# Towards Efficient and Fine-Grained Traceability for Supply Chains using Blockchain Technology

***Abstract*—In today's era, traceability has become a crucial aspect in the supply chain sector, ensuring the safety, quality and origin of products while maintaining transparency and accountability throughout the supply chain. Technologies like Radio-Frequency Identification (RFID) play a pivotal role in augmenting supply chain efficiency by offering real-time visibility into tracking inventory movements. Furthermore, Internet of Things (IoT) sensors provide real-time monitoring of required environmental conditions for product safety. Finally, blockchain technologies have emerged as a promising solution for exchanging, validating, and auditing supply chain data by providing the immutability and integrity necessary for traceability. However, the design space for combining these technologies in a comprehensive traceability solution is very large for complex supply chains, and with solutions offering varying characteristics in terms of tracking granularity, scalability, real-time capability, as well as infrastructure costs. In our paper, we explore this challenging problem by proposing several traceability solutions and studying the trade-offs between scalability, performance, and associated costs. We propose three solutions: a lightweight solution which minimizes cost, a maximum solution which provides detailed traceability at a high granularity, and an intermediate solution which offers a good balance between granularity and performance by leveraging bloom filters and stream processing. To validate our work, we implemented our three traceability solutions in the context of a seafood supply chain for lobsters, using EOSIO as the underlying blockchain platform. Our comprehensive performance evaluation reveals a trade-off between traceability and costs, with our intermediate solution offering a good balance between traceability and cost efficiency.**

***Index Terms*—Blockchain, EOSIO, Traceability, supply chain, IoT data, RFID, seafood, lobster**

## I. Introduction

In the current interconnected world, the demand for robust supply chain traceability has reached unprecedented heights. Companies worldwide grapple with increasing challenges to ensure the quality, safety, and sustainability of their products throughout their journey from the source to the customer [1]. In this context, technologies such as Radio-Frequency Identification (RFID) and the Internet of Things (IoT) play a crucial role in collecting and tracking data related to the quality of products [2]. Broadly speaking, we consider that traceability solutions for supply chains must provide two functionalities: (1) the capacity to track each product as it traverses each step of the supply chain and (2) the capacity to monitor environmental conditions through IoT sensors physically present with the tracked products.

In the midst of these technological advancements, blockchain technology has emerged as a revolutionary solution, promising greater transparency, security, and traceability for supply chains [3]. While offering numerous advantages, the implementation of traceability solutions demands a detailed study to address scalability, performance, and cost challenges. When considered the aforementioned functionalities, a traceability solution which leverages blockchains must determine which data (i.e. with respect to product tracking and IoT data) must be stored on-chain and at which frequency. These design decisions related to on-chain data storage are governed by a trade-off between transparency and performance. Increasing the quantity and frequency of data stored on the blockchain will increase the transparency of traceability, since on-chain information is immutable and can be verified reliably using smart contracts. On the other hand, blockchain technologies typical exhibit poor performance in terms of throughput (low transactions per second), and prohibitive transaction costs when the volume of data stored increases [4].

In this paper, we explore the complex design space surrounding the use of blockchains of supply chain traceability by designing, implementing, and evaluating various traceability methods which differ in granularity, performance, and cost. As a case study, we choose seafood supply chains as an important and representative example [5], especially the lobster supply chain for its economic significance and substantial revenues [6], where traceability is essential for ensuring the environmental conditions needed to maintain the freshness, quality, and sustainability of live lobsters. We present a live lobster supply chain traceability system based on blockchain, leveraging RFIDs, and IoT sensors. This integrated approach aims to provide a transparent and reliable mechanism, verifying the origin, quality, and safety of lobsters throughout their supply chain journey, while also mitigating the risk of theft or loss of products. For the implementation of our proposed solutions, we chose EOSIO blockchain for its superior performance in comparison to other popular blockchains, such as Ethereum, and without requiring transaction fees [7].

The key contributions of our work are as follows:

1) We design a traceability system for tracking products and monitoring IoT data, using the lobster supply chain as a case study. We record events on the blockchain using smart contracts, enabling verification of product origin and quality (Section IV).
2) We present two traceability methods which are situated at the extremes of the design space. The minimum method offers low granularity in terms of time inter-

vals as well as individual product tracking, while the maximum method tries to record every data point on the blockchain (Section V).

3) We propose an optimized solution which provides a good balance between traceability and performance. Our solution employs Bloom Filters for product tracking, combined with a stream processing middleware to efficiently manage IoT sensor data (SectionV-C).
4) We implemented and evaluated our three solutions, using EOSIO as the blockchain platform and Pareto Anywhere as the IoT middleware. Our results demonstrate the trade-off between granularity, performance, and costs (Section VI).

The paper continues with Section II, which explores related works in the field. Following this, Section III provides foundational background information, introducing fundamental concepts used in our study.

## II. Related Works

In this section, we review major works related to our research. First, we focus on papers related to traceability using technology of blockchain. Then, we present related works for supply chain traceability.

