Future Generation Computer Systems 158 (2024) 73–88

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A Blockchain-based Digital Twin for IoT deployments in logistics and transportation

Salvador Cuñat Negueroles *, Raúl Reinosa Simón, Matilde Julián, Andreu Belsa, Ignacio Lacalle, Raúl S-Julián, Carlos E. Palau

Communications Department, Universitat Politècnica de València, 46022, Valencia, Spain

ARTICLE INFO

Keywords:
Digital Twin
FIWARE
Blockchain
Ethereum
Canis Major
Smart logistics

ABSTRACT

Digital Twins are software technologies that enable the modelling of real-world phenomena in digitised environments, representing and monitoring the reality of various processes, including IoT deployments. Since 2017, the use of Digital Twins has been increasing. However, in the road transport and logistics realm, the adoption rate remains low, primarily due to the costs of processing and validating data in centralised scenarios, among other factors. On the other hand, Blockchain technologies were created to provide immutable and decentralised data storage in diverse scenarios, adding a layer of isolation and reliability in heterogeneous solutions. This paper presents a case study proposing a Digital Twin based on open-source Blockchain technologies, such as FIWARE Canis Major. The primary goal is to design and implement a robust and efficient open-source architecture that allows for the control and optimisation of vehicle fleet allocations in logistics/transport companies within supply chain management. This case study aims to showcase the practical application of the proposed solution in a real-world context, providing insights into its eco-friendly and low-cost attributes while opening the door to a large number of additional applications.

1. Introduction

Nowadays, the digital transformation of the transport and logistics sector demands a multifaceted approach, taking into account the myriad factors of the surrounding environment. The ability of obtain, process and transmit data from the physical world is a key factor in the quest to optimise available resources, which directly implies an economic and ecological impact. This process of bridging the physical world with the virtual is generally known as 'Digital Twin' in this context.

The concept of Digital Twin (hereinafter DT) was introduced by Michael Grieves in 2002 [1]. The National Aeronautical Space Administration (NASA) started to also work on DTs concept around 2012 [2]. Over time, and depending on the area of application, the definition of DT has been adapting and evolving. To authors, a DT is, essentially, a virtual representation of a real-world system, object or process which is generally used to monitor and make simulations and/or prediction of those elements' behaviour upon their environment (in real-time). DT technology promises to be an important part of the information systems for Industrial companies in the next decade. The digitisation of data allows companies to be leaders in their sector, reducing costs and increasing the efficiency of their business. In 2019 DT market was quantified at nearly $4 billion and is believed to reach $35 billion by 2025 [3], growing more than 38% every year [4]. Of this amount, approximately 20% will be used in the digitisation of the supply chain [5]. This large investment is due to the compelling need to collect and analyse the data of the companies, develop and implement security systems, the use of artificial intelligence to make simulations, the creation of an environment with IoT devices, etc., to keep competitive.

One of the concerns in DT environment implementations is the need to be able to guarantee the traceability and veracity of data for decision making [6]. In this case, DT technologies lack mechanisms with the ability to verify the data received during the processing and visualisation process. Recently, the development of Blockchain technology has opened new possibilities for DTs. Blockchain provides append-only and immutable storage for transactions that does not depend on a single point of control and is accessible for all the participants of the network. Thus, the use of Blockchain provides a single source of truth and ensures data integrity, transparency, non-repudiation and auditability. DTs can benefit from this technology in different ways [7]. This paper aims to fill this gap by exploring how Blockchain can serve as a verifiable, decentralised, and immutable historical database for DT information, preserving critical information and providing traceability and data provenance. Additionally, it investigates how Blockchain can create a reliable data-sharing environment for DTs, highlighting that

* Corresponding author.
E-mail address: salcuane@upv.es (S. Cuñat Negueroles).

https://doi.org/10.1016/j.future.2024.04.011
Received 1 August 2023; Received in revised form 5 April 2024; Accepted 7 April 2024
Available online 17 April 2024

the articles presented in the related works, to be discussed in Section 2, tend to remain within the realm of theoretical exploration and have not explored in depth an assumable solution to a case study.

