ChainGuard: Verified Data Intake for a Track & Trace Blockchain

André Avelar*, Samih Eisa*, Orlando Remédios†, Miguel L. Pardal*
{andre.avelar, 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*—Maintaining a reliable record of goods movement is essential to supply chain management systems. Ensuring this reliability becomes more challenging as supply chains expand and many stakeholders must collaborate to provide data. Blockchain technology enables a multi-owner system that ensures data integrity through its immutable, transparent, and decentralized features. However, in the supply chain, data is collected from sensor devices at business sites or during transport along the supply chain. This dependence on single-source data undermines the decentralization and security guarantees of blockchain systems.**

This work proposes *ChainGuard*, a system to verify data intake from single sources in blockchains using a decentralized verification process guided by stakeholder-defined rules. It detects errors to ensure compliance with quality criteria. A prototype focused on location sensors was developed and tested in a real-world traceability system monitoring for commercial fruit shipments. The prototype successfully identified three types of suspicious data patterns, achieving low performance impact compared to a system without data verification.

***Index Terms*—Supply Chain Traceability, Route Certification, Oracles, Single Sources of Data, Blockchain**

I. Introduction

In modern supply chains, transparency and traceability are priorities for businesses and consumers. Sensor technologies have advanced with the advent of the Internet of Things [1], enabling seamless data collection across supply chains. However, no single entity can reliably store and protect this data due to the inherent complexity of supply chains, which involve multiple entities, dispersed locations, and numerous transactions. This complexity presents challenges in achieving transparency and accountability, critical for operations like quality assurance, product recalls, fraud prevention, and anti-counterfeiting [2].

Centralized databases are vulnerable to tampering and unauthorized alterations, by intruders or by a malicious owner. Blockchain technology offers a decentralized and tamper-resistant alternative [3]. By distributing a ledger across nodes, the blockchain mitigates single points of failure and ensures data integrity, providing a strong support for a transparent supply chain system. However, implementing blockchain-based traceability faces challenges, especially if using public blockchains. An intermediate approach is to use a *permissioned* blockchain where the system is still decentralized but access is restricted to authorized participants. This works well for supply chain scenarios because the entities involved already collaborate and have a basic level of trust.

In this work, we focus on the challenge of integrating real-world data into permissioned blockchain systems, collected from sensor devices. While a decentralized blockchain can ensure reliable data storage, the trustworthiness of the recorded data depends on the sensors that are external sources. For example, a compromised or malfunctioning sensor monitoring a crate of fruits can report incorrect data. Thus, data must be verified, *before* it is accepted into the blockchain. Without reliable verification mechanisms, data inaccuracies can compromise the overall system integrity.

Blockchains need special mechanisms to fetch and verify external data, known as *Oracles* [4]. There are centralized or decentralized oracles, but there are unresolved issues of security, performance, and cost [5].

This paper introduces *ChainGuard*, a system for securing and decentralizing data intake verification in blockchain-based traceability, from single sources of data along the supply chain. The data is assumed to come from a single source, i.e. only one device is capable of reporting the position and status of goods, in the supply chain, at a certain point in time. *ChainGuard* allows stakeholders of the permissioned blockchain to collaboratively define and execute data verification rules, leveraging the decentralized execution capabilities already present in the blockchain, known as *smart contracts* or *chaincode*.

A prototype was tested with real-world fruit supply chain datasets to identify incorrect data patterns and evaluate how data certification affects performance. The prototype was integrated with Hyperledger Fabric [6], a permissioned blockchain, suited to a multiple organization supply chain, and includes auditing tools to review the verification process and its results.

The remainder of this document is organized as follows: Section II reviews related work. Section III outlines the proposed system. Section IV describes the evaluation methodology and presents the experimental results obtained using the prototype.

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).

II. Related Work

Recent research has explored blockchain-based supply chain systems [7], [8]. Industry-specific implementations include coffee [9], baked goods [10], and pharmaceuticals [11]. Other studies focus on integrating blockchain with IoT (Internet of Things) and AI (Artificial Intelligence) for real-time data collection [12]. Blockchain's role in sustainable supply chain management has also been studied [7].

IoT devices are commonly used in agri-food traceability systems to collect shipment data [13]. These devices resemble oracles in their role of importing off-chain data. Unlike oracles, IoT devices often function as single points of failure. They provide data without cross-verification, introducing vulnerabilities. Practical and economic constraints limit the use of multiple sensors per shipment. This reliance on single sensors makes systems susceptible to errors or tampering [14].