### A. Traceability using Blockchain

Authors in [9] propose a real-time Internet of Things (IoT) data sharing and monetization framework, emphasizing the integration of the MQTT protocol with Ethereum smart contracts to achieve a robust traceability mechanism. They present three solutions with varying costs and levels of effectiveness. Maximum Traceability involves rigorous data logging in the blockchain, offering robust fraud detection but at higher operational costs. Minimal Traceability opts for frugality, registering only data size in the blockchain, sacrificing some fraud detection reliability for reduced costs. Bloom Filter-Based Traceability strikes a balance by introducing representative logging, storing data hash positions in a Bloom filter on the blockchain with a moderate cost, ensuring effective fraud prevention. Each of these traceability mechanisms addresses a specific trade-off between reliability and cost, providing valuable insights for the development of our own traceability system.

### B. Supply chain traceability

Employing a blockchain tracing system is crucial for enhancing transparency within the supply chain [10]. In fact, authors in [11] concentrated on simulating order management in the cheese supply chain, comparing two scenarios. One scenario excluded new technologies, while the other integrated blockchain with RFID ans IoT. Their findings demonstrated that the adoption of these technologies, not only contributes to advancements in sustainability within the cheese supply chain but also ensures the certification, authentication, and improved traceability of the final product, eliminating the need for external intermediaries.

Omar, Ilhaam A., et al [12] propose a blockchain-based solution, designed for tracking Personal Protective Equipment (PPE), ensures comprehensive traceability and quality assurance throughout the supply chain. The system utilizes three smart contracts—Registration, Lot, and Order Management—for real-time tracking of ownership transfers and orders.

In another approach, Lin et al. introduced a decentralized supply chain system [13], leveraging the EPCIS network and Ethereum blockchain to address the challenges of data explosion. This system employs an on-chain and off-chain data management model for enhanced performance.

Authors in [14] introduced a blockchain-based strategy to address challenges in the agri-food supply chain, specifically focusing on product traceability, trading party credibility, and delivery mechanisms. The devised system ensures comprehensive recording of all transactions on the blockchain, with subsequent data upload to the InterPlanetary File Storage System (IPFS).

However, these solutions faces a limitation, handling only 15 transactions per second due to Ethereum's Proof-of-Work (PoW) consensus algorithm.

TrustChain [15], propose an approach to ensure the authenticity of recorded data in the supply chain by assigning trust and reputation scores in real-time. This study also faced limitations in terms of throughput and latency due to the application of Hyperledger Fabric.

In [8], authors compares the efficiency of Delegated Proof-of-Stake as a consensus mechanism for a blockchain-based pharmaceutical supply chain with other existing consensus protocols in terms of processing time. The results show that DPoS has a significantly higher throughput compared to Proof of Work and Practical Byzantine Fault Tolerance.

[7] introduces a Blockchain-IoT model for a food traceability system, connecting each product to its source for detailed raw material information. The authors implement the system on both Ethereum and EOSIO. Analysis reveals EOSIO's suitability over Ethereum.

## III. Background

In this section, an overview of EOSIO blockchain focusing on the types of resources utilized. Additionally, a concise explanation of Bloom filters is presented, followed by an introduction to Substreams.

### A. EOSIO

EOSIO is blockchain protocol designed to facilitate the development and execution of decentralized applications (DApps) by offering a robust infrastructure for smart contracts [16]. One of its notable features is the use of Delegated Proof-of-Stake (DPoS) consensus, providing fast transaction speeds and scalability and making it a popular choice for various decentralized applications.

When it comes to resource consumption, users on EOSIO network need to stake tokens to access resources like CPU, NET, and RAM. CPU, indicating processing power, is vital

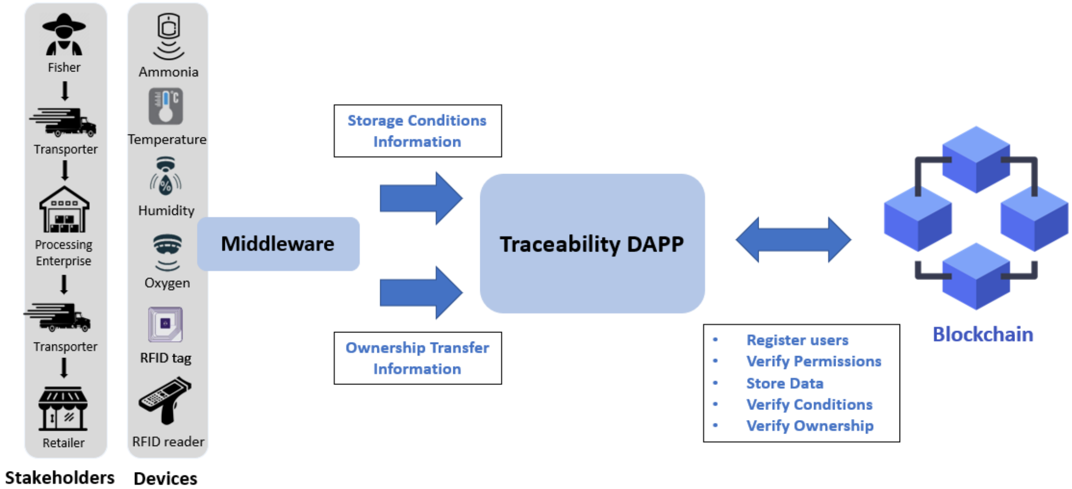


Fig. 1: Proposed System Architecture

for executing transactions, while NET represents network bandwidth. RAM, on the other hand, is crucial for storing data on the blockchain. This resource allocation system ensures that users have a direct stake in the network's operation, preventing misuse and encouraging responsible usage.