In the logistics domain, for example, traceability and immutability are essential for operational efficiency and transparency. The integration of Blockchain with DTs in logistics can be revolutionary, offering innovative solutions for planning, cost reduction, and trust enhancement in the supply chain. This paper delves into how Blockchain-enabled DTs can transform logistics by providing detailed tracking at each logistics stage. Every transaction, movement of goods, or status change is immutably recorded on the Blockchain, enabling real-time visibility for all stakeholders. This comprehensive monitoring from origin to destination addresses the current limitations in logistics operations by pinpointing delays and issues promptly. Additionally, the immutable record-keeping characteristic of Blockchain aids in fraud prevention, marking a significant advancement over existing systems. Through this research, we aim to fill the current gap in DT applications by demonstrating the effective application of Blockchain technology in ensuring data veracity and traceability, particularly in logistics. This revision aims to clearly state the gap in the current state-of-the-art and how this paper addresses it, using the context of Blockchain technology in Digital Twins, particularly in logistics.

The rest of the paper is organised as follows: In Section 2, related work is discussed. Section 3 introduces the case under investigation, presents the case study, focusing on the integration of Blockchain technologies in a smart logistics environment. The findings of the investigation are presented in Section 4, followed by a detailed discussion in Section 5. The paper concludes in Section 6, summarising key insights and outlining directions for future research.

2. Related work

In this section, a literature review has been done to analyse the state-of-the-art of the use of DTs. First, it is worth mentioning the databases used to obtain the information. Mainly, those have been ResearchGate and ScienceDirect, which already give a clear vision of the current state of technology. The established search criteria were the following keywords: “digital twin”, “road transport”, “transport industry”, “logistics”, “Internet of Things”, “state-of-the-art” and “literature review”. Finally, those papers published in English from 2010 to 2022, related to applications of DTs in general and, in specific, the road logistic transport, were analysed.

To learn about the evolution of interest in and implantation of DTs, another more generalised search has been carried out on ScienceDirect covering the period between 2010 and 2022. This search includes review articles, research articles and book chapters, and its objective is to know the number of publications related to DTs over time. Fig. 1 shows the results obtained. As can be seen, the number of publications began to increase significantly since 2017, when DTs started to become a trending technology. Noticeably, in 2021 the number of publications increased by approximately 33% compared to the previous year. This is because DTs along with IoT devices are a key technology in the digitisation of data in the Industry 4.0. IoT devices are an important part of DTs architecture, and by 2025 more than 75 billion of them are expected to be connected to the Internet [8].

After analysing minutely all the selected papers, it is concluded that DTs are used in a wide range of areas [9]. The most prominent application domains identified in the literature review are the following:

- Smart cities: cities have more population and a higher consumption of resources every day. A correct use of those resources is necessary in order to guarantee a sustainable future. Along with IoT devices spread through the streets of the cites, DTs can help manage these resources, simulating possible future scenarios and allowing action to be taken more quickly. They can also help manage the maintenance of public places and architectural heritage such as historical monuments, buildings, parks, museums, etc.

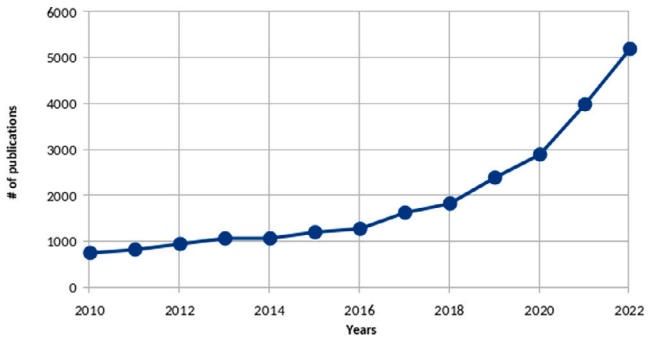

Fig. 1. Number of publications related to DTs in the period 2010–2022 according to ScienceDirect with the exposed criteria.

