Dependable Food Traceability Data through Blockchain and Database Integration

Lourenço Preto*, Samih Eisa*, Orlando Remédios†, David R. Matos*, Miguel L. Pardal*
{lourenco.preto,david.r.matos,miguel.pardal}@tecnico.ulisboa.pt, samih.eisa@inesc-id.pt, orlando.remedios@sensefinity.com
*INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal,
†Sensefinity, Lisbon, Portugal

***Abstract*—Traditional databases have been important for supply chain businesses, helping them store and manage data for logistic operations. However, growing demands for transparency and traceability, especially in food products, have exposed the risk of data manipulation by dishonest participants in the supply chain. Blockchain technology has the potential to solve these issues by providing a decentralized system to store and share product data, ensuring integrity without any single organization having unchecked control.**

This work explores the integration of traditional database systems with blockchain technology, focusing on a supply chain scenario, to achieve dependable food traceability data. The integration leverages specific design patterns to efficiently structure and organize system implementation. To evaluate the effectiveness of this integration, a food traceability application was developed based on a real-world use case. The results show that integrating databases with blockchain, using design patterns, can protect data from tampering and make the overall system dependable.

***Index Terms*—Data Dependability, Blockchain, Database, Design Patterns, Supply Chain, Traceability**

I. Introduction

Databases have been important for supply chain businesses, helping them store, manage, and retrieve large volumes of data required for operation. However, today's need for more product transparency and security has shown the limitations of using centralized databases [1].

Systems controlled by a single owner are vulnerable to tampering and unauthorized changes, especially if the owner is compromised. Blockchain technology addresses this by using a decentralized, tamper-resistant design [2]. It distributes a ledger across multiple nodes, owned by different organizations, to eliminate single points of failure and ensure data integrity, making it suitable for supply chains with multiple business partners. In other words, it allows traceability data to become dependable. Furthermore, the use of *permissioned* blockchains (c.f. Section II-B) limits access to trusted participants only, balancing decentralization with control in collaborative settings.

This work was supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020 and by Project Blockchain.PT– Decentralize Portugal with Blockchain Agenda, (Project no 51), WP 1: Agriculture and Agri-food, Call no 02/C05-i01.01/2022, funded by the Portuguese Recovery and Resilience Program (PPR), The Portuguese Republic and The European Union (EU) under the framework of Next Generation EU Program, and by the European Union's Horizon 2020 research and innovation programme under grant agreement No 952226, project BIG (Enhancing the research and innovation potential of Técnico through Blockchain technologies and design Innovation for social Good).

From a technical perspective, a blockchain can be perceived as a type of distributed database that facilitates the secure and decentralized storage of information. It provides a secure and immutable way to store records, ensuring product transparency throughout the supply chain phases. This immutability and auditability, from origin to consumption, allow all stakeholders to track the entire process. However, the performance of blockchain has limitations [3]. Blockchains are not well suited for storing large volumes of data, as every node must hold a full copy of the ledger. Writes are slow because of consensus mechanisms, and query capabilities are limited, making complex data retrieval slow and inefficient compared to traditional databases. These limitations make blockchain less practical for applications requiring high data throughput or frequent, complex queries.

The motivation of this work is to combine the strengths of both technologies, demonstrated in a supply chain management scenario: improving the security and transparency of data through the use of blockchain, and allowing efficient access to data through the use of databases. The integration is achieved gradually, inside the system source code, through the use of design patterns [4] that provide reusable solutions for recurring tasks within the program. A test application was developed for a real-world food supply chain system, showcasing the practicality of these design patterns and their specific advantages and disadvantages. The experimental results allow us to discuss trade-offs, offering insights for system performance and dependability that extend beyond the test example.

II. Background

This Section describes key concepts and investigations relevant for this work.

A. Supply Chain

The concept of supply chain refers to the network of organizations, individuals and resources involved in the upstream and downstream movement of goods or services from their site of origin to their point of consumption [5]. This process encompasses all the steps needed to bring a product or service from its producer to the ultimate consumer. A representation of the supply chain is shown in Figure 1.

