Martinez-Rendon, C., Camarmas-Alonso, D., Carretero, J. et al. On the continuous contract verification using blockchain and real-time data. *Cluster Comput* (2021).

DOI: 10.1007/s10586-021-03252-0

On the continuous contract verification using blockchain and real-time data

Cristhian Martinez-Rendon[1], Diego Camarmas-Alonso[1]., Jesus Carretero[1], Jose L. Gonzalez-Compean[2]

Cristhian Martinez-Rendon cristhma@pa.uc3m.es, Diego Camarmas-Alonso dcamarma@inf.uc3m.es, Jesus Carretero jcarrete@inf.uc3m.es, Jose L. Gonzalez-Compean joseluis.gonzalez@cinvestav.mx

1 Computer Science and Engineering Department, University Carlos III de Madrid, Av. Universidad, 30, Leganes, 28911 Madrid, Spain
2 CINVESTAV Unidad Tamaulipas, Km. 5.5 Carrera Soto la Marina, 87130 Victoria, Tamaulipas, Mexico

Abstract
Supply chains play today a crucial role in the success of a company's logistics. In the last years, multiple investigations focus on incorporating new technologies to the supply chains, being Internet of Things (IoT) and blockchain two of the most recent and popular technologies applied. However, their usage has currently considerable challenges, such as transactions performance, scalability, and near real-time contract verification. In this paper we propose a model for continuous verification of contracts in supply chains using the benefits of blockchain technology and real-time data acquisition from IoT devices for early decision-making. We propose two platform independent optimization techniques (atomic transactions and grouped validation) that enhances data transactions protocol and the data storage procedure and a method for continuous verification of contracts, which allows to take corrective actions to reduce operational costs and increase benefits in current supply chains. An automatic deployment of a large-scale distributed business logic system using virtualized appliances is also proposed. Evaluation results show the feasibility, or the solution proposed.

Keywords: IoT applications, Blockchain technology, Monitoring data, Cloud computing

1 Introduction

Supply chains today play a crucial role in the success of a company's logistics, whether it is self-sufficient or cooperating with other organizations. It includes the processes and activities that give value to a product from its creation until it reaches the final consumer. For this reason, in the last years, multiple investigations focus their studies on incorporating new technologies to the supply chains in order to obtain greater benefits in the companies applying improvements in their processes and management of the data. Two of the most recent and popular technologies of the last years, Internet of Things (IoT) and blockchain, are specially interesting to be incorporated in the supply chains and to satisfy the demands of the market for real-time information, transaction security and trust in the operations.

Several authors, such as [1], have highlighted the importance of adopting IoT in supply chains to reduce the operating costs, to improve the capacity to respond to the changing needs of both the environment and customers, and to facilitate decision-making. The key factor lies in the agility of data acquisition from the IoT environment in real-time, which allows for faster processing of the data as soon as it is available.

The use of blockchain technology has also had a deep interest in applicability, not only in the Supply Chain but also in Healthcare System, Digital Right Management, Insurance, Financial System, or Real Estate [2]. The interest is due to the characteristics of the technology, which provides multiple benefits such as the complete operation of the system without relying on intermediaries or centralized entities, decisions made by all participants in the network, system transparency, immutability, reliability, and decoupled verification and processing of the transactions. In [3] and [4], the authors studied some cases using blockchain in supply chain management system. As a result, they concluded that it was emphasized that enabling secured transactions without trusted third parties in an automatic way makes legal and regulatory decisions much simpler. One of the areas that could benefit most is the logistics of food supply chains. Authors, such as [5], describe how to use blockchain technology in food supply chains and mention the main benefits of applying it, among which are more

intelligent decision-making both for the client, the producer and auditors. Moreover, the possibility of achieving inter-organizational trust is a major advantage towards the adoption of blockchain technology in supply chains management [6].

Given the number of factors and variables that influence the supply chain and allow an efficient and profitable development for the companies, it is necessary to model this type of systems to understand their internal behavior and to support the decision-making process in each stage of the supply chain. As an example of this, authors such as Ketzenberg [7] proposed a model to help the operational decision-making process of a company, specifically to decide the number of new products to order based on the uncertainty of demand, the return and recovery of products, and capacity usage. Other authors, such as [8], present a model to minimize the cost of transporting apples and pears in Spain. To do this, they plan the routes of the fruit from the warehouses to the processing plants, and the decision variables are focused on distribution and inventory. The research work in [9] reflects the interest and need to model this type of system. They examine more than 28 operational research models proposed to solve decision problems related to the fresh fruit supply chain. The study classifies the models according to different criteria (decision level, modeling approach, decision variables, applicability, etc.) and concludes that there is a lack of holistic approaches to the design and management of supply chains.

All the models presented above focus their attention on the improvement of internal supply chain processes, whether at the operational, tactical, or strategic level, assuming that they are carried out adequately and without infringing the law. Similarly, data is not monitored and is not considered reliable management of the information exchanged between the participants of the supply chains. One of the main challenges facing supply chains is trust in the data and processes that run there, because it is not possible to guarantee at the moment whether supply chain participants are altering, falsifying, or deleting data for their benefit. This aspect is very important in supply chains of products requiring controlled conditions in preparation, transport, and delivery, like food or some medicines [10]. As an example, the World Health Organization (WHO) has indicated that about 600 million diseases were transmitted by food and there were about 40,000 deaths in 2010 [11]. In these scenarios, the blockchain technology plays a fundamental role in the management of the supply chains. This technology, that provides a decentralized and persistent database [12], is a source of secure, verifiable, and reliable information that would allow participants (consumers, suppliers, supervisors, etc) of a supply chain to know in advance the possible causes of a failure or contingencies that may have suffered a specific product from its origin to the end user.

However, the use of blockchain technology in supply chains has considerable challenges today [13], one of which is the time required to process and confirm a transaction on the network, since all participants must approve the transactions. Therefore, if you do not have an efficient system to process transactions, the scalability of the system could be compromised. The challenge is much greater if, in addition to considering a scenario where you have a secure supply chain (incorporating blockchain technology), you consider a high-demand supply chain (incorporating IoT devices) to obtain all the benefits this entails. Achieving the adoption of these new technologies to traditional supply chains is not trivial. Firstly, the IoT environment devices can capture large volumes of information in short periods. This generates a challenge in the infrastructure required to analyze and manage the captured data, which is not prepared in many companies [14]. Secondly, the volume of information generated by the IoT devices incorporated in the supply chains directly affects the number of transactions that need to be registered and verified in the blockchain, generating scalability and performance issues [15].

Some recent works have contributed to the development of optimization techniques to increase performance of blockchain technology [16–18]. However, these optimizations are platform-specific and do not contribute to a generic solution. Other research works have proposed platforms and systems that allow verification of supply chain processes and data [10, 19–21]. However, these works consider the verification process in the final stage of the supply chain, which prevents, limits or delays the use of corrective measures to cope with failures in the supply chains. A last concern is that many companies lack the necessary architecture (objects, networks, data services).

In this paper we propose a model for continuous verification of contracts in supply chains using the benefits of blockchain technology and the importance of real-time data acquisition from IoT devices for early decision-making to take corrective actions to reduce operational costs and increase benefits in current supply chains. The proposed solution uses temperature, GPS and speed sensors installed in a road transport vehicle because, as indicated in [22], most studies carried out focus on improving road transport of temperature-sensitive products.

Main original contributions of the paper are:

- Two platform independent optimization techniques (atomic transactions and grouped validation) that enhances data transactions protocol and the data storage procedure.
- A method for continuous verification of contracts in real-time.

– A formal method to describe the business logic and the different actors involved.
– A technological solution for automatic deployment of a large scale distributed business logic system using virtualized appliances, including a usable Web interface to assist in deployment and visualization of data and results.
– A reliable verification, involving trusted third parties in the loop, that allows very fast conflict resolution.

The rest of the paper is organized as follows: Sect. 2 describes the state of the art of the works related to the topics of the paper; Sect. 3 describes the model and architecture of the proposed solution. The main components of the proposed solution are then detailed: Sect. 4 described distributed deployment, Sect. 5 described real-time data acquisition, and the contract verification process is described in Sect. 6. Finally, Sect. 7 is presented with conclusions and future work.

2 State of the art

In recent years, there has been significant interest in blockchain technology and, therefore, there are many studies on supply chain scenarios where the use of this technology is very interesting [23–25].

Following Pournader [26], applications of blockchain technology in supply chains, logistics, and transport can be classified using 4 Ts: Technology, Trust, Trade, and Traceability. We follow that approach in this section to classify the solutions in the state of the art related to our proposal.

2.1 Technology

Technology category includes studies that deal with blockchain technology, such as performance analysis on the different available development platforms [27, 28], as Ethereum [29] and Hyperledger Fabric [30], and those studies proposing optimizations to increase the performance and scalability of these tools [31, 32]. Those aspects are especially important for those environments including IoT devices, which send a big volume of data in a short time, making the management and storage of the data more difficult.

Since blockchain consensus protocols are very complex, several publications, pursuing to increase blockchain performance, have carried out a thorough study to determine the bottlenecks that exist when a new transaction is processed and stored into the database [17, 33, 34]. Following the bottlenecks detected, different optimizations have been proposed to reduce the latency in transaction processing and storage. Those optimizations mainly consist in the implementation of local caches, parallelization of the transaction validation, or massive and parallel writing and reading in the database that stores the world state [18, 35–38]. Other interesting improvements, try to reduce the size of the messages by separating the transaction header from the body, which contains the relevant content of the transaction, to send only the transaction identifier and the body, thus reducing the size of the transaction [37]. Finally, another possible optimization is to use a hash table in memory to store the world status, thus increasing performance as access is made to memory and not to hard disk [39].

In addition to these optimizations that affects directly to the transaction validation phase, there are other optimizations that are applied to the blockchain network configuration to increase the number of transactions that can be executed. One possible improvement proposed [40] consists of dividing the peer nodes into two different roles: Commitment and Endorsement, allowing the peer nodes to require fewer resources and to be able to scale the network on demand. Other optimizations consist of adjusting parameters, such as the number of channels on the network, the database used to store the world state, or the resources available to the nodes (e.g. number of CPUs or the network bandwidth) among other parameters. Thakkar [17] made a performance analysis using this type of optimizations and noticed that the number of transactions increased with some of the proposed configurations, thus increasing the performance.

As shown above, most works propose to modify the architecture of the development tool so that the validation phase has a shorter execution time, as it is the main bottleneck detected in blockchain technology [17, 18, 37]. However, these optimizations depend on the development tool and the consensus protocol used to validate the transactions. Thus, they are not generic optimizations.

2.2 Trust

The second classification category is Trust provided by the blockchain technology. Studies in this category analyze the different development tools and network taxonomies available to determine which one provides better security, confidentiality, immutability, decentralization, anonymity, or auditability of data. Related to this category, [26] and [41] compare the different network taxonomies available in blockchain technology: public, private, and consortium.

In public blockchain networks, the transactions are visible to all participants and allow a completely decentralized blockchain network to be established. Furthermore, the high number of participants in the network guarantees the immutability of stored data, because it is nearly impossible to modify the data as they are replicated by all participants. However, this information is accessible to all participants, and therefore reduces confidentiality because anybody can carry out and validate transactions without having to be a member of a particular organization. The Ethereum development tool [29] uses this type of network taxonomy. The most successful example of usage of Ethereum are the blockchain implementation modern cryptocurrency developments, like bitcoin [42]. Another example of application is the platform called BanQu, which aims to connect the unbanked to the global economy through mobile phones and a blockchain network [20]. That platform allows people to record their financial transactions in the blockchain history owning their data. The main characteristics of the former platforms are profitability, transparency using secure and auditable records, implementation anywhere, and sustainability. However, public and open networks have several documented problems. For example, a study on BanQu [43] indicated that it is not clear where the decentralized ledger is stored. Additionally, the infrastructure needed for the technology deployment in the region of the intended customers is challenging. Another study [44] showed that Bitcoin transactions can be linked to users, which compromise anonymity. Moreover, the scalability of those solutions is generally poor, as they do not have any limit for block size. Trying to overcome this problem, a new protocol designed to scale have been proposed in [45].

