



Blockchain-Based Traceability and Peer-to-Peer Trading for Urban Fish Farming

Omolola A. OGBOLUMANI^{1*}, Benita AGBAGWARA²

^{1,2}Department of Electrical and Electronics Engineering, University of Lagos, Lagos, Nigeria

^{1*}oagbolumanischolar@gmail.com, ²benitaudrey@gmail.com

Abstract

Aquaculture now supplies many of the fish consumed worldwide, yet the origin of the fish usually cannot be verified. Species substitution remains widespread despite decades of labelling regulation, with recent market surveys reporting mislabeling in roughly one of every three samples. In small-scale production systems, the traceability record consists of paper ledgers and trust-based relationships that can be reconstructed after the fact, and existing digital platforms address this by centralising records, which relocates the trust problem rather than removing it. This paper presents a traceability architecture in which provenance records are made tamper-evident by cryptographic commitment rather than by institutional trust. Environmental telemetry captured continuously at the pond is aggregated into a per-shipment evidentiary document, stored off-chain on content-addressed distributed storage, and committed on-chain as a Keccak-256 digest, so that any later alteration of the record invalidates its published commitment. Successive custody stages embed the content identifier of the preceding stage, producing a hash-linked chain in which tamper-evidence propagates forward rather than remaining confined to a single record. The settlement is in fiat money and done at the time of the transaction using the decentralized price oracle, which ensures that the agreed value of the shipment is not affected by the fluctuations in exchange rate. This paper discusses the design of the mechanism, its threat model, the guarantees it provides, and the trust it requires.

Keywords: Aquaculture, Blockchain, Food and Water Nexus, Food traceability, Smart contracts.

1.0 Introduction

Fish is one of the most widely traded food commodities in the world. It is also one of the commodities most mislabeled. This phenomenon is well evidenced in various parts of the world. One study of 376 fish samples collected from three cities in Mexico found a mislabeling rate of 30.8 percent [1], while a global meta-analysis found similar mislabeling levels across markets and product types [2]. These mislabeling rates seem consistent, and the regulatory measures put in place to tackle this issue do not eliminate substitution because the verification process requires documentary evidence that can always be reconstructed after sales. This behavior has a strong economic foundation. Fish substitutes are usually lower value than the labelled fish, and the observable price difference drives this fraudulent act [3]. This activity has been used to introduce Illegal, Unreported, and Unregulated catches into legitimate supply chains [4]. In addition, this process distorts the market signals used in fisheries management because abundance in the market could be an indication of substitution and not an abundance in the stock itself [5],[6]. The dependency extends beyond fisheries management. Integrated assessment models of resource use in food systems take reported production and throughput figures as given inputs, so the environmental and economic conclusions they generate inherit whatever distortion is present in the underlying records [7]. The problem is sharpest at the small-farm end of the sector. In Nigeria, aquaculture has been among the fastest-growing agricultural subsectors, yet post-harvest handling remains largely undocumented and post-harvest losses are substantial [8],[9]. Technical research aimed at food production in the region has concentrated on the production side of the problem, on the allocation of land, water and energy inputs and on the energy provisioning that intensive production requires [10]-[12]. Producers who wish to sell into certified or export channels face an obvious barrier rather than a production one. They cannot demonstrate the handling history that certification requires. The instruments available to them, which are handwritten ledgers, verbal assurances and buyer relationships, are precisely the instruments that cannot survive adversarial scrutiny. Certification schemes demonstrate that the barrier is evidentiary rather than behavioral. Where labelling is verified under a certification scheme, mislabeling falls dramatically. There is a mislabeling below one per cent across 27 species in 18 countries under one such scheme [13]. The difficulty is that certification is expensive and administratively demanding, which places it out of reach of exactly the producers who would benefit most from it.

While digital traceability systems reduce the load of record-keeping, they fail to solve the problem of trust itself. A traceability database is only as trustworthy as its operator. The system cannot protect itself from tampering with the records at any later point in time; hence the guarantee of a claim by a producer is dependent on the trustworthiness of the traceability platform instead of verifiable proof. Blockchain arrives in this space as one

element of a wider digitalisation agenda for food, energy and water systems that also encompasses sensing, connectivity and machine learning, and it is distinguished from the others in that it addresses trust rather than measurement [14]. Blockchain-powered traceability has been suggested many times as a possible solution to this problem, and the systematic reviews illustrate that quite a few pieces of work have been done on this topic [15], [16]. There are three shortcomings that have driven this work. One of them is that many proposed solutions rely on the ability of participants to securely store their keys and authorize on-chain transactions. But the assumption is false in case of the producers, who need traceability the most, and thus, such systems deny themselves from their potential users. The second problem is that the literature considers the separation between on-chain and off-chain storages as a matter of scalability [17], while not analyzing what the construction ensures. Committing a digest on-chain establishes that a document has not changed since the moment of commitment. It does not establish that the document was accurate when it was written. The distinction matters for what a traceability system can honestly claim, and it is frequently elided. Third, proposals that incorporate payment almost universally denominate value in the native asset of the chain. For a producer selling perishable goods, agreeing a price in a volatile asset transfers exchange-rate risk onto the party least able to bear it.

Traceability is defined as the ability to obtain all information related to a product throughout its life cycle through recorded identification [18]. The term is intentionally noncommittal about the form the record takes, and most of the practical problems stem precisely from this noncommitment. While a record meeting the criteria set forth may still be ineffective as evidence, as it might be falsifiable. Research into seafood fraud gives both an idea of the scale of the issue and the deficiencies of documentary controls. There are misrepresentations of events in the seafood industry where substitution is widespread and not rare [19]. The global situation confirms persistence in different regions and eras [2]. Labeling format is very importance. Samples that had a label written on them were mislabeled significantly less frequently than the samples with verbal labeling [1]. This piece of research is especially pertinent to the current project, as it shows that the quality of the record itself, and not just the seller's honesty, plays a part in the outcome. In terms of regulation, there has been an attempt to introduce mandatory requirements for labeling content. Under the European Union (EU) Regulation 1379/2013, labels must provide the trade name, scientific name, production method, and catch/farming area for fishery and aquaculture products. However, the enforcement of these regulations relies on documentary trails, verified infrequently, and when verified, checked against the records kept by the very people being examined. The catch documentation schemes have been proposed as a more robust method [20], but they are subject to the same basic weakness of having the record and its owner inseparable. Thus, a verification problem emerges. Molecular methods, like DNA barcoding [21], allow establishing species identity reliably and remain the reference method for substitution detection. However, they cannot reveal the handling history, origin, and conditions of farming and transport of the product. For these properties, the record is the only evidence, making the integrity of the record the key factor.

