much faster than a typical manual inspection and payment cycle. This indicates efficiency gains for stakeholders.

b) Non-conformant lot scenario: One condition (temperature) went out of range for a while, simulating a partial failure or sensor failure. A missing document was also tested by omitting the cleaning certificate for the transport container. The ML flagged the temperature anomaly, giving a low conformity passRate. Additionally, the smart contract logic detected the missing document hash (it expected a cleaningCertHash to be present, which it wasn't). Either of these alone would trigger a hold. In the test, both occurred, so the outcome was clearly not to auto-accept. The contract gave the buyer the opportunity to open a dispute. In a real-world sense, this corresponds to quarantining the batch for further inspection. The traceability system here provided forensic detail: via the blockchain, one could see an event DisputeOpened by Buyer at 10:05, reason: temp_anomaly" (the reason can be encoded as an event parameter or inferred from data). The buyer and farmer could use the off-chain data repository to dig deeper (e.g. check exactly when and where the temperature rose, see if any handling mistakes happened). Ultimately, the batch would likely be rejected or downgraded in this scenario. The system ensured that this decision was backed by objective, tamper-proof evidence, reducing potential conflicts. The presence of a trusted data trail helps avoid the "blame game" – it's clear if a logistics provider's truck lost cooling at a specific time, for example, focusing accountability appropriately.

After running these scenarios, an audit trail reconstruction was performed as a final validation. The role of an external auditor was simulated (with access permissions). A reconstruction of the full story was then performed using the available on-chain and off-chain records. By querying the Ethereum logs for the lot's ID, all on-chain events were obtained: creation, dispatch, data anchors, receipt, and outcome (settled or disputed). Each event had references (IDs, hashes). Then each hash's corresponding content from the off-chain storage was fetched: the sensor data summaries (which included ML results), the actual sensor time-series, and the documents. A verification process was conducted to ensure that each hash matched and each digital signature was valid. This process allowed us to verify compliance with every contractual condition, which traced "who did what and when" with precision (Farmer submitted data at time X, system auto-settled at time Y, etc.). The chain of custody from farm to processor was unbroken and cryptographically verifiable end-to-end. This level of auditability is a marked improvement over traditional systems, where auditors must trust paper logs

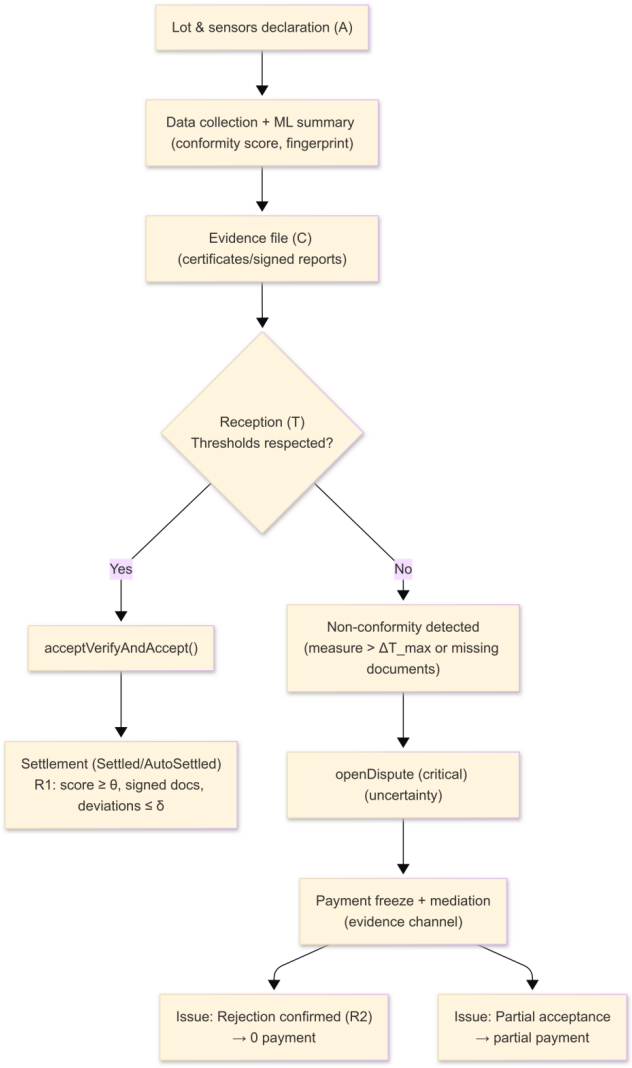

Fig. 7. Outcome workflow for compliant vs. Non-compliant lots.

or disparate databases that could be incomplete or altered. It also illustrates how the design achieves explicable decisions: even the automated actions were explainable by linking to specific data points (e.g., "auto-settlement occurred because all required criteria, as documented in hash H, were satisfied").