- *Traceability:* Points out the difficulty of tracking the provenance of a certain product due to the fact that it goes

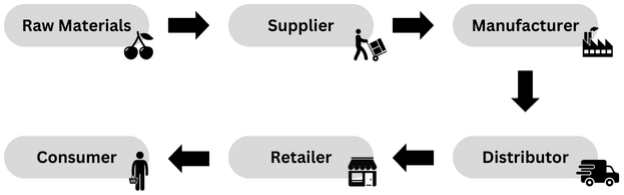

Fig. 1. Example of a food supply chain process.

through multiple organization borders and cross different physical spaces before reaching the end customer;

- *Transparency:* Refers to the need of a stakeholder to track a product. By using blockchain, it is possible for every participant to monitor the location of a product at any point in the supply chain, providing a complete audit log of a product from its origin to consumption;
- *Security:* The supply chain is susceptible to the introduction of counterfeit products due to the high number of locations that a product goes through before reaching its end point. Since each record is linked to the others through hashes, altering a single transaction would require changing the entire chain;
- *Trust between trading partners:* It is hard for all the partners to trust each other, or to fully trust in a central authority. Blockchain offers a decentralized governance, where it is decided in a consensus, if the information goes to the blockchain;
- *Compliance:* Every stakeholder involved in the supply chain must comply with certain standards involving the safety and integrity of the products, otherwise their reputation and success might suffer consequences. The blockchain protects information from tampering, what enables organizations to show that they comply with those requirements.

B. Blockchain Technology

A blockchain is a digital ledger composed of sequential blocks linked through cryptographic hashes. Each block contains a set of transactions with timestamps, and altering a block's data disrupts the entire chain, ensuring integrity and security [6]. Blockchain is characterized by being distributed, decentralized, secure, and immutable. Its distributed nature ensures that records are stored across multiple network nodes. Decentralization allows collective management by nodes, eliminating reliance on a single authority. Security ensures data integrity and non-repudiation, while immutability prevents changes to recorded data.

Blockchains are categorized as public (permissionless) or private (permissioned). Public blockchains, like Bitcoin, allow anyone to join the network, participate in transactions, and access the ledger [7]. Private blockchains, such as Hyperledger Fabric, restrict access to specific participants, often operating within organizations.

Key blockchain platforms include Bitcoin, Ethereum, and Hyperledger. Bitcoin serves as a public, decentralized network for peer-to-peer cryptocurrency transactions [8]. Ethereum supports decentralized applications and smart contracts, offering a programmable environment for developers [9]. Hyperledger focuses on cross-industry collaboration with tools tailored for integrating blockchain into existing systems [10].

Consensus algorithms ensure trust in blockchain networks without a central authority. In Proof-of-Work (PoW), miners solve cryptographic puzzles to validate transactions and earn rewards. In Proof-of-Stake (PoS), validators are selected based on staked cryptocurrency, encouraging honest behavior. Practical Byzantine Fault Tolerance (PBFT) achieves consensus deterministically through structured rounds of agreement.

Smart contracts are self-executing scripts on blockchains, automating agreements without intermediaries. They execute predefined conditions, reducing errors and inefficiencies, and must be deterministic to ensure consistency across decentralized nodes [11].

C. Design Patterns

Design patterns act as templates for addressing specific types of problems rather than offering fixed lines of code or pre-made solutions. Furthermore, design patterns are associated with "forces", which explain their effectiveness in certain contexts. These forces represent the benefits of using a particular pattern, such as adaptability, performance or efficiency [12].

Design patterns can be separated into three main categories: creational, structural and behavioral.

- Creational Patterns: focus on the instantiation of objects, providing flexibility in object creation. For example, the Singleton pattern ensures that only one instance of a class exists to manage shared resources;
- Structural Patterns: organize classes and objects to form larger and more complex systems, emphasizing how components can be combined to create adaptable functionalities. They help define relationships between components, maintain architectural integrity, and support intricate user interfaces;
- Behavioral Patterns: focus on the roles and interactions of objects within a system. They define how objects collaborate and communicate to perform their functions, enhancing extensibility and adaptability by allowing dynamic behavior changes without modifying code.

III. Blockchain Design Patterns

This Section explores design patterns that have been emerged for integrating the blockchain and traditional databases, categorized by their features.

A. Data Management Patterns

Data management patterns in blockchain systems aim to leverage the immutability and transparency of blockchains while addressing limitations like high storage costs and latency. A key approach is *on-chain and off-chain data management*, where critical data is stored on-chain, and large or frequently accessed data resides off-chain in databases or

decentralized storage systems like IPFS. Recent work, such as in [13], demonstrates how integrating NoSQL databases with blockchain can enhance throughput and scalability. Similarly, in [14] the authors explore combining SQL capabilities with blockchain for analytical processing, making it suitable for multi-party cooperation scenarios. These patterns highlight the balance between decentralization and efficiency, critical for applications like healthcare records and supply chain management.

