

Article

Who Tracks What, and How Deep? Granular Traceability in Multi-Firm Supply Chains via Blockchain-Based Tokenization

Vangelis Malamas ^{1,*}, Georgios Lagoumitzis ², Panayiotis Kotzanikolaou ^{2,*}, Theodore G. Voutsinas ¹ and Thomas K. Dasaklis ¹

¹ School of Social Sciences, Hellenic Open University, 26331 Patras, Greece; voutsinas.theodoros@ac.eap.gr (T.G.V.); dasaklis@eap.gr (T.K.D.)

² Department of Informatics, University of Piraeus, 18534 Piraeus, Greece; geolag@hotmail.gr

* Correspondence: bagmalamas@eap.gr (V.M.); pkotzani@unipi.gr (P.K.)

Abstract

Background: Traceability has gained significant importance in supply chain (SC) networks, especially driven by the growing need for transparency, regulatory compliance and consumer assurance. Although blockchain technology addresses several traditional traceability challenges, granularity, interoperability and integration with existing Enterprise Resource Planning (ERP) systems remain open problems. **Methods:** In this paper, we propose an integrated blockchain-enabled framework aimed at granular traceability in SC networks. The proposed framework supports the tokenization of the Bill of Materials (BoM) and introduces append-only versioned tokens that capture state evolution across SC stages. Security and privacy are managed through a baseline Role-Based Access Control (RBAC) policy, while EPCIS GS1 event modeling supports standardized data exchange between ERP middleware and the blockchain layer. To examine its applicability, we implement a proof-of-concept prototype and apply the framework to an illustrative wheat SC scenario. **Results:** Finally, we provide a controlled performance and feasibility analysis based on a private Ethereum/Ganache environment. **Conclusions:** The results should be interpreted as proof-of-concept evidence of technical feasibility rather than as production-grade validation.

Keywords: supply chains (SC); granular traceability; blockchain; tokens; Enterprise Resource Planning (ERP)

1. Introduction

Supply chain (SC) traceability has attracted significant attention during the last years. SC traceability assures the growing consumer demands for transparency and authenticity, especially in globalized SC networks [1]. Its importance could be attributed to several factors, such as the need for regulatory compliance, quality assurance, consumer safety, and protection from counterfeit products and SC attacks. All these requirements are further exemplified in the case of safety-sensitive sectors such as food or pharmaceuticals, where traceability is of critical importance [2]. A key aspect of SC traceability systems is the supported granularity level of traceable units. From a theoretical point of view, granularity refers to the level of detail at which events are recorded and products are identified through a traceability system across the SC. Practically, granularity answers ‘how fine or coarse the trace is’: for example, item vs. batch, step vs. process, address vs. city, and hourly vs. daily updates. Arguably, finer granularity means smaller traceable units across the SC, more precise event links, richer context, and more frequent updates for the information



Academic Editors: Samudaya Nanayakkara, Srinath Perera and Kasuni Vidanagamachchi

Received: 8 May 2026

Revised: 23 June 2026

Accepted: 29 June 2026

Published: 6 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

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

attributed to traceable goods, at the cost of handling larger volumes of data. A useful way to understand granularity in the context of SC networks is across core identification elements and their levels, such as stakeholder, process, traceable asset, transport, location, etc. [1]. Formally, some frameworks proposed in the literature quantify granularity by combining precision, breadth and depth into indicators such as external/internal trace units, one-to-one vs. many-to-many ID conversions, information content, update frequency, and forward/backward tracking distance [1]. An important aspect in SC granularity is the trade-off between the cost incurred (resources required) and the level of granularity (information precision). Obviously, finer granularity means smaller units, richer quality of traceability data, stronger process control, and the ability for range-limited recalls that improve product safety [3]. At the same time, it leads to increased costs in terms of data capture, storage, system maintenance, and staff workload [4].

Traceability relies on sound information exchange mechanisms in SC networks; thus, information sharing and interoperability of traceability-related information remain pivotal in SC contexts [5]. Since traceability and visibility (real-time situational awareness) are both based on information sharing, without shared information across actors, neither accurate backward tracking nor forward tracking is possible in SC networks [6]. In addition, information sharing reduces transaction costs, aligns decision-making and enables better coordination of time-sensitive processes, especially for traceability purposes [7]. Enterprise Resource Planning (ERP) systems provide a transaction-based, integrated platform for synchronizing orders, inventory and shipments. This structured integration enables traceability by ensuring consistency of identifiers as lots and batches and by serving as an operational system of record within each organization. By embedding traceability data (batch numbers, dates, and quality attributes) in ERP workflows, firms can optimize replenishment, reduce bullwhip effects, and quantify the monetary value of policy coordination [8]. Therefore, traceability without information sharing is meaningless, as data must flow upstream and downstream of the SC to reconstruct the product's path and make quick and cost-effective decisions in recalls, quality assurance and compliance.

Recent empirical studies further support the strategic relevance of blockchain-enabled traceability and digital information processing in supply-chain contexts. Rashid et al. [9] shows, using structural equation modeling, that blockchain-enabled supply-chain practices positively affect supply-chain performance through the mediating effects of supplier trust, traceability and transparency. Similarly, [10] demonstrate that information processing capability and digital supply-chain practices positively influence supply-chain risk management and resilience. These findings reinforce the view that traceability should not be treated only as a technical recording mechanism but also as a broader information-processing and trust-building capability that supports coordination, risk management and supply-chain performance.

Motivation: Despite the widespread adoption of ERP systems, end-to-end granular traceability and cross-organizational interoperability remain open challenges, especially in complex and globalized SC networks. Heterogeneous data schemes, the lack of global identifiers, asynchronous state updates, and weak auditability hinder accurate tracking across stages and complicate data exchange among disparate ERPs. Blockchain technology has emerged as a key enabler for the establishment of sound SC traceability mechanisms [11], as it may support traceability of information exchange and interoperability among different ERP systems.

Blockchain technology is particularly effective in managing trusted information sharing in modern SCs [12]. Blockchain-enabled traceability provides operational assurance across SC by recording transactions and product movements in an immutable shared ledger that all stakeholders can verify in real time, strengthening provenance, authenticity and

regulatory compliance while reducing fraud and manipulation [13–15]. The transparency achieved by blockchain-enabled applications in SC networks improves accountability and helps assign responsibility for failures, mitigating moral hazards in logistics and elevating delivery standards. It also supports sustainability reporting by monitoring environmental and social impacts throughout the SC [16]. Empirically and from an operational perspective, these characteristics lower recall costs, improve credibility with consumers and increase the efficiency of coordination among partners [14,17]. However, despite recent advances, most blockchain-based SC traceability implementations fail to take into account aspects of granularity, a key determinant for reliable and operationally meaningful traceability solutions in SC networks [1].

Contribution: To address these granularity and interoperability limitations, this paper proposes an integrated blockchain-enabled traceability framework that supports standardized, middleware-mediated data exchange between ERP systems and a blockchain-based integrity layer. The approach leverages a Bill of Materials (BOM)-driven tokenization model, where append-only versioned tokens represent components, stakeholders and processes across successive supply-chain stages. This design supports high-resolution traceability from raw materials to the final product while limiting on-chain duplication by anchoring references and hashes rather than full operational payloads. The framework enforces a baseline Role-Based Access Control (RBAC) policy and integrates the EPCIS/GS1 standard to support semantic interoperability across the supply-chain network.

The specific key contributions of this work are as follows:

- *A BOM-Driven Token Architecture for Granular Traceability:* We introduce a multi-layer model utilizing three token classes—Stakeholder (SH), Product (PD), and Process (PC)—minted per transformation stage k (where $k = 1 \dots K$ represents the sequential steps in the supply chain). These tokens are cryptographically linked and synthesized into a final synthesized token (ST_X), establishing a granular, state-aware, and composable provenance record.
- *Middleware-Supported ERP Interoperability and Data Integrity:* We define an integration pipeline using ExpressJS-based middleware APIs, Prisma ORM for database abstraction, and EPCIS/GS1 event standardization to reconcile data between simulated ERP systems and the blockchain layer. This validates the architectural integration pattern, while direct integration with commercial ERP platforms remains future work.
- *An Append-Only Versioned Smart-Contract Blueprint:* We develop and deploy a suite of smart contracts on private Ethereum (Ganache) that support event-driven, append-only updates with historical verifiability. This preserves previous token states and provides RBAC-governed access without destructive state modifications.
- *Proof-of-concept Validation Through an Illustrative Wheat Supply-Chain Scenario:* We demonstrate the technical feasibility of the framework through a wheat supply chain scenario and controlled prototype measurement. The evaluation confirms deterministic rule execution and prototype-level synchronization, while production deployment, commercial ERP integration and large-scale benchmarking are identified as future work.

The remainder of this paper is structured as follows. Section 2 discusses in detail relevant blockchain-based traceability solutions in the broad agricultural domain along with critical technical determinants of blockchain-enabled solutions in SC traceability. Section 3 presents the proposed blockchain-enabled traceability framework using a use case scenario. Section 4 presents the implementation details of the proposed framework, along with performance and security analysis. Section 5 outlines the broad managerial implications of the proposed framework and further exemplifies its vital contribution compared to similar

approaches presented in the literature and also outlines several limitations as fruitful areas for future research. The paper ends with some concluding remarks and key takeaways.

2. Literature Review

We review two key streams in the literature for blockchain-based SC traceability mechanisms. The first stream covers blockchain-based traceability solutions in the wide agriculture domain, with a special focus on traceability solutions for seeds and crops. While our proposed framework is fundamentally domain-agnostic, we place special emphasis on the agricultural and food supply chains, as they represent one of the most demanding environments for granular traceability and serve as the primary case study for our evaluation. The second stream examines the critical determinants of blockchain-enabled solutions, such as the use of tokens, granularity frameworks and the EPCIS standard.

2.1. Blockchain-Enabled Traceability Studies in Wheat/Grains SC Management

In the case of seed SC traceability, the authors in [18] present SeedChain, a blockchain-driven framework that leverages Ethereum blockchain technology to create a decentralized application (DApp) for tracking seeds from breeders to farmers. Some of the key technical aspects include the use of Non-Fungible Tokens (NFTs) linked to approved seeds, which enable farmers to verify the origin and quality of their purchases. In the case of soybean SC, [19] addresses the issue of asymmetric information by proposing a blockchain solution. The authors use smart contracts to automate the agreement, reducing the reliance on intermediaries and increasing the efficiency of the soybean SC. In the case of SC crop production, [20] introduces a blockchain-enabled solution for SC crop production, focusing on improving traceability and transparency from planting to harvest and distribution. Existing studies mainly focus on blockchain-enabled supervision, multi-chain coordination and IoT-assisted traceability mechanisms [21]. These approaches demonstrate the applicability of blockchain in agricultural ecosystems but primarily emphasize transaction integrity and visibility rather than dynamic traceability granularity or compositional product transformations.

The application of blockchain technology in the wheat SC has been explored in a limited number of studies. The work of [22] introduces a blockchain-based model that integrates NFC tags, IoT devices, and smart contracts to create an end-to-end traceability solution, ensuring that all transactions from seed supply to consumer purchase are securely recorded. In the case of grain and oil SC, [23] explores the integration of blockchain with data fusion analysis, focusing on enhancing traceability and safety. The work in [24] introduces a reliable traceability model for the grain and oil SC, integrating blockchain with machine learning to detect anomalies and improve data accuracy. Recent studies have increasingly shifted from simple provenance recording toward adaptive and quality-aware traceability architectures. For example, [25] examine blockchain-facilitated quality traceability in geographically indicated agricultural products, highlighting the role of blockchain in coordinating inspection mechanisms, supplier incentives and quality assurance processes. Finally, [26] proposes a blockchain-based solution for managing strategic grain reserves. It utilizes Hyperledger Fabric to create a decentralized and tamper-proof ledger that records all transactions related to the procurement, storage and distribution of grain reserves.

Although the aforementioned studies demonstrate the growing applicability of blockchain technology in agricultural and grain-oriented supply chains, most existing approaches primarily focus on transaction immutability, provenance recording and product visibility within isolated traceability workflows. In contrast, limited attention has been given to dynamically adjustable traceability granularity, compositional token synthesis across successive product transformations, and interoperability with heterogeneous ERP

ecosystems through standardized EPCIS-compliant event orchestration. Moreover, existing implementations rarely provide a unified framework capable of simultaneously integrating process-aware tokenization, stakeholder accountability and bi-directional synchronization between enterprise systems and blockchain infrastructures.

2.2. Literature Review on Granularity Frameworks, Tokens and EPCIS in Blockchain-Enabled SC Traceability Solutions

The integration of EPCIS with blockchain has emerged as a solution to improve SC traceability, transparency and security across industries. For example, [27] introduces a blueprint-based tokenization system that captures EPCIS events, such as creation, aggregation and transformation, and translates them into blockchain tokens. This approach allows tracking complex transformations in dynamic SCs while ensuring an immutable event history. Similarly, [28] focuses on security by introducing a continuous authentication mechanism for storing essential traceability information on the blockchain while maintaining detailed data for product monitoring throughout the supply chain. In addition, the work in [29] emphasizes compliance, particularly in halal product traceability. Their framework leverages smart contracts within the EPCIS and blockchain system to automate halal compliance verification throughout the SC, ensuring transparency while addressing global halal standardization challenges. Finally, [30] utilizes Hyperledger Fabric to store essential traceability data on the blockchain, while larger datasets remain in the off-chain EPCIS repository. This architecture reduces blockchain overhead while maintaining high levels of security and data integrity.

Recent research increasingly explores tokenized representations of physical assets to support fine-grained and verifiable traceability across supply chain networks. In these approaches, blockchain tokens act as digital proxies of products, batches or components, enabling immutable provenance tracking, compositional transformations and life-cycle-aware monitoring. This transition reflects a broader shift from static provenance recording toward dynamic and process-aware traceability architectures. Literature also emphasizes importance of token standards and scalability in blockchain-based traceability systems. For instance, [31] introduce a token-based system for supply chain environments, where each workpiece is tracked as it moves through production. The work of [32] discusses theoretical design of modular, blockchain-based traceability systems, emphasizing token standardization to ensure interoperability across systems and industries. Finally, [33] apply NFTs in SC management by enabling secure and trustworthy digital representations of physical goods. This supports better traceability and transparency, particularly in complex SCs where maintaining the integrity of product history is essential.

Another important research direction concerns interoperability between blockchain infrastructures and enterprise information systems. Several studies recognize EPCIS-compliant event modeling, semantic interoperability and middleware-driven synchronization as critical requirements for scalable enterprise deployment. EPCIS-based architectures allow blockchain systems to maintain standardized event semantics while supporting integration with heterogeneous ERP platforms and distributed operational environments [12,30,34].