The experimental validation of the prototype demonstrates that the integrated blockchain-IoT-ML architecture functions as intended: it provides real-time traceability, enforces data-driven smart contract logic, and handles both nominal and off-nominal scenarios by either streamlining the workflow (for conformant lots) or flagging issues (for non-conformant ones). Improved efficiency (instantaneous settlement vs. potential delays) and improved trust through transparency (immutable evidence replacing he-said/she-said) were observed.

V. Discussion

A. Advantages and Novelty

The proposed system offers several notable benefits over traditional systems and even over earlier blockchain or IoT-only solutions, Improved Transparency and Trust, The use of a public blockchain (Ethereum) as the foundational ledger ensures that no single party can modify or hide any important information. This decentralization is crucial in multi-player supply chains where trust may not be present. All stakeholders (from farmers to retailers to regulators) can have access to an identical, tamper-proof record of key supply chain events, which improves trust in the data.

1) Data integrity and provenance: The hybrid approach of anchoring means that every piece of data (sensor readings, certificates, etc.) has a cryptographic provenance which is a strong defense against fraud. For instance, any attempt to change a food safety certificate or manipulate temperature records leads to hash mismatch.

2) Automation and efficiency: The synergy between smart contracts, sensor data and ML allows a degree of process automation that is novel in this domain. Routine operations like shipment acceptance and payment release can be executed automatically if some conditions are verified, which reduces administrative paperwork and speeds up the supply chain.

3) Accountability and auditability: Every action on-chain is linked to a cryptographic identity (linked to an organization or individual) and timestamped. This level of accountability ("who did what, when") is enhanced compared to centralized databases where logs can sometimes be edited or lost.

4) Modularity and extensibility: The proposed model is reusable and can be extended to different supply chains or to incorporate additional technologies. For example, the IoT layer could be expanded to include new sensors (GPS trackers for real-time location, imaging devices for visual quality checks, etc.).

B. Limitations and Mitigation Strategies

Despite its benefits, the system has certain limitations and challenges that need to be addressed:

1) Blockchain throughput and cost: In high-volume food supply chains, directly writing everything to the Ethereum mainnet could become costly and slow, since thousands of events occur daily, which significantly increase fees (gas costs) per transaction. In order to mitigate this issue, the system anchors data at a coarse granularity (per batch per phase rather than per sensor reading) and stores only hashes (small payloads). In this way the number of transactions and data size are reduced. On the other hand, only critical events and summaries are on-chain. This "selective anchoring" is the important identified mitigation.

2) Data privacy and confidentiality: While the system avoids putting sensitive details on-chain, the presence of certain events or hash values can reveal some information. For example, if a dispute event is visible on a public ledger, competitors might infer that a problem occurred with a shipment between certain parties. To mitigate this issue, pseudonymization is applied (using IDs instead of real names), and sensitive reasons are not revealed explicitly on-chain. Also, all actual content remains off-chain protected by access controls.

3) IoT device reliability and security: The maxim "garbage in, garbage out" applies – if IoT sensors report incorrect data (due to malfunction or tampering), the blockchain will

faithfully record bad data, and smart contracts might act on it. Blockchain doesn't guarantee data accuracy at the source. The model mitigates this risk through multiple actions:

a) Calibration and validation: Ensuring devices are calibrated and periodically checked mitigates systematic errors, and every calibration is anchored on-chain to increase confidence.

b) Anomaly detection (ML and rules): The ML layer and simple thresholds catch many outliers. If a sensor suddenly gives an extreme reading, the system flags it. Also, the verification process reveals important information about the quality of the collected data.

c) Edge computing and fail-safes: The approach relies on connectivity for data to be anchored. If a connection is lost for a long time, data might not get anchored promptly. This issue is mitigated by local storage on devices and backfilling once online. One could also incorporate mesh networks or alternative communications like LoRaWAN for remote areas.

4) Adoption and ecosystem challenges: A technical system is only as good as its adoption by stakeholders. Implementing this in an actual food supply chain means aligning multiple independent parties to use the platform and trust its outcomes. There could be resistance due to cost, complexity, or fear of data sharing. This is a limitation beyond technology; it involves governance, and possibly regulatory push:

- Start with pilot programs in a controlled environment (a single company's supply network or an industry consortium like Walmart did with IBM) to demonstrate value.
- Education and training are important so the users trust the system. The system can be designed with user-friendly interfaces (mobile apps for farmers to see their data).
- Regulatory alignment: Regulations are very important to push stakeholders to adopt such technologies.
- The architecture can also integrate with existing systems via APIs, it doesn't require ripping out the existing databases. For instance, if a company already has an ERP system logging batches, that system can feed into the blockchain layer through an adapter.