In our project, we are leveraging EOSIO to create a supply chain traceability solution that meets the specific needs of our ecosystem.

### *B. Bloom Filter*

Bloom Filters are commonly used in applications where search speed and minimized memory usage are essential [17]. They perform by using a series of hash functions to represent a set of elements. They are designed to quickly answer a crucial question: "does a given element belong to this set?"

Unlike many other data structures, Bloom Filter uses only a small number of bits to represent a large set of elements. Elements are added over time by applying several independent hash functions to each element and setting the corresponding bits in the array. The size of a Bloom Filter is determined by the number of elements to be stored and the acceptable probability of false positive responses.

The main idea behind Bloom filters is to minimize the number of false positive responses (false positives), while enabling rapid response to queries. This property makes them ideal for our supply chain traceability use case, as they reduce the time needed to search for specific information while maintaining an acceptable level of accuracy.

### *C. Substreams*

Substreams by StreamingFast [18] is a powerful platform for the parallel transformation of streaming data. It enables developers to write modules in Rust, leverage community modules, and offer extremely high-performance indexing, while allowing data to be routed to multiple destinations. Substreams offers an optimal and revolutionary experience in managing the flow of blockchain data based on gRPC, protobuf, and the StreamingFast Firehose [19].

Firehose, is responsible for extracting data from blockchain nodes. Substreams collaborates seamlessly with it to execute massive operations on historical blockchain data in an extremely parallelized manner.

This combination offers a powerful blockchain indexing technology, with a particular focus on low latency extraction, modularity and ease of use.

## IV. CASE STUDY: LIVE LOBSTERS TRACEABILITY

In this paper, we use the seafood supply chain of live lobsters as a representative example to evaluate our traceability solutions. For this work, we have conducted a field study in Gaspésie (Quebec, Canada), in order to understand the characteristics of this supply chain. Each fishing season, the permitted period for lobster harvesting, spans ten weeks within a year. During this time frame, approximately 15,000 lobsters are in circulation, being both caught and transferred. Managing this process is critical, considering the elevated risk of lobster mortality if precise storage conditions are not adhered to.

Thus, we conclude that this supply chain has two important characteristics: (1) each lobster must be tracked during the entirety of the supply chain, from when it is caught all the way to its delivery at the grocery store, (2) proper environment conditions must be respected at all times for temperature, humidity and ammonia.

Based on these observations, we propose an architecture which integrates RFID for inventory tracking, IoT sensors for environmental monitoring, and a blockchain platform for managing and securing the collected data. However, we believe that our work in this paper is applicable for a wide variety of other supply chains, such as other seafood [5], food products [11], or cold chains such as pharmaceutical logistics [8]. In this section, we present our system architecture followed by our method of ownership transfer design to ensure better traceability.

### A. System Architecture

In our supply chain, comprising of fishermen, transporters, processing enterprises, and retailers (end customers), the focus is put on maintaining the freshness and quality of lobsters. Our system architecture, depicted in Figure 1, seamlessly integrates stakeholders and cutting-edge technologies to achieve precise traceability and maintain superior lobster quality.

The stakeholders in our supply chain include fishermen responsible for lobster harvesting, transporters managing logistics, processing enterprises handling tasks like deshelling and cleaning, and retailers serving as the final destination.

To enhance traceability, we employ Radio-Frequency Identification (RFID) as a tracking device. Fishermen affix RFID tags to each lobster and corresponding containers, referred to as Logteks, facilitating product grouping into lots. Figure 2, captured in Gaspésie by our team, illustrates a Logtek holding a set of lobsters. Each lobster will be individually identified through an RFID tag attached to its claw, acting as a digital fingerprint and recording ownership transitions across the supply chain. This meticulous tracking not only simplifies traceability but also ensures transparency and accountability.

Fig. 2: Lobsters packed in a Logtek

Our system incorporates sensor technology in each Logtek, measuring vital factors for lobster well-being, such as temperature, humidity, dissolved oxygen, and ammonia concentration. Specific value ranges are set for each parameter, ensuring quality. If values fall outside the designated range, indicating a storage condition violation, our system detects and identifies the responsible party.

Figure 1 illustrates the architecture, showcasing stakeholders, devices, and an IoT middleware. The IoT middleware standardizes data before transmission to our decentralized application (DApp). This middleware processes and transmits IoT devices data, providing ownership transfer information and storage conditions details to our decentralized application.

In our traceability DApp, users can register, permissions are verified, data is stored, and conditions and ownership are verified. This comprehensive integration of technologies ensures a robust and transparent lobster supply chain.

### B. Dual-Check Method for the Transfer of Ownership

In order to enhance traceability, we have devised a dual-check method for the transfer of ownership. This method ensures a robust tracking mechanism when products transition from one owner to the next, involving the execution of two crucial actions, as illustrated in Figure 3.