In contrast to public networks, private networks only have participants who are members of a particular organization [46]. This removes the anonymity of the participants in the transactions, because they must authenticate their identity with a certificate. In addition, despite the network's being decentralized, a reduced number of participants can give rise to a centralized network vision. This also has an influence on immutability, because it is a lot easier to modify a transaction, but the immutability level is still high. However, the confidentiality of the data is greater, since only some participants will view them.

The last type in the network taxonomy are consortium networks [47], which are a combination of the two previous types of networks. In these networks, data is public, but only a specific group of participants is involved in transaction validation. An example of a development tool that can use this type of networks is Hyperledger Fabric. Those networks, using consortium blockchain and smart contract technologies, are now being used for IoT systems to achieve secure data storage and sharing. An example of their application in vehicular edge networks is shown in
[48].

Finally, it is relevant to highlight the studies carried out by [49, 50], and [51], where they compare different development platforms, such as Hyperledger Fabric and Ethereum, focusing on the reliability of the data stored in the transactions and the network type implemented. As shown in those works, private networks offer better reliability because only certain actors, who are trusted within the organization, participate in the validations, thus enhancing also security as they do not allow participants who do not belong to the organization to view the content of the transactions. Anyway, security in IoT networks is nowadays a major concern. Thus, several works have proposed to include blockchain solutions in IoT systems to increase security in data access, transmission, and storage [52, 53], since blockchain-based solutions ensure trust and stability [54]. Puri et al [55] has recently proposed to use smart contract-based policies to enforce privacy and security for the Internet of Things.

2.3 Trade

A major trend of blockchain applications includes works using cryptocurrencies, such as Bitcoin [56], to facilitate financial transactions and supply chain financing [57, 58]. A major application of blockchain technology in this category is the Chinese ''One Belt One Road'' (OBOR) initiative to manage logistics and transport operations [59]. In this initiative that seeks to connect China with numerous countries in Eurasia, Africa, and Oceania, the authors claim that the application of blockchain

technology for the transfer of funds in a considerably shorter time, and with lower costs compared to traditional methods, seems to be a viable option for the financing of projects throughout the OBOR region. Furthermore, the use of a unified cryptocurrency can reduce transaction costs and create a single commercial market for OBOR countries [60].

Another are of interest is focused on increasing the energy consumption efficiency of peer-to-peer electricity trading in intelligent electrical grids to increase sustainability [61, 62]. Blockchain is used to provide reliability, trust, security and privacy through consortium networks.

[19] propose a blockchain-based token deposit system to solve the business need in the maritime logistics industry. It uses Ethereum as a technology platform and provides a RESTful API for business integration. The system has had the participation of large carriers such as Sealand of Maersk and Cosco, shippers such as Li&Fung Logistics, BASF, JF Hillebrand, Esprit, etc. However, the proposed system suspended its operations in October 2019 for two reasons: firstly, the volume of transactions was too low to continue operating commercially; and secondly, there were design problems, as, according to the authors, some shipments were not executed according to the reservation and the shippers were often unable to confirm their reservations during the high season due to performance problems.

As may be seen, even with the many advantages provided, the usage of blockchain for trade still has several challenges that can compromise the success of this technology [63].

2.4 Traceability

Ensuring sustainable and ethical operations of the supply chain needs traceability and security of inventories. As blockchain technology provides both features, there are many applications described in the literature, as shown below.

A smart logistics architecture for the supply chain using of Ethereum Smart Contracts, microservices, automatic learning, and big data was proposed in [64]. The solution proposed includes a logistics planner and a supervisor of the contracts that establish how the assets should be managed. While the study describes an end-to-end solution that encompasses contract negotiation and monitoring, the scenario described is unrealistic as the scenario is very small (e.g., a single consumer, an asset). Additionally, the study does not describe the cost that the use of blockchain could imply in the traceability of assets. On the other hand, the authors do not apply measures, such as throughput or latency, to give an idea of the amount of data that can be processed with the proposed solution.

Bext360 [21] is a platform focusing on supply chains to provide a traceable footprint from producer to consumer. Among the main features of the platform, outside those inherent to the blockchain technology, it highlights the possibility of making configurable solutions and offers APIs that allow Bext360 technology to be integrated into websites, supply chain management systems, point of sale systems and more. Data is collected at each stage of the supply chain and chained to an immutable record. However, as mentioned by the authors [43], it is not clear how the data is manipulated in the later stages of the supply chain, so that the consumer may not have a complete picture of the supply chain.

Guidelines for building a traceability system with reliable information on the agrifood supply chain are presented in [65]. They propose to use RFID (Radio Frequency Identification) technology to implement the acquisition, circulation, and exchange of data in the production, processing, storage, distribution, and points of sale of the supply chain. On the other hand, they assure that the blockchain technology allows us to guarantee that the information that is shared and registered is reliable and authentic. However, the authors do not implement a prototype that would allow them to prove the benefits they mention. Their work focuses on indicating where each RFID tag should be located, what information should be recorded on each tag and the blockchain, and how to use this information in each of the stages of the food supply chain.

Studies of the use of the blockchain for the provenance and traceability of the Internet of Things integrated with food logistics were presented in [5] and [66]. They describe different cases implemented in various parts of the world. One of the cases studied is a test of the usage of blockchain by Walmart for monitoring pork in China and mangoes in Mexico [10]. The tests determined that the use of blockchain can reduce the information tracking time from one week to 2.2 s.

The design of a secure logistic method, with anonymous and light verification, is presented in [67]. It focuses on 4 main phases for process improvement: session key generation phase, order request phase, package pickup phase, and product transfer phase. The authors use lightweight encryption technology to protect information and avoid filters during the delivery process. They also incorporate Elliptic Curve Cryptography (ECC) to generate session keys that are used to secure data transmissions and the BAN logic model [68] to test the correctness of the scheme with mutual authentication. On the other hand, they achieve non-repudiation and data integrity by using digital signatures. Finally, they perform a descriptive analysis

of the calculation, storage, and communication costs for the buyer, the seller, the logistics, and the distributor in each phase. However, they do not show an implementation of the proposal in a real-world scenario.

2.5 Blockchain and supply chains

The blockchain technology provides important advantages such as traceability or trust which is very useful to improve the reliability of supply chains, mainly for sensitive products such as food or medicine, although it can also be used for any other goods. For those reasons, different studies apply blockchain to supply chains.

For example, blockchain has been used in agri-food supply chains to certify and warranty the origin of goods to prevent frauds [5, 69–71]. Furthermore, it has been used in medicine supply chains, because many of them are sensitive to temperature or humidity, among other factors. Therefore, it is essential to ensure that the medicine is in optimal condition when it is consumed, following the Good Distribution Practice of medicinal products for human use (GDP). For example, in [72] they use sensors to take measurements that are sent to a cell phone, which records the measurement in the ledger. Moreover, blockchain technology has been also used in supply chains for high value items to prevent counterfeiting, such as in the case of Diamonds [73].

However, as explained in [74, 75] and [4], most works focus on studying and describing conceptual or theoretical models of potential applications of this technology in supply chains, but they do not explain the technical details to implement the model or analyze its performance to determine if it is feasible to use blockchain in production of the described use case.

2.6 Limitations of previous work

A summary of the features of the solutions analyzed is presented in Table 1. Their main strengths and limitations using major features are discussed below.

1. Generality and performance We have identified the main performance bottlenecks in blockchain and the solutions that have been proposed in the literature to solve them to obtain better performance and scalability of the Blockchain platform. The performance limitations are evident in terms of transaction volume, latency, and block size, or in bytes of transactions related to traditional systems such as Visa or Mastercard [26] where about 50,000 t/s are processed. This demand for transactions from traditional applications has led to strong efforts to optimize each development platform or a specific consensus protocol. However, this approach prevents a generic coupling of the solutions proposed to other blockchain platforms.
2. Blockchain-IoT integration The design of most of the works analyzed does not allow the integration of blockchain technology with systems that support the data load produced by different IoT devices. The majority of the proposals consider a predefined scenario or an already established and unchanging data set over time.
3. Monolithic deployment A major limitation of the proposal found is the lack of dynamic deployment of the blockchain network together with the solution they propose. In many studies, the deployment of verification networks and the proposed solution are in the same infrastructure, so that it is not possible to have a decoupled, distributed, and decentralized system. In this sense, current work does not consider factors such as latency resulting from the geographical distribution of network nodes or scenarios in which possible network nodes or solution components fail.
4. Manual and supervised deployment Part of the studies analyzed deploy the blockchain network manually through the knowledge of an expert in the area. This dependence on the expert's knowledge limits the scalability of the system in scenarios with a complex network, considering additionally that companies or entities not an expert in the use of blockchain could not deploy a solution for lack of technical knowledge in the area.
5. Use of virtual machines With the advance of technology, different studies have highlighted the advantages and benefits obtained by the containers against the use of virtual machines [76, 77].
6. Interoperability and integration with existing systems There are currently different implementations of blockchain designed by several companies. Communication between these implementations is not possible at this time, which prevents their interoperability and widespread adoption for data storage and analysis of higher-order and capacity [78].

7. Post-registration verification Some works consider the registration of transactions in real-time, and other accumulate data before performing the registration. However, the validation of these accumulated data occurs in the final stages of the value chain, limiting or preventing the use of early corrective measures to reduce the cost of damage caused.

This paper aims to address the problems discovered in the literature reviewed by designing, building, and deploying a system that includes all the fields analyzed in Table 1.

Table 1 Summary of state of the art. Technology (BC: Blockchain, IoT); Deployment (Di: Distributed, Co: Container); Verification (RT: Real-Time, Cn: Continuous); Visualization (Da: Dashboard)

	Tec		Dep		Ver		Vis
	BC	IoT	Di	Co	RT	Cn	Da
Baliga [27]	✓		✓				
Nasir [28]	✓						
Pournader [31]	✓						
Saberi [32]	✓						
Tian [17]	✓						
Javaid [18]	✓		✓				
Gorenflo [37]	✓		✓				
Arumugam [64]	✓				✓		
Treat [65]	✓	✓					
Leung [19]	✓						✓
Gadnis [20]	✓				✓		✓
Jones [21]	✓	✓					✓
Kamath [10]	✓	✓		✓	✓		

3 Design of the proposed solution

This section describes how the proposed system have been designed to overcome the problems or limitations described in the previous section. Four main phases are considered to perform a continuous verification of contracts using blockchain and real-time data: definition, deployment, data acquisition, and contract verification. In the definition phase, the components and elements of the proposed architecture are declared using a mathematical model described below. The deployment phase is described in Sect. 4 and the operational phase, which includes the data acquisition and contract verification process, is described in detail in Sects. 5 and 6 respectively.

3.1 Mathematical model of the proposed solution

We have developed a mathematical model to represent the components and entities of the four previously mentioned phases and the necessary operations to carry out a continuous verification of contracts using blockchain and realtime data in different business logic. This section describes the operational model proposed and the notations required for its implementation in the definition phase, the deployment phase. and the operational phase.

3.1.1 Definition phase

Blockchain Network The definition of the blockchain network consists of establishing the number of nodes, that will make up the network, and the roles that each node will have, as well as the consensus protocol to be used, the communication channel, the certification bodies, and those participants in charge of ensuring consensus. Additionally, this phase carries out the generation and creation of artifacts that guarantee security and provide the main characteristics of a blockchain network. The blockchain network deployment definition is made in the following way:

1. Organizations ! Org An organization represents a set of participants in the blockchain network with a unique purpose. For example, in a supply chain one organization may represent the process of data acquisition, another organization may be in charge of data processing, and another organization of product distribution. They are denoted as Org ¼ fOrg$_i$g 8i in1...n.
2. Nodes From the verifiability network point of view, each organization has a set of Nodes N that are part of a verifiability network and that access this network through a REST API. There are different types of nodes in the verifiability network:

   (a) Peer nodes, N_{peers} Set of nodes of the verifiability network that issue transactions. It is denoted as N_{peers}½j ¼ x, where x 2 N ¼ fN_1;N_2;N_3;...g: For simplicity purposes, this parameter has the same value for all organizations along the paper. For example, if you have a configuration of 2 organizations Org ¼ fOrg_1;Org_2g and jN_{peers}j ¼ 2, it means that each organization will have two peer nodes, resulting in 4 peer nodes in the verifiability network. The general formula is given below:

$$N^{\circ}\ Peers\ nodes = \underbrace{|Org|}_{N^{\circ}\ Organizations} * \underbrace{|N_{peers}|}_{N^{\circ}\ Peers\ by\ Org}$$

ð1Þ

   (b) Orderer nodes, N_{or} Set of nodes of the network in charge of establishing a mechanism of consensus between the peer nodes at the moment of issuing a transaction. The objective of the ordered nodes is to have a verifiable network of ordered and consistent records. It is denoted as N_{or}½l ¼ y, where y 2 N ¼ fN_1;N_2;N_3;...g:

   (c) Certifying authority nodes, N_{ca} Set of nodes that issue the certificates and private keys to all the peer nodes belonging to a certain organization. Each organization has a certification authority node. Therefore, certification authority nodes are denoted by: N_{ca} ¼ fN_{caorg1}; N_{caorg2}; ...;N_{caorgZ}g where Z ¼ jOrgj.