Blockchain is one of several technologies applied to food systems under the fourth industrial revolution, alongside the internet of things, precision agriculture and machine learning, with provenance among its principal proposed applications [14]. The use of distributed ledger to achieve food provenance dates back 2017, when hazard analysis techniques were utilized alongside blockchain and radio frequency identification [22]. There have been further developments within the area with the inclusion of many products. Examples include soybean provenance using a public blockchain [23]; rice provenance [24], and an analogous technique for textiles [25]. Recent works have seen integration of smart contracts with distributed ledgers to facilitate agricultural food supply chain processes [26],[27]. A review of the literature found convergence in findings from these studies. For example, it was discovered that blockchain increases food traceability and operations within the supply chain but, most of the works have been framework proposals, not actual systems developed [15].

A review of blockchain use cases across global fishery and aquaculture value chains drew a similar conclusion, observing that there is a significant gap between proposals and reality [16]. A traceability system for sustainable seafood observed stakeholder coordination as the main implementation challenge [28]. Two criticisms recur. The first is empirical thinness, since reviews repeatedly note the scarcity of implemented and evaluated systems relative to conceptual proposals. The second is the adoption gap. Studies of practitioner uptake find that cost, integration difficulty, and coordination across stakeholders dominate the barriers to adoption, and these barriers fall hardest on small producers [29],[30]. The distributed ledgers address a narrower class of operational problems than the enthusiasm surrounding them suggests, and the clarity about which problem is being solved is a precondition for useful application [31]. Storing substantive traceability data directly on-chain is economically impractical at scale, leading to growing adoption of hybrid architectures in which detailed records are maintained off-chain while only their cryptographic digests are committed to the ledger. Comparative analyses of hybrid and fully on-chain designs show that the on-chain digest acts as a digital fingerprint: any alteration of the underlying off-chain record generates a different digest, enabling the modification to be detected [17]. A comparable architecture combining a content-addressed storage layer with a unified smart contract was shown to reduce gas consumption relative to multi-contract designs [27]. However, the security implications of this architecture are often under-examined. An on-chain digest guarantees integrity only after commitment; it does not establish data authenticity or protect

against pre-commitment manipulation. Nevertheless, such systems are often described as trustless or tamper-proof. This work therefore examines this distinction explicitly, since overstating the guarantee can undermine the trust the system is intended to establish.

Smart contracts lack any means for accessing off-chain data directly. Oracles provide the solution, but the utilization of oracles reintroduces centralization and counterparty risks that ledgers were intended to eliminate. The issues can be referred to as the oracle problem, and it is often neglected in applied blockchain research [32]. Decentralized oracle networks address the issue by gathering data from several different sources. An empirical survey of a widely used decentralized network found that price feeds represent its predominant use case [33]. The connection with traceability emerges in two related aspects. Sensor telemetry entering the provenance record is itself oracle input, and no cryptographic construction can establish that a sensor reported honestly. Deployed monitoring systems make the point concrete. In an instrumented growing system reporting temperature, humidity, pH, and light to a remote server, the reading traverses a low-cost sensor, a microcontroller, and an intermittent network link before any record exists, and connectivity failures at that stage are a reported operational reality rather than a hypothetical one [34]. Separately, a price oracle used for settlement introduces a distinct and more bounded dependency. The present work uses an oracle only for the second purpose and treats sensor trust as an explicit residual assumption rather than as a solved problem. Recurring smart-contract vulnerabilities include reentrancy, access-control failures, arithmetic errors, and unchecked external calls [35],[36]. Reentrancy remains a persistent source of loss, alongside other well-documented attack classes and mitigation patterns [37], [38]. Yet traceability studies that present contract code often omit security analysis. In payment-bearing deployments, where traceability contracts also function as financial contracts, this omission is significant. This work therefore evaluates the security of the proposed design explicitly

The review studies establish four key points:

1. Seafood fraud is persistent, and documentary controls have not resolved it.
2. Blockchain-based traceability is widely proposed but thinly implemented.
3. Hybrid storage is the accepted architecture, but its security semantics are under-specified.
4. Oracle and contract-level security receive little attention in applied traceability work.

The gap this paper addresses lies at the intersection of these issues. To the best of our knowledge, there is no reported system that combines a custodial access model suitable for producers with blockchain competence, a hash-linkage construction that propagates tamper-evidence across custody stages, fiat-denominated settlement through a price oracle, and an explicit statement of the resulting threat model. The remainder of the paper describes such a system.

2.0 Methodology

2.1 Design rationale

The architecture responds to three constraints that are characteristics of small-scale aquaculture, and each constraint drives a specific design decision. The first constraint is that participants are not blockchain-literate and cannot reasonably be expected to manage key material or to interpret transaction fees. The system therefore adopts a custodial signing model, in which participants authenticate to a conventional web application programming interface and the backend signs transactions on their behalf. This trades a measure of decentralisation for accessibility, and the consequences are analysed. The second constraint is that the evidentiary payload, which consists of telemetry sampled continuously across a harvest window, is far too large to store on chain economically. The system therefore adopts a hybrid storage model. The full record is serialised and stored off-chain, and only its digest is committed on-chain. The third constraint is that produce is priced in local fiat terms while settlement occurs in a volatile asset. The contract therefore denominates shipment value in United States dollars at eight decimal places and converts to the native asset at payment time using a decentralised price feed, accepting payment within a defined tolerance band. The resulting architecture, shown in Figure 1, comprises of four layers that separate sensing, application logic, storage, and settlement responsibilities.