Beyond purely technical traceability architectures, recent empirical supply-chain studies also highlight the organizational relevance of blockchain, traceability and digital information processing. [9] examines the relationship between blockchain-enabled supply-chain practices and supply-chain performance and shows that supplier trust, traceability and transparency mediate this relationship. This finding is relevant to blockchain-based traceability systems because it links technical capabilities such as shared records and visibility with organizational outcomes such as trust and performance. In a complementary direction, [10] examines digital supply chains and information processing capability from

a supply-chain resilience perspective, showing that digital supply-chain practices and visibility-oriented information processing support risk management and resilience. These studies do not propose granular tokenization, EPCIS-compliant event orchestration or ERP-blockchain middleware integration; however, they strengthen the managerial motivation for architectures that improve trusted information sharing, traceability visibility and risk-aware coordination across supply-chain actors.

Granularity is another critical factor in ensuring precise tracking and verification across blockchain-enabled SCs. Several studies highlight its role in enhancing transparency, accountability and operational trust. The work presented in [35] discusses how different levels of granularity can be adapted to the needs of various stakeholders, ensuring that traceability information remains accessible at the appropriate level of detail. Similarly, [1] argues that traceability granularity should be dynamically adjusted according to factors such as product type, operational risk and regulatory requirements. A common theme across recent studies is the need for compositional and token-based mechanisms that support both fine-grained and generalized traceability depending on SC requirements. For instance, [36] emphasizes the importance of adjusting granularity through the definition of batch sizes, while [27] discusses token-based models that map object-related events at different levels of granularity. More recent studies further examine the implications of component-level versus product-level tracing, highlighting that traceability depth significantly affects visibility, coordination efficiency and scalability in multi-stage SC environments [37].

Another significant aspect of granularity concerns the management of large volumes of traceability data. Towards this direction, [38] proposes a space-efficient logging mechanism that supports high traceability granularity without overwhelming storage resources. Similarly, [39] proposes a hybrid framework utilizing IPFS to support large-scale data produced in agricultural supply-chains. Recent research also highlights the need for fine-grained traceability architectures capable of balancing operational visibility with scalability and synchronization overhead [40]. Granularity is also explored in domain-specific SCs such as halal [41], grain and oil [42] and carbon footprint traceability [43]. Although existing approaches demonstrate the importance of granular traceability, most current solutions primarily focus on isolated provenance recording or static token representations. Limited attention has been given to dynamically adjustable granularity mechanisms that preserve semantic interoperability, process-aware transformations and synchronized integration with heterogeneous ERP environments.

2.3. Open Issues Identified in the Literature and Positioning of Our Work

Existing work on blockchain-based SC traceability exhibits several structural limitations. First, none of the existing approaches concurrently addresses the following: (i) fine-grained traceability with token-based representations, (ii) EPCIS-compliant event modeling, and (iii) open integration interfaces with existing ERP systems. Second, although some studies adopt EPCIS standards, they typically fail to account for traceability granularity, treating events as isolated records rather than as components of a coherent, evolving process. Third, most token-based models rely on rigid abstractions—such as batch- or NFT-level representations that under-specify process semantics, including accountability (who performed an action and under what conditions). As a result, granularity is often defined in an ad hoc manner that fails to remain consistent across transformation stages.

The framework proposed in this paper attempts to go beyond these limitations through a generalizable second-layer architecture that preserves existing ERP infrastructures while adding a blockchain-based composition and verification layer for provenance. Metadata flow between ERP-like systems and the blockchain layer through middleware APIs, Prisma

ORM and containerized services, enabling prototype-level synchronization for orders, inventory and customer records. A compositional token triad, Stakeholder Token (SH), Product Token (PD), and Process Token (PC), is minted at each transformation and synthesized into a Synthesized Token (ST_x) that encapsulates lineage references, actors, materials, parameters and timestamps. RBAC policies regulate access to blockchain-enabled functions, while smart contracts automate token creation, updates and linkage. We deliver a reproducible proof-of-concept implementation and an illustrative wheat SC scenario, demonstrating the technical feasibility of dynamic granularity and EPCIS-oriented interoperability under controlled conditions. Therefore, our approach provides a pipeline for SC traceability where we (a) account for established granularity requirements from the literature, (b) translate these requirements into a technical and operational framework, and (c) demonstrate feasibility through a wheat SC proof of concept, from which we derive context-specific managerial implications.

As shown in Table 1, prior studies address important but partial aspects of blockchain-enabled supply-chain traceability. Some works focus on EPCIS-compliant event modeling, while others emphasize tokenization, traceability granularity or enterprise-system interoperability. However, the comparison indicates that these capabilities are usually examined separately. In particular, token-based approaches rarely provide EPCIS-compliant event semantics and ERP synchronization, while EPCIS-oriented studies generally do not support dynamic token-based granularity across product, batch and component levels. This study addresses this gap by integrating tokenization, EPCIS-based event semantics, dynamic traceability granularity and ERP interoperability within a unified blockchain-enabled supply-chain traceability framework.

Table 1. Comparison of prior blockchain-enabled supply-chain traceability studies.

Study	Tok.	EPCIS	ERP	Dyn. Gran.	Main Focus
[36]	Partial	No	No	Partial	Batch-level traceability and product transformation tracking.
[35]	No	No	No	Yes	Granularity-aware blockchain traceability for stakeholder-specific visibility.
[1]	No	No	No	Yes	Dynamic adaptation of traceability granularity according to product characteristics, operational risk and regulatory requirements.
[30]	No	Yes	Partial	Partial	EPCIS–blockchain architecture with off-chain repository support.
[38]	No	No	No	Yes	Space-efficient logging for highly granular traceability data.
[42]	No	No	No	Yes	Blockchain-enabled traceability framework for grain and oil supply chains.
[43]	No	No	No	Yes	Carbon-footprint traceability and environmental information tracking.
[41]	No	No	No	Yes	Granular traceability framework for halal supply chains.
[27]	Yes	Yes	No	Yes	Tokenization of EPCIS events for blockchain-based traceability and transformation tracking.
[28]	No	No	No	No	Secure and uninterrupted authentication mechanism for traceability information.
[12]	No	Yes	Partial	Partial	EPCIS-based blockchain interoperability and enterprise synchronization.
[29]	No	Yes	No	Partial	Smart-contract-based halal compliance verification using EPCIS and blockchain.
[32]	Yes	No	No	No	Token standardization and modular blockchain traceability design.
[34]	No	Yes	Partial	Partial	Event-driven EPCIS-compliant blockchain integration and interoperability.
[33]	Yes	No	No	No	NFT-based digital representation of physical supply-chain assets.
[37]	No	No	No	Yes	Product-level and component-level traceability depth and visibility.
[40]	No	No	No	Yes	Fine-grained traceability architectures and scalability trade-offs.

Table 1. Cont.

Study	Tok.	EPCIS	ERP	Dyn. Gran.	Main Focus
[31]	Yes	No	No	Yes	Token-based workpiece tracking in production environments.
[39]	No	No	No	Yes	Hybrid blockchain–IPFS architecture for large-scale agricultural traceability data.
[44]	Yes	No	No	Partial	Blockchain and machine learning-based drug traceability in the pharmaceutical supply chain.
[5]	No	No	No	Partial	Blockchain-based traceability framework for the manufacturing and textile supply chain.
This study	Yes	Yes	Yes	Yes	Unified framework integrating tokenization, EPCIS semantics, dynamic traceability granularity and ERP interoperability.

3. Proposed Framework

3.1. High Level Description

In this paper, we propose a general framework based on blockchain and the use of tokens, for product traceability combined with additional information about their production processes along a SC. The high-level architecture and general operation of the proposed framework are illustrated in Figure 1. As shown in the figure, the framework builds on existing infrastructure, essentially acting as a second-layer solution. By using blockchain technology, data integrity is ensured throughout all phases of the SC. The system follows an interconnected process; for each phase that contributes to the product transformation, a token is created that incorporates the relevant information. The tokens at each intermediate product are tracked in the blockchain, leaving an irrevocable record of each transformation.

Building on the framework introduced in [45], traceability granularity is defined as the joint resolution at which (i) product units, (ii) supply-chain processes and (iii) participating stakeholders are uniquely identified, monitored and recorded within the traceability information model. Granularity therefore emerges from the combined specification of product characteristics, process observability and stakeholder engagement, rather than from any single system parameter. Traceability granularity is determined by (i) the size of the traceable unit, (ii) the frequency and type of recorded events, and (iii) the scope of process and stakeholder attributes bound to each traceable state. In the sequence, we elaborate on the proposed granularity scheme. The notations used in the proposed framework are shown in Table 2.

Table 2. Notation used in the proposed framework.

Symbol	Meaning	Description
SH	Stakeholder Token	Represents the identity, role and accountability attributes of a supply-chain actor.
PD	Product Token	Represents a product, sub-product, batch, lot or traceable product unit.
PC	Process Token	Represents transformation, storage, transport or processing events.
TP_i	Traceability Token at stage i	Generic token instance associated with a specific supply-chain stage.
ST_x	Synthesized Token	Aggregated provenance token for the final product or composed product state.

Multiple stakeholders, such as manufacturers, suppliers, and distributors, participate in tasks that transform, store or move the product. These transformation processes are recorded at each stage, and the traceability record is progressively extended until the final product is produced. The final product, labeled as ‘Product X’ in Figure 1, is linked to all intermediate components and stages. The synthesized token ST_x does not duplicate all intermediate payloads; instead, it aggregates verifiable references to the relevant component

tokens and their version histories, enabling reconstruction of the product's provenance when needed.

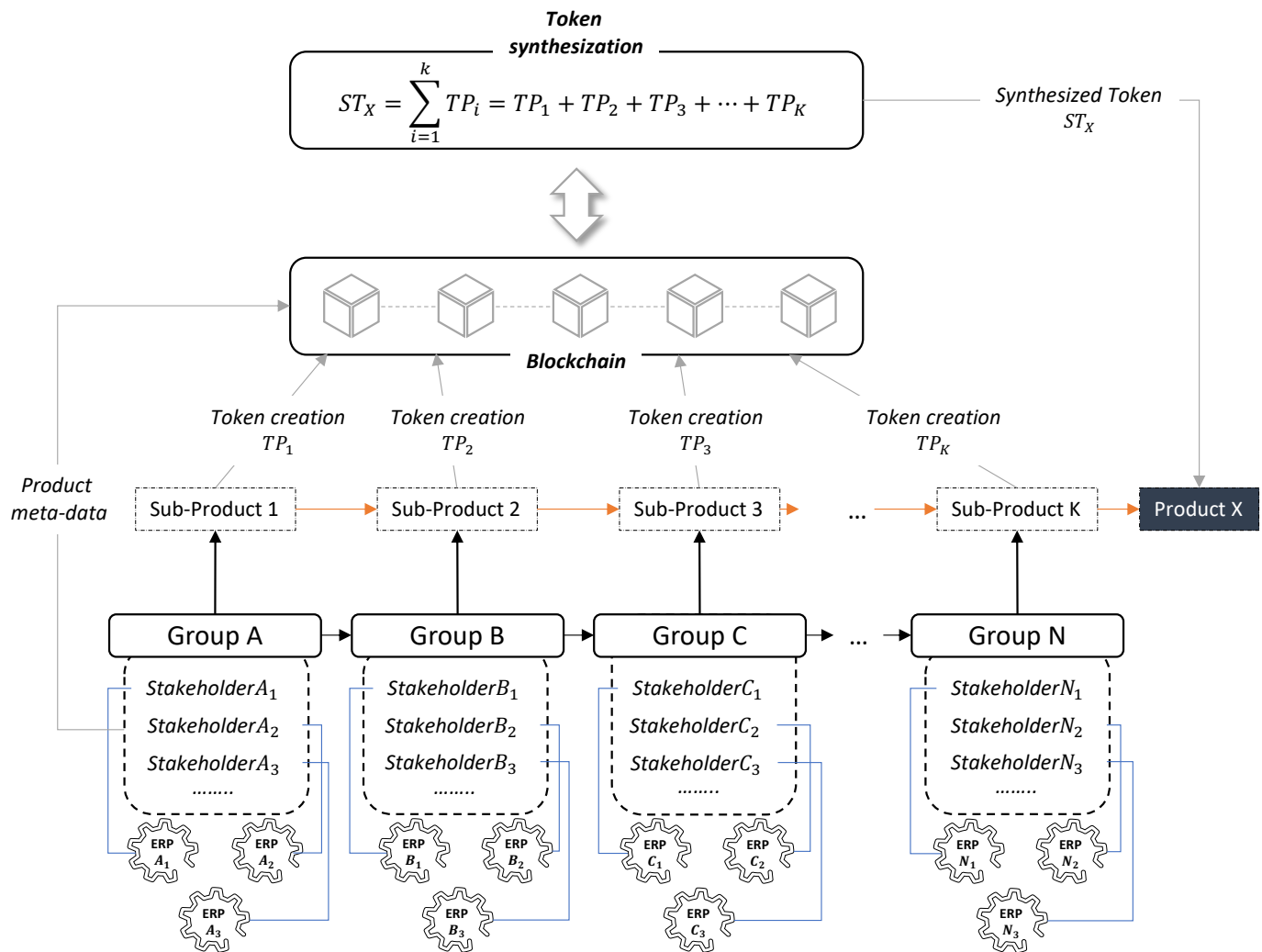


Figure 1. High-level architecture.

Formally, each traceability token TP_i , shown in Equation (1), is modeled as a tuple consisting of immutable identity attributes, an initial state, and an append-only sequence of version records. Each version v_i^k , shown in Equation (2), cryptographically links to its predecessor and to the corresponding EPCIS event, ensuring both temporal ordering and data integrity across token updates. Algorithm 1 formalizes the notion of traceability granularity as an explicit constraint on how product units, process events and stakeholder attributes are recorded throughout the supply chain.

$$TP_i = \langle I_i, M_i^0, \{v_i^1, v_i^2, \dots, v_i^k\} \rangle \quad (1)$$

$$v_i^k = (t_k, H(v_i^{k-1}), H(e_k), \Delta_k) \quad (2)$$

In order to enable the storage and transfer of information between the individual phases, the framework creates tokens (in each phase) which are stored on the blockchain. As illustrated in Figure 1, as the product evolves through the different phases, new tokens are created that are bound to the previous ones, thus maintaining continuity in the traceability chain. In particular, for each sub-product a distinct token ($TP_1, TP_2, \dots, TP_K$) is created, in correlation with each stage of the SC, in which the relevant information is stored. During the process of embedding the information in the tokens, metadata is extracted from

the stakeholders' ERP systems, which contain critical information: for example, stakeholder specific data and processing details. Where applicable, these ERP-resident records are stored automatically from IoT sensor streams (e.g., temperature, humidity, geolocation) through gateways into the EPCIS/ERP pipeline and then anchored on-chain. In particular, all traceability updates are encoded in EPCIS-compliant events, which act as a standardized execution layer between the ERP systems and the blockchain. Each movement or transformation of the product is first expressed as an EPCIS event with validated business steps and dispositions, and only then is it anchored on-chain through token updates. This approach ensures that blockchain state transitions follow globally standardized supply-chain process approaches while preserving interoperability among heterogeneous ERP systems.