   (d) Client nodes, N_{cli} Set of nodes providing the access interface between the participant entities and the physical network of verifiability. Through those nodes, the authorized participants execute each transaction in the network. In addition, clients are associated with a specific organization, which may have a different number of them: N_{cli} ¼ f$N_{cliorg1}$;N_{caorg2};...;N_{caorgZ}g where Z ¼ jOrgj being $N_{cliorgk}$½j ¼ c; 8j in 1..N, where c 2 N ¼ fN_1;N_2;N_3;...g:

3. Smart Contract, SmC A file that includes the functionalities required to carry out the business logic (queries and functions definition). Even if the definition of the business logic is specific of each business, we here describe the model with our logistic of truck shipments. Below we show the participants (company), the fleet of trucks of each company, the routes of trucks, the contracts, and the shipments that will be registered as transactions in the blockchain verifiability network. The definition of each of the entities and elements mentioned above is done in the following order:

   (a) Participant creation A participant Pa is defined a tuple of values, such as

\id; type; email; address; quantity[. The \id [allows the unique identification of the participant (in our case a unique name), the \type[identifies the role in the business (in our case can be grower, importer, or truck); \email[and \address[attributes are used to contact the participant.

(b) Fleet of trucks creation A fleet Fle is established as a set of trucks for transporting assets. The fleet is denoted as Fle ¼ ftru$_1$; tru$_2$; ...;tru$_n$g, where each tru$_i$ is a transport truck including a tuple of \id; sensors[values. The sensors of each truck are denoted as ST$_t$ ¼ fSe$_1$; Se$_2$; ...;Se$_n$g, where t is the truck identifier and each Se$_i$ is a sensor identifier incorporated in the truck, which can be temperature, GPS, speed, etc.

(c) Route creation ! Rou A route is established as a set of GPS points that a certain truck (Tru$_i$) must follow. The route is denoted as Rou$_{sd}$ ¼ fpoi$_1$;poi$_2$;...;poi$_n$g where s is the origin point and d the destination point, and where each poi$_i$ is a GPS point formed by the tuple of values \latitude; longitude;date;time[. The \latitude[and \longitude [attribute indicate the position of the truck, and the \date[and \time[attributes are set to allow time and space restrictions on the routes that trucks must follow.

(d) Contract configuration A contract C establishes the parameters agreed upon by the different participants for the execution and fulfilment of a transaction. A contract is denoted as C ¼ fPas;arrival time;price;penalty;Resg where Pas ¼ fPa$_1$;Pa$_2$;...;Pa$_n$g is the set of participants that establish the contract,
\arrival time[is the time limit established for the fulfilment of the same, \price[is the value of the contract, \penalty[is the amount to pay if the contract is violated, and Res ¼ fRe$_1$;Re$_2$;...Re$_n$g denotes the set of constraints to which the contract is subject. For example, the value of Re$_1$ could correspond to a minimum value of temperature that can receive an asset or Re$_n$ could be the maximum value of the acceptable speed of the truck that transports the asset.

(e) Shipping configuration ! Shi Shipments are made by trucks (Tru) and are tied to contracts (C) and established routes (Rou). A shipment is denoted as Ship$_{id}$, asset, status, unit count, C$_{id}$, tru$_{fi}$, Rou$_{id}$}, where: id is the shipment identifier; asset is the goods or object being transferred shipment; status can be created, in transit or delivered; unitcount determines the number of assets being transported; C$_{id}$ corresponds to the contract identifier to which the shipment is tied; tru$_{fi}$ identifies the truck i of the fleet f that makes the shipment; and Rou$_{sd}$ is the route identifier that the shipment must follow.

(f) Establish queries The model is designed to consult the current and historical values of the records of each of the elements or entities belonging to the business logic.

3.1.2 The deployment phase

In the deployment phase of the proposed system, the elements that make up the model are containers and managers. They are responsible for configuring, creating, and deploying the artifacts necessary for the deployment and operation of the verifiability network:

1. Container Denoted as C$_o$, it represents a virtual machine in which all the technological requirements of a certain system or application are encapsulated to be deployed and executed in any infrastructure. Even if a container might include several nodes, in the design proposed, each node of the network is encapsulated in a single container to optimize distribution and location of resources. Thus, the total number of containers deployed by the system is determined by the values established in N$_{or}$, N$_{cli}$, N$_{peer}$, and the number of organizations |Org|. The general formula for estimating the number of nodes deployed is shown below:

$$N^{\circ}\ Deployed\ C_o = \underbrace{|N_{ca}|}_{A} + \overbrace{2 * (N^{\circ}\ Peers\ nodes)}^{B} +$$

$$\underbrace{(2 * N_{or})}_{C} + \overbrace{(N_{cli})}^{D} + \underbrace{1}_{E}$$

Where A is the number of Certifying authority nodes; B is the number of Peer and DB nodes (for each peer node there is a database node); C is the number of Consensus nodes (for each orderer node two auxiliary nodes are deployed: Kafka and zookeeper); D is the number of Client nodes; and E is one container where the intelligent contract is initiated.

2. Managers are all those scripts, applications, or programs in charge of executing each of the phases of the proposed system: declaration, deployment, operation (data acquisition and verification). To carry out these phases, three types of managers are devised:

   (a) Configuration Manager Process executing the configuration scripts needed to implement the verifiability network. These scripts are generated by the proposed system through a web application from the parameters defined by the user in the definition phase.

   (b) Creation manager. Manager that run the different creation scripts needed to generate: (1) the security elements (private keys, public keys, certificates, etc) of each of the nodes and users that will make up the blockchain network; (2) the communication channel through which messages will be exchanged between each of the nodes; (3) the first block of the blockchain network called the genesis block and (4) the intelligent contract that will be deployed in the verifiability network. The Hyperledger Fabric platform is integrated into the system proposed for this purpose.

   (c) Deployment Manager Program in charge of receiving the configuration files and deploying, or activating, the services needed for the correct execution and operation of the verifiability network. In our system, the Docker platform [79] is used for this purpose.

3.1.3 The operational phase—data acquisition

Each Tru_i truck collects data with their respective sensors ST_i ¼ fSe_1;Se_2;...;Se_ng where i corresponds to the truck's identifier and n is the number of sensors it may have installed. In turn, each sensor has associated a frequency rate (fr_{sei}) in seconds to determine the frequency of data collection. For example, the Se_1 sensor has an associated frequency rate fr_{se1}, if the value of fr_{se1} is 60 s, it means that every minute the Se_1 sensor belonging to ST_i of the Tru_i truck is obtaining data from its environment. The frequency rates denoted by FR_i ¼ ffr_{se1};fr_{se2};...;fr_{sen}g may vary from each other, regardless of whether they belong to the same Tru_i truck.

The proposed system is designed so that a transaction (Tx_{se}) is performed to record the data collected for each sensor, following the frequency rate of each ST_i sensor, in the blockchain verifiability network. Each transaction is denoted as Tx_{se} ¼ fSe_{id}; Shi_{id}; timestamp, valueg, where \$Se_{id}$ [is the sensor identifier, \$Shi_{id}$ [is the shipment identifier, \timestamp[is the data capture date, and \value[is the set of values collected by the sensor. In this context, the number of transactions to the blockchain, issued by a shipment ($Tx_{shiitruckj}$) is given by:

$$Tx_{shi_i\_truck_{j-k}} = \sum_{i=1}^{n} fr_{se_i}$$

Where i corresponds to the number of shipments made by the truck j of the fleet k, and n corresponds to the total number of sensors installed on the truck j.

Therefore, the number of blockchain transactions that are made is given by:

$$\#\_Tx\_data\_collected = \sum_{k=1}^{nf}\left(\sum_{j=1}^{nt}\left(\sum_{i=1}^{ns} Tx_{shi_i\_truck_{j-k}}\right)\right)$$

Where nf corresponds to the number of fleets, nt is the number of trucks, ns is the number of shipments, i corresponds to the id of shipment made by the truck j of the fleet
k.

3.1.4 The operational phase—contract verification

In this phase, two elements are considered to carry out the verification process: (1) the penalty clauses; and (2) the type of verification to be performed.

1. A Penalty clause is defined to adjust the value of the contract established according to the values received from each sensor and the compliance with the restrictions established in the contract. A penalty clause is declared per sensor in the model and it is defined as the following three-element function
   Pen_i ¼ fx;cond;penaltyg, where x is the value received in real-time from the sensor, cond is the conditional function that must be met to record the value of x as valid, and penalty determines the value of the penalty to be deduced from the total value of the contract.
2. Type of verification The verification process consists of contrasting the data acquired and registered in the blockchain, with the restrictions stated in the contract for such data. It also applies the penalty clauses if required. The proposed system incorporates three types of contract verification:

   (a) Verification by identifier. In this case only one record identified by a unique id in the network is analyzed. This type of verification is useful when
   the origin and time of the failure is known, and it is conceived to verify the restrictions and possible penalties for a single record in particular.
   (b) Lot verification. Since all records in the network have by default the time stamp for transaction creation, the lot verification is meant to recover all records in the blockchain which time stamp is within a specified time interval. This interval could be previously defined by the user or could be calculated automatically to execute the verification periodically.
   (c) Results verification. This type of verification is thought to reuse the results of previous verifications and can be useful in subsequent queries for different purposes and reduce costs of running again previously executed verification processes. As different actors may need to verify the blockchain records, and since the records in the blockchain are immutable, it does not make sense to repeatedly run verification processes on already verified set of records as they would always give the same results.

The general architecture of the proposed design is shown in Fig. 1. In the first part, there is an administrator who is responsible for defining and configuring each of the elements and entities that make up the blockchain network and business logic. In the second part, the different managers (creation, configuration, and deployment) receive the information provided by the administrator and automatically perform (scripts deploy_network_N.sh and deploy_logic_N.sh) the necessary processes to implement the blockchain network (part 3, Fig. 1) and run the business logic within it (part 4, Fig. 1).

The proposed design is generic and allows the deployment of any business logic. In Fig. 1, the business logic consists of transport organizations, each one having a fleet (Fle) of trucks that transport different types of assets. For the sake of simplicity, Fig. 1 shows a Business Logic i, where each organization i has one truck (Tru_i) with three sensors associated ST_i ¼ fSe_1;Se_2;Se_3g, where Se_1 is used to control the speed of Tru_1, a GPS sensor Se_2 is used to control the route that Tru_1 must follow, and Se_3 is a temperature sensor for the food that Tru_i transports. In this scenario, a truck Tru_i is assigned to make two deliveries (Shi_1 and Shi_2) that are subject to a contract (C_1 and C_2 respectively). For each contract, the arrival time for the goods, the price to be paid for the contract, and the route (Rou_{sd}) that Tru_i must follow are established. Also, the participants who established the contract Pas ¼ fPa_1;Pa_2g where Pa_1, in this case, corresponds to a Grower and Pa_2 to an Importer. Finally, the restrictions Res ¼ fRe_1;Re_2;Re_3g are established in both C_1 and C_2, where the restriction Re_1 establishes the minimum and maximum temperature values that can receive the food transported by Tru_i, the restriction Re_2 establishes the maximum speed value and the restriction Re_3 establishes that each GPS point of Tru_i must be within a maximum distance allowed to some point that belongs to the Rou_{sd} route.