Blockchains face significant challenges when interacting with real-world data, including issues of data reliability, trustworthiness, and integration. External data, such as stock prices or weather conditions, exists off-chain and is dynamic. This challenge is referred to as the *Oracle Problem* [4].

Oracles act as intermediaries to fetch, verify, and relay data to blockchains. They enable smart contracts to access reliable external data [15]. Oracles work well for public data that can be independently verified by multiple nodes, such as weather updates in cities. However, these methods are less effective for private datasets in agri-food traceability systems because the locations are rural. In such areas, it is typically practical and economical to deploy only a few sensors. Additionally, managing devices in the fields is challenging due to the lack of power and connectivity.

The key question is how blockchain systems can securely import data from a single source. This issue has not been thoroughly explored in the current literature. While some studies address data reliability in blockchain-based supply chains [16], there is limited research on reducing the risks associated with relying on a single data source.

III. ChainGuard

The *ChainGuard* system is designed to address the challenges of integrating external data, specifically location information, into blockchain-based traceability systems. Figure 1 illustrates the concept of data intake verification. The system name is derived from rules to "guard" the "chain" from bad data. Rather than allowing sensor data to be directly recorded onto the blockchain ledger, *ChainGuard* intercepts all incoming data and processes it through its data testing modules to evaluate and validate its authenticity before committing it to the ledger.

Thus, *ChainGuard* is a general-purpose solution designed to integrate with any blockchain-based traceability system. Its architecture is blockchain-agnostic, independent of specific programming languages or supporting technologies, making it adaptable to a wide variety of use cases. In this work, we implemented *ChainGuard* using Hyperledger Fabric, with its chaincode written in the Go programming language.

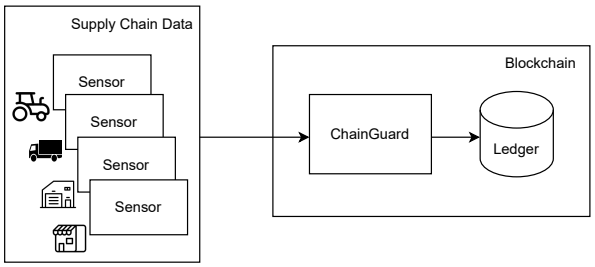

Fig. 1. *ChainGuard* overview.

The blockchain deployment utilized a test network provided in the 'release-2.5' version of the official Hyperledger Fabric documentation [17]. This network included two organizations, each operating a peer node, and a single orderer node to manage the ordering service.

A. Architecture

The internal architecture of *ChainGuard* has two main components: *On-chain* and *Off-chain*. These components collaborate to ensure secure and accurate integration of location data into the blockchain ledger. Figure 2 provides an overview of the system's internals.

1) On-Chain: refers to the components of the system that operate directly on the blockchain. It consists of three primary elements: CGSC, CM, and Data Testing.

ChainGuard Smart Contract (CGSC) serves as the system's central interface, facilitating all interactions with client applications. Users submit location data or query information through the CGSC, which manages updates to the World State. The World State includes route information, represented as a collection of points that detail the movement of a product through the supply chain. Each point records the location of a product at a specific time.

Certification Manager (CM) ensures that all incoming location data is validated against predefined rules agreed upon by stakeholders. After applying the rules, the CM generates a certificate, known as a *GuardCert*, which records the rule outcomes that were passed or failed. Even if the data fails all rules, it is still stored in the ledger along with its *GuardCert*, enabling further analysis while alerting users to its low reliability. This approach ensures no potentially valuable data is discarded prematurely.

Data Testing is performed on input data, for example, location data from transport vehicle tracking. A set of predefined rules are applied to identify suspicious data, such as limits on average speed between two points to identify suspicious data (e.g., exceeding 120 km/h). These rules are designed collaboratively by stakeholders to reflect realistic expectations for supply chain operations. By decentralizing rule evaluation among multiple blockchain participants, allows all participants to trust but also that the intended rules were properly evaluated against the data.