5) Sustainability considerations: While this work is motivated partly by sustainability (reducing waste through better monitoring, etc.), one might point out that blockchains (especially proof-of-work ones) have high energy use. Ethereum has transitioned to proof-of-stake (as of 2022), drastically cutting energy usage.

VI. Conclusion

In this paper, a comprehensive Web3-based architecture for end-to-end food supply chain traceability, integrating Ethereum blockchain smart contracts, IoT sensor networks, and machine learning analytics into a unified system is presented. Through the design and prototyping of this architecture, it has been demonstrated how combining these technologies can overcome many limitations of traditional traceability methods and add novel capabilities for data-driven decision-making in supply chain operations.

The system achieves real-time, auditable traceability by anchoring key supply chain events and sensor data on a public blockchain, ensuring the integrity, transparency, and tamper-resistance of records. At the same time, it preserves practical efficiency and confidentiality by keeping detailed data off-chain under secure management, linking the two realms with cryptographic hashes to guarantee provenance. It separates the business logic into a modular Ethereum smart contract (DCC) for sensor data and a CRC embodying a contractual state machine, which allowed for coupling evidence with actions (such as payment settlement) in a flexible manner.

By deploying low-cost IoT sensors across the supply chain, from farms to transport to storage, the system provides more visibility into environmental conditions that affect product quality and safety. The IoT layer feeds a continuous stream of data into the traceability records, enabling stakeholders to monitor conditions in real-time. The integration of an ML analytics layer introduces predictive and diagnostic intelligence in addition to detecting anomalies and predicting conformity to quality standards, effectively functioning as a virtual quality inspector that works in tandem with the physical sensors.

The findings have several implications for the future of supply chain management:

1) Enhancing food safety and quality: With end-to-end traceability and real-time monitoring, issues can be identified and addressed much faster than before, preventing contaminated or spoiled products from ever reaching consumers.

2) Building consumer trust: The ability to provide verifiable proof of a product's journey (including compliance with temperature controls, organic certifications, etc.) can boost consumer confidence.

3) Operational efficiency and automation: Automating routine checks and settlements can reduce transaction friction in supply chains, shortening payment cycles and reducing the manual workload on logistics and quality control personnel.

4) Generalizability: While the model focused on the food/agriculture domain, the architecture is applicable to any supply chain where traceability, authenticity, and condition monitoring are important (pharmaceuticals, chemicals, high-value electronics, etc.). The modular design means it can be adapted with minimal changes.

References

[1] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*. Online manuscript, 2008. Available: https://bitcoin.org/bitcoin.pdf

[2] J. George, A. Kamilaris, A. Fonts, and F. X. Prenafeta-Boldú, *Food traceability on blockchain: Walmart's pork and mango pilots with IBM*. International Journal of Information Management, 2019, vol. 52, art. 101967. doi: 10.1016/j.ijinfomgt.2019.05.007

[3] B. C. Cheong, *Leveraging blockchain for enhanced transparency and traceability in sustainable supply chains*. Discover Analytics, 2025, vol. 3, no. 1, p. 6. doi: 10.1007/s44257-025-00032-7

[4] A. Kumar, R. Liu, and Z.-J. Shan, *Blockchain: a silver bullet for supply chain management? Technical challenges and research opportunities*. Decision Sciences, 2020, vol. 51, pp. 8–37. doi: 10.1111/deci.12396

[5] F. Casino, J. Deckert, and I. Podnar Žarko, *A systematic literature review of blockchain-enabled supply chain traceability implementations*. Sustainability, 2022, vol. 14, no. 4, art. 2439. doi: 10.3390/su14042439

[6] F.-J. Ferrández-Pastor, J. Mora-Pascual, and D. Díaz-Lajara, *Agricultural traceability model based on IoT and blockchain: Application in industrial hemp production*. Journal of Industrial Information Integration, 2022, vol. 29, art. 100381. doi: 10.1016/j.jii.2022.100381

[7] International Organization for Standardization, *ISO 9000:2015 — Quality management systems — Fundamentals and vocabulary*. Geneva, Switzerland: ISO, 2015. Available: https://www.iso.org/standard/45481.html