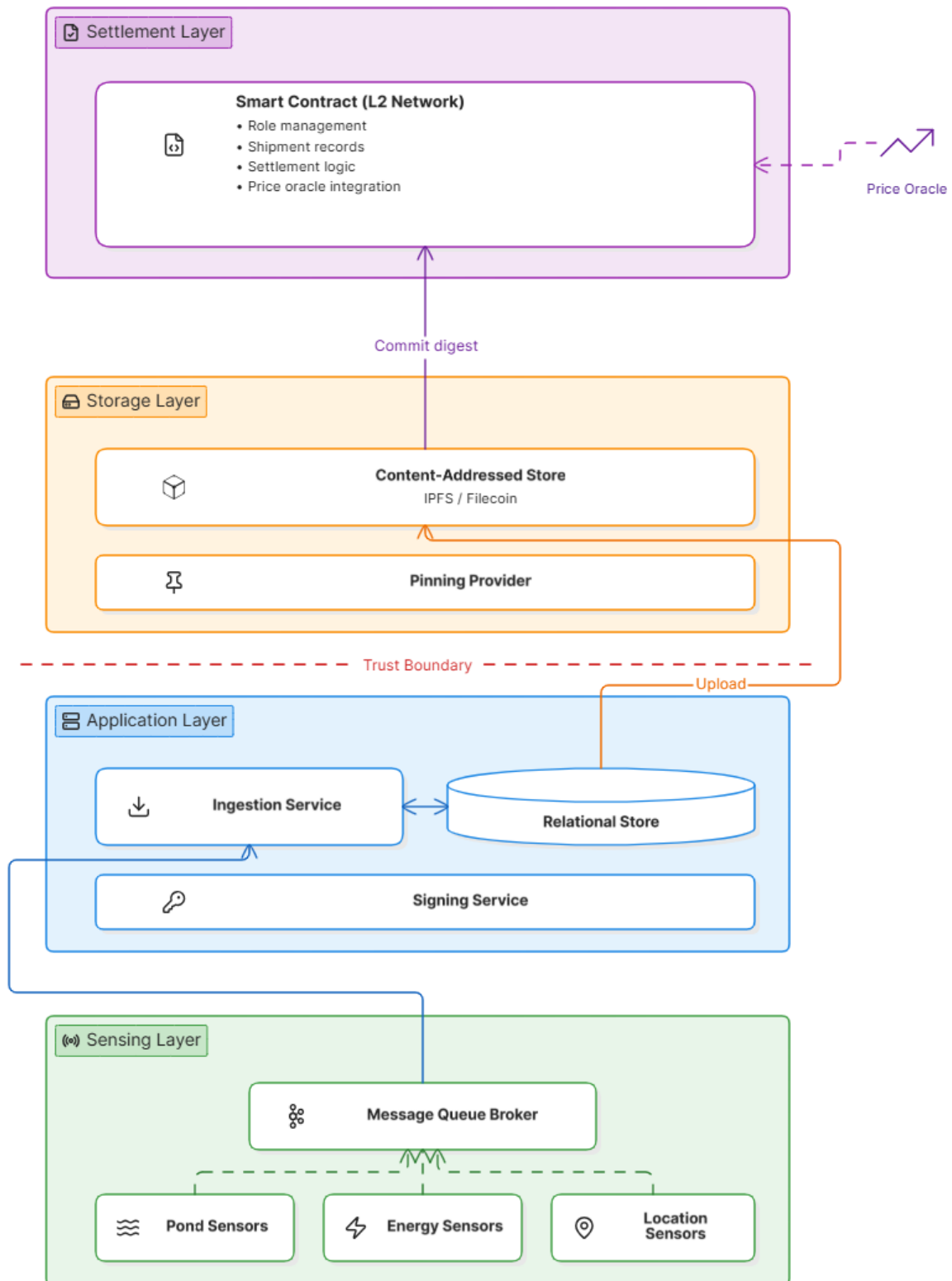


Figure 1: Four-layer system architecture

The trust boundary shown in Figure 1 separates the application layer from the storage and settlement layers, reflecting the custodial model.

2.2 Architecture

The system has four layers. The sensing layer consists of message-queue telemetry devices deployed at ponds and, where applicable, in transport. Devices publish to a managed broker over a transport-layer-secured connection on three topics, which carry pond telemetry, energy telemetry and location telemetry respectively. Pond readings carry temperature, relative humidity, acidity and dissolved oxygen, each with an associated timestamp.

Subscriptions use quality-of-service level one, which provides at-least-once delivery. This design places more importance on completeness over duplication elimination on the assumption that intermittent rural connectivity makes message loss a more likely failure mode than message duplication. The application layer is a web service that subscribes to the broker as a hosted background process, persists readings through an object-relational mapper to a relational store, and pushes live updates to connected clients over a persistent socket connection. Readings are associated with ponds through a sensor-to-pond mapping held in the database rather than through the structure of the topic hierarchy, which allows a flat topic hierarchy to serve an arbitrary number of farms. The storage layer is a content-addressed distributed file system, with pinning delegated to a managed provider. Self-hosting was evaluated and rejected. Continuous availability of the evidentiary document is a functional requirement, because verification fails outright if the document cannot be retrieved, and the operational cost of guaranteeing that availability independently exceeded the constraints of the project. The residual centralisation that this introduces is also addressed. The settlement layer is a smart contract deployed to an Ethereum-equivalent Layer 2 network. Selection was made based on transaction cost observed informally during development relative to the Ethereum main network and to a comparable Layer 2 network. No controlled cross-chain benchmark was conducted, and the selection should therefore be read as an engineering judgement rather than as an empirical finding.

2.3 Shipment creation and notarisation

A producer submits a harvest declaration that identifies a pond and a time window. The service first verifies pond ownership against the authenticated identity and rejects the request otherwise. It then retrieves all sensor readings for that pond within the declared window and assembles a single structured document containing two members. The first member holds shipment metadata, which comprises farm identity, harvest time, weight, price, species, harvest method, feed type, feeding schedule, and fish count. The second member holds the ordered sensor log. This document is then forwarded to the storage layer that produces a content ID. The same serialized document is then hashed with Keccak-256, which is the native hash function of the Ethereum Virtual Machine, and the hash is then provided on-chain along with the price in fiat currency. On successful transaction execution, the following information will be recorded by the service locally: content ID, shipment ID on-chain, and transaction hash. The sequence of operations from producer submission to on-chain notarisation is illustrated in Figure 2.