B. Smart Contract Design Patterns

Smart contract design patterns focus on creating scalable, secure, and upgradeable contract systems. Patterns such as *smart contract factory* simplify deploying multiple similar contracts, while *upgradeable contracts* address immutability issues by enabling updates via proxy mechanisms. Recent innovations like in [15] introduce relational database principles into blockchain environments, facilitating contract-driven database interactions. These patterns ensure the reliability and adaptability of smart contracts in applications ranging from token systems to decentralized finance (DeFi).

C. Identity and Access Management Patterns

Decentralized identity and access management patterns are vital for ensuring secure and transparent user interactions in blockchain systems. *Decentralized Identifiers (DIDs)* and *multisignature (multisig)* approaches are widely used to enhance security and control. A survey on blockchain-database integration, in [16], emphasizes the growing need for blockchain-based identity solutions to enhance trust and reduce reliance on centralized systems. These patterns underpin applications in user authentication, governance, and secure transaction approvals.

D. Scalability and Performance Patterns

Scalability remains a critical challenge for blockchain systems. Patterns such as *state channels* and *sidechains* help offload transaction processing from mainchains, improving speed and reducing costs. Additionally, in [17] the authors introduce sharding and off-chain mechanisms to enable scalable blockchain databases. These patterns are essential for high-performance applications in real-time systems like Internet of Things (IoT) applications and payment networks.

E. Security and Trust Patterns

Security and trust patterns ensure the integrity and authenticity of blockchain systems. *Blockchain as a verification layer* and *atomic swaps* facilitate tamper-proof data verification and trustless asset exchanges. In [15], the authors demonstrate how combining blockchain and database technologies can enhance data integrity and trustworthiness. These patterns are foundational for applications like cross-chain trading and secure financial systems.

F. Governance and Collaboration Patterns

Governance patterns like *Decentralized Autonomous Organizations (DAOs)* and *consensus mechanisms* enable transparent and decentralized decision-making. A comprehensive overview in [16] highlights how these patterns drive innovation in decentralized governance systems. They are crucial for applications like community-driven projects and distributed resource management.

G. Integration and Interoperability Patterns

Integration patterns focus on seamless interaction between blockchain and external systems. *Oracles* and *database integration patterns* are instrumental in connecting blockchains with real-world data and external databases. In [14] the authors highlight the potential of SQL-powered blockchains for improving multi-party cooperation. These patterns enable interoperability, enhancing blockchain's usability in complex ecosystems such as DeFi (Decentralized Finance), IoT, and cross-chain applications.

IV. Implementation

We developed a real-world food supply chain application showcasing the practicality and potential of design patterns. Given the scenario, multiple organizations need to handle confidential data. So, we selected a private blockchain technology, Hyperledger Fabric, that offers more data control than public blockchains.

The implementation requirements were primarily defined to support a real-world food traceability project focused on tracking cherries throughout their lifecycle. Also, the solution integrates blockchain technology with the database model, to test out the proposed patterns.

A. Use Case Scenario

A simulation scenario was developed to mimic the type of information a supply chain business might want to protect on a blockchain. In this case, location information that represents a product's location at a time. The information may include: *_id:* Unique identifier of the location point; *ClaimID:* Claim or event associated with the product; *ProverID:* Entity or device responsible for providing the claim; *Location:* Geographical coordinates of the product's location; *Timestamp:* Date and time of when this location point data was captured. The following are the objectives of the simulation scenario:

- Use hybrid storage pattern that uses both an off-chain and an on-chain storage. The off-chain storage is used to store large volumes of data that do not need the high level of security provided by blockchain technology, such as inventory records, images or large documentation. The on-chain storage allows critical data, such as ownership information, to be stored on the blockchain. This ensures that the important information is immutable and transparent to all the stakeholders, providing integrity and resistance to data tampering;
- Implement design pattern that uses blockchain technology to guarantee the integrity and immutability of large

datasets that cannot be stored on-chain. Since some of the information is kept off-chain, it is always susceptible to attacks. Calculating the hash of that information and storing it on-chain, provides data integrity even when the participants do not trust each other. Any stakeholder can verify the accuracy of the information by simply generating the hash value of the data kept off-chain and comparing it with the one stored on-chain;
- Adopt design pattern that focuses on ensuring non-repudiation and trust among stakeholders in a blockchain network. Each party signs the information they want to put on-chain using their private key, and other participants can verify the authenticity by checking the signature with the public key of the signer. This method prevents impersonation, as only the owner of the private key can create that specific signature. Additionally, it ensures non-repudiation, meaning the party that signed the data cannot later deny having done so.