Any authorized participant of a Business Logic can perform transactions to the Blockchain Network through the client nodes (e.g. N_{cli1} and N_{cliN}). The transactions that they can perform are determined by the Smart Contract (SmC) established in the definition and configuration phase. Similarly, part 5 of Fig. 1 shows that a government entity, auditor or verifying organization, can access through a client node ($N_{cliauditor}$) to the data of the blockchain network with the purpose of verifying the contract and detecting errors to execute any decision-making process based on a reliable and immutable source.

3.2 Architecture of the solution

This section presents the result of implementing the model described and the processes that are executed from that model. In the first phase, an administrator user is responsible for defining and configuring each of the elements of the model (organizations, nodes, participants, contracts, etc.) through a web service. As a result of this configuration the proposed system generates configuration files needed to deploy the business logic and the blockchain network for data verification (see part 1, Fig. 2).

This phase of deployment has two main purposes: to implement a blockchain network ready to receive transactions, and to register the base elements of the business logic in the network that allow starting the day-to-day operations of one or several organizations. Both purposes are performed through the automatic execution of a script (network_N.sh, see part 2, Fig. 2). Each of the processes carried out in the deployment phase is described in detail in the Sect. 4.

After the deployment phase, the blockchain network is in operation and ready to receive transactions from the business logic. In this work, a use case of a truck fleet for food transportation was created. Each truck captures and registers in the blockchain network real-time data from speed, temperature, and GPS sensors. Contracts based on these data were established, for the fulfillment of routes and management of the quality of the product that was transported, to support the decision-making process in case of eventualities and problems along the supply chain (bad condition of the cargo, deviation of routes, evasion of responsibilities, etc) that directly influence the costs of the operations for different transport companies.

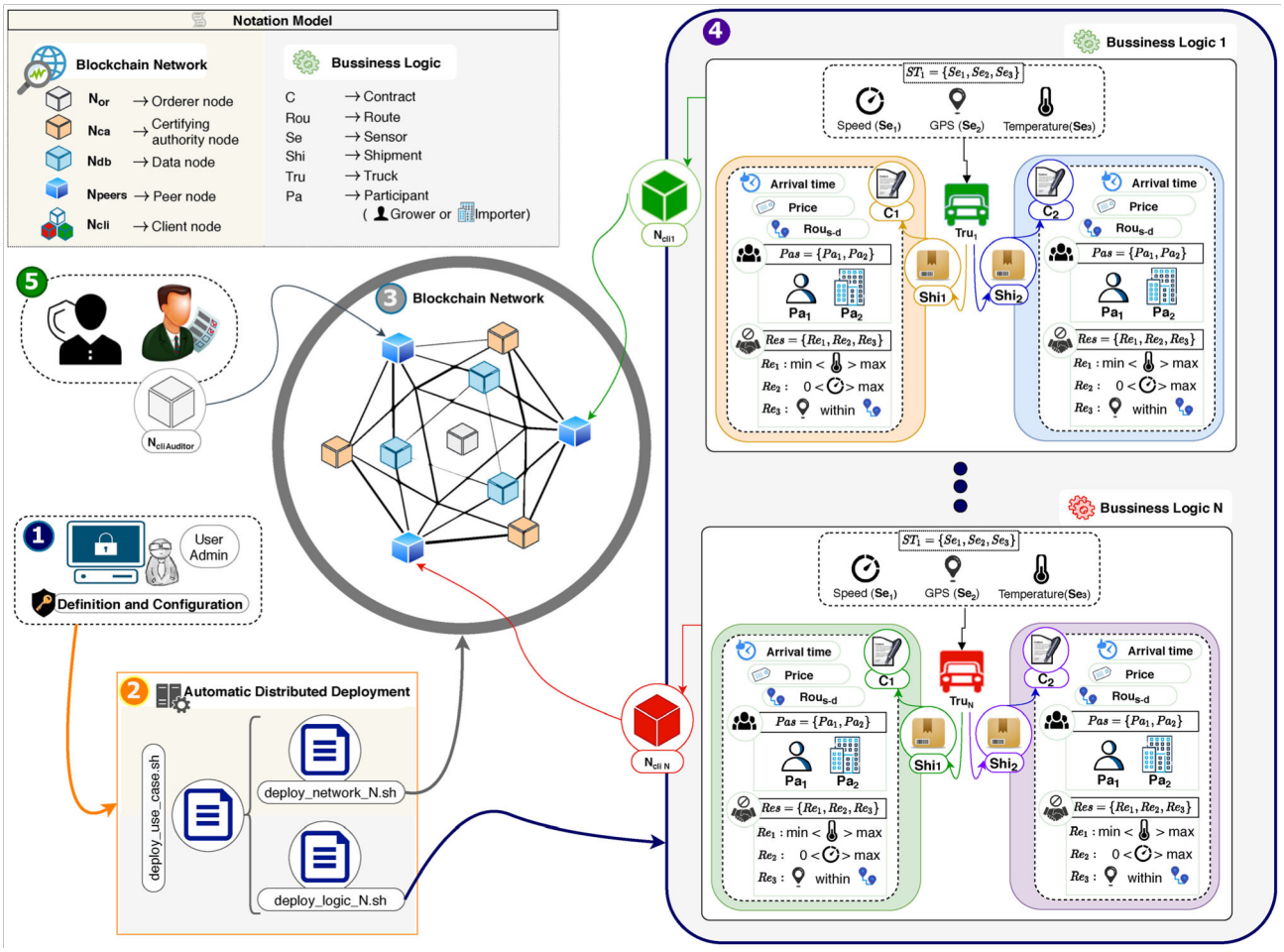


Fig. 1 General architecture with the designed model

Simultaneously to the data acquisition, a contract verification process is carried out periodically (see part 4, Fig. 2). The penalty clauses are applied as soon as the contract restrictions are not fulfilled. The results of the verification and application of penalties are also registered in the blockchain network for traceability purposes, which allows government entities, auditors, or entities external to the organization participating in the business logic to verify the status of each of the contracts established in the system. In the following sections, the deployment, data acquisition and contract verification phases are described in detail.

4 Distributed deployment solution

This section details the second phase shown in Fig. 2, which represents the automatic deployment of the blockchain network and the business logic. During this deployment phase, all the components and participants that are part of the system are defined, configured, and implemented. To carry out this description, we are going to explain first how the deployment of the blockchain network is performed (Sect. 4.1) and then the deployment of the business logic (Sect. 4.2).

4.1 Automatic blockchain network deployment

The deployment of the blockchain network is carried out using scripts and JSON and YAML configuration files, allowing the automation of the process. Firstly, the cryptographic material has to be generated, because it is necessary to identify the participants during the validation of a transaction (Hyperledger Fabric requires the identification of all the participants as it is a private network). Then, all the necessary nodes, such as orderer nodes (N_{or}), peer nodes (N_{peers}), or client nodes (N_{cli}), among others, are defined and deployed for the correct operation of the blockchain technology. Those nodes are configured and packaged in containers (C_o) to simplify their deployment and to enhance scalability in different environments and infrastructures, especially in shared environments. These advantages are due to the fact that the containers prevent interference between the software package in the container and the software installed in the computer, as this system is self-contained [80]. Furthermore, these nodes can be deployed locally, that is all nodes in the same computer, or they can be deployed in a distributed system to allocate the nodes in different computers and avoid a single point of failure. We have used Dockers to develop our prototype and Docker Swarm [81] to carry out the deployment of the distributed network by executing multiple containers in different computers, enabling them to communicate with each other at any time.

Once all nodes of the network have been deployed, the communication channels, to transmit and replicate the transactions among all the peer nodes in the network, are created. Finally, the smart contracts (SmC) are installed and instantiated in the

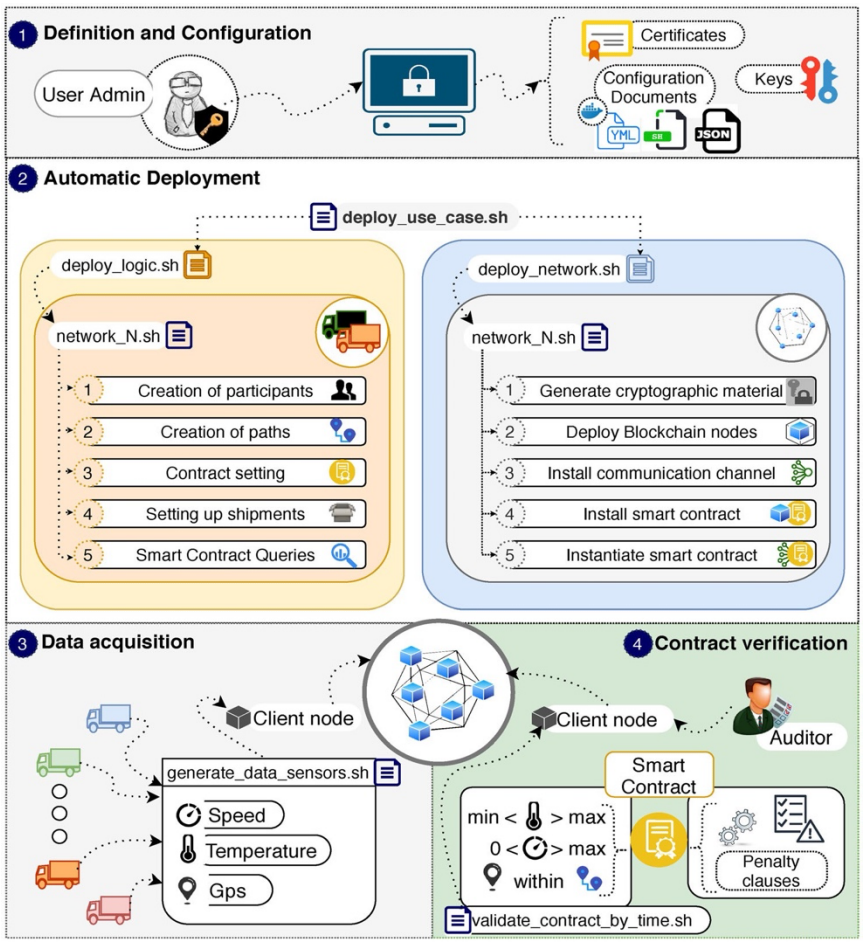


Fig. 2 Definition phases of the designed model: definition, deployment, acquisition and verification of data

peer nodes, allowing network participants (truck, grower, auditor, etc.) to carry out transactions

4.2 Business logic deployment

The business logic deployment, which is composed of each of the entities and/or elements that interact with the blockchain network for trading assets, is made after the blockchain network has been established. Scripts are used to automate the full process, just as we have shown for the deployment of the blockchain network.

The different participants who will interact in the network are defined, as well as the contract terms that must be complied and the queries that can be carried out, are defined in this phase. It is important to notice that, to define all these entities, transactions are carried out in the blockchain network to define participants' features or attributes or contract terms.

In the use case presented in this paper, the scenario is composed by entities that allow the trade and transport of foods, which are monitored, in real-time, by a temperature sensor, a speed sensor, and a GPS sensor, to always verify that the terms of the contract are complied with. Therefore, to deploy the business logic, firstly all the participants (Pa) who are members of the network and carry out transactions are created. Those entities include growers, importers, trucks, and sensors (ST_i ¼ fSe_1;Se_2;...;Se_ng). When the participants have been created, the path (Rou) that each truck must follow during the transport of the goods is created by submitting a transaction that contains all the GPS points of the route. Then, the contract (C) that governs the trade of the goods and the conditions that each truck have to fulfill for the shipping. The shipment (Shi) is then created, where the type of goods to be transported, the quantity, the status of the order, etc. are stipulated. Finally, it is possible to query all those entities to see the data that has been stored in the world status or in the ledger.

After this phase, the system should be ready to receive real-time data from the sensors and to verify them to ensure that the terms of the contract are fulfilled.

5 Real-time data acquisition

This section presents the design decisions and optimizations proposed in our work to manage and store the data produced by the IoT devices in the blockchain network. Management and data storage is implemented after the blockchain network and the contract logic have been fully deployed, as shown in Fig. 2. The use case presented above (goods transport with trucks including three sensors) is used to evaluate the optimizations.