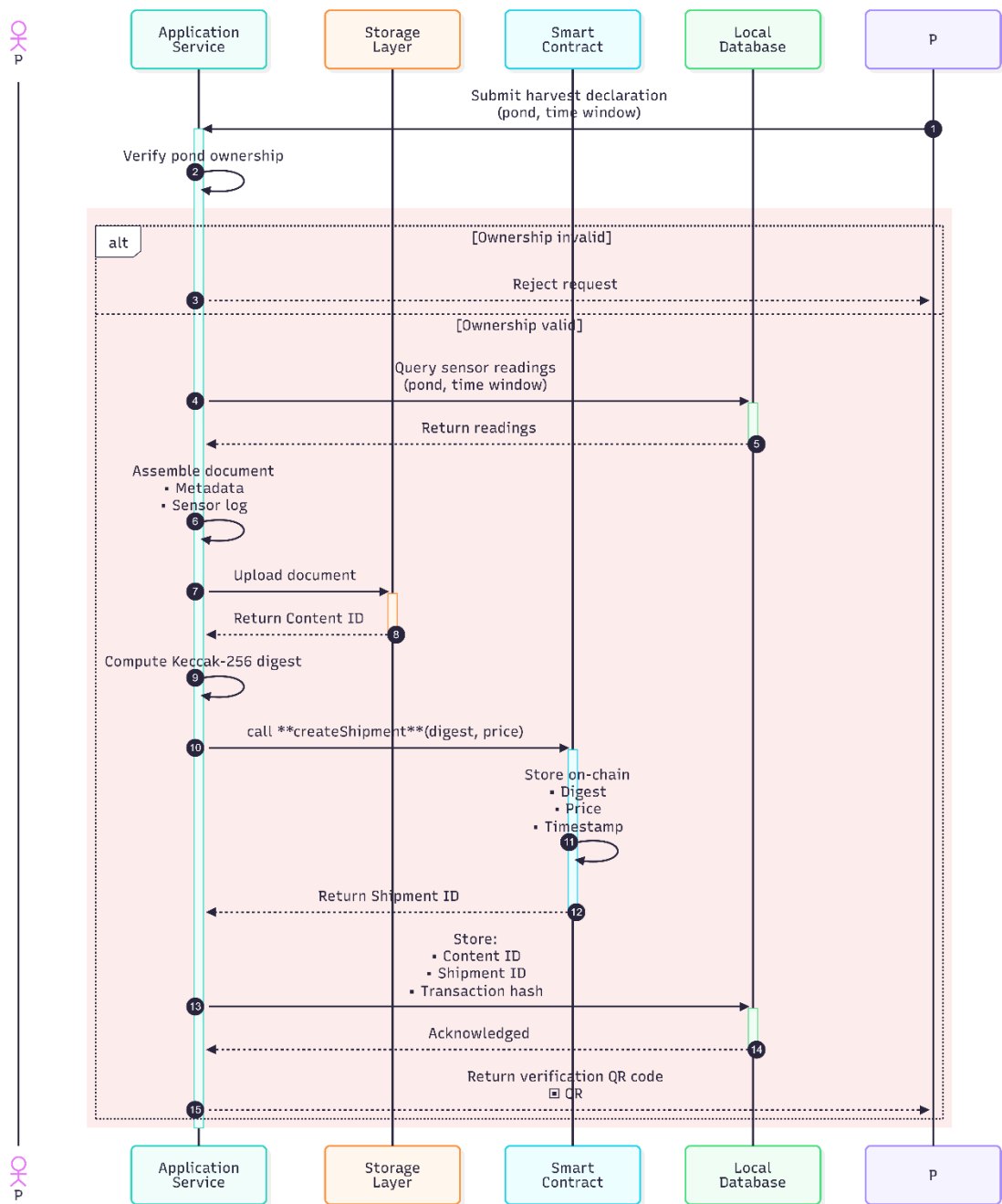


Figure 2: Shipment creation and notarisation sequence.

This sequence ensures that every provenance record is independently verifiable through the content identifier and on-chain commitment.

There are two specific characteristics of this design, which relate to the strength of the guarantee and which are stated explicitly. One of them is canonicalisation. The hash digest is computed for the exact byte sequence output by the serializer and then verified by computing the hash again, based on the document retrieved from the storage gateway. Since the storage is content-addressable, the retrieved bytes are equal to the uploaded ones and the check can be performed successfully. Still, this design depends on consistent serialization of the underlying data because any entity that re-serializes the document by changing either its order of members or its whitespaces will produce a new digest and therefore will produce a false negative result. This problem is avoided in this implementation by not having re-serialization at all, but if implemented with such a feature, the system requires a canonical form. The other characteristic relates to the storage encoding. The hash digest is stored on the chain as a variable-length string instead of being stored as a thirty-two-byte fixed-size field. Hexadecimal representation of sixty-four characters takes three fields (one length field and two data fields) in contrast with the raw representation, which takes only one field.

2.4 Multi-stage custody and hash linkage

Processing represents a distinct phase of custody. Whenever the processor documents a phase, the system creates a second document with information about the processing time, method of processing, type of packaging, storage conditions, number of packages, and processor identity. Importantly, the new document will also have the content identifier from the previous farm-phase document as a field. Because the content identifier itself is a digest of the farm phase document, the processor document is tied to the farm phase document not by a key but by content. Hence, each phase document becomes a link in a hash chain terminating with the harvest record, and all the links are separately committed to the chain. This construction ensures that any tampering propagates along the chain. An attacker who tampers with the harvest telemetry will change the farm phase document and thus its content identifier, which in turn will make the embedded reference in the processor phase document incorrect and will change its digest. Figure 3 demonstrates how this hash-linkage construction causes a single alteration at the farm stage to invalidate every downstream commitment.