B. Solution Architecture

Users engage with the blockchain network through a command-line interface (CLI), which simplifies the process of reading and writing location data in both the blockchain and off-chain storage. A high-level overview of the solution architecture is depicted in Figure 2.

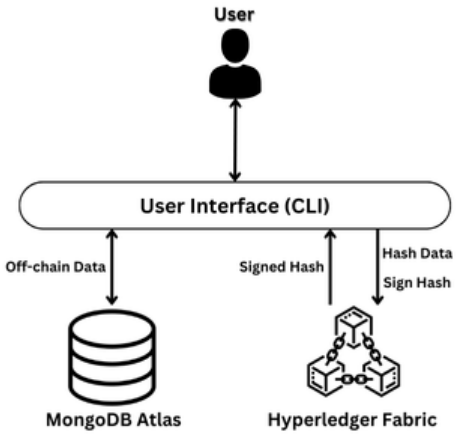

Fig. 2. High-level view of the solution architecture

C. Implemented Chaincode

The implemented chaincode includes all necessary operations for storing and retrieving location points on the blockchain, featuring two key functions: *WriteLocationPoints* and *ReadLocationPoints*.

- *WriteLocationPoints*: This function records new location points for a specific product. It accepts two parameters: the product identifier and a string representing the signed hash of the file containing the location points;
- *ReadLocationPoints*: This function retrieves the location points associated with a specific product. It accepts the product identifier as the only parameter, verifies its existence in the ledger, and retrieves the signed location points linked to it.

D. Client Application

The client application uses the following command to invoke the implemented chaincode.

- *Write*: allows users to store new location points information in the MongoDB Atlas database. The process involves reading the local file containing the location points, connecting to the MongoDB instance, and inserting the data;
- *Read*: enables users to retrieve and display location points stored in the MongoDB Atlas database. The application first connects to the MongoDB instance and reads the location points data. Upon success, the JSON string containing the location points is logged;
- *ProtectedWrite*: allows users to write location points to both Hyperledger's blockchain and the MongoDB Atlas database. It connects to the blockchain network, retrieves the smart contract instance for adding location points data, reads the location, hashes the information, signs it with the user's private key, and writes it to the blockchain;
- *ProtectedRead*: enables users to read location points from the blockchain and the off-chain. It uses a smart contract to access the signed location points information stored on the blockchain and retrieves the certificate used to verify the signature. The signature of the location points data is then verified using this certificate, and then the data is compared to the one stored off-chain.

E. Data Schema

By utilizing the Electronic Product Code Information Services (EPCIS), different partners can consistently exchange information across systems, thereby providing a unified view of the movement of physical objects throughout the supply chain [18]. In this work, a simplified version of the EPCIS model was employed, concentrating on essential data that addressed immediate needs.

Each main event type in EPCIS encompasses fields representing five dimensions: 1) object involved in the event;2) date and time at which the event occurred; 3) location where the event took place; 4) business context of the event; 5) condition of the object involved in the event.

An EPCIS contains a field called "events". This is where the records related to the completion of a certain stage in the business process are stored. In the context of this project, there can be three types of events: an `ObjectEvent`, a `TransformationEvent` and an `AggregationEvent`.

- `ObjectEvent` Captures information about an event involving one or more objects identified by EPC identifiers or from the EPC class;
- `TransformationEvent` Captures information about an event in which one or more physical or digital objects identified by EPC identifiers or from the EPC class, are

fully or partially consumed as inputs, and a new object is produced as output;
- `AggregationEvent` Describes an event where an object has been aggregated with another. In this event, the objects are grouped within a "container" entity that identifies the aggregation itself.

V. Evaluation

This Section presents the evaluation of the implementation.

A. Integrity Testing

The off-chain integrity test involves tampering with the off-chain storage to determine if the system can detect the alteration. The goal is to compare the hash of the off-chain data with the corresponding hash that is stored in the blockchain. If there is a mismatch, then the system will recognize that the information was altered and cannot be trusted.

B. Performance Testing

The performance test evaluates the proposed solution based on the system's response time and latency. In *ReadEvent* operation, it is evident that the operation mainly maintains fast response times even as the number of products increases, in the case when using the blockchain.