Algorithm 1 Granularity-Constrained Token Updates via EPCIS Events

Require: traceable unit size u , granularity policy $G = \langle u, \mathcal{E}, \rho, \mathcal{A}_{proc}, \mathcal{A}_{sh} \rangle$, EPCIS event stream $\{e\}$, caller u_c , mapping policy Π

Ensure: append-only token histories $\{v_i^k\}$ consistent with G and EPCIS semantics, or REJECT

```

1: function APPLYGRANULARITY( $G, \{e\}, u_c, \Pi$ )
2:   for all EPCIS events  $e$  do    ▷ (i) EPCIS semantic admissibility (type  $\rightarrow$  operation)
3:      $(op, \theta) \leftarrow \text{MAPEPCISTOOPERATION}(e, u_c, \Pi)$     ▷ Algorithm 2
4:     if  $(op, \theta) = \text{REJECT}$  then
5:       return REJECT
6:     end if
7:     ▷ (ii) Granularity constraints: event type and frequency
8:     if  $e.type \notin \mathcal{E}$  or  $\neg \text{SATISFYFREQUENCY}(e, \rho)$  then
9:       return REJECT
10:    end if
11:    ▷ (iii) Token instance resolution from unit size  $u$ 
12:     $TP_i \leftarrow \text{RESOLVETOKEN}(u, e)$  ▷ create or retrieve the token for this traceable unit
13:    ▷ (iv) Attribute-scope constraint: only scoped fields may enter  $\Delta$ 
14:     $\Delta \leftarrow \text{EXTRACTUPDATESSET}(e, op, \theta)$ 
15:    if  $\neg \text{SCOPEOK}(\Delta, \mathcal{A}_{proc}, \mathcal{A}_{sh})$  then
16:      return REJECT
17:    end if
18:    ▷ (v) Append-only versioned update (no destructive modification)
19:     $T \leftarrow \text{GETTOKEN}(TP_i)$ 
20:    if  $T = \emptyset$  then
21:      return REJECT
22:    end if
23:     $v_k \leftarrow T.history[end]$ 
24:     $v_{k+1} \leftarrow (t, H(v_k), H(e), \Delta)$ 
25:     $\text{APPENDVERSION}(TP_i, v_{k+1})$ 
26:  end for
27:  return true
28: end function
  
```

Finally, the product is represented by a synthesized token ST_x , which is the result of the binding of all tokens from each phase through a process known as token synthesis. A detailed and unchangeable record of every phase of the SC to the final product is provided by this synthesized token, which includes all the information from the previous phases.

3.2. System Assumptions

The design of the framework for multi-token-based tracing relies upon specific assumptions that are critical to ensure its functionality, scalability and security.

We assume that each SC stakeholder either owns an ERP system or has access to an ERP-connected data-entry mechanism, such as a cooperative-managed portal, lightweight

web/mobile interface, middleware gateway, or third-party logistics platform. Larger actors may connect directly through ERP APIs, while actors with limited resources may participate through simplified interfaces that generate standardized EPCIS events. Therefore, the framework does not require all stakeholders to operate full-scale ERP systems, but it does require that traceability-relevant events are captured in a structured and authenticated manner. It is also assumed that each stakeholder is securely connected to the blockchain network. This is required for token generation, token storage, and retrieval at each transformation stage. The blockchain network must be decentralized and distributed to ensure availability at any moment in the event of partial network failures, hence eliminating any single point of failure.

Another basic assumption is the possibility of uniquely identifying each SC product and sub-product. Uniqueness is important for the exact mapping of each SC stage to its respective token in the blockchain. The framework depends on standardized and universally understood identification standards, which will be adopted by all stakeholders. In particular, we adopt the well-established EPCIS standard (integration details are provided in Section 3.3). The framework requires metadata obtained from ERP systems, IoT gateways or data-entry interfaces to be sufficiently accurate and consistent for traceability purposes. However, this is a critical operational assumption rather than a property guaranteed by blockchain itself. We assume that the blockchain network can linearly grow to handle the growing number of tokens and transactions related to the SC complexity and the increase in stakeholders. The network must support this expansion while upholding performance and security criteria. It is presumed that the framework's implementation adheres to all pertinent legislative mandates regarding data storage, privacy, and traceability within the SC. Stakeholders must ensure that their utilization of the blockchain network and data management procedures adhere to industry norms and regulatory requirements, especially in areas with rigorous data protection rules.

3.3. Blockchain and Tokenization Model

The methodological approach of the Tokenization Model extends the research presented in [45]. The BOM works as a hierarchical inventory that includes a detailed record of all the elements, by-products, stakeholders and processes involved in the production of the final product. Based on this model, each element of the SC is accurately represented and easily traceable. In the proposed framework, this is achieved by using three unique categories of tokens: *stakeholder token* (SH), *product token* (PD) and *process token* (PC), as visualized structurally in Figure 2.

Semantic interoperability is enforced by mapping these token types to predefined subset of EPCIS event types. Specifically, *ObjectEvent* governs the creation and state updates of PD tokens, *TransformationEvent* triggers the synthesis or mutation of PC tokens during material transformations, and *AggregationEvent* governs the composition and decomposition of compound products within the BOM hierarchy. This semantic coupling can be expressed as a function that maps EPCIS event types (*e.types*) to admissible blockchain token operations. Algorithm 2 formalizes the semantic coupling between EPCIS events and blockchain token operations. Only events that satisfy predefined semantic and structural constraints are allowed to trigger state transitions, while all others are deterministically rejected.

The structural integration of these tokens is visualized in Figure 2, which illustrates the subtokenization model where SH, PD, and PC tokens are generated and linked at each supply chain stage. This hierarchical approach ensures that both static stakeholder data and dynamic product/process attributes are captured with high granularity.

Algorithm 2 Mapping EPCIS Events to Token Operations**Require:** EPCIS event e , caller u , mapping policy Π **Ensure:** token operation op with parameters θ or REJECT

```

1: function MAPEPCISTOOPERATION( $e, u, \Pi$ )
2:   AUTHORIZE( $u, e$ )
3:   VALIDATEEPCIS( $e$ )
4:   if  $e.type \notin \text{dom}(\Pi)$  then
5:     return REJECT
6:   end if
7:   for all  $f \in \Pi[e.type].requiredFields$  do
8:     if  $e[f]$  is undefined then
9:       return REJECT
10:    end if
11:  end for
12:   $op \leftarrow \Pi[e.type].operation$ 
13:   $\theta \leftarrow \text{EXTRACTPARAMS}(e)$ 
14:  if CHECKSEMANTICRULES( $e, op, \theta$ ) = false then
15:    return REJECT
16:  end if
17:  return ( $op, \theta$ )
18: end function

```

Figure 2 illustrates an example of the tokenization process based on a generic SC consisting of five stakeholder groups: Raw Material Suppliers, Transportation Providers, Warehousing, Manufacturers, and Distributors. The graphic depicts the creation process of the three token types (SH, PD, and PC) at each phase. For each stakeholder group, STs are created to represent the identification and function of the relevant entities. At the same time, PD tokens are created to capture and record the status and characteristics of the items or by-products as they move from one phase of the SC to the next. Simultaneously, PC tokens are generated to record and track information about the processes the products undergo, such as transport, warehousing, and various transformation/processing procedures. In the process, as products flow between phases through the SC—from raw material to final distribution—these tokens are accumulated into a composite token in which the products' details are recorded, ensuring traceability and transparency throughout the transport process to the end-user.

Although SH, PD and PC tokens are designed to evolve over time in order to reflect successive supply-chain events, mutability in the proposed framework is deliberately constrained and formally governed. In particular, token identifiers, ownership bindings, cryptographic references to predecessor states and EPCIS event hashes are immutable once recorded. Mutations are limited to well-defined state attributes (e.g., process parameters, quality metrics, timestamps and status flags) and are triggered exclusively by validated EPCIS event types. Each token update is implemented as an append-only state transition: rather than overwriting prior information, every mutation appends a new version entry that includes (i) a timestamp, (ii) a cryptographic hash of the immediately preceding token state, and (iii) a cryptographic reference to the corresponding EPCIS event payload maintained off-chain. This design establishes a chain of tamper-evident versions in which all historical states remain verifiable and replicable. As a result, external verifiers can reconstruct the canonical state of any token at an arbitrary time t by replaying the ordered sequence of recorded state transitions, starting from the token's creation event. Token synthesis operates on these versioned tokens by aggregating their immutable lineage references rather than mutable attributes, ensuring that synthesized tokens encapsulate complete provenance without duplicating intermediate states. Algorithm 3 describes the synthesis of multiple component tokens into a single aggregated token ST_x .

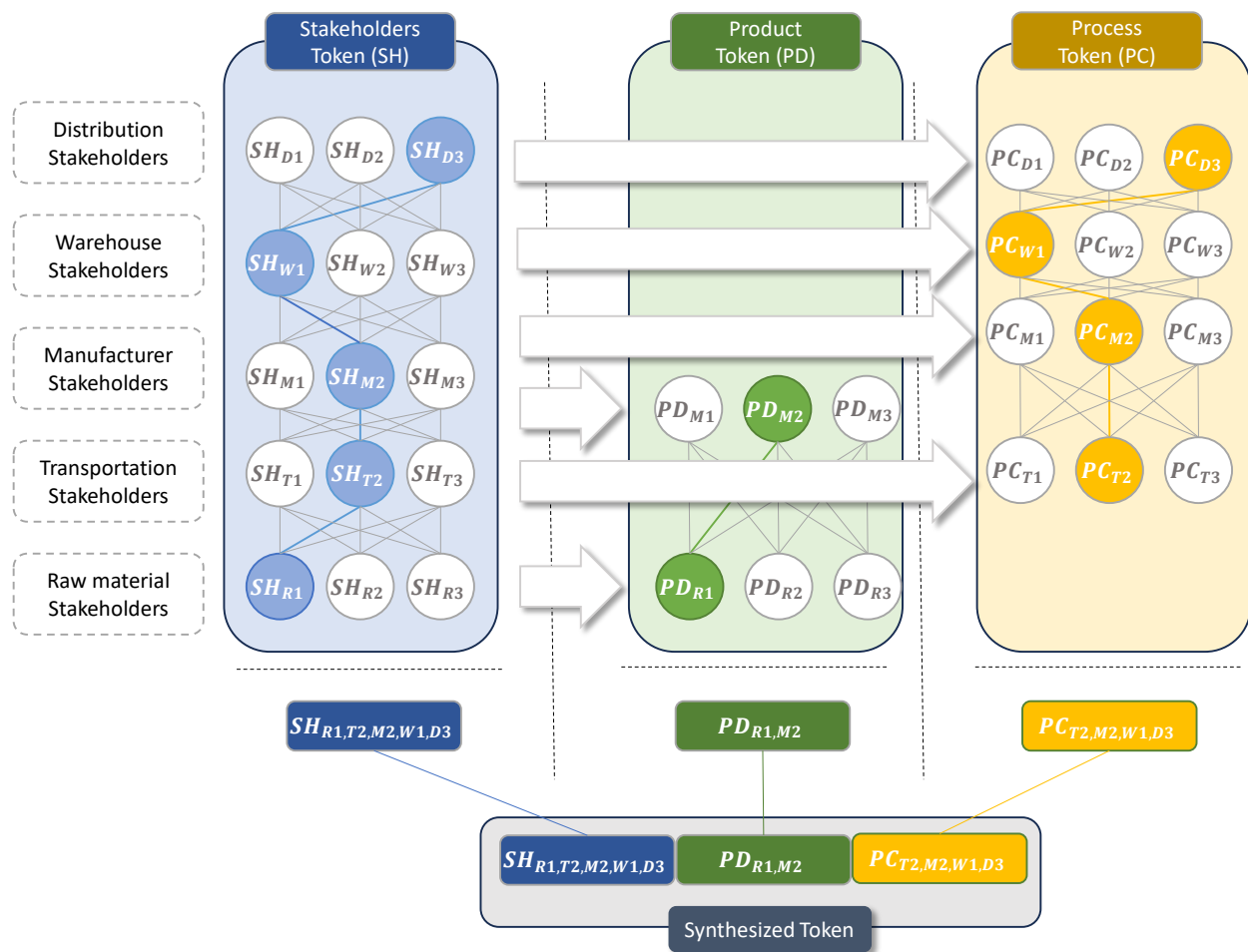


Figure 2. Illustration of the subtokenization model in which stakeholder, product, and process tokens are created at each stage of the supply chain.

Algorithm 3 Synthesizing a Token ST_x from Component Tokens

Require: component tokens $\mathcal{T} = \{TP_1, \dots, TP_n\}$, synthesis metadata M_x

Ensure: synthesized token ST_x or REJECT

```

1: function SYNTHESIZESTA( $\mathcal{T}, M_x$ )
2:    $R \leftarrow []$ 
3:   for all  $TP_i \in \mathcal{T}$  do
4:      $T \leftarrow \text{GETTOKEN}(TP_i)$ 
5:     if  $T = \emptyset$  then
6:       return REJECT
7:     end if
8:      $v_i \leftarrow T.\text{history}[\text{end}]$ 
9:      $R \leftarrow R \cup \{(TP_i, H(v_i))\}$ 
10:  end for
11:   $ST_x \leftarrow \langle R, M_x, ts \leftarrow \text{now}() \rangle$ 
12:  STORETOKEN( $ST_x$ )
13:  return  $ST_x$ 
14: end function

```

The purpose of Algorithms 1–3 is to define the execution logic and rejection conditions that must be preserved by the prototype implementation. Algorithm 1 specifies the admissibility checks for granularity-constrained token updates, Algorithm 2 defines how EPCIS events are mapped to blockchain token operations, and Algorithm 3 specifies the synthesis of component tokens into a single synthesized token. These algorithms are therefore not

treated as purely abstract specifications; their main branches are operationalized in the middleware validation logic and in the smart-contract functions described in Section 4. In particular, invalid EPCIS event types, missing required fields, unauthorized callers, unresolved token identifiers and missing component tokens are expected to terminate with a deterministic rejection rather than producing a partial or inconsistent on-chain state.

In addition, the proposed framework explicitly distinguishes between on-chain integrity guarantees and data authenticity at ingress. While blockchain mechanisms ensure that once recorded, traceability data cannot be altered without detection, the correctness of data entering the system depends on the trustworthiness of upstream sources, namely ERP systems, IoT gateways and EPCIS event generators. In the current design, authenticity at ingress is enforced through organizational trust boundaries, authenticated middleware APIs, and ERP-level access controls rather than through decentralized oracle consensus. This boundary is important because input authenticity is central to supply-chain traceability. The proposed framework guarantees that once an EPCIS event hash is anchored on-chain, subsequent tampering can be detected; however, it does not independently prove that the original ERP record, IoT measurement or human-entered value was correct. This residual risk should be mitigated in practical deployments through authenticated data-entry procedures, ERP audit logs, IoT device identity management, sensor calibration, random inspections, anomaly detection and contractual accountability among participating organizations. Cryptographic hashes of these events are immutably recorded in token histories, providing tamper-evidence and post hoc verifiability, but not independent validation of sensor or human-reported inputs. Formally, for all $k > 1$, we have

$$H(v_i^{k-1}) = v_i^k.\text{prevHash} \quad (3)$$

$$H(v_i^{k-1}) \neq v_i^k.\text{prevHash} \Rightarrow \text{Tampering Detected} \quad (4)$$

This reflects a deliberate design choice aligned with enterprise traceability deployments, where accountability is enforced through contractual, regulatory and organizational mechanisms rather than anonymous oracle networks.