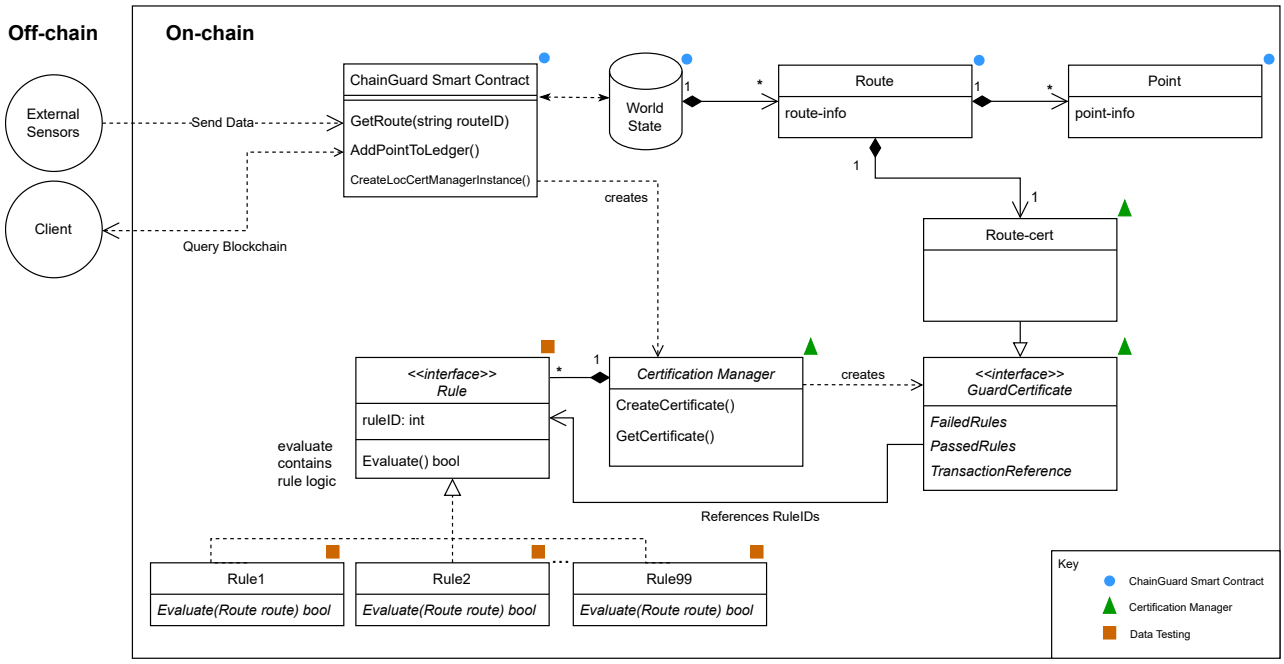

Fig. 2. *ChainGuard* internal architecture.

2) Off-Chain: these components include external sensors that generate input data into the system and client applications that make use of that data. *External sensors*, such as IoT trackers or RFID readers, provide location data for the supply chain. These devices are considered single sources of data, inherently introducing a point of vulnerability. *Client applications* query the blockchain to retrieve information, such as the certified location data of a product.

B. GuardCert Generation

GuardCerts are certificates generated during the data validation process. In Hyperledger Fabric, transactions are endorsed by network peers, with each peer independently validating the transaction and providing a digital signature. The *GuardCert* encapsulates the results of the rule evaluations and includes a Transaction Reference, which links it to the blockchain transaction where it was created. This ensures that every *GuardCert* is verifiable and tamper-resistant. Since all *GuardCerts* have associated transactions that describe how several nodes endorsed their creation, *ChainGuard* uses the ID of those transactions as a placeholder for all the digital signatures that may be found inside the transaction itself. These references to the transaction where the certificate was created are called Transaction References. This concept has been used in other projects, such as Blockcerts [18].

C. Auditing Feature

The *ChainGuard* auditing feature leverages Transaction References to enable stakeholders or auditors to verify the validity of a *GuardCert*. By examining the corresponding blockchain transaction, auditors can review the digital signatures of endorsing peers and validate that the certification process was executed as intended. A provided script facilitates this process, extracting and analyzing transaction details to confirm the authenticity of endorsements.

Similar to what is provided with Blockcerts [18] system, *ChainGuard* is accompanied by a script that enables auditors to use transaction IDs to easily access transactions details. To use it, the raw hexadecimal representation of the transaction must first be acquired. This can be done through the QSCC (Query System Chaincode) smart contract that is present in all Hyperledger Fabric blockchains. The `GetTransactionByID` function of QSCC is called to acquire information about a transaction on a specific channel with a certain ID.