- Industrial sector (including road transport): in this area DTs are used for the control and maintenance of industrial facilities and the monitoring of relevant assets (e.g., trucks, as in our case). Thanks to IoT sensors, they are able to collect real-time data on the state of the machinery and simulate its future operation to reduce and prevent possible failures that produce stops in the production chain. A relevant sub-case of this category is the maritime sector. In this domain DTs are mainly applied to control cargo containers, which carry IoT sensors to monitor their position, temperature, humidity, etc. In this way, the condition of the goods can be controlled without the need to open the container and break the seal. Regardless, according to [10], DTs should focus on the digitisation of port facilities and the integrated management of their elements.
- Manufacturing industry: DTs are employed to optimise and reduce costs throughout all the life cycle of a product, encompassing in this term from the conception of the idea and the design to the production line and logistics aspects.
- Medicine: in combination with IoT sensors, this is the fastest growing area in the use of DTs. More precise control of hospitalised patients, simulation of different treatments or remote surveillance are the most common applications of DTs in the health care of people.

However, authors have observed that the particular sector in study (road transport) is an application domain where DTs have little implementation. This is due to the high acquisition and maintenance costs that this type of tool currently has in this area, especially considering that there is a lack of open, easy-to-access technology to deploy DTs. The proposed solution aims to change this situation by lowering the costs of utilisation, to widely use DTs and IoT devices in road logistic transport.

Generally, all DTs are formed by a basic structure of operation, which is composed of four basic pillars: (i) data acquisition and management, (ii) modelling the reality, (iii) simulation and forecasting based on such models and (iv) presentation and visualisation of the “physical element”. Relatedly, different types of DTs focus on one of those pillars or another, depending on the application and the final use of the system.

In recent years, there has been an increasing interest in integrating Blockchain technology with DT solutions [7]. This aims to address the need for secure and confidential data sharing while ensuring the integrity of information within DT environments. Industrial research has designed various solutions for different use cases, which have focused on transparency, trust, and the security of transactions and showcase the potential of this integration. Moreover, architectural frameworks have been proposed to facilitate efficient and secure real-time data management and guarantee transactional integrity in the data exchange across diverse industrial applications. Thus, the combination of DT with

Blockchain was addressed in the literature review by performing a new search focused on publications related with both DT and Blockchain. Those papers published in English where a solution combining both technologies was proposed were analysed (Appendix A).

The most prominent application domain for such solutions is the manufacturing sector. For example, Dietz, Putz and Pernul [11] analysed the requirement for secure DT data sharing and proposed a solution based on Distributed Ledger Technology that focused mainly on confidentiality and data integrity. ManuChain [12] combines Blockchain and DT in decentralised manufacturing environments. Makerchain [13] integrates DT and Blockchain to enable security and trust in manufacturing service transactions among different makers. Manufacturing Blockchain of things (MBCoT) [14] defines an architecture for secure, data-driven, autonomous, decentralised manufacturing. Huang et al. [15] designed a solution for product life cycle management that combined DT and Blockchain. Hasan et al. [16] proposed a solution for the creation of DTs based on Blockchain and smart contracts. Hemdan and Mahmoud [17] designed BlockTwins, which is a Blockchain-based framework for securing the transactions between physical and virtual assets in manufacturing. EtherTwin [18] combines DT and Blockchain for data sharing and information management and is available as an open-source prototype. Finally, Salim et al. [19] designed a framework based on DT and Blockchain to enable the early detection of bot activity in smart factory environments and prevent the infected nodes from sending data to the network.

Another domain where the combination of Blockchain and DT has been considered is construction. For instance, Lee et al. [20] proposed the use of Blockchain to add security and shared traceability to the transactions to the DTs and ensure the integrity of the data. Another example is the solution proposed by Hunhevicz, Motie and Hall [21], which combines DTs of buildings and smart contracts for performance evaluation and digital payments to implement performance-based contracts in construction.