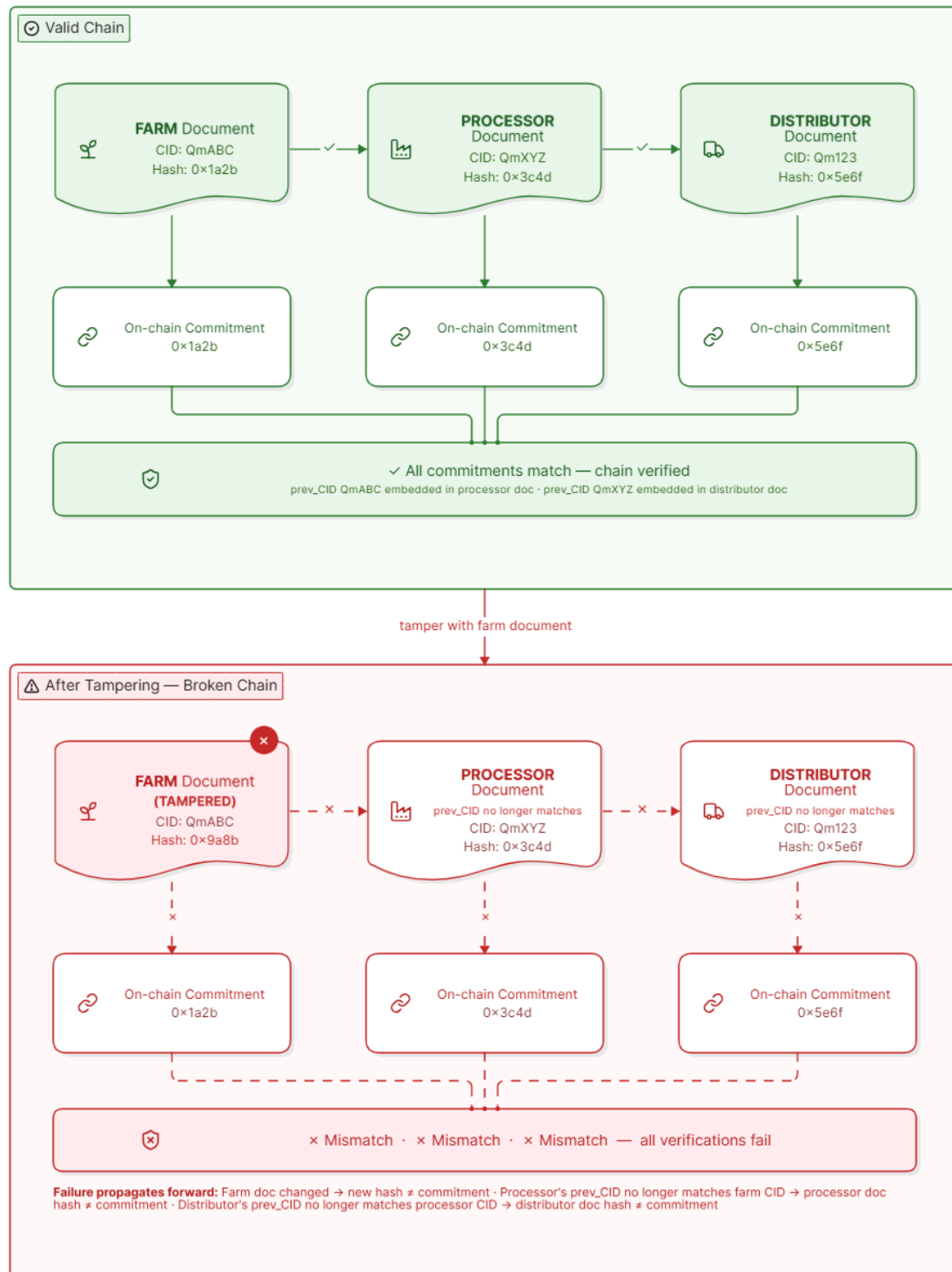


Figure 3: Forward propagation of tamper-evidence across custody stages

The broken chain in the lower half of Figure 3 illustrates why the construction prevents undetectable retroactive modification.

2.5 Access control

There are four roles defined by the contract: unassigned, producer, intermediary, and consumer. The deploying address is hardcoded as an immutable administrator that is the only actor able to assign roles. Role assignment prevents using the zero address, the self-assignment of the administrator, and assigning any role above the maximum valid role. Those three prevent the use of straightforward paths for privilege escalation and accidental lockout. There are two modifiers controlling the other surface. First, it limits all the actions to addresses with the assigned role. Second, it limits the creation of records to producers and intermediaries so that consumers cannot create records.

2.6 Settlement logic

Settlement represents the security-critical path. Validation takes place in the payment function before all the state changes happen. The payment function checks if the shipment exists, hasn't been paid for, the caller doesn't own it, and the caller hasn't created it. After that, it reads the rate from the price feed and calculates the amount of the native asset needed for the payment as the fiat price multiplied by 10^{18} and divided by the read rate. Payments can take place within one percent of tolerance in both directions. The tolerance band allows some price change that happens between client quotation and transaction execution, limiting the magnitude of mispricing transferred in either direction. All state changes happen in an atomic manner. The old owner is noted, the ownership and owner role are changed, the paid flag is turned on, and the payment is added to the seller's internal balance. No external call is made during this function. Sellers retrieve funds through a separate withdrawal function that zeroes the caller's balance before transferring, which is the pull-payment pattern and an ordering that closes the reentrancy path that would otherwise exist. The contract additionally reverts on both the receive and the fallback paths, so that native asset sent directly to the contract address is rejected rather than trapped. Error handling uses thirteen typed custom errors rather than revert strings. Beyond the reduction in gas cost, this allows the backend to decode the four-byte error selector returned by a failed call and to surface a specific and human-readable explanation to the user, which is a practical necessity given that the intended operators do not read blockchain error output. The lifecycle of a shipment through the contract's state transitions is visualized in Figure 4.

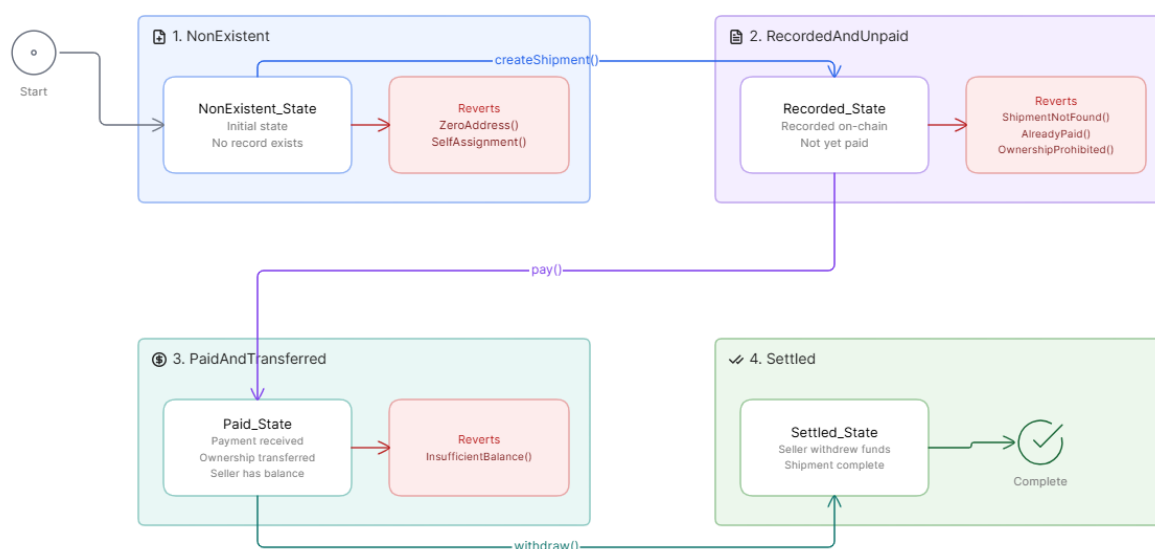


Figure 4: Contract state machine for a shipment

The state machine in Figure 4 captures the four distinct stages of a shipment's lifecycle, with guard conditions enforcing the business logic.