We have used the use of blockchain technology for information management because it allows the decentralization of information, preventing a single point of failure, ensuring the persistence, anonymity, and auditability of the data. Those are critical points of failure in the security and reliability of IoT devices, because they have limited hardware and therefore cannot implement security mechanisms that require a lot of resources, making the data unreliable. However, as IoT devices are capable of sending high volumes of data in a short time, the design of the network should be scalable and have a high throughput to be able to manage the data in real-time, preventing data loses or delays in storage. As our goal was to implement an efficient closed network, the Hyperledger Fabric platform was chosen because it provides better performance [31]. We used the version 1.4.2 of this tool and the Kafka protocol [82]. As for the hardware used to deploy the network, a general-purpose computer has been used for each organization in the network, which allowed us to see the performance when using general-purpose computers and not large data centers.

The blockchain network deployed in this work follows the architecture shown in Fig. 3. For the purpose of evaluation, we composed a blockchain scenario with three different organizations connected by a single channel. The transactions registered in one of the organizations are replicated and stored in the other organizations within the network. Each organization is composed of two peer nodes that are responsible for the transaction validation and storage in their corresponding ledger. Each organization have three clients, which in this use case will be trucks that have three sensors (Temperature, GPS, and Speed). The trucks, which must belong to the fleet of a specific organization, execute the transactions to send the sensors' measurements to the network.

Sensors may send data with a different frequency. In our use case, the temperature sensor sends a measurement every second, while the GPS sensor sends the truck's location every 10 s, and the speed sensor every 5 s, allowing to know in real-time the status of the foods and the truck, to make decisions quickly, as will be explained in the following lines.

To cope with the problem of sending IoT data to the peers, firstly we designed the network described in Sect. 5.1 following the traditional methodology, where all data of each shipment are stored consecutively in blocks of the ledger (concatenated records). However, as shown below, due to low scalability and data loss for transactions, we proposed two new optimizations

in order to solve the former problems, as well as to increase the blockchain network throughput. Those optimizations are described in Sect. 5.2.

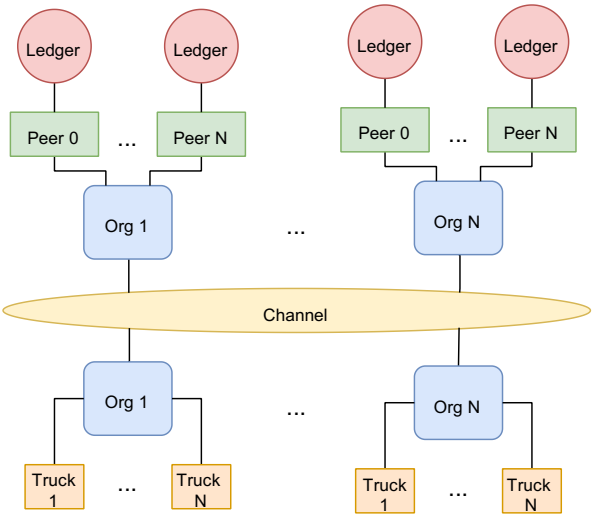


Fig. 3 Generic blockchain network design

5.1 Concatenated records

The first proposed network design consists of storing the data concatenated in a block of the chain, as in traditional implementations. In other words, when a sensor takes a new measurement, all the measurements previously taken by that sensor must be recovered, new data are added to the existing measurements for the shipment in a new block and, finally, the transaction is stored in the ledger. This methodology allows to simplify data recovery when a query is executed, because it is only necessary to get the last transaction, also called world state, realized in a specific shipment by the sensor whose measurements are consulted (e.g. temperature). This query can be carried out because the last transaction contains all the values read by the sensor, until that moment, for this shipment.

We made two experiments with this concatenated records solution to measure performance (see Sect. 5.1.1) and to verify the operation correctness (see Sect. 5.1.2) when this network is built for the use case defined. We focus on those two aspects as they are critical metrics of the blockchain network in this use case, as we must cope with a massive amount of data in real-time coming from the trucks equipped with IoT devices.

5.1.1 Performance analysis

We made a performance evaluation of the blockchain network using the concatenated records solution when it is subjected to an intense workload, but without concurrence. In other words, only one sensor of one shipment from those created in the blockchain network will send data. A temperature sensor per truck, performing a transaction every 0.5 s, is used to send a large amount of data to stress the system. To evaluate the performance provided by the blockchain network, we measured the transaction processing average time in five different workload scenarios: 1000, 5000, 10,000, 20,000 and 40,000 transactions respectively.

As shown in Fig. 4, the average execution time of a transaction increases as the total number of executed transactions raises. Therefore, it can be concluded that this implementation offers low scalability because transaction time increases quickly. The reason for this behaviour is that the block size increases for every new transaction, as new data are stored concatenated with the previous values for that sensor, which means a longer time to replicate the transaction in all the peer nodes of the network.

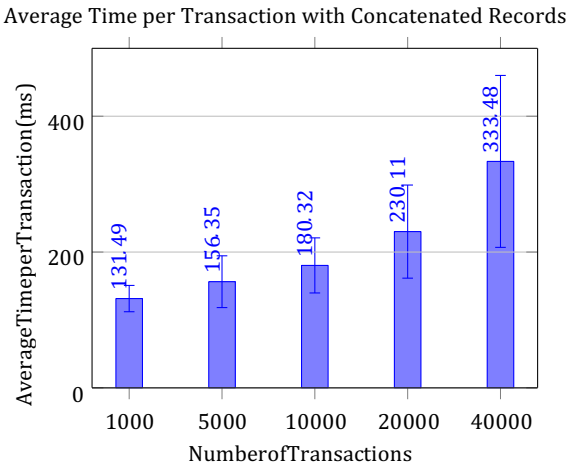


Fig. 4 Average time per transaction with concatenated records with different workloads

5.1.2 Concurrency analysis

In addition to measure performance, we studied the behavior of the blockchain network when several sensors from trucks send data at the same time, creating concurrent transactions. The purpose of this experiment is to determine the number of errors that occur when transactions are executed concurrently and, furthermore, to determine the percentage of data that has not been recorded in the ledger despite the transactions having taken place.

For this reason, the network scenario shown in the Fig. 3 has been used to perform this analysis, particularized with a total of nine trucks in the network with their three corresponding sensors, so that in this study there are twenty seven sensors carrying out transactions at the same time, which ensures the existence of concurrency. To study the number of errors that happen in execution and the volume of data that is not recorded, a test of 1-h duration was executed, with the sensors sending information periodically with the following rates: temperature sensors 1 s, GPS sensors 10 s, and speed sensors 5 s.

To alleviate the load of the validator, in Hyperledger Fabric the transactions can be validated individually or in blocks of transactions, by modifying the parameter MaxMessageCount. It means that, instead of validating immediately each transaction, the system waits until a certain number of transactions is available (e.g. 10) to make delayed validation. This feature allows the system to make optimizations, however the problem is that until a transaction is not validated, it will not be stored in the ledger. Thus, it is not available to other partners in the blockchain network, creating delays in information availability. For this evaluation, we tested different values for MaxMessageCount (1, 2, 3, 5, 15, 25 and 100) to see the effect of this feature.

Figure 5 shows the errors produced during the validation of transactions using concatenated records. This number is very high with the frequencies defined in our experimental study, especially for individual evaluation. It also shows that when the size of the block of transactions is increased, the number of errors that appear during the transaction execution decreases and the performance is increased. This behavior is due to the effect of validating multiple transactions simultaneously. The trade-off is that, as we increase MaxMessageCount, we also decrease the data availability for the peers. For example, if a GPS sensor is sent every 10 sand the parameter MaxMessageCount is set to 50, those position would be only available in the ledgers with an average delay of 250 s.

After studying how many errors are produced during execution, the percentage of data that have not been recorded in the tests described in the previous lines was analyzed. For this purpose, the average value of unrecorded data will be calculated for each type of sensor and each test. The results obtained in these tests are shown in Fig. 6, where the losses are worst for high-rate sensors, as majority of the unrecorded data comes from the temperature sensors where the failure rates are almost always greater than 50%. However, when analyzing the results obtained in the GPS and speed sensors, the failure rate is lower and, in some cases, there is no data loss.

This behavior is caused by the method of storing the data received from the sensors because the data is stored in a concatenated format and the previous measurements must be obtained first in order to add the last measurement. If the

frequency of data sent is high, the previous transactions may not have been validated yet and, therefore, they are not available for querying because they are not stored in the ledger, which implies that the last data are not available and the data that were read between the last transaction stored in the ledger and the last one carried out

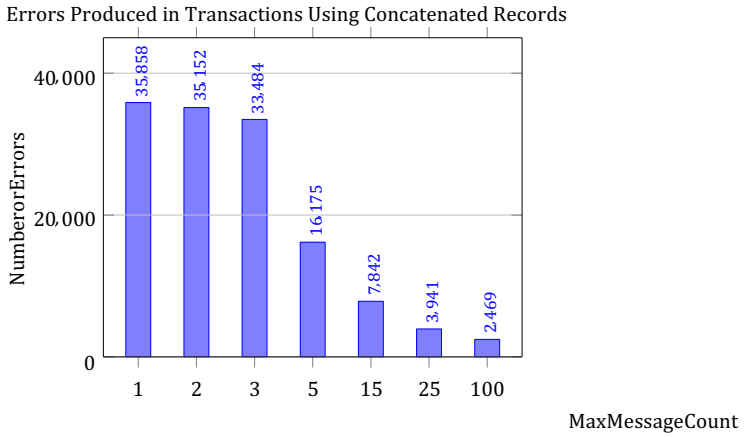
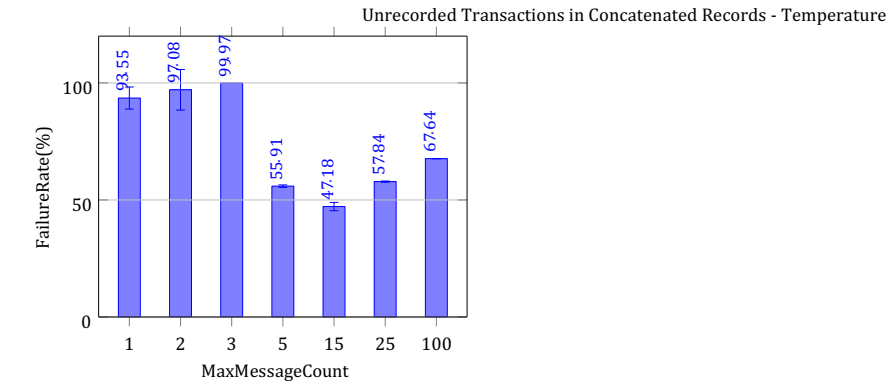
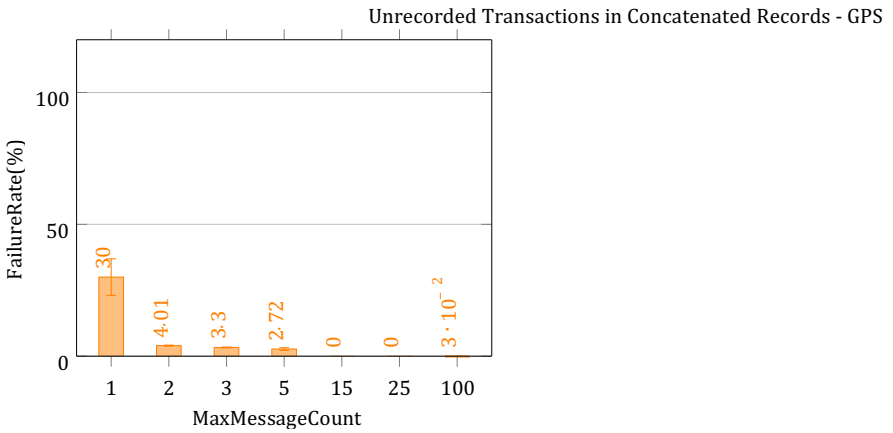


Fig. 5 Execution errors using concatenated records with different validation block sizes





Unrecorded Transactions in Concatenated Records - Speed

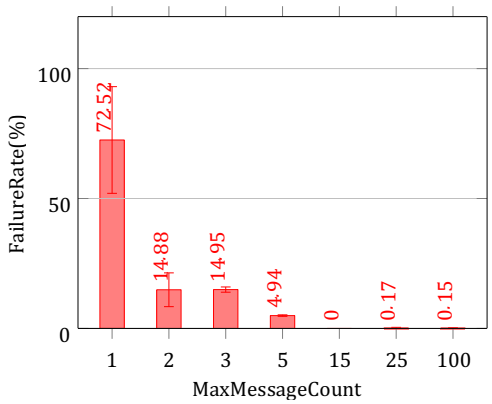


Fig. 6 Unrecorded temperature, GPS and speed sensor transactions with different validation block size

are lost. Moreover, as previously mentioned, the majority of the losses are produced in the temperature sensors, as these sensors have a higher sampling frequency (1 Hz) than other sensors, while the GPS sensors, with frequency of 0.1 Hz, show less transactions unrecorded. Those results show that the traditional blockchain platforms are unable to record high speed transactions due to protocol latency and data storage costs.