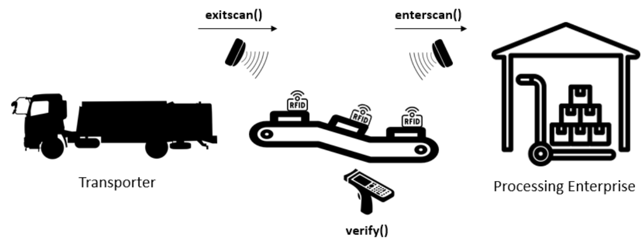


Fig. 3: Dual-Check Method for Ownership Transfer

The first action, denoted as `exitscan()`, signifies that the transporter, acting as the initial owner, has successfully delivered the products. The second action, `entrescan()`, is executed by the subsequent owner (e.g., a processing enterprise in the figure 3) to confirm the receipt of the products. Both `entrescan()` and `exitscan()` are triggered when the RFID reader of the respective actor detects the RFIDs associated with the products.

This method enables the detection of potential violations, losses, or any malicious intent where an actor falsely claims to have dispatched products that have not been received by the next entity. Furthermore, the implemented mechanism serves as a formidable deterrent against counterfeit products. Any actor attempting to introduce counterfeit items would be unable to produce the authentic ownership history, providing an additional layer of security. In instances where the receiving actor suspects foul play, the option to reject products is facilitated by the `rejectscan()` function.

Between the `entrescan()` and `exitscan()` actions, the receiving actor has the opportunity to conduct verifications on the state of the received products. This includes examining the ownership history of the products to validate their origins and assessing the storage conditions throughout the entire supply chain.

The implemented dual-check mechanism not only enhances traceability but also empowers entities within the supply chain to ensure the integrity of product transfers and proactively address any discrepancies or concerns that may arise.

In the next section, we'll explore how these functions are uniquely implemented to establish three distinct traceability models. Following that, we'll compare their effectiveness and costs.

## V. PROPOSED TRACEABILITY SOLUTIONS

Our supply chain traceability project is built upon an innovative approach, which involves the development of three distinct methods: MAX-MAX-Trace, MIN-MIN-Trace, and

BF-Substreams-Trace. In this section, we provide detailed descriptions of each of these solutions, outlining the property transfer tracing method and the storage conditions tracing method for each. We specify the data recorded on-chain and off-chain, along with the associated verification process for each solution.

### *A. MAX-MAX-Trace : Maximum of traceability for ownership transfer and storage data*

The MAX-MAX-Trace method aims to achieve the highest level of traceability for both products and storage-related information throughout the supply chain. It emphasizes the comprehensive retention of data within the blockchain.

*1) Ownership transfer traceability:* For each product recorded in the blockchain, a detailed ownership history is maintained. This history includes essential elements such as owner, state (delivered, accepted, or rejected), and timestamp. An entry is appended to this historical record every time `exitscan()`, `enterscan()`, or `rejectscan()` action is executed. Given the secure and immutable nature of these records in the blockchain, they serve as trustworthy and unalterable evidence of the product's journey.

The complete history recorded for each lobster in the blockchain ensures the assurance of its origin, providing a transparent account of its journey from capture to its current point in the supply chain.

*2) Storage data traceability:* Sensor data is logged in the "entries" table of the smart contract. Every entry undergoes a meticulous verification of storage conditions. If an entry fails to comply with specified conditions, the condition status for concerned products is set to false. Simultaneously, the ID of the non-conforming entry is documented in the entriesId list, providing swift access for subsequent verification. The integrity of this information is bolstered by its secure and unchangeable status within the blockchain.

### *B. MIN-MIN-Trace : Minimum of traceability for ownership transfer and storage data*

designed for achieving a minimum level of traceability at a lower cost, the approach involves two key aspects:

*1) Ownership transfer traceability:* All product details are recorded in a centralized database, and to maintain traceability, the `exitscan()` function registers the quantity of products delivered, while the `enterscan()` function logs the received quantity in the blockchain.

**Verification Process:** The verification process consists of carefully examining the records on the blockchain, to ensure that the quantity of products delivered corresponds to the quantity received. Verifying the coherence of these blockchain records allows the system to detect inconsistencies, offering insights into potential losses, violations, and responsible entities.

It's crucial to note that in the MIN-MIN-Trace method, the blockchain doesn't play an active role in ensuring product provenance; rather, product provenance relies on off-chain data.

*2) Storage data traceability:* All sensor data is recorded in the entries table of the centralized database. However, the blockchain specifically tracks and counts non-conforming entries—those not adhering to the prescribed storage conditions. This focused approach aims to fortify the system against potential falsification, alteration, or deletion of the most sensitive sensor data.

**Verification Process:** Verification involves comparing the count of non-conforming entries in the centralized database with the count stored in the blockchain. While effective in preventing the removal or modification of non-conforming entries, this method doesn't ensure the accuracy of the values within these entries. Therefore, there's no guarantee that these values haven't been adjusted to approximate conforming values while still remaining non-conforming, and such adjustments may go undetected.

### *C. BF-Substreams-Trace : Bloom filter-based traceability for ownership transfer and Substreams-based traceability for storage data*

To address the limitations of the other two proposed models and ensure dependable traceability with lower resource consumption compared to MAX-MAX-Trace, this method combines the use of Bloom filters for product ownership traceability with Substreams for storage-related information traceability, offering a balanced approach.