In a scenario where the database was populated with 100 different events, the results of the *ReadEvent* operation presents similar results to the one tested regarding the blockchain performance.

In a hybrid scenario, storages were populated with 100 different events and the results of the *ReadEvent* operation present much slower response times compared to the one tested regarding the blockchain performance and the database performance.

VI. Conclusion

Blockchain technology enhances supply chain management by ensuring data immutability and transparency across all stages. This transparency allows for the verification of a product's authenticity and integrity. By combining blockchain with traditional databases, information can be strategically split into on-chain and off-chain storage. This approach leverages blockchain's security benefits while optimizing storage space by keeping larger datasets off-chain. Also, using query engines improves efficiency. The end result is an overall system that can store traceability data in a dependable way, achieving a balance between robust security and practical performance.

References

[1] R. Adams, B. Kewell, and G. Parry, "Blockchain for good? digital ledger technology and sustainable development goals," *Handbook of sustainability and social science research*, pp. 127–140, 2018.

[2] P. Dutta, T.-M. Choi, S. Somani, and R. Butala, "Blockchain technology in supply chain operations: Applications, challenges and research opportunities," *Transportation research part e: Logistics and transportation review*, vol. 142, p. 102067, 2020.

[3] Z. Ge, D. Loghin, B. C. Ooi, P. Ruan, and T. Wang, "Hybrid blockchain database systems: design and performance," *Proc. VLDB Endow.*, vol. 15, no. 5, p. 1092–1104, Jan. 2022. [Online]. Available: https://doi.org/10.14778/3510397.3510406

[4] X. Xu, H. D. Bandara, Q. Lu, I. Weber, L. Bass, and L. Zhu, "A decision model for choosing patterns in blockchain-based applications," in *2021 IEEE 18th International conference on software architecture (ICSA)*. IEEE, 2021, pp. 47–57.

[5] J. T. Mentzer, W. DeWitt, J. S. Keebler, S. Min, N. W. Nix, C. D. Smith, and Z. G. Zacharia, "Defining supply chain management," *Journal of Business logistics*, vol. 22, no. 2, pp. 1–25, 2001.

[6] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," *arXiv preprint arXiv:1906.11078*, 2019.

[7] Q. Lu, X. Xu, Y. Liu, I. Weber, L. Zhu, and W. Zhang, "ubaas: A unified blockchain as a service platform," *Future generation computer systems*, vol. 101, pp. 564–575, 2019.

[8] H. Vranken, "Sustainability of bitcoin and blockchains," *Current opinion in environmental sustainability*, vol. 28, pp. 1–9, 2017.

[9] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020.

[10] A. H. Mohammed, A. A. Abdulateef, and I. A. Abdulateef, "Hyperledger, ethereum and blockchain technology: a short overview," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 2021, pp. 1–6.

[11] M. Alharby, A. Aldweesh, and A. van Moorsel, "Blockchain-based smart contracts: A systematic mapping study of academic research (2018)," *2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB)*, pp. 1–6, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:195884358

[12] E. Gamma, R. Helm, R. E. Johnson, and J. M. Vlissides, "Design patterns: elements of reuseable object-oriented software," 1994. [Online]. Available: https://api.semanticscholar.org/CorpusID:54007465

[13] F. Carrozzino, M. Fiore, and M. Mongiello, "Development of an hybrid blockchain and nosql platform to improve data management," *ArXiv*, vol. abs/2305.03592, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258547237

[14] J. Doe and A. Smith, "achain: A sql-empowered analytical blockchain as a database," *IEEE Transactions*, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10154133

[15] S. Nathan, C. Govindarajan, A. Saraf, M. Sethi, and P. Jayachandran, "Blockchain meets database: Design and implementation of a blockchain relational database," *Proc. VLDB Endow.*, vol. 12, pp. 1539–1552, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:67877079

[16] A. Jones and B. Lee, "A survey on the integration of blockchains and databases," *Springer*, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s41019-023-00212-z

[17] Z. Hong, S. Guo, E. Zhou, W. Chen, H. Huang, and A. Y. Zomaya, "Gridb: Scaling blockchain database via sharding and off-chain cross-shard mechanism," *Proc. VLDB Endow.*, vol. 16, pp. 1685–1698, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258190937

[18] J. Ahn, H. Gaza, J. Lee, H. Kim, and J. Byun, "Oliot epcis: An open-source epcis 2.0 system for supply chain transparency," *SoftwareX*, vol. 23, p. 101477, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352711023001735