3.4. Use Case: Wheat Supply Chain

To demonstrate the feasibility of the proposed framework, this section adopts an illustrative wheat SC scenario. The scenario is used to validate the logic of the proposed architecture and prototype implementation; it does not rely on real transaction data from industrial wheat supply-chain actors. Several factors make the wheat SC a suitable scenario for demonstrating the applicability of the proposed framework. Wheat is among the most widely consumed grains in the world, and a significant element in the diet of the world's billion-plus citizens. Wheat and all the products based on it, such as flour and bread, are significant in developing and developed countries. Due to the demand, there is an immediate need to offer a consistent mechanism to ensure food quality, safety, and authenticity in various markets.

The wheat supply chain (WSC) serves as our primary case study, involving a multi-stage process from farming and milling to manufacturing and retail distribution (as depicted in Figure 3). Our framework applies the SH, PD, and PC token model to each stage, ensuring that transformation parameters (e.g., storage temperature, milling specifications) and stakeholder accountability are immutably captured. This process culminates in a synthesized token that encapsulates the product's entire provenance, verifiable by end-consumers through a QR code on the final product. Consumer QR verification is designed as a filtered disclosure interface rather than as full access to the underlying traceability graph. Consumers may access selected information such as product origin, batch or lot identifier,

high-level transformation milestones, certification status, and relevant quality or safety claims. By contrast, commercially sensitive information—including supplier contracts, exact quantities, internal process parameters, detailed timestamps, private stakeholder identifiers and business relationships—remains accessible only to authorized supply-chain participants according to RBAC policies and off-chain access controls.

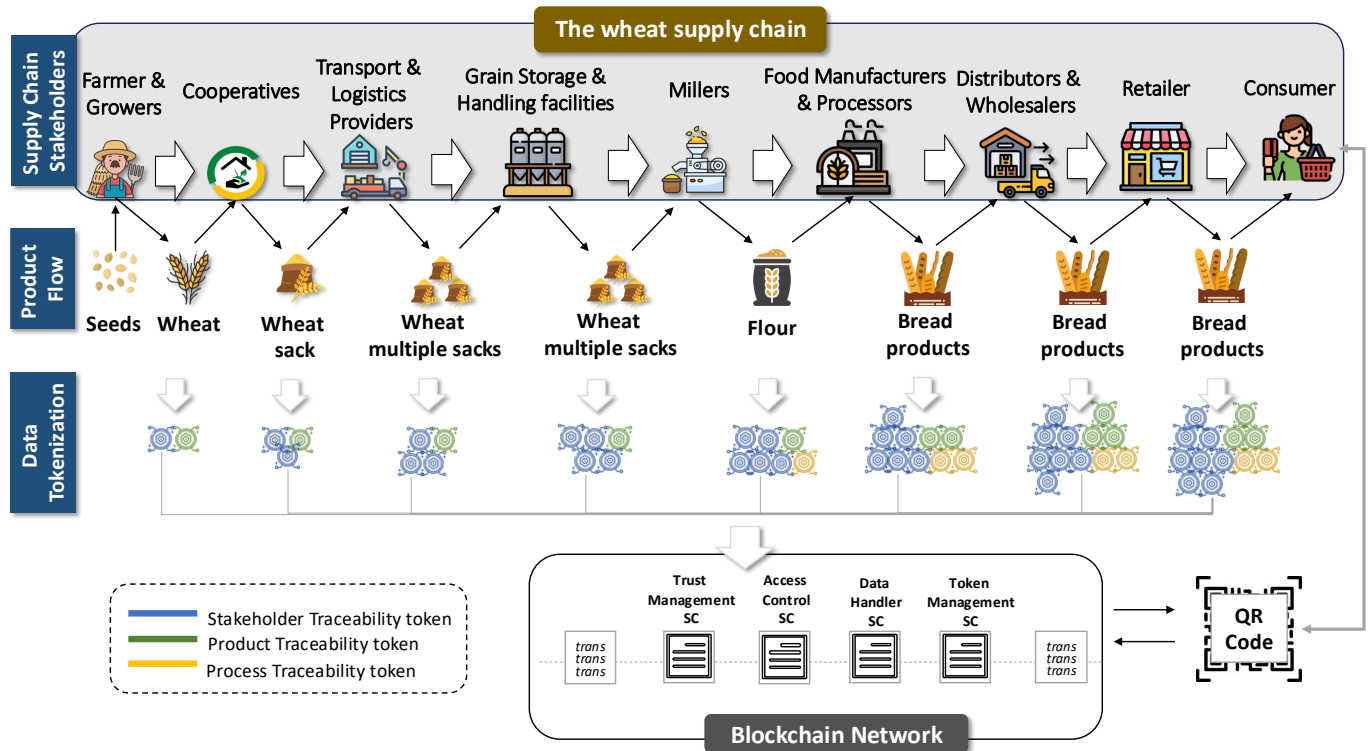


Figure 3. Architecture integration in wheat supply chain.

4. System Implementation and Performance

The primary objective of this deployment is to demonstrate the efficiency of the proposed blockchain-based SCM model. The model provides end-to-end product and process traceability at all the various stages in the SC. The stakeholders are in a position to monitor the transactions in real-time through tokenization, smart contracts, and decentralized ledger mechanisms, and the integrity and transparency in the data are ensured. The framework was evaluated using the wheat SC scenario described in Section 3. The implementation should be interpreted as a proof-of-concept validation under controlled conditions. The objective is to validate the architectural feasibility of EPCIS-oriented token updates, ERP-like middleware synchronization, RBAC-controlled smart-contract operations and provenance reconstruction. The prototype does not constitute a production deployment and does not include direct integration with commercial ERP platforms such as SAP, Oracle or Microsoft Dynamics. Similarly, the private Ethereum/Ganache environment is used to isolate smart-contract logic and application-layer behavior, not to reproduce the latency, throughput, consensus or governance characteristics of a production blockchain network.

4.1. Main Actors and Components

The design is constructed with a role-based access control (RBAC) mechanism that organizes users into three different groups, i.e., Administrators, Moderators, and Users, with varying levels of permissions and interactions with the blockchain:

- *Administrators* have the highest level of access in the system. Their responsibilities include the assignment of user permissions, revoking user permissions, enabling or disabling user registration in the blockchain, and monitoring general system operations. Administrators can also search for orders and customers in the blockchain as well as the local database.
- *Moderators* are similar to the administrator role, although with restrictions. Moderators cannot revoke or assign user permissions or toggle the blockchain user registration status. However, they can search for orders and customers and retrieve the company's inventory in the blockchain.
- *Users* have the lowest privilege level in the system. They cannot be assigned functions for interacting with the blockchain: for example, query the blockchain or query inventory data. Their operations are limited to basic ERP functions, with blockchain functions completely hidden in their interface.

This RBAC model is intentionally implemented as a baseline access-control prototype. It demonstrates how blockchain-enabled functions can be restricted according to predefined roles, but it does not fully capture the governance complexity of real multi-firm supply chains. In production settings, access policies would need to be extended with organization-specific roles, contractual permissions, attribute-based access control, selective disclosure rules and possibly channel- or consortium-specific governance mechanisms. Therefore, the current RBAC model should be understood as a minimal enforceable access-control layer rather than a complete supply-chain governance model.

Each role comes with specific permissions, which are enforced by application logic built within the smart contracts published on the blockchain. Unauthorized operations are prevented by dynamic control mechanisms for access, which check permissions before allowing specific features or executing specific actions. The system consists of two main layers: the ERP layer and the Blockchain layer. The entire source code is fully reproducible and has been made publicly available on GitHub (<https://github.com/eeddieg/Blockchain-Based-ERP-Interoperability-in-Supply-Chain-Management>, accessed on 6 May 2026).

The ERP layer consists of the basic functions of procurement, inventory management, order processing, logistics, and supplier relationship management. The integration between ERP simulated data and the blockchain is supported by middleware APIs, which act as intermediaries. The APIs support prototype-level synchronization between simulated ERP records and blockchain events, ensuring that updates performed in the controlled testbed are reflected consistently across the middleware, local database and blockchain layer. A central component of the integration is the use of Prisma ORM (object-relational mapping), which supports the management and interaction with the underlying ERP databases. Prisma provides a high-level data abstraction of the database schema, which supports the interaction between the middleware and the ERP system while optimizing the database queries and the update operations. This ensures changes in the ERP system, for example, order status changes or inventory changes, are always correctly represented in the blockchain, ensuring data integrity and correctness.

The Blockchain layer was implemented in a private Ethereum blockchain network using Ganache, which offers a local blockchain simulation for development and testing. The selection of a private, permissioned blockchain for this enterprise supply chain prototype is grounded in established blockchain selection methodologies (e.g., [46]). In a multi-actor supply chain, participants are known but not fully trusted, requiring smart contract functionality and access control without the transaction cost (gas) overhead and public exposure typical of permissionless networks. Therefore, a private Ethereum network fulfills the need for a shared, tamper-evident ledger while preserving privacy and minimizing operational costs during prototype development. Seven smart contracts, written

in Solidity, were deployed in the private network to automate the predefined business rules and processes. The smart contracts govern critical operations such as order creation, status update, token generation, and token synthesis. By embedding business rules in the blockchain itself, smart contracts enhance the traceability of the transaction and automate processes that otherwise require manual verification.

The system architecture consists of eleven servers for the ERP system, along with a Blockchain network running multiple nodes in the Ganache simulation environment. The servers, which were built with the ExpressJS framework, are hosts for the APIs by which the ERP systems interact with the blockchain. All parts of the system, including ERP servers, blockchain nodes, middleware APIs and front end, are containerized with Docker to ensure portability, scalability and ease of deployment. Docker simplifies deployment by wrapping each piece of the system in its own container so each of the services will run the same, regardless of the underlying hardware or cloud environment.

The frontend was developed using NodeJS, the VueJS framework, TypeScript, and Bootstrap. It offers a user interface to the stakeholders for order tracking, inventory status, and the interactions between the ERPs and the blockchain. The frontend also includes synchronization tracking monitoring tools between the ERP layer and the blockchain layer, which provide real-time visibility into the flow of products and data throughout the SC.

4.2. Registration and Inventorying

The data exchange and processes in the proposed model establish a structured interaction between the blockchain and ERP systems, ensuring synchronized and transparent operations across both layers.

The Registration and Inventorying phase is the entry point for getting the operations of a company onto the blockchain, with the purpose of having any data relevant to the company—inventory to customer and order data—stored securely and trackable. The process begins with the assessment of the company status in the blockchain network. On the administrator's first-time login, the system queries whether the company is already registered. In the absence of such a registration, the blockchain registration process is called, which makes the company visible and live in the decentralized network. The flow diagram presented in Figure 4 shows the procedure.

In the process of registration, the critical company information—inventory, customer lists, order history—gets transferred from the ERP database to the blockchain. The blockchain ledger stores this information immutably, providing a secure, tamper-proof record of the transactions. The synchronization is significant because the blockchain acts as a tamper-evident shared integrity layer for selected traceability events, while ERP systems, EPCIS repositories, middleware services and IoT gateways remain part of the broader information infrastructure. The system also ensures that each customer in the company's database is registered in the blockchain. The system automatically registers any customer yet to be registered for consistency. This is significant for the integrity of interactions in the SC since each customer placing or receiving an order should be traceable in full in the blockchain. Once the customer registration is complete, the system continues to register each order pertaining to each customer. Each order is verified to determine whether it is already in the blockchain. In the absence of it, the system registers the order, including the details pertaining to it such as order ID, details of the products, and the dates of the transaction. The order registering process is central in ensuring that the whole lifecycle of each transaction, ranging from the placement of the order to its completion, is recorded in the blockchain. This simplifies it for the internal stakeholders and the external partners in the SC to track the status and movement of orders with precision.

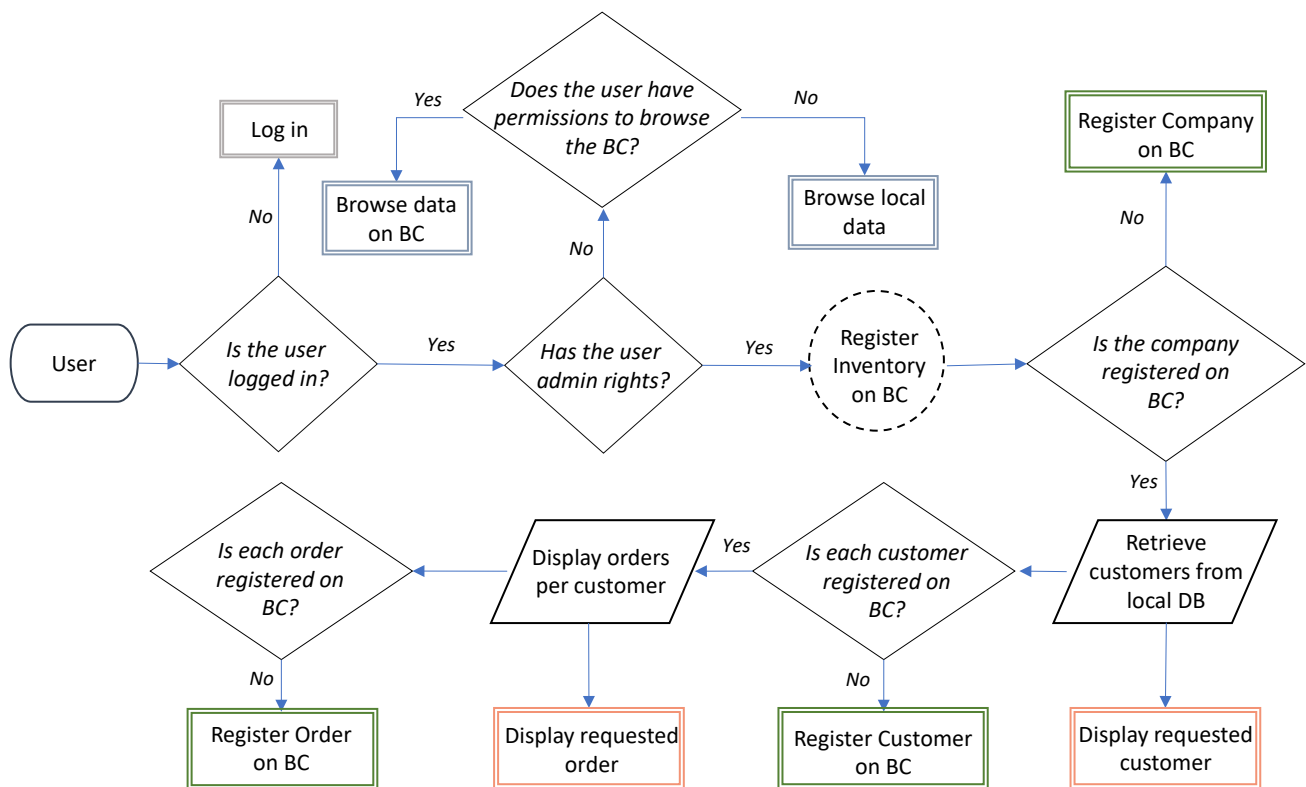


Figure 4. Sequence diagram for Registration and inventorying.