With the acquired hexadecimal representation of the transaction, an auditor can use the script to parse it and read about the details of the transaction, including the endorser signatures. These signatures can be validated with a script with the same structure as Algorithm 1.

D. Trust and Threat Model

We assume that *ChainGuard* is deployed within a permissioned blockchain network. All stakeholders are registered entities with cryptographic identities certified by a Membership Service Provider (MSP) and operate one or more peers. Each peer holds a unique public-private key pair issued by a Certificate Authority (CA). The system assumes that peers do not share their private keys. It is also assumed that the endorsement policy used will accept a transaction

Input: $TxInformationPayload$
$certificates, signatures, message \leftarrow$
$parsePayload(TxInformationPayload)$;
for $i \leftarrow 0$ **to** $len(signatures) - 1$ **do**
  $pubKey \leftarrow extractPubKey(certificates[i])$;
  $hash \leftarrow$
   $sha256.Hash(append(message, certificates[i]))$;
  $result \leftarrow$
   $ecdsa.Verify(pubKey, hash, signatures[i])$;
  $print(result)$
end

Algorithm 1: Pseudocode of signature validator script.

endorsed by a majority of organizations and that a majority of organizations will behave in good faith. Thus, we assume that the blockchain provides data integrity, non-repudiation, authentication and auditability. Incoming data, most likely provided by IoT devices, is considered to be tainted, as it may be erroneous or intentionally manipulated. However, each sensor is assumed to sign its data cryptographically, providing a traceable origin for all input data. In other words, we known who is writing the data but next we need to verify it. The *ChainGuard* certification process uses decentralized execution of rules to detect anomalies, flagging suspicious data. Peers are partially trusted, although the majority is assumed to be operating in good faith, the rest may try to introduce false transactions into the ledger. Since the system is auditable, permissioned and every action can be traced to the source, this behaviour is discouraged since, once detected, the organization would incur significant reputational damage.

IV. Evaluation

As part of a European project to bring blockchain traceability to the agri-food supply chain, we evaluated the effectiveness of *ChainGuard* in real-world scenarios that were designed to trace the locations of cherries and almonds during transport from farm to consumers across multiples regions in Portugal. Six traceability scenarios, described in Table I, were analysed. Figures 3 and 4 are visual representations of two of these routes. The goal was to use *ChainGuard* in real-world scenarios to verify compliance with expected routes.

TABLE I
Fruit traceability routes.

Route	Description
Route 1	Cherries being shipped from a farm to a retailer near Lisbon
Route 2	Truck picking up cherries from multiple farms
Route 3	Cherries shipped from a processing plant to a distribution center near Porto
Route 4	Smaller produce pickup, similar to Route 2
Route 5	Cherries shipped from processing plant to a supermarket in Porto
Route 6	Almonds shipped to Spain for processing and returned to storage in Portugal

A. Evaluation Rules

For the prototype implementation of *ChainGuard*, we developed three rules:

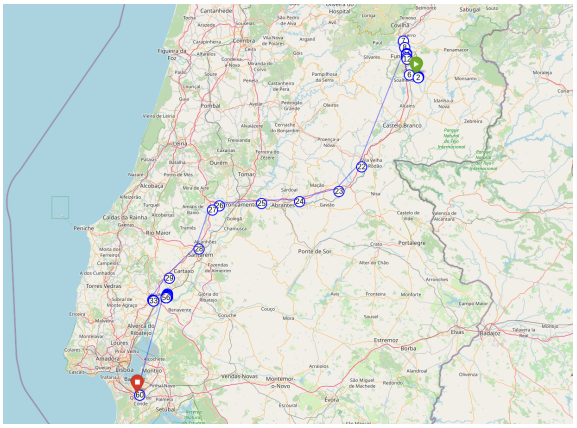

Fig. 3. Route 1 - Cherries being shipped from farm to a retailer near Lisbon.

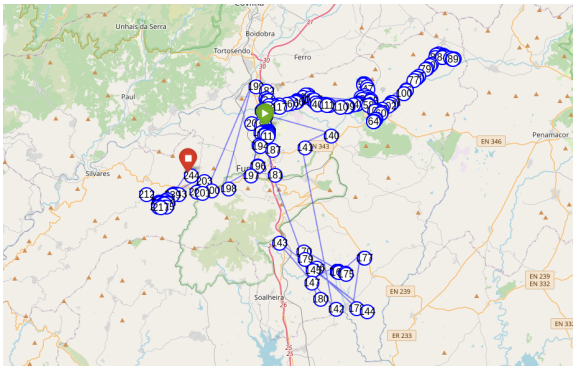

Fig. 4. Route 2 - Truck picking up cherries from farms.