*1) Ownership transfer traceability:* In this approach, Bloom filters are explored for ownership transfer traceability of products. For each transfer of ownership, two Bloom filters are generated. The first filter indicates that the original owner has transferred these products to the new owner, while the second filter indicates that the new owner has received these products from the original owner.

To facilitate up to 5,000 product ownership transfers between two parties throughout the fishing session, we designed a Bloom filter with a size of 47,965 bits and a hash rate of 7 [20], aiming for a false positive probability of $10^{-2}$.

To implement this solution, we create an array consisting of 750 elements of 64 bits each. For each product ownership transfer, we combine its RFID with that of the corresponding LogTek and its fishing date. This concatenation is performed to keep a record in the blockchain of the lobster's freshness based on its fishing date and to maintain a link with the associated LogTek.

The sha3 function is then applied to hash this concatenated RFID. To determine its position within the Bloom filter, the resulting hash is divided by the filter size. It's crucial to note that this sequence iterates seven times for each product, ensuring the acquisition of seven distinct positions within the Bloom filter.

After identifying the position by dividing the hash by the filter size, we then derive the positions in our array. We do this by dividing the position by 64 to locate the element needing modification in the array. Following that, we take the remainder after dividing the position by 64 to pinpoint the

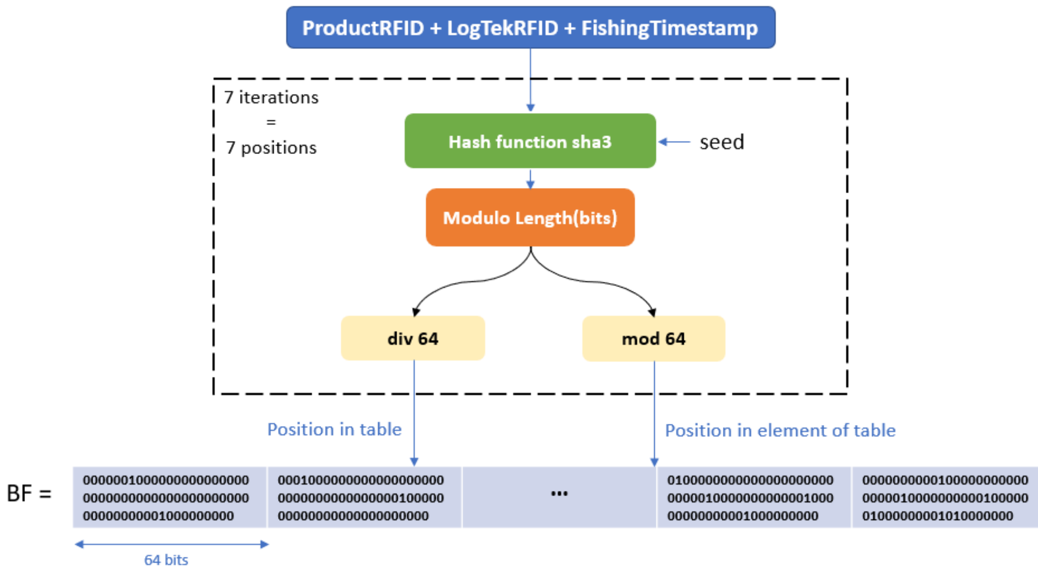


Fig. 4: Bloom Filter Data Insertion

exact bit within that element that requires alteration. Figure 4 visually represents these steps for a clearer understanding.

**Verification Process :** To verify the consistency of ownership transfer, the Dice coefficient is employed [21]. Calculated as:

$$DiceCoefficient = \frac{2h}{(a+b)} \tag{1}$$

This formula assesses the similarity between two Bloom filters. Here, $h$ represents positions set to 1 in both filters, and $a$ and $b$ represent positions set to 1 in each individual Bloom filter. A higher Dice coefficient indicates increased similarity.

The ownership transfer's coherence is determined by evaluating the Dice coefficient, serving as a similarity metric between the two Bloom Filters. A Dice coefficient of 1 signifies a perfect match, ensuring a seamless transfer with no loss, no anomalies, or risk of counterfeiting. This rigorous approach guarantees the legitimacy of products and verifies the absence of unauthorized transfers.

It's noteworthy that the provenance in this method can be verified by the blockchain. This is achieved by checking if the product exists in the Bloom filter of the fisherman who caught the lobster. Therefore, a function `exists()` is created to search for an element in a Bloom filter.

*2) Storage data traceability:* Figure 5 illustrates data storing and verification steps. First, the `setentry()` function records sensor data in a centralized database. It then sends this data to the blockchain.

The blockchain increments the entry counter for the corresponding actor involved in the process.

From the output of the transaction, the block number is extracted and stored in the entry's record in the centralized database. This creates a trace for future verification.

**Verification Process:** Substreams is employed for verification. Using the recorded block number, Substreams accesses the corresponding block on the blockchain, filters, and extracts parameters specific to that transaction.

The extracted values are compared with the corresponding entries stored in the centralized database. If the values match, it indicates no tampering or falsification of data. Finally, to ensure that no entries have been deleted, the blockchain counter value for each actor is compared with the total entries in the centralized database for that specific actor.