5.2 Proposed optimizations in transaction processing

As shown in the previous section, there are problems to be solved in order to use the blockchain technology in a network gathering real-time data from IoT devices. Those problems are related to performance, scalability, and possible data losses when transactions are processed. The former problems have been detected in previous works, but most solutions proposed to cope with them try to enhance the Internals of the blockchain platform used (validation, protocols, etc.).

For the sake of usability, and to increase adoption of the solution, we propose in this paper two generic optimizations in transaction processing that are compatible with any blockchain platform used to implement the blockchain network, as modifying the internals platform is not needed for the optimizations to take effect. These improvements are: using atomic records to store the data sent by the sensors and changing the validation block size.

5.2.1 Atomic records

This first optimization proposed in this work consists of storing each transaction carried out by a sensors in an atomic form. With this method, we create a new block for each transaction from a sensor. This block includes the minimum information needed for the transaction to be executed. For example, their content can be the sensor identifier, the value of the measurement taken, and the identity of the truck where the sensor is installed. For this purpose, when a sensor sends a new transaction, a new transaction is added for this sensor after the most recent one. This avoids the need of recovering all values previously stored by the sensor to concatenate the new value afterward, as it happened in the model described in the previous lines.

The effect of this optimization is critical, as the size of the transactions made by each sensor remains constant over time and the transaction replication time in all the peer nodes is also constant, as it only contains the last value and the minimum necessary information. Thus, scalability should be ensured. As the blockchain technology provides immutability and auditability, we provide the functionalities needed to get the history of values of a given sensor since it was created. Therefore, it is possible to execute queries to see the content of the last transaction, a set of transactions, or to recover all the transactions executed, which is also an advantage over the previous model that always returned the complete history of a sensor requiring further post-processing.

To evaluate the correctness and performance of this optimization, we run the same experiments devised to analyze the performance of the concatenated model (Sect. 5.1.1). The results shown in Fig. 7 prove that the proposed optimization provides very good performance and scalability of the blockchain network as the number of transaction increase, since the average time taken to execute a transaction is kept constant and lower than in the concatenated case. This effect is especially

clear in the load test that carries out 40,000 transactions, where the time is reduced to less than half. Furthermore, it should be noted that as the execution time remains constant regardless of the total number of transactions executed, the scalability of the network is enhanced.

In contrast with the traditional blockchain, this optimization does not originate data losses, even making immediate validation of each transaction, because the transactions are atomic and the previously recorded values do not have to be recovered as they are stored concatenated. However, although all the data is correctly recorded, we observed that, when the block size is too small and the data frequency high, the blockchain network is not able to validate the transactions that are generated in due time, generating delays in the data availability for queries, because a transaction is available for querying only when it has been validated.

5.2.2 Changing the transaction block size

To cope with the delays in the data availability, we proposed a second optimization technique in our model: modifying the maximum number of transactions that can be stored within a validation block. It also uses atomic transactions to ensure that they are independent of each other. This optimization allows increasing the performance of the blockchain network since the validation process (the phase that produces the highest latency) is not carried out every time a transaction is made, but by groups of transactions, which reduces the number of times the validation process is performed on the network.

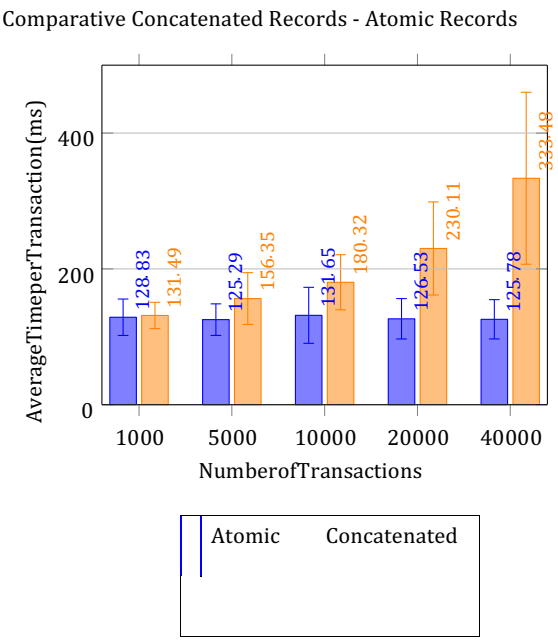


Fig. 7 Comparison between concatenated records and atomic records with different workloads

Figure 8 shows the delays generated in validations with the different block sizes in a scenario of 27 sensors sending data for an hour. The longest delay occurs when the transactions are validated individually, which is 295 min. However, when transaction block sizes are increased to validate a group of 15, 25, and 100 transactions, the processing delay is not generated because the system is able to manage all the transactions that are produced on time and the transactions are stored in the ledger in a short period of time since they were created.

This optimization allows the existence of concurrency in the system and increases the performance, because, by adjusting the block size, the system is able to validate a large number of transactions in a short time, being these stored in the ledger and available for querying in due time.

6 Periodic contract verification

The contract verification process consists of recovering the data stored in the blockchain network, that have been registered by the different participants or entities of business logic, and subsequently, determining if each of the records accomplish the contract terms, which have been established in the definition phase by the user administrator. In case of failure to comply with the constraints, the system proceeds to execute the penalty clauses established on the final value of the shipment (see Sect. 3.1.4). In all the supply chain applications for transport reviewed, contract verification is made once the products are delivered o whether there is a contract breach in the delivery date. This may be due to business logic, but in most cases it is due to the fact that data are not reported in real-time to the actors and then they cannot verify the contract. In our case, our solution is able to collect data in real-time and submit the transactions to the blockchain network so that data are registered immediately or with a small delay.

Having data registered in real-time, we propose to run a continuous contract verification process, with a periodicity that can be defined by the business logic. Figure 9 shows the verification processes that are executed to determine whether a contract is valid or has violated the restrictions. In the first instance, the contract (C) for a shipment (Shi) is started when the truck (Tru) with the shipment allocated starts the route. From that moment, data from its sensors (Se) are collected in real-time and recorded on the blockchain network. Simultaneously, when the contract is started, a periodic verification process is also started for the contract. This verification process retrieves data from the network for analysis and every time (T), being T a period defined for every shipment, determines if the restrictions Res 2 C are fulfilled. If the restrictions are satisfied, a successful verification process is recorded for this period. Since the duration of a shipment will be, in general, longer than the verification period, the process will be repeated for the series of subsequent periods in the shipment. This procedure is executed repeatedly in a closed cycle until the status of the whole shipment is delivered. In other words, the truck has reached its destination and the product has been delivered to the final consumer. If all the records of the verification process are correct, then no alert is issued, and the resulting record of the verification will be a valid contract. In this case, the contract is deemed to have been satisfactorily terminated.

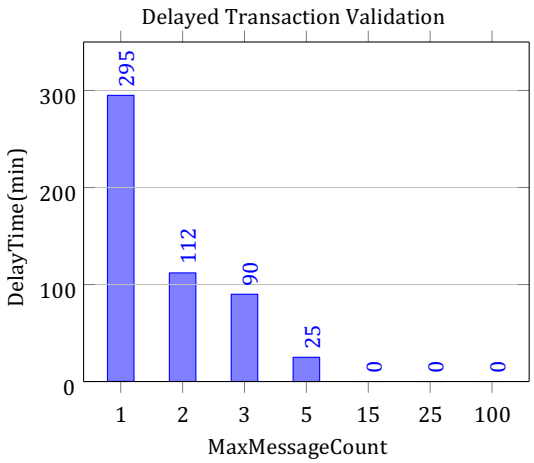


Fig. 8 Delayed transaction validation with different validation block size

To achieve the closed cycle between the data capture and the periodic verification procedure, the administrator user in the definition phase establishes the periodicity parameter T, which is the period in which the contract verification process is executed. Any record in the blockchain written during this time interval and related with the contract will be verified. The result of each periodic contract verification is also recorded in the blockchain, as a new record that includes the results of the verification performed. This approach has two main advantages: near real-time status of each shipping and scalability in contract verification.

ig. 9 Continuous contract verification flowchart

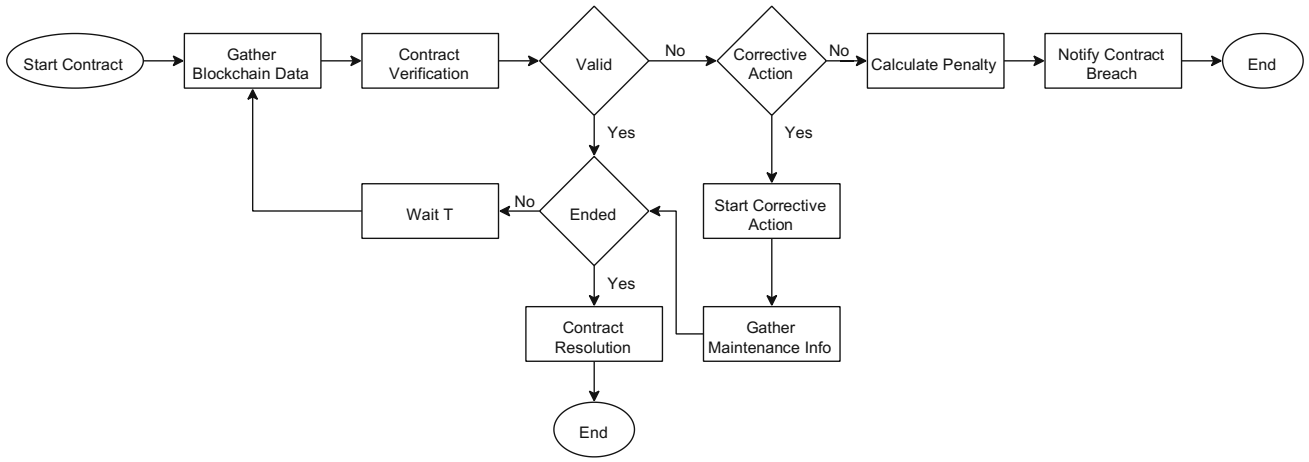


Fig. 9 Continuous contract verification flowchart

As contracts are verified every T for a shipping, a periodic verification process might determine in near realtime that the contract verification is not valid in the interval analyzed ½t i;ti þ T due to errors in data capture or conditions breaching the contract, In this case, the error condition is also recorded in the blockchain to ensure nonrepudiation and the participants are notified of the incidents found to immediately take corrective measures (if possible). After that, the system evaluates if the contract has ended. In case not, it continues the periodic contract verification until the delivery of the shipment. Otherwise, if it is not feasible to apply the corrective or maintenance measures, the penalty clauses established in the contract are executed. Following this procedure, and depending on the violations committed, the final value of the contract is established, and the breaches of the contract are notified to the interested parties. The penalty clauses are part of the proposed system model and were described in Sect. 3.1.4. The process of verification and application of sanctions is summarized in Algorithm 1: Every result of a verification process is saved as a new record in the blockchain, in this sense, a result variable is declared (step 1) that will store all the transactions that do not comply with the established restrictions. By default, the proposed system executes the verification process automatically every verification period (T) in seconds (step 1 pseudocode). Then, the current system time cts is retrieved (step 2) and the time stamp of the next verification interval is calculated as pts ¼ timestamp of T seconds before that cts (step 3). The variables pts and cts denote the initial time and end time respectively of the time interval to be analysed. Then all blockchain records that were created for the shipment of the contract within the previously calculated time interval are analysed one by one (step 4). For each record, its contract is identified (step 5) and the sensor identifier is extracted (step 6) which collected the data sent in the transaction. Knowing the sensors involved in the execution of the current C_i contract, the restrictions defined by Res 2 C that must be met are consulted (step 7). If the current transaction (Tx_{sei}) does not comply with the restrictions of the contract (step 8), the penalty clauses are consulted to apply them to the final value of the contract price (steps 9 and 10) and the result variable records the current transaction and an invalid contract status as a warning for subsequent verification.