1) **Average Speed Rule**: Checks whether the speed between consecutive points in a route exceeds a predefined limit (e.g., 120 km/h).
2) **Geofencing Rule**: Checks whether all points in a route are within a specified geographic boundary (e.g., within Portugal).
3) **Outlier Rule**: Checks if any point is suspiciously distant from the rest of the route.

These rules were chosen to address anomalies that could occur in the supply chain traceability data, which may indicate erroneous data or potential tampering.

During evaluation, we applied the rules and we assume that all data have been gathered from sensors tracking crates of the fruits. The transportation is exclusively conducted by transport trucks operating on national roads and highways, with no movement occurring outside Portugal. The limits defined by the rules reflect what is deemed acceptable under these conditions, with an added margin of error. Specifically, the maximum average speed is capped at 120 km/h, the geographic boundaries are restricted to Portugal, and any point more than 100 km away from its preceding and succeeding points, if those points are within 50 km of each other, is flagged as an outlier. Any routes containing points that violate these rules will fail the validation. Based on the given scenario and having manually inspected the routes, Table II outlines the

expected results.

TABLE II
EXPECTED RESULTS OF RULE EVALUATIONS ON TEST ROUTES.

Route	Average Speed Rule	Geofencing Rule	Outlier Rule
Route 1	✓	✓	✓
Route 2	✗	✓	✓
Route 3	✓	✓	✗
Route 4	✓	✓	✓
Route 5	✓	✓	✓
Route 6	✗	✗	✓

B. System Performance

To evaluate the system's performance, a total of 669 data points, corresponding to all points in the six routes were added to the blockchain ledger one at a time. The experiments were designed to compare two scenarios:

1) **Adding Points Without Certificate Creation**: The system processes 669 incoming data points and stores them on the blockchain without generating certificates.
2) **Adding Points With Certificate Creation**: The system processes 669 incoming data points, generates certificates by evaluating the defined rules, and stores both the data and certificates on the blockchain.

With respect to the performance metrics, we measured the following metrics in each scenario.

- **Total Processing Time**: The total time taken to process all data points in each scenario.
- **Mean Time Per Point**: The average time needed for a single point to be added to the ledger.
- **Mean Certificate Creation Time**: The average time specifically spent on generating certificates in the second scenario.
- **Maximum Time Per Point**: The most amount of time spent adding a point to the ledger.
- **Minimum Time Per Point**: The least amount of time spent adding a point to the ledger.

These tests were performed on the test network provided in the official Hyperledger Fabric documentation. It used two peers and one orderer node. The used machine had: Ubuntu 22.04.4 LTS 64-bit; AMD® Ryzen 5 4500u with radeon graphics × 6; 16 GiB RAM; 256,1 GB SSD.

Table III summarizes the results for both scenarios, and the data is shown visually in Figures 5 and 6.

TABLE III
PERFORMANCE OF ADDING 669 POINTS TO BLOCKCHAIN LEDGER, WITH AND WITHOUT CERTIFICATION, IN SECONDS.

Metric	Without Certification	With Certification
Total Processing Time	1408.434	1420.11
Mean Time Per Point	2.10529	2.12274
Mean Certificate Creation Time	N/A	0.01745
Maximum Time Per Point	2.504	2.655
Minimum Time Per Point	2.078	2.104

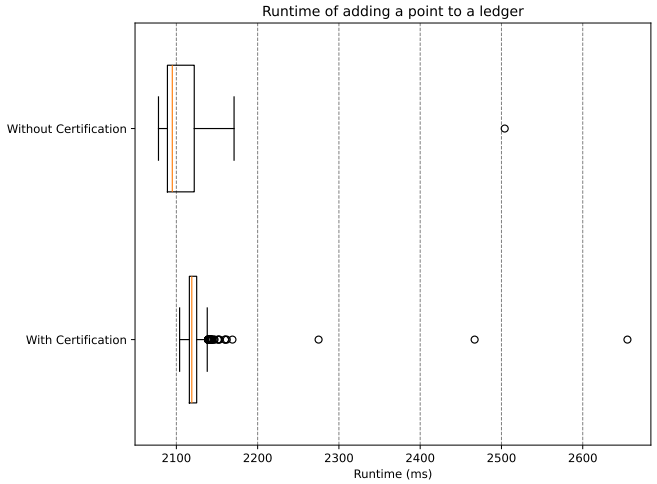

Fig. 5. Performance results box plot.