Another fundamental purpose of this phase is the inventory registration. Inventory management in the blockchain is significant for the sake of providing the visibility of the assets available as well as the inventory levels to the respective stakeholders. The company inventory is automatically recorded on the blockchain, with near real-time updating of the inventory records under controlled prototype conditions. The blockchain ensures propagation of such updates to the respective participants, ensuring transparency throughout the entire SC. The process of registration and inventorying is also tightly coupled with the RBAC mechanism. Based on their permissions, users such as administrators and moderators can see inventory and customer data locally or in the blockchain itself. Without the required permissions by the user, the blockchain-based operations are hidden from their interface, ensuring no unauthorized disclosure of confidential data.

4.3. Tokenization

The tokenization process of the system in general is presented in Figure 5. The process begins with the admin login, in which the administrator initiates the workflow. After successful login, the administrator initiates the inventory registration process.

The system verifies the company's resources for whether or not they have been logged onto the blockchain properly. Without the company's resources, the system performs the registration of the present resources, which makes them visible and accessible in the decentralized network. The process ensures that the company's assets are transparently logged in the blockchain so other stakeholders such as the suppliers or consumers can see the latest information regarding the present products. Additionally, the system retrieves information regarding the present vendors in the blockchain. This is a significant feature, as it allows the company to choose the suitable partners or suppliers for the completion of their procurement requirements. Once the necessary resources are established, the system can create a mock order to simulate the order management process. The mock order is locally stored in the company database in order for the ERP system to maintain internal

consistency. In addition, the system records the mock order in the blockchain. The consistency between the decentralized blockchain and the local ERP data at each level of the SC is maintained by the two-step registration. Once the mock order has been registered, the system generates a transaction receipt with the necessary details showing the registration in the blockchain, which can be used as a verifiable proof of transaction execution. It is also stored locally in the ERP system to maintain an internal record of the blockchain transaction. In the current prototype, order, vendor and inventory records are represented through simulated ERP datasets. This allows reproducible testing of the middleware and smart-contract workflow but does not capture the full semantic complexity, customization and integration constraints of commercial ERP systems.

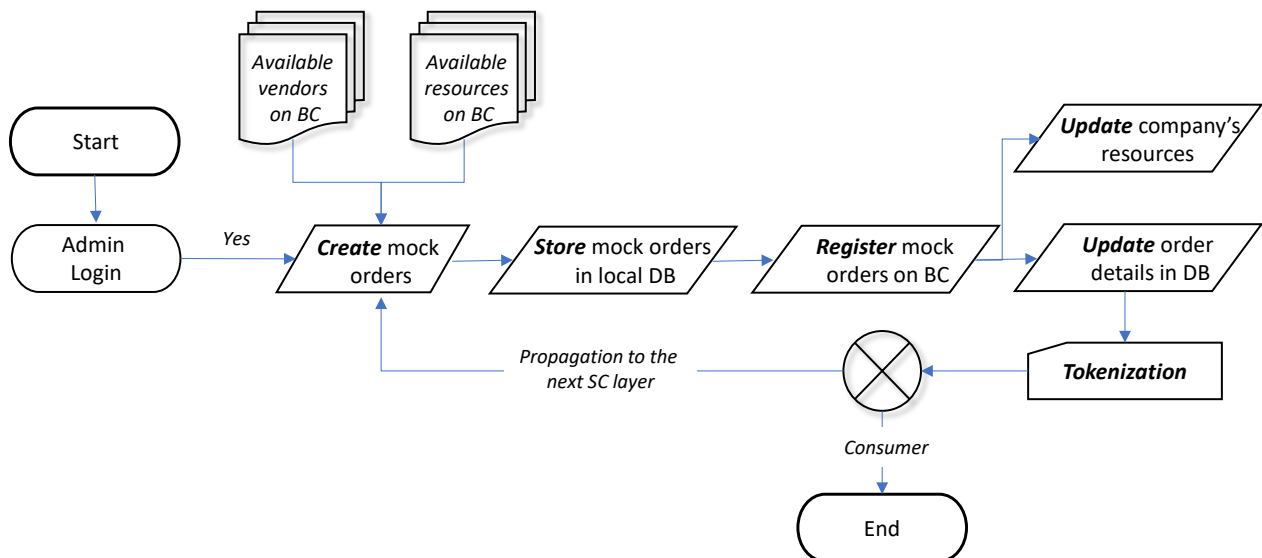


Figure 5. Tokenization flow.

The propagation of data to the next layer in the SC is an important outcome of this phase. Once the assets of the company and mock orders are inserted into the blockchain, the data is propagated to the respective stakeholders in the SC. The propagation makes the data fully visible and traceable such that each stakeholder in the transaction can view the correct data. Figure 6 presents the tokenization process in detail in the proposed blockchain-based ERP system. The tokenization call is the beginning of this phase, which invokes the generation or retrieval of a token object as a unique identifier for a process or a product in the blockchain. The token is the key reference for traceability in the life cycle of the SC.

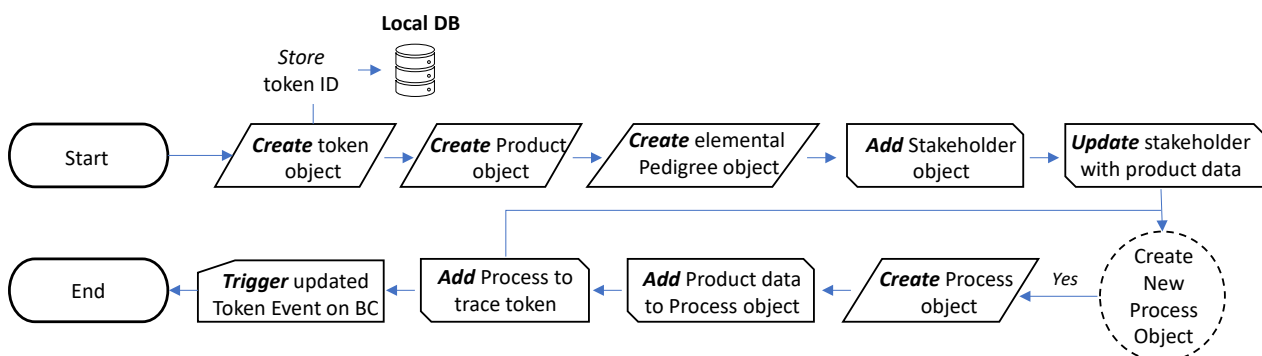


Figure 6. Basic steps of the tokenization process.

Once the token is generated or pulled out of the blockchain, the token ID is saved in the local database (DB) in real time, mapping the token on the blockchain to the company's

internal ERP system. The dual storage ensures synchronization between the blockchain and the ERP system, allowing for seamless integration with the possibility of almost real-time synchronization between the two layers. The local DB is used as a staging area for token operations, ensuring consistency between the blockchain records and the records in the local area. The second operation is the creation of a Product object, which encapsulates the relevant details of the product to be traced. The object stores critical information such as the specifications of the product, the identifiers, and the relevant metadata required for the identification of the product along the SC. Upon the creation of the product object, a Stakeholder object is generated, which represents the entity handling or interacting with the product at the specified stage in the SC. The stakeholders can be the suppliers, manufacturers, logistics providers, or end-consumers.

Once the Stakeholder object is established, the system logs the stakeholder with the product data so the relationship between the entity responsible for the product and the product itself can be established. This relationship is necessary for traceability, as it can track the product itself and the stakeholder role in the SC. The system creates an Elemental Pedigree object at the same time, which is a record of the history of the product, including any processes or transformations the product might have undergone throughout its lifecycle. This object is established in order to maintain a full view of the history of the product, with full transparency and traceability. The Process object is the other key component in this phase, which captures any specific actions or changes made to the product, such as manufacturing operations, quality checks, or shipping procedures. The system then associates Product Data with the Process object, enriching the trace token with accurate data about the lifecycle of the product. The newly created process is added to the Trace Token, ensuring that each change or process made to the product is securely and immutably stored on the blockchain.

The system checks at this point whether there are other processes needed. In the case of new processes, the system creates other Process objects and associates them with the trace token. This ensures the full lifecycle of the product—from raw material purchase to final delivery—is comprehensively documented. Once the required product, stakeholder, and process data are logged, the system creates an updated token event in the blockchain. The event ensures the most up-to-date state of the token, including the related data and transactions, is securely updated in the blockchain. The tokenization process is finalized when the updated token event is created, completing the registration and ensuring that the full history of the product, stakeholders, and processes is securely stored and verifiable in the blockchain. This process operationalizes the SH, PD, and PC tokens described in Section 3. By binding each create or update operation to a validated EPCIS event (ObjectEvent, TransformationEvent, or AggregationEvent), the system ensures standardized event capture and cross-ERP semantic consistency with minimal on-chain overhead.

4.4. Smart Contracts

For implementation purposes, we have created seven fully functional smart contracts that integrate the system's functions, all of which can be found on github (<https://github.com/eeddieg/Blockchain-Based-ERP-Interoperability-in-Supply-Chain-Management/tree/main/blockchain/contracts>, accessed on 18 November 2025). Six of them (namely *processes.sol*, *registerCompany.sol*, *registerOrder.sol*, *registerUser.sol*, *stakeholders.sol* and *transactions.sol*) act as auxiliary contracts that ensure the registration of users, companies, stakeholders and the automation of the related processes and transactions. The main contract is *MutableTokens.sol*, which integrates all the functionality of the tokenization process. The main functionalities of this contract are shown in Table 3 below.

Table 3. Append-only tokens smart contract main functions.

Function	Description
<i>createToken</i>	Creates a new token with specified metadata and an associated creation timestamp.
<i>updateToken</i>	Updates an existing token by appending a new version entry to its history.
<i>changeStatusToken</i>	Changes the activation status of a token, enabling or disabling it as required.
<i>createProduct</i>	Creates a new product entity including its descriptive attributes, timestamp, and hierarchical layer.
<i>addProcess</i>	Associates a process identifier with a token, enabling traceability of operational steps.
<i>addStakeholder</i>	Associates a stakeholder identifier with a token, enabling traceability of involved actors.
<i>getTokenById</i>	Returns the complete metadata and state of a specified token.
<i>getProduct</i>	Retrieves the stored information of a specified product.
<i>getProcessesToken</i>	Returns the list of process identifiers associated with a given token.
<i>getStakeholderToken</i>	Returns the list of stakeholder identifiers associated with a given token.
<i>retrieveHash</i>	Computes and returns a cryptographic hash of the token based on its attributes and block data for integrity verification.

The MutableTokens contract automates the synthesis of SH, PD, and PC tokens into a single verifiable state, as shown in the system output in Figure 7. By recording cryptographic references to pre-validated EPCIS events rather than storing full event payloads, the contract maintains auditability and semantic alignment while optimizing for on-chain storage efficiency.

Actual Trace Token Data

token id	1
is mutable	false
name	L1-Company_A_orderid_8+L2-Company_A_orderid_5+L3-Company_A_orderid_7+L4-Company_A_orderid_5+L5-Company_A_orderid_5+L6-Company_A_orderid_5+L7-Company_A_orderid_5+L8-Company_A_orderid_5
information	Token for WHEAT, WHEAT_SACK, WHEAT_SACK_PALLETE_PRE, WHEAT_SACK_PALLETE_POST, FLOUR, BREAD_PRODUCT, DISTRIBUTED_BREAD_PRODUCT, BREAD_PRODUCT_CONSUMER
timestamp	["1726407861328", "1726408566084", "1726408986437", "1726409123403", "1726409332070", "1726409521375", "1726409774175", "1726410069390"]
pedigree id	[1, 2, 3, 4, 5, 6, 7, 8]
process id	[1, 2, 3, 4, 5, 6, 7, 8]
stakeholder id	[1, 2, 3, 4, 5, 6, 7, 8]
owner	0x1Fd644b3b7f98be0D608525AA08e58020AD84DFE
active	true
hashIPFS	0x00

Figure 7. System token output.

It is important to clarify that update operations supported by the MutableTokens contract do not imply destructive modification of token state. In particular, the `updateToken()` function does not overwrite existing data structures but instead records a new version entry that is cryptographically linked to the prior state of the token. Algorithm 4 defines the controlled mutability mechanism adopted for token updates.

Each update appends a new timestamped state snapshot to the token's on-chain history, while preserving all previously recorded versions. This append-only execution model ensures that token evolution remains fully auditable, as historical states cannot be altered or removed without invalidating subsequent cryptographic references. The `retrieveHash()` function further enables independent verification by allowing external parties to recompute integrity hashes for specific token versions and compare them against on-chain records.

To clarify how the formal logic presented in Algorithms 1–4 is connected to the prototype, Table 4 maps each algorithmic component to the corresponding implementation elements and exception-path checks. The validation focused on controlled application-level and smart-contract-level failure conditions that are directly relevant to traceability

correctness. These tests do not emulate production network failures, validator outages or commercial ERP data conflicts; instead, they verify that invalid inputs and inconsistent token operations are rejected without creating partial or inconsistent blockchain states.

Algorithm 4 Append-Only Versioned Update of Token TP_i

Require: token identifier TP_i , EPCIS event e , mutable update set Δ

Ensure: updated token state or REJECT

```

1: function UPDATETOKEN( $TP_i, e, \Delta$ )
2:    $T \leftarrow \text{GETTOKEN}(TP_i)$ 
3:   if  $T = \emptyset$  then
4:     return REJECT
5:   end if
6:    $h_e \leftarrow H(e)$ 
7:    $v_k \leftarrow T.\text{history}[\text{end}]$ 
8:    $h_k \leftarrow H(v_k)$ 
9:   if  $\text{VALIDATEMUTABLE}(\Delta, T.\text{mutableSchema}) = \text{false}$  then
10:    return REJECT
11:  end if
12:   $v_{k+1} \leftarrow (ts, h_k, h_e, \Delta)$ 
13:   $T.\text{history} \leftarrow T.\text{history} \cup \{v_{k+1}\}$ 
14:   $T.\text{mutable} \leftarrow \text{APPLY}(T.\text{mutable}, \Delta)$ 
15:   $\text{STORETOKEN}(T)$ 
16:  return  $T$ 
17: end function

```

The exception-path validation showed that the prototype follows a fail-closed behavior: When a required condition is not satisfied, the corresponding operation is rejected and the previous valid token state is preserved. This is particularly important for traceability applications because failed updates, invalid EPCIS mappings or incomplete token synthesis operations should not result in partially updated provenance records. Therefore, the algorithms are connected to the implementation not only through the successful execution path but also through explicit rejection paths that preserve state consistency and auditability under controlled failure conditions.