[8] International Organization for Standardization, *ISO 22005:2007 — Traceability in the feed and food chain — General principles and basic requirements for system design and implementation*. Geneva, Switzerland: ISO, 2007. Available: https://www.iso.org/standard/36297.html

[9] Official Journal of the European Communities L31/1–24, *Regulation (EC) No 178/2002 of the European Parliament and of the Council of 28 January 2002 — General principles and requirements of food law*. 2002. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32002R0178

[10] M. Montecchi, K. Plangger, and M. Etter, *Blockchain and supply chain management: Practices, challenges and opportunities*. Journal of Business Research, 2019, vol. 100, pp. 365–380. doi: 10.1016/j.jbusres.2018.07.027

[11] S. Mondal, S. Dey, and P. T. Helo, *Blockchain-inspired RFID-based information architecture for food supply chain*. Journal of Food Engineering, 2019, vol. 263, pp. 92–101. doi: 10.1016/j.jfoodeng.2019.02.004

[12] V. Natanelov, S. Cao, M. Foth, and U. Dulleck, *Blockchain smart contracts for supply chain finance: Mapping the innovation potential in Australia–China beef supply chains*. Journal of Industrial Information Integration, 2022, vol. 30, art. 100389. doi: 10.1016/j.jii.2022.100389

[13] C. Thota, S. Kollias, and G. Leontidis, *Multi-source deep learning for food package verification*. arXiv preprint, 2020, arXiv:2001.10335. Available: https://arxiv.org/abs/2001.10335

[14] A. Nogales, D. Díaz, and Á. García-Tejedor, *Predicting food alert risks using deep learning techniques*. arXiv preprint, 2020, arXiv:2009.06704. Available: https://arxiv.org/abs/2009.06704

[15] F. Tian, *A supply chain traceability system for food safety based on HACCP, blockchain & Internet of Things*. in *Proc. 14th Int. Conf. on Service Systems and Service Management (ICSSSM)*, 2017. doi: 10.1109/ICSSSM.2017.7996119

[16] M. Kollia, P. Stevenson, and S. Kollias, *Intelligent food supply chain with deep learning*. arXiv preprint, 2021, arXiv:2105.00333. Available: https://arxiv.org/abs/2105.00333

[17] K. Biswas, V. Muthukkumarasamy, and W.-L. Tan, *Blockchain-based wine supply chain traceability system*. in *Future Technologies Conference (FTC)*, 2017, pp. 56–62. Available: https://ieeexplore.ieee.org/document/8246460

[18] F. J. Ferrández-Pastor, J. M. García-Chamizo, J. Nieto-Hidalgo, and J. Mora-Martínez, *Developing ubiquitous sensor network platform using Internet of Things: Application in precision agriculture*. Sensors, 2016, vol. 16, no. 7. doi: 10.3390/s16071141

[19] K. Addou, M. Y. El Ghoumari, S. Achkdir, and M. Azzouazi, *A decentralized model to ensure traceability and sustainability of the food supply chain by combining blockchain, IoT, and machine learning*. Mathematical Modeling and Computing, 2023, vol. 10, no. 2, pp. 498–510. doi: 10.23939/mmc2023.02.498

[20] Microsoft Azure, *What is Azure Internet of Things (IoT)?*. 2021. Available: https://docs.microsoft.com/es-es/azure/iot-fundamentals/iot-introduction

[21] B. C. Cheong, *Blockchain-enabled supply chain management: A review of security, traceability, and data integrity amid the evolving systemic demand*. Applied Sciences, 2023, vol. 15, no. 9, art. 5168. doi: 10.3390/app15095168

[22] P. Kerschke-Risch, *The horsemeat scandal: The unknown victims of economically motivated crime*. Journal of Victimology, 2017, vol. 5, no. 9, pp. 63–84. https://doi.org/10.12827/RVJV.5.03

[23] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, *Blockchain-based traceability in agri-food supply chain management: A practical implementation*. in *2018 IoT Vertical and Topical Summit on Agriculture – Tuscany (IOT Tuscany)*, 2018, pp. 1–4. doi: 10.1109/IOT-TUSCANY.2018.8373021

[24] J. Sunny, N. Undralla, and V. M. Pillai, *Supply chain transparency through blockchain-based traceability: An overview with demonstration*. Computers & Industrial Engineering, 2020, vol. 150, art. 106895. Available: https://www.sciencedirect.com/science/article/pii/S0360835219308963

[25] Amazon Web Services, *IoT services for industrial, consumer, and commercial solutions*. 2021. Available: https://aws.amazon.com/iot/