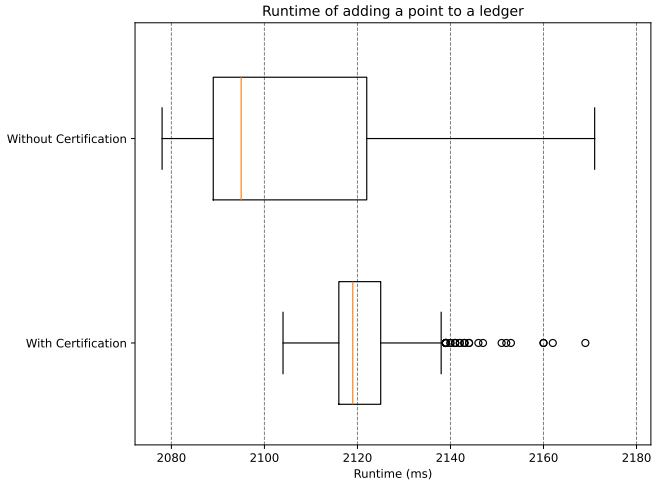

Fig. 6. Performance results, excluding the top 4 outliers.

C. Rules Compliance

After adding all the points to the ledger, the created certificates were exported. The outcomes of the rules for each route were identical to the predictions in Table II.

The results of Routes 1, 4, and 5 passed all rules, while Route 2 failed the Average Speed Rule, Route 3 failed the Outlier Rule, and Route 6 failed both the Average Speed and Geofencing rules.

These findings demonstrate *ChainGuard's* effectiveness in detecting anomalies in traceability data. The Average Speed Rule successfully identified implausibly high speeds in Routes 2 and 6, indicating errors in position data collection. Such anomalies could occur in real deployments and might be exploited by malicious actors.

In Route 6, a point with a very short distance and high average speed did not cause a fail because it was within the expected margin of sensor error, and such cases are accounted for by the rule. The Geofencing Rule performed as expected,

flagging Route 6 for containing points outside the specified geographic region, while other routes passed the test.

The Outlier Rule effectively identified an outlier in Route 3 without false positives. The erroneous point was likely introduced during remote maintenance of the sensors, exemplifying how non-malicious data manipulation could enter the blockchain and highlighting the risks associated with single points of failure in traceability infrastructure.

D. Discussion

Overall, the rules accurately flagged the behavior they were meant to detect. The routes with no outstanding suspicious data passed all rules, while routes with suspicious average speeds between points, locations out of geographic bounds and noticeable outliers, were flagged by the system. For the system to perform as intended, its owners must input rule parameters that are likely to provide useful information. For example, setting the maximum average speed above 30 Km/h is likely to result in more false positives and setting it at 200 Km/h will probably lead to not flagging suspicious positions.

Depending on the implementation, a route possessing a certificate with several failed rules could incite an investigation into that particular shipment, an automatic exclusion of the product from being sold or the auditing of a particular company involved in the supply chain. The evaluation of the implemented rules demonstrates that they are effective in detecting potential problems with gathered data, which might inform decisions by stakeholders regarding the products themselves. The modular and extensible design of the rules allows for continuous improvement and adaptation to evolving requirements in the supply chain.

The results also show that in the *ChainGuard*, the certificate creation adds an average of 17.45 milliseconds per point to the processing time, representing a modest 0.83% increase in runtime compared to without certification. For the 669 points, this equates to an additional 11.7 seconds over a total processing time of 23.47 minutes. Despite four large outliers, most processing times remain within a narrow range. Interestingly, while certification introduces a performance cost, it also results in more consistent runtimes, as indicated by a narrower inter-quartile range (IQR) in the "With Certification" scenario compared to the "Without Certification" scenario.

In the context of traceability systems, this small overhead is acceptable given the enhanced security guarantees provided and because sensors typically send data every few minutes, meaning that an extra runtime measured in milliseconds does not significantly impact overall performance. Therefore, *ChainGuard* effectively delivers increased security through certificate creation while maintaining reasonable performance, even when processing large numbers of data points.