Table 4. Algorithm-to-implementation mapping and exception-path validation.

Algorithm	Prototype Component	Tested Exception Path	Expected Behavior
A1	Middleware updateToken()	validation; Invalid event; policy or scope violation; unresolved traceable unit	Update rejected; no new version appended.
A2	EPCIS validator; mapping policy II; contract calls	Unauthorized caller; missing fields; unsupported or mismatched event	Operation rejected; no token linkage created.
A3	Component lookup; ST_x creation logic	Missing/inactive token; invalid identifier; incomplete lineage	Synthesis aborted; histories unchanged.
A4	updateToken(); hash and version logic	Non-existent token; malformed payload; overwrite attempt; invalid previous hash	Transaction reverted; valid state preserved.

4.5. System Performance and Scalability Evaluation

4.5.1. Operational Cost Analysis

Economic feasibility is critical for enterprise blockchain adoption. We evaluate the framework's operational costs by measuring gas consumption for core smart contract functions, assuming an April 2026 gas price baseline. By utilizing event logging and minimal on-chain data storage (storing only critical hashes while keeping bulk data in

legacy ERPs), the system minimizes transaction overhead and maximizes scalability. To quantify the economic impact of the suggested system, Table 5 reports USD fee estimates for representative smart contract operations.

Table 5. Gas costs and execution costs of smart contract methods (with April 2026 ETH/USD exchange rate).

Method	Tx Gas	Exec. Gas	Slow (USD)	Avg. (USD)	Fast (USD)
<i>createToken()</i>	60,000	50,000	0.162	0.180	0.268
<i>updateToken()</i>	55,000	45,000	0.148	0.162	0.245
<i>changeStatusToken()</i>	30,000	25,000	0.083	0.088	0.134
<i>createProduct()</i>	65,000	55,000	0.176	0.194	0.296
<i>addProcess()</i>	40,000	35,000	0.111	0.125	0.185
<i>addStakeholder()</i>	42,000	37,000	0.116	0.130	0.194
<i>getTokenByID()</i>	20,000	15,000	0.051	0.056	0.088
<i>getProduct()</i>	25,000	20,000	0.065	0.074	0.111
<i>getProcessToken()</i>	28,000	23,000	0.074	0.083	0.125
<i>getStakeholderToken()</i>	30,000	24,000	0.079	0.088	0.134
<i>retrieveHash()</i>	18,000	12,000	0.046	0.051	0.074

We keep the intrinsic gas usage (transaction and execution gas) identical to our prior measurement and update only the price input to reflect current market conditions. As of April 2026, we use an ETH/USD rate of \$1923.17 and three gas levels aligned with widely used trackers: slow (0.76 Gwei), average (0.84 Gwei) and fast (1.26 Gwei). The reported USD values are computed as $\text{USD} = (\text{Tx Gas} + \text{Exec Gas}) \times \text{GasPrice}_{\text{Gwei}} \times 10^{-9} \times \text{ETH/USD}$. It is also worth noting that fees fluctuate with network demand and ETH price and Table 5 provides a reproducible snapshot methodology. The following gas-cost estimates are reported to compare relative smart-contract execution complexity across functions. They should not be interpreted as operational cost estimates for a production blockchain deployment.

In order to calculate the efficiency of the underlying traceability mechanisms, we have also measured end-to-end latencies using wall-clock timestamps for three main operations: (a) ERP-to-chain commit, (b) chain-to-ERP synchronization and (c) lineage queries for k-hop product histories. The computational complexity of a lineage query is bounded by the number of traversed tokens and the imposed hop limit, resulting in a worst-case complexity proportional to the explored lineage depth as depicted in (5).

$$T_{\text{lineage}} = \mathcal{O}(\min(n, k \cdot d)) \quad (5)$$

We executed each experiment 30 times on the Ganache EVM private network, with autominer enabled and the block time set at 0 s. We used multiple blockchain nodes running on Intel i7 with 16GB of RAM machines (Santa Clara, CA, USA). We measured both the median and the 95th percentile (p95) timings to capture typical performance and upper-bound responsiveness under feasibility conditions.

As depicted in Table 6, the median commit latency from ERP to chain was approximately ≈ 90 ms (p95: 125 ms), and chain-to-ERP synchronization had a median of 65 ms (p95: 95 ms), which is low-latency responsiveness in our deployment setting. Lineage traversal queries were below sub-second performance even at 10 hops (median: 410 ms; p95: 620 ms), which suffices for practical exploration of merge/split product histories. In addition, consistency validation had 100% lineage completeness and no duplicate-ID admissions under idempotent retries ($n = 200$ events). These feasibility-grade outcomes show that the proposed stack provides both responsiveness and correctness at demo scale, while we recognize the need for additional optimization for industrial-level throughput.

Table 6. End-to-end execution latency under feasibility testing (private EVM, automining enabled).

Operation	Path	Definition	Median (ms)	p95 (ms)	n
Commit token	ERP → Chain	API call to on-chain commit	90	125	30
Update token	ERP → Chain	API call to state update	95	130	30
Emit EPCIS event	ERP → Chain	Event persisted on-chain	88	118	30
Sync to ERP view	Chain → ERP	On-chain update to UI	65	95	30
Lineage query (5 hops)	ERP → UI	Query resolution (5 hops)	210	320	30
Lineage query (10 hops)	ERP → UI	Query resolution (10 hops)	410	620	30

The evaluation environment (Ganache with automining enabled and zero block time) was deliberately chosen to isolate the correctness, determinism and application-layer overhead of the proposed ERP–blockchain integration rather than to emulate production blockchain conditions. As a result, the reported latency figures abstract away dominant sources of delay and variance in operational deployments, including consensus finality, block scheduling, network propagation and mempool contention. Therefore, the reported figures should be interpreted as proof-of-concept measurements rather than indicators of production performance. While the framework enables dynamic control of traceability granularity at the information-model level, assessing the throughput, backlog behavior and performance implications of fine-grained traceability under realistic network conditions is beyond the scope of this prototype and is left for future work.

4.5.2. Parametric Assessment of Traceability Workloads

To further assess the operational implications of the proposed framework, we conduct a parametric evaluation of traceability workloads under increasing supply-chain complexity and traceability granularity. Rather than benchmarking blockchain platforms at the infrastructure level, the analysis focuses on the information-processing effort required to resolve traceability queries and recall scenarios, which constitutes a core concern in industrial engineering and supply-chain operations. The evaluation models traceability as a workload reconstruction problem, where the objective is to identify affected products, upstream sources and responsible stakeholders following a disruption or recall event. A common parametric model is applied across alternative traceability architectures, enabling a consistent comparison of how traceability effort scales as the number of transformations, actors and traceability resolution increase.

Workload Model

Let the supply chain consist of S stages and T transformation steps, where products evolve through aggregation, processing or splitting operations. Traceability granularity is represented by G , corresponding to batch-, lot- or unit-level tracking, while A_i denotes the number of actors involved at transformation step i . A recall event is assumed to affect a fraction R of downstream products, requiring backward and forward traceability resolution. The traceability workload is evaluated along three dimensions: (i) coordination steps required to reconstruct lineage, (ii) query latency as a function of lineage depth, and (iii) growth of traceability data as granularity increases.

Recall Resolution Effort

The recall resolution effort is approximated by the number of coordination or traversal steps required to reconstruct complete lineage information across T transformations. For different traceability architectures, the effort can be expressed as follows:

$$E_{\text{ERP}} = \sum_{i=1}^T A_i, \quad (6)$$

$$E_{\text{EPCIS}} = T + J, \quad (7)$$

$$E_{\text{NFT}} = \sum_{i=1}^T N_i, \quad (8)$$

$$E_{\text{Proposed}} = T, \quad (9)$$

where J denotes the number of cross-organizational joins required to correlate EPCIS event repositories, and N_i denotes the number of batch-level tokens traversed at transformation i . In the proposed framework, recall resolution requires direct traversal of synthesized token lineage, resulting in effort that scales linearly with the number of transformations.

Trace Query Latency

Trace query latency is modeled as a function of lineage depth k . Using the feasibility measurements reported in Section 4.5, we approximate the end-to-end query latency as follows:

$$L(k) = k \cdot \ell, \quad (10)$$

where ℓ represents the measured per-hop traversal latency. For the private EVM deployment used in this study, ℓ corresponds to the median lineage traversal cost observed in Table 3. Architectures requiring additional joins or token dereferencing incur higher effective per-hop latency due to off-chain reconciliation or token proliferation.

Data Volume Growth and Redundancy

Traceability data growth and system redundancy are evaluated as a function of the number of supply chain stakeholders (S), product units or tracked batches (P), and transformation processes (Pr). Traditional architectures, such as full event anchoring or NFT-per-batch models, scale multiplicatively, generating redundancy proportional to $\mathcal{O}(S \cdot P \cdot Pr)$ as every node duplicates complete histories or token lineages [1]. In these baseline models, data overhead is heavily exacerbated by secondary variables: J represents the number of cross-organizational joins required to correlate fragmented, off-chain EPCIS event repositories, while N_i denotes the number of individual batch-level tokens generated at each transformation step i , leading to severe token proliferation across the network. Conversely, off-chain centralized or EPCIS databases limit baseline redundancy to $\mathcal{O}(P \cdot Pr)$ but lack decentralized auditability. The proposed framework drastically reduces this footprint by utilizing the synthesized token (ST_x), which aggregates cryptographic references rather than duplicating payloads. Consequently, the on-chain data volume scales additively as $\mathcal{O}(S + P + Pr)$, mitigating state explosion and token proliferation even as traceability resolution shifts from batch-level to unit-level tracking. Specifically, in the proposed architecture, the redundancy remains exceptionally lower at the synthesized-token level because ST_x stores references to data components rather than duplicating full histories on-chain.

Figure 8 illustrates the scaling behavior of recall resolution effort, trace query latency, and traceability data growth under increasing supply-chain complexity and granularity. To clarify the trade-offs of the proposed architecture, Table 7 compares it conceptually against alternative traceability architectures.

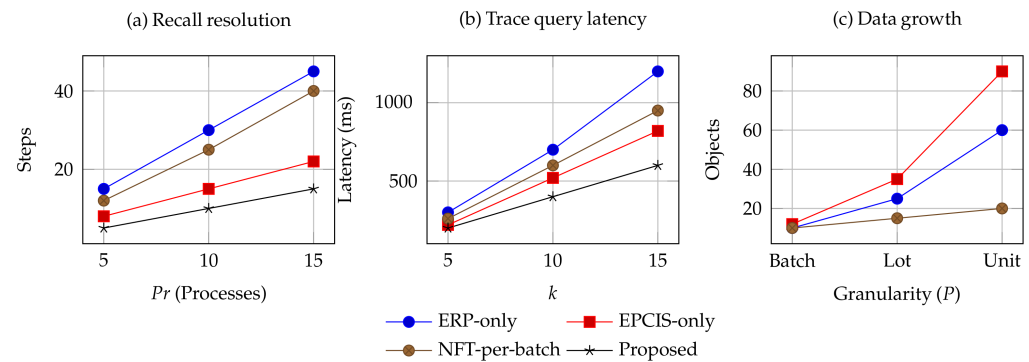


Figure 8. Parametric comparison of traceability workloads: (a) recall resolution effort versus number of transformations Pr , (b) trace query latency versus lineage depth k , and (c) traceability data growth as granularity increases.

Table 7. Conceptual comparison of alternative traceability architectures.

Architecture		Auditability	Granularity (Formula)	Redundancy	ERP Integration	Privacy Control	Implementation Complexity
Centralized database	off-chain	Low-medium	Configurable: $\mathcal{O}(P \cdot Pr)$	Low	High within one firm; weak across firms	High	Low
EPCIS repository without blockchain		Medium	Medium-high: $\mathcal{O}(P \cdot Pr) + J$	Medium	Medium-high	Medium-high	Medium
Blockchain-EPCIS with full event anchoring		High	Medium-high: $\mathcal{O}(S \cdot P \cdot Pr)$	High	Medium	Medium	High
Blockchain-EPCIS with off-chain payloads and hashes		High for anchored events	Medium-high: $\mathcal{O}(S \cdot \sum N_i)$	Low	Medium	Medium-high	High
Proposed framework		High for token lineage and event hashes	High: $\mathcal{O}(S + P + Pr)$	Low	Prototype-level middleware support	Baseline RBAC	High

4.6. Security and Privacy

Security and privacy are among the most important factors in the uptake of enterprise blockchain-based solutions, particularly in supply chain (SC) management applications. In this section, we elaborate on the critical capabilities of the proposed blockchain-based tracking system, which is developed specifically to address primary security requirements such as data confidentiality, data integrity, system availability, non-repudiation, and resistance to malicious attacks. For the purposes of our analysis, we rely on the STRIDE framework. By highlighting these security and privacy features, we demonstrate how the system supports trust, resilience, and reliable operation while safeguarding sensitive information and enabling secure SC processes.

A token TP_i maintains an append-only history $T.history = \{v_i^1, \dots, v_i^k\}$, where each version is defined as $v_i^k = (t_k, H(v_i^{k-1}), H(e_k), \Delta_k)$. Let $GETTOKEN(TP_i)$ return the on-chain token state and history. Let $AUTHORIZE(u, e)$ denote the on-chain RBAC check (or middleware-assisted on-chain check) that returns true if and only if caller u has permission to invoke the operation associated with EPCIS event e under policy Π . We assume that $H(\cdot)$ is resistant to collisions.

4.6.1. Authorization and Access Control Mechanisms

SCs enforce RBAC, restricting the execution of fundamental functions to authorized participants. Access control policies prevent unauthorized changes to SC records, ensuring that only authenticated manufacturers, distributors, or regulators can execute core SC functions. From a threat-modeling perspective, this design directly mitigates spoofing

and privilege escalation risks (STRIDE), as each operation is cryptographically bound to a verified identity and role. By enforcing authorization logic at the smart contract level, the system prevents unauthorized entities from impersonating legitimate actors or invoking restricted functions, thereby ensuring that all state transitions are attributable to authenticated and authorized participants.