### *D. Comparaison and Discussion*

For MAX-MAX-Trace, the blockchain meticulously records each product individually, along with its comprehensive history, storage conditions, and ownership changes throughout the supply chain. This ensures that our retailer, acting as the customer, can verify the origin, quality, and safety of the product throughout its entire supply chain journey.

As for MIN-MIN-Trace, fewer details are recorded. While it guarantees the same number of products transferred between actors and ensures security, it falls short in detecting counterfeiting and doesn't provide the exact values guaranteed by the blockchain for quality assurance.

Turning to BF-SUB-Trace, product security and provenance rely on the application of Bloom Filters. If the similarity is not equal to 1, it signals potential issues like product loss, counterfeiting, or unauthorized additions. A second verification using the `exists()` function helps pinpoint the exact malicious act and responsibilities. However, it's essential to acknowledge the possibility of false positives with Bloom Filters. In the Substreams process, if the block number stored in the centralized database doesn't correspond to the block containing the sought transaction or contains different values

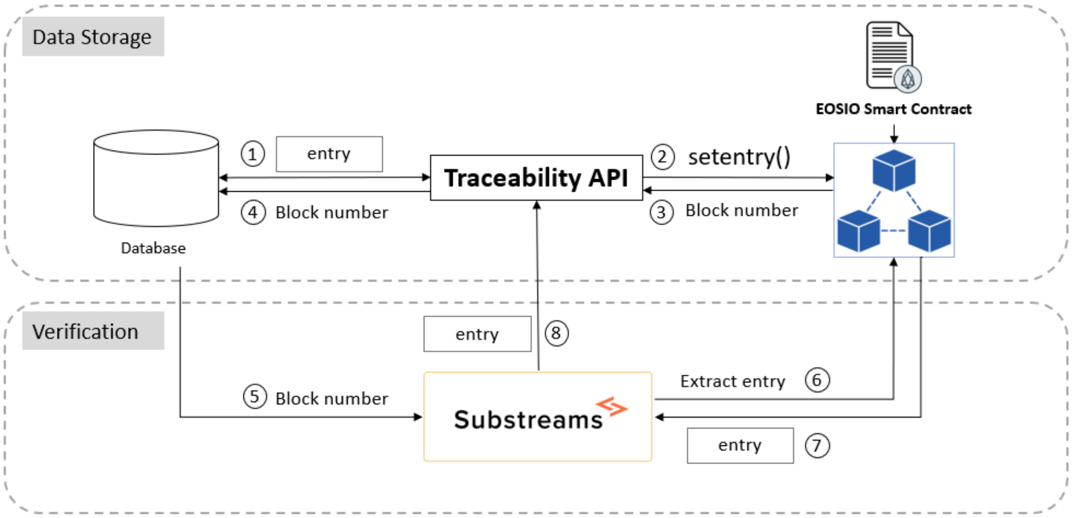


Fig. 5: Verification Process with Substreams

than those on the blockchain, it indicates an alteration—an act of malice. Likewise, a discrepancy in the number of entries between the database and the blockchain signals a deleted entry.

## VI. Evaluation and Result

In this section, We detail our experimental set-up, present performance results and engage in a concise discussion.

### A. Setup

Our test environment consists of a hardware setup featuring a PC with an Intel(R) Core(TM) i7 CPU 2.00 GHz and 8GB of installed RAM. The software environment is illustrated in Figure 6 and includes:

*FlexSim* : A robust simulation platform designed for modeling and analyzing complex processes that we used to simulate our supply chain.

*Pareto Anywhere* [22] : developed by reelyActive, is a middleware solution designed to integrate and manage data from various IoT (Internet of Things) devices, including RFID (Radio-Frequency Identification) tags. In our implementation, lacking access to IoT devices, we simulated IoT data and forwarded it to Pareto Anywhere for processing before transmitting it to our DApp.

*Jungle Testnet* : A test environment we use to evaluate our approach under conditions similar to those on the main network. We've deployed our contracts, written in C++, on this network.

*eosjs* : A Javascript library for interacting with the EOSIO blockchain enableing communication with smart contracts.

*MongoDB* : Utilized for the storage and management of off-chain data, MongoDB excels in high-speed writing and flexible query capabilities.

*Substreams* : Fueled by EOS Nation's Firehose for Jungle Testnet, we efficiently extracted, filtered, and delivered blockchain data.

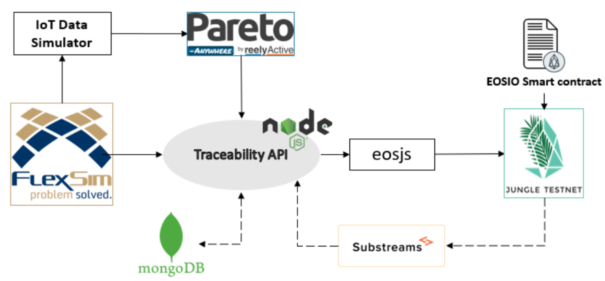


Fig. 6: Experimental Setup

This diverse set of tools ensures a comprehensive testing and development environment for our supply chain traceability project.

### B. Performance evaluation

To evaluate the performance and compare the operating costs of the proposed models, we have developed test scenarios for the supply chain using our simulation. These scenarios are developed by manipulating two key parameters: the number of products in transit and the frequency of sensor data.