2.7 Verification

Each shipment yields a quick-response code that encodes a verification address parameterised by the on-chain shipment identifier and the storage content identifier. The code is generated at error-correction level Q, which tolerates approximately twenty-five per cent occlusion, on the assumption that labels applied to chilled or handled produce will degrade in transit. Resolving the address triggers the verification routine. The document is retrieved from the storage gateway, its Keccak-256 digest is computed, the shipment record is read from the contract, and

the two digests are compared. A match resolves to the document itself and a mismatch returns an explicit failure. Verification is performed on the server, which means that the verifying party is trusted to perform the comparison faithfully. Because both inputs are independently and publicly retrievable, however, any third party can reproduce the comparison without the cooperation of the platform.

3.0 Results and Discussion

3.1 Threat model and security analysis

The threat model comprises four adversary types. A dishonest supply-chain participant seeks to alter a provenance record after publication, or to publish a record that misrepresents handling conditions. A dishonest platform operator controls the backend, the database, and the custodial keys. An external network attacker can intercept or inject traffic between components. A malicious buyer or seller seeks to extract value through the settlement path. It is assumed that Keccak-256 is collision-resistant and preimage-resistant, that the underlying chain provides transaction finality and censorship resistance consistent with an Ethereum-equivalent Layer 2 network, and that the price oracle is not fully compromised. Sensor hardware is not assumed to be honest.

3.2 Retroactive modification of records

This represents the primary threat that the design seeks to counteract. A hacker who modifies an evidentiary document that has been made public changes its digest, but the commitment to the document on the blockchain is left untouched and, as per the presumption of irreversibility, is not amenable to modification. Verification hence is bound to fail. So far, the document's identifier is a digest; the hacker is also incapable of replacing it at its address on the blockchain. The guarantee is precise and worth stating precisely. The system establishes that a record has not changed since the moment of commitment. It does not establish that the record was accurate at that moment. This distinction is frequently blurred in the traceability literature, and blurring it overstates what any such system can deliver.

3.3 Falsified sensor input

A participant who controls sensor hardware can report fabricated readings, and those readings will be faithfully committed. No cryptographic mechanism in this architecture detects that. The system converts a record that could be altered at any time into a record that is fixed at the moment of publication, which shifts the adversary's problem from undetectable retroactive revision to falsification in advance and in real time. It also creates a durable artefact against which later physical or laboratory evidence, such as molecular species verification, can be contrasted. That is a meaningful narrowing of the attack surface rather than a solution to the problem, and mitigations that require attestation or signing at the sensor are identified.

3.4 Compromise of the platform operator

The custodial model centralizes authority. A participant key holder can generate records for any participant, while a server-side verifier can declare a successful comparison regardless of what the comparison is. There are two restrictions that limit the extent of the possible harm from the above design. First, all the commitments are public and therefore allow another third party to fetch the document and on-chain commitment and redo the comparison independent of the operator. This means that a dishonest verifier can be found out by anybody who tries to verify. Second, the on-chain entry is append-only and hence an operator can fake new records but not silently alter existing ones. Therefore, any wrongdoing is always documented. However, this is still the major problem with the design, and the proper way to characterize it is to say that the protocol guarantees tamper-evidence of the supply chain but not of the platform itself. Non-custodial signing and client-side verification eliminate the above problems and are considered high-priority tasks for further development.

3.5 Settlement-path attacks

Reentrancy is a well-established smart-contract vulnerability that has been formally characterized and continues to be reported as a persistent security concern [35, 37]. The present design mitigates it structurally rather than defensively. The payment function makes no external call, and the withdrawal function zeroes the caller's balance before transferring, so that an attacker who re-enters the withdrawal observes a zero balance and reverts. The separation of payment from withdrawal also prevents a seller from blocking settlement through a reverting receive path, because payment credits an internal balance rather than transferring immediately. Access control is enforced through role checks on every state-changing function. Direct transfers are rejected on both the receive and the fallback paths, which prevents the balance of the contract from being manipulated independently of the accounted internal balances. Oracle manipulation is the residual settlement risk. The one per cent acceptance band bounds per-transaction exposure to error in the reported rate, but a sustained oracle compromise would mis-denominate all settlements. This dependency is inherent to any on-chain contract that references off-chain price data [32]. Two known limitations remain. The administrator role is immutable and unrecoverable, so loss of the

deploying key permanently disables role assignment. Role assignment is also a single-signature action, so compromise of the administrator permits arbitrary participant admission. Multi-signature administration and a role-revocation path are identified.

3.6 Scope of the present evaluation

The analysis presented is a design-level security argument. No controlled empirical evaluation of the deployed system is reported in this work. Specifically, we do not report measured gas costs, end-to-end latency distributions, throughput under load, or the results of automated vulnerability scanning or formal verification of the contract. The system has been implemented and deployed and the implementation is available for inspection, but performance characterisation and empirical security assessment remain future work.

3.7 Discussion

The central design claim of this paper is that tamper-evidence, rather than decentralisation, is what small-producer traceability requires, and that the two can be separated. A producer with no blockchain literacy, no key material and no capacity to pay transaction fees nonetheless obtains a publicly verifiable and append-only provenance record. The cost of that arrangement is a trusted operator in the write path. For a producer whose current alternative is a paper ledger, this remains a substantial improvement in evidentiary standing. The framing that matters is therefore not trustless against trusted, but rather which trust assumptions are being made and whether those assumptions are visible to the parties who rely on them. The forward-propagating hash linkage has an implication worth drawing out. Because alteration at any stage invalidates every downstream commitment, a verifier at the retail end detects tampering that occurred at the pond without needing to know that a farm-stage record exists or where to locate it. Verification effort is therefore constant in the number of custody stages while detection coverage spans all of them. This property matters in supply chains where the verifying party has no relationship with, and no visibility into, upstream actors, which is the ordinary condition of seafood retail.