Lemma 1 (Authorization safety). *Assuming that all state-transition functions in smart contracts are guarded by $\text{AUTHORIZE}(u, e)$ and that private keys are not compromised, no unauthorized caller can induce a valid state transition (i.e., append a new version entry v_i^{k+1} to any TP_i).*

Proof. A valid state transition requires successful execution of the corresponding smart contract operation (e.g., *createToken*, *updateToken*). Each such operation is preceded by $\text{AUTHORIZE}(u, e)$, which deterministically rejects callers lacking the required role or permission. Under the assumption that private keys are not compromised, an adversary cannot generate a transaction that verifies as originating from an authorized identity and therefore cannot pass $\text{AUTHORIZE}(u, e)$. Consequently, the transition cannot occur. $\square$

4.6.2. Confidentiality and Privacy

Blockchain networks use pseudonymization techniques to protect participant identities while maintaining transaction transparency. Smart contracts enforce access control policies, maintaining business data security while enabling traceability. Manufacturers and distributors in SCs are represented by hashed identifiers in order to preserve business confidentiality while enabling oversight. From a threat-modeling perspective, these mechanisms primarily mitigate information disclosure risks (STRIDE), as they limit the exposure of sensitive business identities and transactional metadata to authorized entities while preserving verifiability for accountability and auditing.

Proposition 1 (Bounded on-chain identity disclosure). *If stakeholder identities are represented on-chain only via hashed identifiers (e.g., $ID_u \mapsto H(ID_u)$) and no plaintext identifiers are stored on-chain, then the blockchain state reveals at most these pseudonymous identifiers and associated activity patterns.*

Proof. The blockchain contains only hashed identifiers by construction. As a result, direct disclosure of plaintext identities does not occur from on-chain data alone. However, linkage through auxiliary information (e.g., timing, transaction volumes, or counterparties) remains possible and is not prevented by hashing. $\square$

4.6.3. Data Integrity and Immutability

Blockchain technology ensures strong data integrity guarantees. Transactions are cryptographically recorded on-chain and cannot be altered without detection, rendering them tamper-resistant. This ensures that order placements, product status updates, and approval records are preserved as permanent and verifiable artifacts throughout the SC lifecycle. From a threat-modeling perspective, these mechanisms mitigate tampering threats (STRIDE) by enforcing append-only state transitions and cryptographic linkage between successive records.

Lemma 2 (Tamper-evident token history). *Let TP_i have history entries $v_i^1, \dots, v_i^k$ stored on-chain. If an adversary modifies any past entry v_i^j (for some $1 \leq j \leq k$) without rewriting subsequent history, then verification of the chain of hashes fails, i.e., there exists $\ell > j$ such that $H(v_i^{\ell-1}) \neq v_i^\ell.\text{prevHash}$.*

Proof. Each v_i^ℓ stores $prevHash = H(v_i^{\ell-1})$. If v_i^j is altered to $\tilde{v}_i^j \neq v_i^j$, then, under collision resistance, $H(\tilde{v}_i^j) \neq H(v_i^j)$ except with negligible probability. As a result, the linkage check fails at $\ell = j + 1$. $\square$

Lemma 3 (Event anchoring integrity). *For any recorded version v_i^k , if the referenced EPCIS event payload is altered off-chain from e_k to $\tilde{e}_k \neq e_k$, then the on-chain reference $H(e_k)$ does not match $H(\tilde{e}_k)$ except with negligible probability.*

Proof. The version stores the event hash $H(e_k)$. Any modification to the payload changes the hash unless a collision is found. Under collision resistance, $H(\tilde{e}_k) = H(e_k)$ occurs with negligible probability. $\square$

4.6.4. Availability and Resilience

Blockchain technology enhances system availability and resilience by mitigating risks associated with denial-of-service (DoS) attacks through its decentralized architecture. Because the ledger operates across multiple distributed nodes, critical operations—such as order placement, dispatch tracking, and ownership verification—remain functional even in the presence of partial node failures or targeted disruption attempts. This distributed execution model preserves continuity of service and reinforces overall system robustness.

Proposition 2 (No single point of failure in the ledger layer). *In a distributed blockchain network with N nodes, ledger availability persists under failure of up to f nodes, provided that the remaining nodes satisfy the network's liveness assumptions.*

Proof. This property depends on the consensus protocol and deployment configuration rather than on the tokenization mechanism itself. $\square$

4.6.5. Non-Repudiation and Accountability

Blockchain transactions are cryptographically signed using private keys, binding each action to its originating entity. This ensures non-repudiation and prevents participants from denying their involvement in recorded operations. All interactions—including SC exchanges, contractual actions, and product transfers—are immutably recorded on the ledger, enabling traceability and accountability. From a threat-modeling perspective, this mitigates repudiation risks (STRIDE) by providing verifiable, tamper-resistant evidence of actions, while audit trails support post hoc verification.

Lemma 4 (Non-repudiation under signature security). *Assume that the digital signature scheme used by the blockchain is existentially unforgeable under chosen-message attacks (EUF-CMA) and that private keys are not compromised. Then, any successfully mined token operation that appends v_i^{k+1} is attributable to the signer of the corresponding transaction, and that signer cannot plausibly deny issuing it.*

Proof. Each on-chain state transition is triggered by a transaction signed with the caller's private key. Under EUF-CMA security, an adversary cannot forge a valid signature for an honest user's public key. Therefore, if the ledger accepts a transaction as signed by identity u , it must have been produced using u 's private key. Because the ledger is append-only and verifiable, the transaction provides persistent evidence of origin. $\square$

Table 8 summarizes the coverage of STRIDE threat categories by the proposed mechanisms and formal guarantees. The STRIDE analysis provides a structured threat-modeling assessment of the proposed framework. The lemmas and propositions specify the security

properties expected under standard cryptographic assumptions and correct implementation of the access-control logic. In real deployments, practical security testing will require formal smart-contract verification, penetration testing or production security certification.

Table 8. STRIDE threat coverage by system mechanisms and formal guarantees.

STRIDE Threat	Mitigation Mechanism	Formal
Spoofing	Role-based access control enforced by SC and cryptographic identities	Lemma 1
Tampering	Append-only token history and cryptographic hash chaining	Lemma 2
Tampering	On-chain anchoring of EPCIS event hashes	Lemma 3
Repudiation	Digitally signed transactions and immutable ledger	Lemma 4
Information Disclosure	Pseudonymized on-chain identities and access-controlled data exposure	Proposition 1
Denial of Service	Decentralized ledger replication and consensus-based availability	Proposition 2
Elevation of Privilege	Smart-contract-level authorization checks bound to predefined roles	Lemma 1

5. Discussion

In this paper, we presented a blockchain-based framework to offer advanced granularity features in SC traceability while taking into account interoperability requirements among disparate ERP systems. The proposed solution relies upon a relatively lightweight traceability mechanism versatile enough to accommodate multiple traceability scenarios in real-life applications. The solution also uses append-only tokens and smart contracts, which are designed to support product, stakeholder, and process traceability throughout the SC. The use of append-only tokens allows tracking complex product transformations while addressing limitations of previous approaches that were either redundant or difficult to scale. By elevating EPCIS from a passive interoperability format to a semantic control layer that constrains token mutations through standardized event types and business steps, the framework strengthens process correctness, interoperability and auditability without increasing on-chain complexity. We provided a proof-of-concept prototype implementation of the proposed framework in the wheat SC and demonstrated its practicality and impact on SC transparency, trust and efficiency.

The implementation details provided for the proof-of-concept wheat SC showed how blockchain technology can be designed to track agricultural products from production to consumer. Using a tokenization model based on the BOM, the framework is designed to capture data at each stage of the SC, aiming to enable provenance and authenticity, while supporting standardized data sharing through EPCIS. The RBAC mechanism ensures that only relevant stakeholders can access the required information without compromising data security, a key concern in globalized multi-stakeholder SC environments. Arguably, this research provides a solid foundation for addressing long-standing issues related to SC data consistency, process transparency and operational efficiency through a secure solution adaptable to different industries.

The proposed framework differs from similar studies by combining three pivotal elements: (1) EPCIS-driven standardization, (2) dynamic granularity control and (3) advanced tokenization using append-only tokens into a unified system for ERP interoperability in SC management. Unlike many existing studies that address these aspects in isolation, our approach integrates EPCIS to capture standardized SC events, granularity mechanisms to tailor the level of traceability detail based on contextual needs and a tokenization model that tracks product transformations and stakeholder interactions. This unified approach enhances data consistency and transparency while bridging the interoperability gap among disparate ERP systems and stakeholders.

Although other studies outline important approaches for defining granularity levels and structuring blockchain-based traceability through immutable tokens, our research operationalizes these concepts within a tangible use-case scenario. The integration of

EPCIS ensures ERP interoperability while append-only tokens and smart contracts achieve dynamic end-to-end traceability. In addition, the proposed framework resolves several impediments flagged in the literature, including limited ERP integration, lack of shared identifiers and insufficient support for reproducible deployments. Through the bi-directional ERP–ledger synchronization pipeline, the BOM-aligned SH/PD/PC tokens and the fine-grained on-chain RBAC model, the framework significantly extends existing theoretical and operational approaches.

A comparison with related studies further substantiates the orchestration of the proposed framework, particularly regarding token usage. Although recent studies demonstrate the potential of tokenization for traceability, most remain limited to static asset representation or conceptual prototypes. For example, [47] introduced a compositional token logic but relied on immutable tokens and lacked support for dynamic granularity or ERP interoperability. Similarly, many tokenization schemes rely on immutable ERC-721/1155 assets, forcing re-minting during transformations and fixing granularity upstream the SC [48]. In contrast, our second-layer design uses append-only SH/PD/PC tokens with synthesis, ERP/EPCIS anchoring and on-chain RBAC to preserve lineage, capture process state and dynamically adjust granularity without token proliferation. While Harish et al. [33] introduced digital asset tokens for logistics financing, their approach does not model the transformation logic required for material traceability. To our knowledge, no prior study combines append-only compositional tokens, EPCIS interoperability and dynamic granularity control within a unified operational system.

The proposed framework also introduces several innovations regarding granularity. Within the framework, granularity is treated not as a fixed system property but as a controllable information-processing variable that mediates the trade-off between traceability precision, interoperability and operational cost. Alamsyah et al. [41] proposed a traceability system structured around fixed compliance checkpoints, while Zhang et al. [42] improved traceability resolution through trusted identification mechanisms but maintain a uniform granularity level across stages and users. Similarly, Haouari et al. [49] advance transportation transaction storage without addressing ERP interoperability, lifecycle-aware granularity or fine-grained on-chain access control. In contrast, our framework leverages elemental and compound append-only tokens integrated with EPCIS to encode SC events and transformations with fine-grained or coarse-grained detail as required. This design ensures traceability information remains configurable and aligned with operational practices and ERP systems.

5.1. Managerial Implications and Generalization

The proposed blockchain-based ERP interoperability framework improves SC management by increasing transparency, efficiency, and trust among stakeholders. By leveraging blockchain's immutable and decentralized nature, the framework is designed to ensure that transactions, product transformations and stakeholder interactions are recorded on a secure and tamper-proof ledger. However, it should be noted that these practical industry benefits and the scalability of the system remain to be fully validated in real-world settings. This level of transparency addresses one of the most critical challenges in SC management, namely verifying the origin and authenticity of products. The wheat SC use case demonstrated that such an approach improves accountability throughout the SC, from raw material suppliers to distributors and end customers.

Apart from high-level managerial implications, the proposed solution also contributes to operations management practice. The framework converts SC traceability data into stateful records since each event (who/what/when/where/how) is bound into specialized tokens and later synthesized into an auditable product pedigree. Therefore, SC granularity

becomes an adjustable mechanism and logisticians gain precise access to key information across the SC. For high-risk or high-value products, managers may “zoom in” to lot/item level and capture additional process details, whereas for routine products they may “zoom out” to the batch level with fewer events. Since these decisions are configuration-based (token scope, event frequency, and RBAC) on top of existing ERP systems, traceability detail can be adjusted without rebuilding operational infrastructures. In this way, ERPs remain the operational system of record while blockchain acts as the composition and verification layer. The integration of blockchain with existing or legacy ERP systems also promotes operational efficiency by streamlining information exchange among stakeholders. Traditional SC operations often struggle with data silos and inconsistencies, especially when multiple ERP systems are involved. The proposed framework can help mitigate these barriers through tokenization and smart contracts that automate processes such as inventory management, order tracking and quality control. This automation reduces manual interventions, minimizes errors and may reduce reconciliation effort and support more timely transaction processing under appropriate deployment conditions. At the same time, increased granularity reallocates accountability by binding product states and transformations to specific actors and standardized events, thus narrowing liability attribution in cases of non-compliance or quality failure. The combination of EPCIS-standardized events and immutable token histories also shifts regulatory inspection from retrospective documentation toward continuous event-driven auditability. However, higher granularity and supporting infrastructure introduce non-trivial information capture and governance costs. As such, managers should treat granularity not merely as a technical parameter but as a strategic governance variable shaping data sovereignty, regulatory oversight, risk exposure and cost allocation across the SC.

Finally, the proposed framework can be generalized and adapted across multiple industries. Its modular design and transparency make it suitable for domains such as healthcare, agriculture and manufacturing where traceability, accountability and data integrity are critical. Furthermore, the platform’s open-source implementation fosters collaboration and allows researchers and practitioners to extend or customize the framework according to sector-specific requirements. The proposed solution can therefore be adapted by preserving the logic of the second-layer architecture, EPCIS event model, ERP anchoring, RBAC and token synthesis while adapting sector-specific identifiers, payloads and granularity policies.

5.2. Limitations and Future Research

While the proposed blockchain-based ERP interoperability architecture delivers significant contributions, several limitations must be acknowledged to contextualize its scope and guide future research. First, in the presented proof-of-concept prototype implementation, the ERP layer is simulated through ExpressJS-based middleware, Prisma ORM and local databases to demonstrate the feasibility of the integration pattern. To fully evaluate the schema’s heterogeneity, customization, data conflicts, failure recovery mechanisms and organizational constraints on real ERP ecosystems, further validation is required through direct connectors to commercial ERP platforms.

Second, the performance evaluation relies on a private Ethereum environment with automining and zero block time, which is suitable for isolating contract and application-layer behavior but does not capture production blockchain constraints. Future work should evaluate the framework under realistic block times, concurrent workloads, multi-node deployments, and alternative permissioned blockchain configurations.

Third, the framework does not eliminate the “garbage-in, garbage-out” problem. Blockchain anchoring ensures tamper-evidence after data have been recorded, but it does

not independently verify the correctness of ERP records, IoT measurements or human-entered data at the point of entry. This limitation is especially relevant in agricultural supply chains involving small farmers, cooperatives and heterogeneous data-entry practices. Future deployments should combine authenticated middleware APIs, ERP audit trails, IoT device attestation and contractual accountability mechanisms.

Fourth, GDPR and data-erasure requirements remain challenging in blockchain-based traceability systems. The proposed architecture minimizes exposure by storing personal or commercially sensitive payloads off-chain and anchoring only hashes or references on-chain. Nevertheless, immutable on-chain references may still create legal and technical tensions with rectification and erasure rights. Therefore, GDPR compliance should be treated as a combined architectural, organizational and legal governance issue rather than as a problem solved solely by off-chain storage. Techniques such as redactable ledgers, chameleon hashes, privacy-preserving commitments and zero-knowledge proofs represent promising directions for future research.