V. Conclusion

This paper introduced *ChainGuard*, a decentralized information verification system addressing blockchains reliant on single sources of data. It enforces stakeholder-agreed rules to verify data, ensuring that all records added to the ledger are trustworthy. *ChainGuard* was tested with real-world agricultural traceability data, tracking fruit shipments. The system detected three distinct location data anomalies.

A performance comparison with direct ledger entry showed *ChainGuard* had minimal impact, measured at less than 1%. These results confirm its effectiveness in enhancing data security and integrity while maintaining efficiency in blockchain data intake.

References

[1] D. Uckelmann, M. Harrison, and F. Michahelles, eds., *Architecting the Internet of Things*. Springer, 2011. ISBN: 978-3642191565.

[2] F. Dabbene, P. Gay, and C. Tortia, "Traceability issues in food supply chain management: A review," *Biosystems engineering*, vol. 120, pp. 65–80, 2014.

[3] 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.

[4] G. Caldarelli, "Understanding the blockchain oracle problem: A call for action," *Information*, vol. 11, no. 11, p. 509, 2020.

[5] S. K. Ezzat, Y. N. Saleh, and A. A. Abdel-Hamid, "Blockchain oracles: State-of-the-art and research directions," *IEEE Access*, vol. 10, pp. 67551–67572, 2022.

[6] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. A. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolic, S. W. Cocco, and J. Yellick, "Hyperledger fabric: a distributed operating system for permissioned blockchains," *Proceedings of the Thirteenth EuroSys Conference*, 2018.

[7] V. Paliwal, S. Chandra, and S. Sharma, "Blockchain technology for sustainable supply chain management: A systematic literature review and a classification framework," *Sustainability*, vol. 12, no. 18, p. 7638, 2020.

[8] A.-A. A. Sharabati and E. R. Jreisat, "Blockchain technology implementation in supply chain management: A literature review," *Sustainability*, vol. 16, no. 7, p. 2823, 2024.

[9] A. Alamsyah, S. Widiyanesti, P. Wulansari, E. Nurhazizah, A. S. Dewi, D. Rahadian, D. P. Ramadhani, M. N. Hakim, and P. Tyasamesi, "Blockchain traceability model in the coffee industry," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 9, no. 1, p. 100008, 2023.

[10] L. Cocco, K. Mannaro, R. Tonelli, L. Mariani, M. B. Lodi, A. Melis, M. Simone, and A. Fanti, "A blockchain-based traceability system in agri-food sme: Case study of a traditional bakery," *IEEE Access*, vol. 9, pp. 62899–62915, 2021.

[11] A. Musamih, K. Salah, R. Jayaraman, J. Arshad, M. Debe, Y. Al-Hammadi, and S. Ellahham, "A blockchain-based approach for drug traceability in healthcare supply chain," *IEEE access*, vol. 9, pp. 9728–9743, 2021.

[12] Y. P. Tsang, K. L. Choy, C. H. Wu, G. T. S. Ho, and H. Y. Lam, "Blockchain-driven iot for food traceability with an integrated consensus mechanism," *IEEE access*, vol. 7, pp. 129000–129017, 2019.

[13] R. J. Lehmann, R. Reiche, and G. Schiefer, "Future internet and the agri-food sector: State-of-the-art in literature and research," *Computers and Electronics in Agriculture*, vol. 89, pp. 158–174, 2012.

[14] N. N. Misra, Y. Dixit, A. Al-Mallahi, M. Bhullar, R. Upadhyay, and A. I. Martynenko, "Iot, big data, and artificial intelligence in agriculture and food industry," *IEEE Internet of Things Journal*, vol. 9, pp. 6305–6324, 2022.

[15] S. N. Steve Ellis, Ari Juels, "Chainlink: A decentralized oracle network," 2017. Accessed on 18.12.2023.

[16] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics and Informatics*, vol. 36, pp. 55–81, 2019.

[17] Hyperledger Foundation, *Using the Fabric test network*. Release 2.2. https://hyperledger-fabric.readthedocs.io/en/release-2.2/test_network.html.

[18] M. M. Lab and H. Credentials, "Blockcerts." https://github.com/blockchain-certificates, 2024. Accessed: 2024-08-02.