Oracle-denominated settlement addresses an objection that is routinely raised against commerce settled in volatile assets, and that is rarely answered in the traceability literature. A producer who agrees a price at harvest and receives payment several days later is otherwise exposed to exchange-rate movement across that interval. Denominating in fiat and converting at payment time transfers that exposure to the holder of the volatile asset rather than to the producer. The cost is the oracle dependency, and the trade is favourable in this specific direction, because the party least able to hedge is the party relieved of the exposure. A final point of interpretation concerns what the system contributes to the fraud problem. It does not detect species substitution, and molecular methods remain necessary for that purpose. What it provides is a fixed and independently checkable claim about handling history, against which physical evidence can later be compared. Where certification schemes reduce mislabeling by attaching an audited claim to a product, the present architecture reduces the cost of producing a claim that is durable enough to audit.

4.0 Conclusion

Custodial key management is the principal architectural limitation. Signing authority resides with the backend, which bounds the guarantee of the system to tamper-evidence against the supply chain rather than against the platform. Migration to non-custodial signing through account abstraction with sponsored transactions would preserve the fee-free user experience while returning signing authority to participants. This is the highest-priority direction for further work. Server-side verification should be moved to the client. A browser-based verifier that retrieves the document, computes the digest locally and reads the contract through a public remote-procedure-call endpoint would eliminate the trusted comparison step entirely, using inputs that are already public. Sensor trust remains an open assumption. Telemetry is trusted at the point of capture, as discussed. Cryptographic signing at the sensor, ideally with hardware-backed key storage, would extend the integrity boundary from the backend outward to the device itself. Storage centralisation is a further limitation. Pinning is delegated to a managed provider. Content addressing means that the provider is unable to alter documents undetectably, but it is able to withhold them and withholding causes verification to fail. Redundant pinning across independent providers, or a storage arrangement with an on-chain availability commitment, would reduce this exposure. Storage encoding is inefficient. The digest is committed as a hexadecimal string rather than as a fixed-width value, which consumes additional storage slots for every record, as noted. This is straightforward to correct and would reduce the per-record cost of the system. Administrative control constitutes a single point of failure. The administrator is immutable and single signature. Multi-signature administration together with a role-revocation path is needed before production deployment at scale. Empirical evaluation is the most significant remaining gap. The work to be done involves gas utilization measurements per contract operation with fiat-based costs, latency distribution measurement from sensor ingest through storage upload to transaction confirmation, tamper resistance tests measuring detection over an adversarial set of samples, automated security testing as well as formal verification of

the contract, and a field study involving the producers to test usability and adoption. Additionally, an experimental comparison of cross-chain costs is needed to complement the non-scientific platform choice made.

The small-scale producers who produce their aquatic products face an evidentiary barrier rather than a production constraint. These producers do not have the capability of meeting the handling history standards of the certified market, and the evidence they have does not withstand adversarial scrutiny. The paper proposes an architecture for making the provenance records of these small-scale producers tamper-evident using cryptographic commitment without requiring any expertise in blockchains from the producer. In this design, there is a custody chain with hashes to create tamper-evidence that travels throughout the chain such that any tampering done at any stage will be noticed at all downstream stages. There is also the oracle-denominated settlement that removes any exposure to exchange rates for the producer. Furthermore, this design presents a clear threat model that separates what cryptographic commitment can prove and what it cannot prove. The former is that the provenance record has not been tampered with since it was created, and the latter is the correctness of the record when it was created. This is because the traceability literature often mixes up these two, and any architecture that claims to deliver more than its promises creates more problems than solving the existing ones.

References

- [1] A. Munguia-Vega, R. Terrazas-Tapia, J. F. Dominguez-Contreras *et al.*, "DNA barcoding reveals global and local influences on patterns of mislabeling and substitution in the trade of fish in Mexico," *PLoS ONE*, vol. 17, no. 4, p. e0265960, 2022.
- [2] G. M. Luque and C. J. Donlan, "The characterization of seafood mislabeling: A global meta-analysis," *Biological Conservation*, vol. 236, pp. 556-570, 2019.
- [3] C. J. Donlan and G. M. Luque, "Exploring the causes of seafood fraud: A meta-analysis on mislabeling and price," *Marine Policy*, vol. 100, pp. 258-264, 2019.
- [4] M. Fox, M. Mitchell, M. Dean *et al.*, "The seafood supply chain from a fraudulent perspective," *Food Security*, vol. 10, no. 4, pp. 939-963, 2018.
- [5] B. I. Crona, T. M. Daw, W. Swartz *et al.*, "Masked, diluted and drowned out: How global seafood trade weakens signals from marine ecosystems," *Fish and Fisheries*, vol. 17, no. 4, pp. 1175-1182, 2016.
- [6] K. Kroetz, G. M. Luque, J. A. Gephart *et al.*, "Consequences of seafood mislabeling for marine populations and fisheries management," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 117, no. 48, pp. 30318-30323, 2020.
- [7] O. A. Ogbolumani and N. I. Nwulu, "Environmental impact assessment for a meta-model-based food-energy-water-nexus system," *Energy Reports*, vol. 11, pp. 218-232, 2024.
- [8] O. A. Adelaja, R. B. Kamaruddin, and L. W. Chiat, "Assessment of post-harvest fish losses Croaker *Pseudotolithus elongatus* (Bowdich, 1825), Catfish *Arius heudeloti* (Valenciennes, 1840) and Shrimp *Nematopalaemon hastatus* (Aurivillius, 1898) in Ondo State, Nigeria," *Aquaculture and Fisheries*, vol. 3, no. 5, pp. 209-216, 2018.
- [9] A. L. Abelti and T. A. Teka, "Intervening fish post-harvest losses to narrow the gap between demand and supply: A review on magnitude of fish post-harvest losses in some Sub-Saharan African countries," *Aquaculture, Fish and Fisheries*, vol. 4, no. 2, p. e168, 2024.
- [10] O. A. Ogbolumani and N. Nwulu, "Integrated appliance scheduling and optimal sizing of an autonomous hybrid renewable energy system for agricultural food production," in *Advances in Manufacturing Engineering: Selected Articles from ICMMPPE 2019*, S. S. Emamian, F. Yusof, and M. Awang, Eds. Singapore: Springer, 2020, pp. 651-660.
- [11] O. A. Ogbolumani and N. I. Nwulu, "Multi-objective optimisation of constrained food-energy-water-nexus systems for sustainable resource allocation," *Sustainable Energy Technologies and Assessments*, vol. 44, p. 100967, 2021.
- [12] O. A. Ogbolumani and N. I. Nwulu, "A food-energy-water nexus meta-model for food and energy security," *Sustainable Production and Consumption*, vol. 30, pp. 438-453, 2022.
- [13] J. Barendse, A. Roel, C. Longo *et al.*, "DNA barcoding validates species labelling of certified seafood," *Current Biology*, vol. 29, no. 6, pp. R198-R199, 2019.
- [14] O. Ogbolumani and N. Nwulu, *Food, Energy, and Water Nexus 4.0: Concepts and Applications*, 1st ed. Boca Raton, FL, USA: CRC Press, 2025.
- [15] R. M. Ellahi, L. C. Wood, and A. E.-D. A. Bekhit, "Blockchain-based frameworks for food traceability: A systematic review," *Foods*, vol. 12, no. 16, p. 3026, 2023.
- [16] F. Tolentino-Zondervan, P. T. A. Ngoc, and J. L. Roskam, "Use cases and future prospects of blockchain applications in global fishery and aquaculture value chains," *Aquaculture*, vol. 565, p. 739158, 2023.
- [17] M. J. Fernández-Iglesias, C. Delgado von Eitzen, and L. Anido-Rifón, "Efficient traceability systems with smart contracts: Balancing on-chain and off-chain data storage for enhanced scalability and privacy," *Applied Sciences*, vol. 14, no. 23, p. 11078, 2024.