6. Conclusions

In this paper, we have presented a second-layer blockchain-enabled traceability framework that could be integrated with ERP-oriented middleware to support verifiable SC traceability and provenance. The framework uses a BOM-aligned approach, an append-only versioned token system consisting of Stakeholder Tokens (SH), Product Tokens (PD), Process Tokens (PC) and a Synthesized Token (ST_x) to represent product lineage through the lifecycle. This approach supports dynamic granularity and controlled access to provenance information, while EPCIS-standardized events provide a common semantic layer between ERP-like systems and blockchain-based integrity anchoring. A reproducible implementation is provided using private Ethereum/Ganache, smart contracts demonstrated with a wheat SC use case. By maintaining core functionalities while adjusting payloads, the proposed framework is adaptable across various sectors and versatile enough to accommodate different SC operational requirements and traceability prerequisites. The benefits of the proposed framework include, among others, policy-driven granularity determinants, automated assurance reporting, faster dispute resolution, improved coordination and efficient operationalization of tasks like product recall. Compared to previous models, this solution offers a unified, ERP-interoperable approach that addresses transformation and governance issues.

Author Contributions: Conceptualization, V.M. and G.L.; methodology, P.K.; software, V.M.; validation, T.G.V. and T.K.D.; formal analysis, V.M.; investigation, G.L.; resources, P.K.; data curation, V.M.; writing—original draft preparation, V.M. and G.L.; writing—review and editing, P.K., T.G.V., and T.K.D.; visualization, V.M.; supervision, P.K.; project administration, T.G.V. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partly supported by the University of Piraeus Research Center.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author(s).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ahmed, W.A.; MacCarthy, B.L. Blockchain-enabled supply chain traceability—How wide? How deep? *Int. J. Prod. Econ.* **2023**, *263*, 108963. [\[CrossRef\]](#)
2. Nawaz, A.; Wang, L.; Irfan, M.; Westerlund, T. Hyperledger sawtooth based supplychain traceability system for counterfeit drugs. *Comput. Ind. Eng.* **2024**, *190*, 110021. [\[CrossRef\]](#)
3. Qian, J.; Li, J.; Geng, B.; Chen, C.; Wu, J.; Li, H. Finding Traceability Granularity Influencing Factors Using Rough Set Method: An Empirical Analysis of Vegetable Companies in Tianjin City, China. *Foods* **2023**, *12*, 2124. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Karlsen, K.; Donnelly, K.M.; Olsen, P. Granularity and its importance for traceability in a farmed salmon supply chain. *J. Food Eng.* **2011**, *102*, 1–8. [\[CrossRef\]](#)
5. Agrawal, T.K.; Kumar, V.; Pal, R.; Wang, L.; Chen, Y. Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry. *Comput. Ind. Eng.* **2021**, *154*, 107130. [\[CrossRef\]](#)
6. Roy, V. Contrasting supply chain traceability and supply chain visibility: are they interchangeable? *Int. J. Logist. Manag.* **2021**, *32*, 942–972. [\[CrossRef\]](#)
7. Lusiantoro, L.; Yates, N.; Mena, C.; Varga, L. A refined framework of information sharing in perishable product supply chains. *Int. J. Phys. Distrib. Logist. Manag.* **2018**, *48*, 254–283. [\[CrossRef\]](#)
8. Kelle, P.; Akbulut, A. The role of ERP tools in supply chain information sharing, cooperation, and cost optimization. *Int. J. Prod. Econ.* **2005**, *93–94*, 41–52. [\[CrossRef\]](#)
9. Rashid, A.; Baber Ali, S.; Rasheed, R.; Amirah, N.A.; Ngah, A.H. A paradigm of blockchain and supply chain performance: A mediated model using structural equation modeling. *Kybernetes* **2023**, *52*, 6163–6178.
10. Rashid, A.; Rasheed, R.; Ngah, A.H.; Pradeepa Jayaratne, M.D.R.; Rahi, S.; Tunio, M.N. Role of information processing and digital supply chain in supply chain resilience through supply chain risk management. *J. Glob. Oper. Strateg. Sourc.* **2024**, *17*, 429–447. [\[CrossRef\]](#)
11. Omar, I.A.; Debe, M.; Jayaraman, R.; Salah, K.; Omar, M.; Arshad, J. Blockchain-based Supply Chain Traceability for COVID-19 personal protective equipment. *Comput. Ind. Eng.* **2022**, *167*, 107995. [\[CrossRef\]](#)
12. Malamas, V.; Dasaklis, T.K.; Voutsinas, T.G.; Kotzanikolaou, P. Blockchain service layer for erp data interoperability among multiple supply chain stakeholders. In *Proceedings of the 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*; IEEE: New York, NY, USA, 2023; pp. 145–150.
13. Agrawal, T.K.; Angelis, J.; Khilji, W.A.; Kalaiarasan, R.; Wiktorsson, M. Demonstration of a blockchain-based framework using smart contracts for supply chain collaboration. *Int. J. Prod. Res.* **2023**, *61*, 1497–1516. [\[CrossRef\]](#)
14. Xu, X.; Chen, X.; Chen, J.; Cheng, T.; Yu, Y.; Liu, S.S. Adoption of blockchain considering platform's information sharing and service effort under the cap-and-trade scheme. *Int. J. Prod. Res.* **2024**, *62*, 6688–6712. [\[CrossRef\]](#)
15. Sunny, J.; Undralla, N.; Madhusudanan Pillai, V. Supply chain transparency through blockchain-based traceability: An overview with demonstration. *Comput. Ind. Eng.* **2020**, *150*, 106895. [\[CrossRef\]](#)
16. Yang, L.; Ni, Y.; Ng, C.T. Blockchain-enabled traceability and producer's incentive to outsource delivery. *Int. J. Prod. Res.* **2023**, *61*, 3811–3828. [\[CrossRef\]](#)
17. Wu, X.Y.; Pu, X.; Fan, Z.P. Blockchain technology adoption strategies of a supply chain considering possible product defects. *Int. J. Prod. Res.* **2024**, *63*, 5190–5216. [\[CrossRef\]](#)
18. Ahuja, R.; Chugh, S.; Singh, R. SeedChain: A Secure and Transparent Blockchain-Driven Framework to Revolutionize the Seed Supply Chain. *Future Internet* **2024**, *16*, 132. [\[CrossRef\]](#)
19. Lakkakula, P.; Bullock, D.W.; Wilson, W.W. Asymmetric information and blockchains in soybean commodity markets. *Appl. Econ. Perspect. Policy* **2022**, *44*, 273–298. [\[CrossRef\]](#)
20. Radeva, I.; Popchev, I. Blockchain-Enabled Supply-Chain in Crop Production Framework. *Cybern. Inf. Technol.* **2022**, *22*, 151–170. [\[CrossRef\]](#)
21. Peng, X.; Zhang, X.; Wang, X.; Li, H.; Xu, J.; Zhao, Z. Multi-Chain Collaboration-Based Information Management and Control for the Rice Supply Chain. *Agriculture* **2022**, *12*, 689. [\[CrossRef\]](#)
22. Rathore, S.; Gupta, N.; Rathore, A.S. Blockchain-based smart wheat supply chain model in Indian context. *Adv. Ser. Manag.* **2022**, *27*, 77–96. [\[CrossRef\]](#)
23. Shao, P.; Kamaruddin, N.S.; Wang, D.; Huang, Y. Integration and Analysis of Data in Grain Quality and Safety Traceability Using Blockchain Technology. *J. Logist. Informatics Serv. Sci.* **2023**, *10*, 47–61. [\[CrossRef\]](#)
24. Xu, J.; Han, J.; Qi, Z.; Jiang, Z.; Xu, K.; Zheng, M.; Zhang, X. A Reliable Traceability Model for Grain and Oil Quality Safety Based on Blockchain and Industrial Internet. *Sustainability* **2022**, *14*, 15144. [\[CrossRef\]](#)
25. Hu, S.; Jin, Y.; Qin, X. Blockchain-facilitated quality traceability and pre-sale inspection: Influencing geographical indication supply chain contracts and farmer quality decisions. *Comput. Ind. Eng.* **2025**, *200*, 110769. [\[CrossRef\]](#)
26. Iftekhhar, A.; Cui, X.; Yang, Y. Blockchain technology for trustworthy operations in the management of strategic grain reserves. *Foods* **2021**, *10*, 2323. [\[CrossRef\]](#) [\[PubMed\]](#)

27. Dietrich, F.; Louw, L.; Palm, D. Blockchain-Based Traceability Architecture for Mapping Object-Related Supply Chain Events. *Sensors* **2023**, *23*, 1410. [[CrossRef](#)] [[PubMed](#)]
28. Malamas, V.; Koutras, D.; Kotzanikolaou, P. Uninterrupted trust: Continuous authentication in blockchain-enhanced supply chains. In *Proceedings of the 2023 8th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*; IEEE: New York, NY, USA, 2023; pp. 1–6.
29. Munawar.; Mugiono, A. Framework for smart contract blockchain in halal traceability, integrity, and transparency. *Int. J. Electr. Comput. Eng.* **2024**, *14*, 2875–2884. [[CrossRef](#)]
30. Li, L.; Qu, H.; Wang, H.; Wang, J.; Wang, B.; Wang, W.; Xu, J.; Wang, Z. A Blockchain-Based Product Traceability System with Off-Chain EPCIS and IoT Device Authentication. *Sensors* **2022**, *22*, 8680. [[CrossRef](#)] [[PubMed](#)]
31. Chen, L. Tokenization-enabled sustainable deep-tier supply chain finance. *Int. J. Prod. Res.* **2026**, 1–23. [[CrossRef](#)]
32. Zülch, F.; Holle, M.; Hofmann, A. Theoretical design of blockchain-based traceability for organic egg supply chains according to regulation (EU) 2018/848. *PLoS ONE* **2024**, *19*, 0304791. [[CrossRef](#)] [[PubMed](#)]
33. Rachana Harish, A.; Liu, X.; Li, M.; Zhong, R.Y.; Huang, G.Q. Blockchain-enabled digital assets tokenization for cyber-physical traceability in E-commerce logistics financing. *Comput. Ind.* **2023**, *150*, 103956. [[CrossRef](#)]
34. Liu, J.; Jiang, P.; Zhang, J. A blockchain-enabled and event-driven tracking framework for SMEs in supply chains. *Comput. Ind. Eng.* **2024**, *188*, 109955. [[CrossRef](#)]
35. Ahmed, W.A.H.; Maccarthy, B.L. Blockchain-enabled supply chain traceability in the textile and apparel supply chain: A case study of the fiber producer, lenzing. *Sustainability* **2021**, *13*, 10496. [[CrossRef](#)]
36. Westerkamp, M.; Victor, F.; Kupper, A. Blockchain-based supply chain traceability: Token recipes model manufacturing processes. In *Proceedings of the IEEE 2018 International Congress on Cybermatics: 2018 IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, iThings/GreenCom/CPSCom/SmartData/Blockchain/CIT*; IEEE: New York, NY, USA, 2018; pp. 1595–1602. [[CrossRef](#)]
37. Babaei, M.; Khedmati, M.; Jokar, M.R.A.; Tirkolaee, E.B. Product tracing or component tracing? Blockchain adoption in a two-echelon supply chain management. *Comput. Ind. Eng.* **2025**, *200*, 110789. [[CrossRef](#)]
38. Qiao, S.; Cao, C.; Zhou, H.; Gong, W. Space-efficient logging for Supply Chain Traceability based on blockchain. In *Proceedings of the 2022 10th International Conference on Advanced Cloud and Big Data, CBD*; IEEE: New York, NY, USA, 2022; pp. 252–257. [[CrossRef](#)]
39. Ahamed, S.S.; Khanam, M.H.; Reddy, T.R.K.; Sree, P.R.; Narayan, G.J.V. A Blockchain—IPFS Hybrid Framework for Scalable and Verifiable Agricultural Supply-Chain Traceability. In *Proceedings of the 2026 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*; IEEE: New York, NY, USA, 2026; pp. 1–5.
40. Romdhane, M.; Zhang, K.; Santa-Eulalia, L.A.D. Towards efficient and fine-grained traceability for a live lobster supply chain using blockchain technology. *Procedia Comput. Sci.* **2025**, *252*, 123–132. [[CrossRef](#)]
41. Alamsyah, A.; Hakim, N.; Hendayani, R. Blockchain-Based Traceability System to Support the Indonesian Halal Supply Chain Ecosystem. *Economies* **2022**, *10*, 134. [[CrossRef](#)]
42. Zhang, X.; Li, Y.; Peng, X.; Zhao, Z.; Han, J.; Xu, J. Information Traceability Model for the Grain and Oil Food Supply Chain Based on Trusted Identification and Trusted Blockchain. *Int. J. Environ. Res. Public Health* **2022**, *19*, 6594. [[CrossRef](#)] [[PubMed](#)]
43. Ju, C.; Shen, Z.; Bao, F.; Weng, P.; Xu, Y.; Xu, C. A Novel Credible Carbon Footprint Traceability System for Low Carbon Economy Using Blockchain Technology. *Int. J. Environ. Res. Public Health* **2022**, *19*, 316. [[CrossRef](#)] [[PubMed](#)]
44. Abbas, K.; Afaq, M.; Ahmed Khan, T.; Song, W. A blockchain and machine learning-based drug supply chain management and recommendation system for smart pharma industry. *Electronics* **2020**, *9*, 852.
45. Dasaklis, T.K.; Casino, F.; Patsakis, C.; Douligieris, C. A framework for supply chain traceability based on blockchain tokens. In *Proceedings of the Business Process Management Workshops: BPM 2019 International Workshops, Vienna, Austria, 1–6 September 2019; Revised Selected Papers 17*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 704–716.
46. Wüst, K.; Gervais, A. Do you need a blockchain? In *Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*; IEEE: New York, NY, USA, 2018; pp. 45–54.
47. Chen, C.Y.; Kang, T.C.; Chan, Y.W.; Yang, C.T.; Chang, C.H.; Tsai, Y.T. An Integrated Framework of Supply Chain Traceability Based on Blockchain Technology. *Lect. Notes Electr. Eng.* **2020**, *551*, 346–351. [[CrossRef](#)]
48. Koustas, S.G.; Jalowski, M.; Reichenstein, T.; Oks, S.J. A blockchain-based IIoT traceability system: ERC-721 tokens for Industry 4.0. *Procedia Cirp* **2023**, *120*, 1280–1285. [[CrossRef](#)]
49. Haouari, M.; Mhiri, M.; El-Masri, M.; Al-Yafi, K. A novel proof of useful work for a blockchain storing transportation transactions. *Inf. Process. Manag.* **2022**, *59*, 102749. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.