In order to accurately measure resource consumption, we have integrated a specific function into our implementations. This function analyzes the results of each transaction, extracts CPU and network bandwidth (NET) consumption, and accumulates these values throughout the scenario. In addition, to evaluate RAM consumption, we compare the amount of RAM used before and after execution, thus isolating the RAM consumption related to the scenario data.

In the first experiment, we manipulated the sensor data frequency parameter, testing four scenarios: reading data every 5 minutes, every minute, every 6 seconds, and every 2 seconds. Figure 7 illustrates the results:

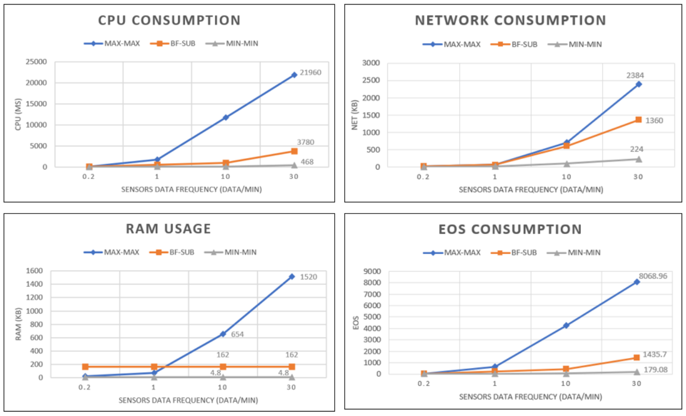


Fig. 7: Sensor data frequency effect on resources consumption

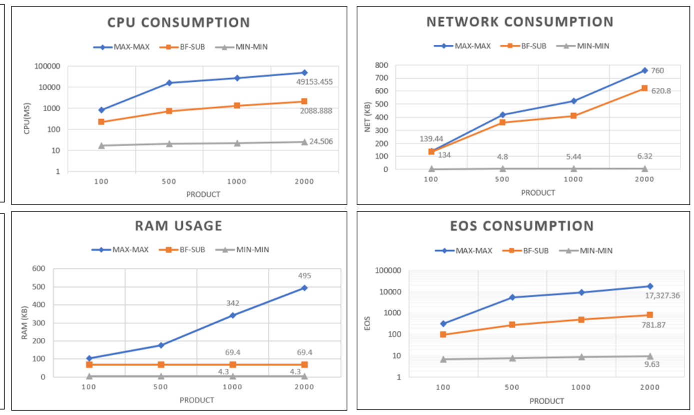


Fig. 8: Number of products effect on resources consumption

MAX-MAX-Trace proves to be the most resource-intensive in terms of CPU and NET consumption. This is attributed to its requirement to record all information, including sensor data, in the blockchain. The consumption escalates significantly with an increase in sensor data frequency, reaching up to 22s of CPU time and 2.4MB of NET for the most exhaustive iteration in our case. In terms of RAM, MAX-MAX-Trace exhibits poor scalability, peaking at nearly 10 times more than the BF-SUB-Trace solution. Consequently, MAX-MAX-Trace emerges as the most EOS-consuming solution.

BF-SUB-Trace, on the other hand, experiences a more modest increase, reaching up to 4s of CPU time and 1.4MB NET. BF-SUB-Trace doesn't require the recording of all events in the blockchain, resulting in consistent RAM values unaffected by this parameter. Therefore, this solution consumes significantly less EOS compared to MAX-MAX-Trace.

MIN-MIN-Trace demonstrates a minimal increase in CPU and NET consumption, with values consistently lower than the other solutions. Regarding RAM, this solution consumes only 4.8KB for any data frequency, as these data are not recorded in the blockchain. Thus, for these iterations, MIN-MIN-Trace incurs a maximum consumption of 180EOS, which is negligible compared to MAX-MAX-Trace's consumption.

In the second experiment, we varied the product quantity parameter. As it is shown in Figure 8, the CPU consumption showed a slight increase for MIN-MIN-Trace, reaching up to 25ms, and for BF-SUB-Trace, peaking at 2s. However, for MAX-MAX-Trace, the increase was significant, reaching up to 49s. This surge is attributed to the addition of each product, along with its details, to the blockchain in the case of MAX-MAX-Trace.

Regarding NET consumption, MAX-MAX-Trace consistently emerged as the most resource-intensive. In terms of RAM, BF-SUB-Trace showcased constant consumption, indicating the fixed size of the Bloom filter, a notable advantage of Bloom filters. RAM consumption remained constant for MIN-MIN-Trace as well, and it was minimal since only the number of transferred products, not the details of each product, were recorded.

Observing these results, MAX-MAX-Trace utilizes a significantly larger amount of EOS compared to BF-SUB-Trace, while MIN-MIN-Trace stands out as the most cost-effective solution.

### C. Discussion

This evaluation allows us to precisely quantify the performance and execution costs associated with each of our models, highlighting the advantages and compromises of each approach.

While MIN-MIN-Trace requires fewer EOS resources, it exhibits a vulnerability in terms of reliability, as certain malicious behaviors, such as the transfer of counterfeit products, may go unnoticed.

MAX-MAX-Trace stands out as the most resource-intensive solution, relying on comprehensive recording of all actions on the blockchain, including the addition of lobsters and LogTeks, as well as the complete history of transfers and storage conditions for each product. This meticulous traceability offers detailed visibility but at a higher cost.