- [18] P. Olsen and M. Borit, "How to define traceability," *Trends in Food Science & Technology*, vol. 29, no. 2, pp. 142-150, 2013.
- [19] M. Á. Pardo, E. Jiménez, and B. Pérez-Villarreal, "Misdescription incidents in seafood sector," *Food Control*, vol. 62, pp. 277-283, 2016.
- [20] G. Hosch and F. Blaha, *Seafood Traceability for Fisheries Compliance: Country-Level Support for Catch Documentation Schemes*, FAO Fisheries and Aquaculture Technical Paper No. 619. Rome, Italy: Food and Agriculture Organization of the United Nations, 2017.
- [21] P. D. N. Hebert, A. Cywinska, S. L. Ball *et al.*, "Biological identifications through DNA barcodes," *Proceedings of the Royal Society B: Biological Sciences*, vol. 270, no. 1512, pp. 313-321, 2003.
- [22] F. Tian, "A supply chain traceability system for food safety based on HACCP, blockchain & Internet of Things," in *2017 International Conference on Service Systems and Service Management (ICSSSM)*, Dalian, China, 2017, pp. 1-6.
- [23] K. Salah, N. Nizamuddin, R. Jayaraman *et al.*, "Blockchain-based soybean traceability in agricultural supply chain," *IEEE Access*, vol. 7, pp. 73295-73305, 2019.
- [24] B. M. Yakubu, R. Latif, A. Yakubu *et al.*, "RiceChain: Secure and traceable rice supply chain framework using blockchain technology," *PeerJ Computer Science*, vol. 8, p. e801, 2022.
- [25] T. K. Agrawal, V. Kumar, R. Pal *et al.*, "Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry," *Computers & Industrial Engineering*, vol. 154, p. 107130, 2021.
- [26] P. Saranya and R. Maheswari, "Proof of Transaction (PoTx) based traceability system for an agriculture supply chain," *IEEE Access*, vol. 11, pp. 10623-10638, 2023.
- [27] A. Sharma, T. Bhatia, A. Sharma *et al.*, "Secure and traceable decentralized agri-food supply chain framework using Ethereum blockchain and IPFS platform," *Transactions on Emerging Telecommunications Technologies*, vol. 36, no. 7, p. e70188, 2025.
- [28] A. Shamsuzzoha, J. Marttila, and P. Helo, "Blockchain-enabled traceability system for the sustainable seafood industry," *Technology Analysis & Strategic Management*, vol. 36, no. 11, pp. 3891-3905, 2024.
- [29] R. Cole, M. Stevenson, and J. Aitken, "Blockchain technology: Implications for operations and supply chain management," *Supply Chain Management: An International Journal*, vol. 24, no. 4, pp. 469-483, 2019.
- [30] D. Roeck, H. Sternberg, and E. Hofmann, "Distributed ledger technology in supply chains: A transaction cost perspective," *International Journal of Production Research*, vol. 58, no. 7, pp. 2124-2141, 2020.
- [31] V. Babich and G. Hilary, "OM Forum—Distributed ledgers and operations: What operations management researchers should know about blockchain technology," *Manufacturing & Service Operations Management*, vol. 22, no. 2, pp. 223-240, 2020.
- [32] G. Caldarelli, "Understanding the blockchain oracle problem: A call for action," *Information*, vol. 11, no. 11, p. 509, 2020.
- [33] M. Kaleem and W. Shi, "Demystifying Pythia: A survey of ChainLink oracles usage on Ethereum," in *Financial Cryptography and Data Security: FC 2021 International Workshops—CoDecFin, DeFi, VOTING, and WTSC*. Springer, 2021, pp. 115-123.
- [34] O. A. Ogbolumani and B. Mabaso, "An IoT-based hydroponic monitoring and control system for sustainable food production," *Journal of Digital Food Energy & Water Systems*, vol. 4, no. 2, pp. 106-140, 2023.
- [35] S. S. Kushwaha, S. Joshi, D. Singh *et al.*, "Systematic review of security vulnerabilities in Ethereum blockchain smart contract," *IEEE Access*, vol. 10, pp. 6605-6621, 2022.
- [36] H. Chu, P. Zhang, H. Dong *et al.*, "A survey on smart contract vulnerabilities: Data sources, detection and repair," *Information and Software Technology*, vol. 159, p. 107221, 2023.
- [37] L. Luu, D.-H. Chu, H. Olickel *et al.*, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, Vienna, Austria, 2016, pp. 254-269.
- [38] S. Sayeed, H. Marco-Gisbert, and T. Caira, "Smart contract: Attacks and protections," *IEEE Access*, vol. 8, pp. 24416-24427, 2020.