The proposed procedure increase scalability as the verification is made for a small subset of transactions (depending on T) and the result is stored. Thus, on the one hand if every step is ok, the final verification process is automatically solved. On the other hand, if there was a contract breach in a cycle, the verification processes do not need to analyze again all the data, are the verification results of every steps are also logged in the ledgers, providing non-repudiation. Even if a third actor would like to verify the whole shipping, she only had to verify the results of the periodic verification processes.

The verification process of the case study proposed in this paper is described below.

6.1 Case study: fleet of trucks transporting food

The case study developed consists of the business logic of food transport. In this business logic different organizations participate, each organization has its fleet of trucks, each truck makes different shipments of assets and incorporates three types of sensors: temperature, GPS, and speed to control the conditions of each shipment.

The verification process consists of determining the status of the shipments made by each truck by verifying whether the records of temperature, speed, and GPS sensors of each truck meet the conditions established in the contract. In particular,

for the temperature recording is verified that it is within the minimum and maximum values established in the contract, for the speed sensor that does not exceed the maximum limit allowed and for each GPS recording is verified that this point is within a minimum distance to any of the GPS points that make up the route that the truck must follow.

The verification process is done incrementally, from checking a shipment to checking the status of a fleet of trucks. As an example, it is possible to check the general state of a truck, that is to say, to check the state of all the shipments that a particular truck has made. For this purpose, the verification history is reused, since there is a record with the verification results of each shipment.

Figure 10 shows an output example of contracts tracking for a fleet of trucks. The list indicates the shipments made by each truck and the contract status of that shipment (Ok or Contract Violation). In the case of contract violation, the number of penalties that have been recorded during the complete verification of the shipment is initially shown.

Each shipment can then be examined in more detail if needed just by clicking on the Penalties icon. The proposed system allows the authorized actors to view the details of the shipment and the contract, the route that the truck should follow, and the details of the result of verification (Fig. 11). The verification process generates as a result all the data captured by the sensors that do not meet the conditions of the contract. As an example, Fig. 11 shows that, in a failed shipment, there was a violation of the contract for temperature data and a GPS location outside the established route.

Thus, our web tool allows the participants of the blockchain network, the government entities, or auditors in charge of supervising the correct execution of the contracts in business logic, to easily consult the data of the blockchain network and the verification results of any business logic, even if our use case was developed for fleets of trucks transporting food.

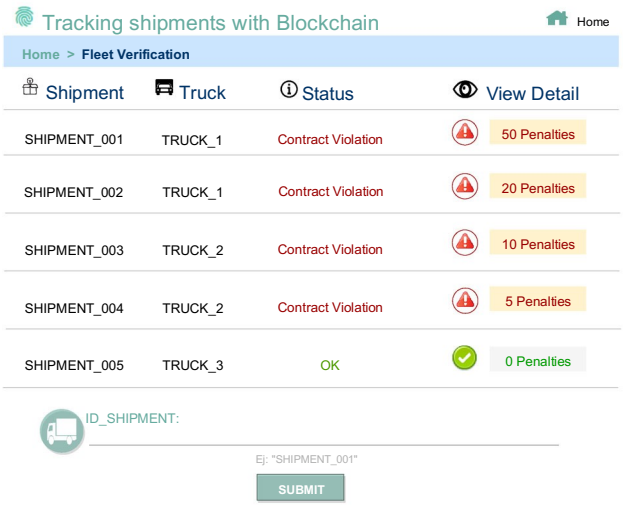


Fig. 10 Search for validated shipments

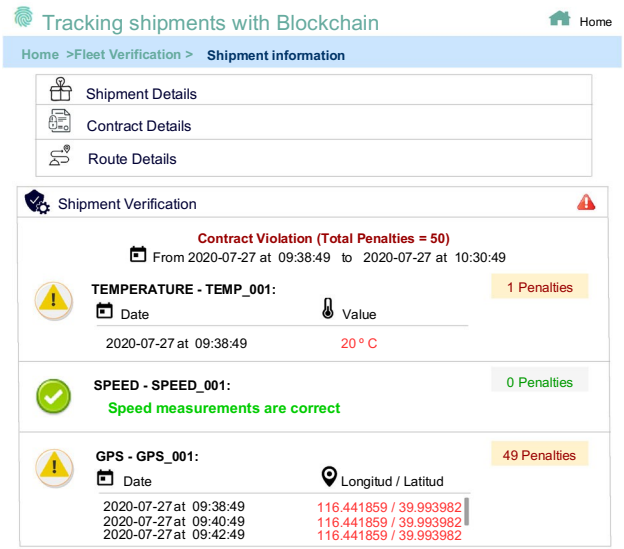


Fig. 11 Detail of a validated shipment

7 Conclusions and future work

In this paper we presented a model for the continuous verification of contracts in the supply chain environment. The solution proposed uses the blockchain technology along with the Internet of Things paradigm to provide a secure, verifiable, and traceable solution for real-time sensor data that can be integrated into supply chains to support logistics.

The main technical contributions of this paper were the following:

1. Mathematical model. We have developed a mathematical model to represent the components and entities of the definition, deployment, data acquisition, and contract verification phases, which are present in supply chains using blockchain and real-time data.
2. Automatic and distributed deployment. The proposed solution, through the integration of scripts and systems such as Docker and Hyperledger Fabric, allows the deployment and implementation of the proposed model in an automated way on local or distributed infrastructures.
3. Blockchain platform independent optimization techniques. We proposed the atomic transactions and grouped validation techniques, which are applied on top of the blockchain platform and that are not dependent on any specific platform. As a result, improvements were obtained in the data collection transactions protocol and the data storage procedure.
4. Improved blockchain transactions validation performance. The optimizations carried out on the transactions allowed to increase the performance and scalability of transactions validation in the blockchain platform, allowing high speed validation and storing the data consistently in near real-time, allowing the data to be available for query in the continuous validation of the contract.
5. Continuous verification of contracts in the supply chain environment. Periodic verification of contract proved to be useful to have near real-time notification of contract breaches and to increase scalability in large supply chain networks.

We run experiments to test the performance and reliability of the solution proposed. The evaluation results show that the optimizations proposed allows to process data request with higher registration rate that the solution provided in by default in blockchain platforms, like Hyperledger. We could also avoid missing data and reduce the transaction processing delays, thus increasing the reliability of the supply chain processes.

Our proposal also has benefits for business logic actors, such as:

– Usability of the solution. The proposed system incorporates a web interface that integrates the operational model described and facilitates its adoption by companies or organizations not specialized in the field of information technology. It guides users without previous knowledge about blockchain and supply chains to deploy their solution, to setup the periodic contract verification, and to perform verification queries.

– Faster conflicts resolution: In case of conflict between two or more participants of the network for the breach of a contract, the web service of the proposed system allows to identify in real-time the problems recorded during the execution of the contracts of each shipment made, in this way, the system can establish an irrefutable culprit of the problem caused taking into account that the records are immutable.
– Reliable verification with third parties: The definition of client nodes employing containers that encapsulate the software necessary to interconnect to the blockchain network allows external entities (auditors, government, or police) to the business logic to perform a reliable and traceable verification of all the registered activities of each organization in the blockchain network.

During the development of this research, we identified different enhancements that could be included in the proposed system as future work. The first one is the study of distributed model costs. As the elements of the solution proposed are encapsulated in virtual containers to achieve a deployment of the system in different infrastructures (edge, fog, cloud), we will study the cost (performance, latency, among others) of distribution of the proposed solution considering scenarios where the nodes are fully distributed geographically. The second one would be to store the ledger in a NoSQL database to increase the blockchain performance, as database engines are designed to be able to manage and store a large number of transactions per second. Finally, we plan to use the data collected from IoT devices to improve maintenance of the supply chain elements, as the continuous verification of the blockchain's records could be used to generate alerts (corrective maintenance), statistics or suggestions (predictive maintenance) to establish measures for the components that are causing losses (for example, sensors or trucks in poor condition).

Acknowledgements
This work has been partially supported by the project ''CABAHLA-CM: Convergencia Big data-Hpc: de los sensores a las Aplicaciones'' S2018/TCS-4423 from Madrid Regional Government and by the Spanish Ministry of Science and Innovation Project ''New Data Intensive Computing Methods for High-End and Edge Computing Platforms (DECIDE)''. Ref. PID2019-107858GB-I00.

References

1. Haddud, A., DeSouza, A., Khare, A., Lee, H.: Examiningpotential benefits and challenges associated with the internet of things integration in supply chains. J. Manuf. Technol. Manage. (2017)
2. Mohanta, B.K., Panda, S.S., Jena, D.: An overview of smartcontract and use cases in blockchain technology. In: Proceedings of the 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–4. IEEE (2018)
3. Roy, D., Bhadra, D., Das, B.: Is blockchain the future of supplychain management?-a review paper. In: Proceedings of International Ethical Hacking Conference 2019, pp. 83–103. Springer Singapore (2019)
4. Gonczol, P., Katsikouli, P., Herskind, L., Dragoni, N.: Blockchain implementations and use cases for supply chains—a survey. Ieee Access 8, 11856–11871 (2020)
5. Pal, A., Kant, K.: Using blockchain for provenance and traceability in internet of things-integrated food logistics. Computer 52(12), 94–98 (2019)
6. Alazab, M., Alhyari, S., Awajan, A., Abdallah, A.B.: Blockchaintechnology in supply chain management: an empirical study of the factors affecting user adoption/acceptance. Clust. Comput. (2020). https://doi.org/10.1007/s10586-020-03200-4
7. Ketzenberg, M.: The value of information in a capacitated closedloop supply chain. Eur. J. Oper. Res. 198(2), 491–503 (2009)
8. Pla, L., Nadal, E.: Optimal transport planning for the supply to afruit logistic center. Springer, New York (2015)
9. Soto-Silva, W.E., Nadal-Roig, E., Gonza´lez-Araya, M.C., PlaAragones, L.M.: Operational research models applied to the fresh fruit supply chain. Eur. J. Oper. Res. 251(2), 345–355 (2016)
10. Kamath, R.: Food traceability on blockchain: Walmarts pork andmango pilots with ibm. J. Br. Blockchain Assoc. 1(1), 3712 (2018)
11. Havelaar, A.H., Kirk, M.D., Torgerson, P.R., Gibb, H.J., Hald, T.,Lake, R.J., Praet, N., Bellinger, D.C., De Silva, N.R., Gargouri, N., et al.: World health organization global estimates and regional comparisons of the burden of foodborne disease in 2010. PLoS Med. 12(12), e1001923 (2015)
12. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview ofblockchain technology: architecture, consensus, and future trends. In: Proceedings of the 2017 IEEE international congress on big data (BigData congress), pp. 557–564. IEEE (2017)
13. Pavithran, D., Shaalan, K., Al-Karaki, J.N., Gawanmeh, A.: Towards building a blockchain framework for iot. Clust. Comput. 23, 2067–2087 (2020). https://doi.org/10.1007/s10586-02003059-5
14. Helo, P., Shamsuzzoha, A.: Real-time supply chaina blockchainarchitecture for project deliveries. Robot. Comput. Integr. Manuf. 63, 101909 (2020)
15. Zheng, P., Zheng, Z., Luo, X., Chen, X., Liu, X.: A detailed andreal-time performance monitoring framework for blockchain systems. In: Proceedings of the 2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), pp. 134–143. IEEE (2018)