In contrast, BF-SUB-Trace consumes significantly fewer resources than MAX-MAX-Trace. This efficiency stems from the constant spatial and temporal efficiency of Bloom filters and the use of Substreams, resulting in significant savings in RAM and CPU. It strikes a balance between traceability level and costs, offering good scalability with moderate expenses.

## VII. Conclusion

In this paper, we propose and implement a blockchain-based traceability system for lobster supply chain. Introducing three distinct traceability models on the EOSIO blockchain—MAX-MAX-Trace, MIN-MIN-Trace, and BF-SUB-Trace—to trace ownership transfer and storage conditions.

The results show different resource consumption for the three models with high costs for exhaustive traceability for MAX-MAX-Trace and MIN-MIN-Trace exhibiting limitations in verification reliability. The proposed Bloom filter and Substreams-based model emerge as the optimal solution, showcasing superior performance and precise traceability.

## References


[1] Kittipanya-Ngam, P., & Tan, K. H. (2020). A framework for food supply chain digitalization: lessons from Thailand. Production Planning & Control, 31(2-3), 158-172.

[2] Callefi, M. H. B. M., Ganga, G. M. D., Godinho Filho, M., Queiroz, M. M., Reis, V., & dos Reis, J. G. M. (2022). Technology-enabled capabilities in road freight transportation systems: A multi-method study. Expert Systems with Applications, 203, 117497.

[3] Song, J. M., Sung, J., & Park, T. (2019). Applications of blockchain to improve supply chain traceability. Procedia Computer Science, 162, 119-122.

[4] Chukwu, E., & Garg, L. (2020). A systematic review of blockchain in healthcare: frameworks, prototypes, and implementations. Ieee Access, 8, 21196-21214.

[5] Blaha, F. & Katafono, K. 2020. Blockchain application in seafood value chains. FAO Fisheries and Aquaculture Circular No. 1207. Rome, FAO.

[6] Facchini Amondarain, F. (2019). Study of the functional and economic feasibility of the application of Blockchain in a food sector company's supply chain (Bachelor's thesis, Universitat Politècnica de Catalunya).

[7] Tripathi, A. K., Akul Krishnan, K., & Pandey, A. C. (2023). A Novel Blockchain and Internet of Things-Based Food Traceability System for Smart Cities. Wireless Personal Communications, 129(3), 2157-2180.

[8] Sharma, S., Bahga, A., Sharma, T., & Krishna, C. R. (2021). Time-Efficient Auditable Blockchain-based Pharma Drug Supply Chain using Delegated Proof-of-Stake. In International Conference on Emerging Technologies: AI, IoT and CPS for Science & Technology Applications (ICET 2021).

[9] Badreddine, W., Zhang, K., & Talhi, C. (2020, May). Monetization using blockchains for IoT data marketplace. In 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 1-9). IEEE.

[10] Wang, H., Zhang, M., Ying, H., & Zhao, X. (2021). The impact of blockchain technology on consumer behavior: A multimethod study. Journal of Management Analytics, 8(3), 371-390.

[11] Varriale, V., Cammarano, A., Michelino, F., & Caputo, M. (2021). Sustainable supply chains with blockchain, IoT and RFID: A simulation on order management. Sustainability, 13(11), 6372.

[12] Omar, I. A., Debe, M., Jayaraman, R., Salah, K., Omar, M., & Arshad, J. (2022). Blockchain-based supply chain traceability for COVID-19 personal protective equipment. Computers & Industrial Engineering, 167, 107995.

[13] Lin, Q., Wang, H., Pei, X., & Wang, J. (2019). Food safety traceability system based on blockchain and EPCIS. IEEE access, 7, 20698-20707.

[14] Shahid, A., Almogren, A., Javaid, N., Al-Zahrani, F. A., Zuair, M., & Alam, M. (2020). Blockchain-based agri-food supply chain: A complete solution. Ieee Access, 8, 69230-69243.

[15] Malik, S., Dedeoglu, V., Kanhere, S. S., & Jurdak, R. (2019, July). Trustchain: Trust management in blockchain and iot supported supply chains. In 2019 IEEE International Conference on Blockchain (Blockchain) (pp. 184-193). IEEE.

[16] Xu, B., Luthra, D., Cole, Z., & Blakely, N. (2018). EOS: An architectural, performance, and economic analysis. Retrieved June, 11, 2019.

[17] Tarkoma, S., Rothenberg, C. E., & Lagerspetz, E. (2011). Theory and practice of bloom filters for distributed systems. IEEE Communications Surveys & Tutorials, 14(1), 131-155.

[18] "Substreams - StreamingFast," https://substreams.streamingfast.io/

[19] "Firehose - Firehose," https://firehose.streamingfast.io/.

[20] "Bloom filter calculator," https://hur.st/bloomfilter/.

[21] Randall, S. M., Ferrante, A. M., Boyd, J. H., Bauer, J. K., & Semmens, J. B. (2014). Privacy-preserving record linkage on large real world datasets. Journal of biomedical informatics, 50, 205-212.

[22] "reelyactive/pareto-anywhere," https://github.com/reelyactive/pareto-anywhere.