16. Dorri, A., Kanhere, S.S., Jurdak, R.: Towards an optimizedblockchain for iot. In: Proceedings of the 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), pp. 173–178. IEEE (2017)
17. Thakkar, P., Nathan, S., Viswanathan, B.: Performance benchmarking and optimizing hyperledger fabric blockchain platform. In: Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 264–276. IEEE (2018)
18. Javaid, H., Hu, C., Brebner, G.: Optimizing validation phase ofhyperledger fabric. In: Proceedings of the 2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 269–275. IEEE (2019)
19. Leung, J., Lee, J.: 300cubits: Blockchain for shipping. (2017). https://www.300cubits.tech/
20. Gadnis, A., Keiser, J.A., Linton, M., Natalenko, S.: Blockchainbased identity and transaction platform (2018). https://banqu.co/. US Patent App. 15/767,969
21. Jones, D., Kingston, D., Willette, A.Q.: Bext360. (2017). https:// www.bext360.com/
22. Konovalenko, I., Ludwig, A.: Event processing in supply chainmanagement—the status quo and research outlook. Comput. Ind. 105, 229–249 (2019)
23. Ferna´ndez-Carame´s, T.M., Froiz-Mı´guez, I., Blanco-Novoa, O., Fraga-Lamas, P.: Enabling the internet of mobile crowdsourcing health things: a mobile fog computing, blockchain and iot based continuous glucose monitoring system for diabetes mellitus research and care. Sensors 19(15), 3319 (2019)
24. Biswas, K., Muthukkumarasamy, V.: Securing smart cities usingblockchain technology. In: Proceedings of the 2016 IEEE 18th international conference on high performance computing and communications; IEEE 14th international conference on smart city; IEEE 2nd international conference on data science and systems (HPCC/SmartCity/DSS), pp. 1392–1393. IEEE (2016)
25. Tripathi, G., Ahad, M.A., Sathiyanarayanan, M.: The role ofblockchain in internet of vehicles (iov): Issues, challenges and opportunities. In: Proceedings of the 2019 International Conference on contemporary Computing and Informatics (IC3I), pp. 26–31. IEEE (2019)
26. Pournader, M., Shi, Y., Seuring, S., Koh, S.L.: Blockchainapplications in supply chains, transport and logistics: a systematic review of the literature. Int. J. Prod. Res. 58(7), 2063–2081 (2020)
27. Baliga, A., Solanki, N., Verekar, S., Pednekar, A., Kamat, P.,Chatterjee, S.: Performance characterization of hyperledger fabric. In: 2018 Crypto Valley conference on blockchain technology (CVCBT), pp. 65–74. IEEE (2018)
28. Nasir, Q., Qasse, I.A., Abu Talib, M., Nassif, A.B.: Performanceanalysis of hyperledger fabric platforms. Security and Communication Networks (2018)
29. Ethreum: Ethereum. https://ethereum.org/
30. Hyperledger-Fabric: Hyperledger fabric. https://www.hyperl edger.org/projects/fabric
31. Pongnumkul, S., Siripanpornchana, C., Thajchayapong, S.: Performance analysis of private blockchain platforms in varying workloads. In: Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN), pp. 1–6. IEEE (2017)
32. Rouhani, S., Deters, R.: Performance analysis of ethereumtransactions in private blockchain. In: Proceedings of the 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 70–74. IEEE (2017)
33. Li, K., Li, H., Hou, H., Li, K., Chen, Y.: Proof of vote: A highperformance consensus protocol based on vote mechanism & consortium blockchain. In: Proceedings of the 2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 466–473. IEEE (2017)
34. Wang, C., Chu, X.: Performance characterization and bottleneckanalysis of hyperledger fabric. arXiv preprint arXiv:2008.05946 (2020)
35. Gao, Z., Yang, L.: Optimization scheme of consensus mechanismbased on practical byzantine fault tolerance algorithm. In: Proceedings of the CCF China Blockchain Conference, pp. 187–195. Springer (2019)
36. Stathakopoulou, C., David, T., Vukolic´, M.: Mir-bft: Highthroughput bft for blockchains. arXiv preprint arXiv:1906.05552 (2019)
37. Gorenflo, C., Lee, S., Golab, L., Keshav, S.: Fastfabric: Scalinghyperledger fabric to 20,000 transactions per second. In: Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 455–463. IEEE (2019)
38. Nakaike, T., Zhang, Q., Ueda, Y., Inagaki, T., Ohara, M.: Hyperledger fabric performance characterization and optimization using goleveldb benchmark. In: Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 1–9. IEEE (2020)
39. Ali, S., Wang, G., White, B., Cottrell, R.L.: A blockchain-baseddecentralized data storage and access framework for pinger. In: Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/ 12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), pp. 1303–1308. IEEE (2018)
40. Manevich, Y., Barger, A., Tock, Y.: Endorsement in hyperledgerfabric via service discovery. IBM J. Res. Dev. 63(2/3), 2–1 (2019)
41. Lin, I.C., Liao, T.C.: A survey of blockchain security issues andchallenges. IJ Netw. Secur. 19(5), 653–659 (2017)
42. Vujicˇic´, D., Jagodic´, D., Randjic´, S.: Blockchain technology, bitcoin, and ethereum: a brief overview. In: Proceedings of the 2018 17th international symposium infoteh-jahorina (infoteh), pp. 1–6. IEEE (2018)
43. Verhoeven, P., Sinn, F., Herden, T.T.: Examples from blockchainimplementations in logistics and supply chain management: exploring the mindful use of a new technology. Logistics 2(3), 20 (2018)
44. Heilman, E., Baldimtsi, F., Goldberg, S.: Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions. In: Proceedings of the International conference on financial cryptography and data security, pp. 43–60. Springer (2016)
45. Eyal, I., Gencer, A.E., Sirer, E.G., Van Renesse, R.: Bitcoin-ng: A scalable blockchain protocol. In: Proceedings of the 13th fUSENIXg symposium on networked systems design and implementation (fNSDIg 16), pp. 45–59 (2016)
46. Guegan, D.: Public blockchain versus private blockhain. Tech.rep, Centre dEconomie de la Sorbonne (2017)
47. Nathan, J., Jacobs, B.: Blockchain consortium networks: addingsecurity and trust in financial services. J. Corp. Account. Finance 31(2), 29–33 (2020)

48. Kang, J., Yu, R., Huang, X., Wu, M., Maharjan, S., Xie, S.,Zhang, Y.: Blockchain for secure and efficient data sharing in vehicular edge computing and networks. IEEE Internet Things J. 6(3), 4660–4670 (2018)
49. Valenta, M., Sandner, P.: Comparison of ethereum, hyperledgerfabric and corda. no. June pp. 1–8 (2017)
50. Sajana, P., Sindhu, M., Sethumadhavan, M.: On blockchainapplications: hyperledger fabric and ethereum. Int. J. Pure Appl. Math. 118(18), 2965–2970 (2018)
51. Saberi, S., Kouhizadeh, M., Sarkis, J., Shen, L.: Blockchaintechnology and its relationships to sustainable supply chain management. Int. J. Prod. Res. 57(7), 2117–2135 (2019)
52. Li, H., Pei, L., Liao, D., Wang, X., Xu, D., Sun, J.: Bddt: useblockchain to facilitate iot data transactions. Clust. Comput. 23, 1–21 (2020). https://doi.org/10.1007/s10586-020-03119-w
53. Khalid, U., Asim, M., Baker, T., Hung, P.C., Tariq, M.A., Rafferty, L.: A decentralized lightweight blockchain-based authentication mechanism for iot systems. Clust. Comput. 23, 2067–2087 (2020). https://doi.org/10.1007/s10586-020-03058-6
54. Alfandi, O., Khanji, S., Ahmad, L., Khattak, A.: A survey onboosting iot security and privacy through blockchain. Clust. Comput. 23, 1–19 (2020). https://doi.org/10.1007/s10586-02003137-8
55. Puri, V., Priyadarshini, I., Kumar, R., Van Le, C.: Smart contractbased policies for the internet of things. Clust. Comput. 24, 1–20 (2021). https://doi.org/10.1007/s10586-020-03216-w
56. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system.Tech. rep, Manubot (2019)
57. Bo¨hme, R., Christin, N., Edelman, B., Moore, T.: Bitcoin: economics, technology, and governance. J. Econ. Perspect. 29(2), 213–38 (2015)
58. Davidson, S., De Filippi, P., Potts, J.: Economics of blockchain.Available at SSRN 2744751 (2016)
59. Zhang, Y.: Developing cross-border blockchain financial transactions under the belt and road initiative. Chin. J. Comp. Law (2020)
60. Tam, B.: An investigation of how the adoption of blockchain inthe one belt, one road initiative will impact China's economy (2019)
61. Kang, J., Yu, R., Huang, X., Maharjan, S., Zhang, Y., Hossain,E.: Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. IEEE Trans. Ind. Inf. 13(6), 3154–3164 (2017)
62. Aitzhan, N.Z., Svetinovic, D.: Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Trans. Depend. Secure Comput. 15(5), 840–852 (2016)
63. Niranjanamurthy, M., Nithya, B., Jagannatha, S.: Analysis ofblockchain technology: pros, cons and swot. Clust. Comput. 22(6), 14743–14757 (2019)
64. Arumugam, S.S., Umashankar, V., Narendra, N.C., Badrinath, R.,Mujumdar, A.P., Holler, J., Hernandez, A.: Iot enabled smart logistics using smart contracts. In: Proceedings of the 2018 8th International Conference on Logistics, Informatics and Service Sciences (LISS), pp. 1–6. IEEE (2018)
65. Tian, F.: An agri-food supply chain traceability system for chinabased on rfid & blockchain technology. In: Proceedings of the Proceedings of the 2016 13th international conference on service systems and service management (ICSSSM), pp. 1–6. IEEE (2016)
66. Latif, R.M.A., Farhan, M., Rizwan, O., Hussain, M., Jabbar, S.,Khalid, S.: Retail level blockchain transformation for product supply chain using truffle development platform. Clust. Comput. 1–16 (2020)
67. Chen, C.L., Lin, D.P., Chen, H.C., Deng, Y.Y., Lee, C.F.: Designof a logistics system with privacy and lightweight verification. Energies 12(16), 3061 (2019)
68. Liang, K., Susilo, W.: Searchable attribute-based mechanismwith efficient data sharing for secure cloud storage. IEEE Trans. Inf. Forensics Secur. 10(9), 1981–1992 (2015)
69. Caballero, R., Rivera, B.: Blockchain: An alternative to enabletraceability in the agricultural supply chain in panama. In: Proceedings of the 2019 7th International Engineering, Sciences and Technology Conference (IESTEC), pp. 46–51. IEEE (2019). https://doi.org/10.1109/IESTEC46403.2019.00017
70. Saurabh, S., Dey, K.: Blockchain technology adoption, architecture, and sustainable agri-food supply chains. J. Clean. Prod. 284, 124731 (2021). https://doi.org/10.1016/j.jclepro.2020. 124731. http://www.sciencedirect.com/science/article/pii/ S0959652620347752
71. Wingreen, S., Sharma, R., et al.: A blockchain traceabilityinformation system for trust improvement in agricultural supply chain. In: Proceedings of the European conference on information systems (ECIS2019) (2019). https://aisel.aisnet.org/ecis2019_rip/ 10/
72. Bocek, T., Rodrigues, B.B., Strasser, T., Stiller, B.: Blockchainseverywhere-a use-case of blockchains in the pharma supplychain. In: Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 772–777. IEEE (2017)
73. Thakker, U., Patel, R., Tanwar, S., Kumar, N., Song, H.: Blockchain for diamond industry: opportunities and challenges. IEEE Internet Things J. (2020)
74. Ben-Daya, M., Hassini, E., Bahroun, Z.: Internet of things andsupply chain management: a literature review. Int. J. Prod. Res. 57(15–16), 4719–4742 (2019)
75. de Vass, T., Shee, H., Miah, S.J.: Iot in supply chain management: a narrative on retail sector sustainability. Int. J. Logist. Res. Appli. 1–20 (2020)
76. Joy, A.M.: Performance comparison between linux containersand virtual machines. In: Proceedings of the 2015 International Conference on Advances in Computer Engineering and Applications, pp. 342–346. IEEE (2015)
77. Zhang, Q., Liu, L., Pu, C., Dou, Q., Wu, L., Zhou, W.: A comparative study of containers and virtual machines in big data environment. In: Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), pp. 178–185. IEEE (2018)
78. Treat, D., Giordano, G., L. Schiatti, L., Borne-Pons, H.: Connecting ecosystems: blockchain integration. (2018). https://www.accenture.com/us-en/insights/blockchain/integration-ecosystems
79. Docker: Docker. https://www.docker.com/

80. Zheng, C., Tovar, B., Thain, D.: Deploying high throughputscientific workflows on container schedulers with makeflow and mesos. In: Proceedings of the 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), pp. 130–139. IEEE (2017)
81. Docker-Swarm: Docker swarm. https://docs.docker.com/engine/ swarm/
82. kafka: Kafka. https://kafka.apache.org/