Finally, Sahal et al. [22] designed a collaborative DT framework based on Blockchain that provides distributed consensus mechanisms and enables real-time data analytic. Four use cases were described, which included smart transportation and smart logistics scenarios. In a more recent work, Sahal et al. [23] proposed the use of Blockchain and collaborative DTs as the basis for a decentralised alerting system that could be used to detect COVID-19 outbreaks.

As can be seen, there is currently no DT combined with Blockchain technologies covering road goods transport companies expectations. However, there have been several initiatives addressing DTs, but no real consensus nor de-facto standard technology stands out. Also, no other previous tentative has incorporated Blockchain for registering certain events as in this work. Therefore, the developed solution has been tackled from a holistic perspective, i.e., selecting specific consolidated technologies in each one of those pillars to build a comprehensive DT without relying on any vendor-lock, isolated single technology. Also, the developed solution is capable of delivering high performance even in a low-resource environment. According to [24], less than 10% of developers create their own software, the others use professional solutions. Furthermore, in less than 25% of cases DTs run in real time [25]. There is, then, a clear gap (with room for improvement) that the proposed solution can fill. Hence, Section 3, as the main source of contribution to the state of the art, will consist of the definition and development of a use case design with the aim of filling this previously defined information gap under the implementation of a Blockchain-based Digital Twin system for IoT deployments in logistics and transportation.

3. Case study design

The structure of this section is as follows. In Section 3.1 the research questions are presented. Next, the use-case selection is explained (Section 3.2), followed by the data-collection procedure (Section 3.3) and the architecture of the solution (Section 3.4). The last point is Section 3.5, where the technical design of the solution is presented.

3.1. Research questions

For this case study, we derive the main research question of *how Blockchain can serve as a verifiable, decentralised, and immutable historical database for DT information, preserving critical information and providing traceability and data provenance* from the introduction. We further develop this into three sub-questions:

- How can DT help in logistic and transportation processes?
- How can blockchain provide traceability and security to DT technologies?
- How scalable is the combination of these technologies to a real-world implementation?

3.2. Case selection

To help answer these questions and to help in the trip assignment process as well as in the monitoring and management of the vehicle fleet, the DT presented in this work is adapted to the emerging IoT scenario and configured based on the assignment decision process in freight transportation (Fig. 2). This subsection will present the use case scenario and the architecture of the proposed solution.

As can be seen from the flow depicted in Fig. 2, the process begins in the logistics company, when a trip is planned and the most efficient solution must be calculated and presented to the company. Here in the Digital Twin, using real-world data, a recommendation is made to the company, which must be reflected again in the DT if the company decided to carry this out. For this, the system should include a distributed ledger register. This requirement comes from the need of absolute traceability of actions. An improper communication or accountability might derive into delays that will directly impact the productivity and the quality of the service. In addition, many goods-carrying companies may depend on the throughput of their transport operations to guarantee their sustainability.

For all the above, verifiability of the actions in a flow such as in this scenario is paramount. Concretely, when a driver-trailer tandem is assigned to operate a cargo order (trip), there are several aspects that should be accounted for and whose execution should not be refuted (Fig. 3):

1. Whenever a transport order is received, a `timestamp` must be created and recorded to ensure the moment of trip request reception.
2. After several calculations, and based on continuously updated information, the DT offers a recommendation (driver-trailer tandem) to fulfil the order through its own recommendation system.
3. Regardless of the result, the (human) user of the system will decide which is the proper assignment to perform. This action is of utmost importance. Therefore, it must also be stored on the distributed ledger.
4. Whenever the transport order (journey, trip) has been completed, a new entry is registered, completing the cargo processing cycle from the traceability perspective. Blockchain technologies are an essential component in the traceability of transactions because by default no environment where data and information is shared can be considered completely secure for intermediaries. Blockchain adds a layer of security that minimises problems of transparency, traceability and trust in the transactions carried out [26].
5. Whereas all the previous can be considered sequential actions, the Blockchain will be storing also as relevant events those moments where a truck leaves off or arrives at a Point of Interest (PoI), meaning that it did upload/download a part of the cargo of the whole order.

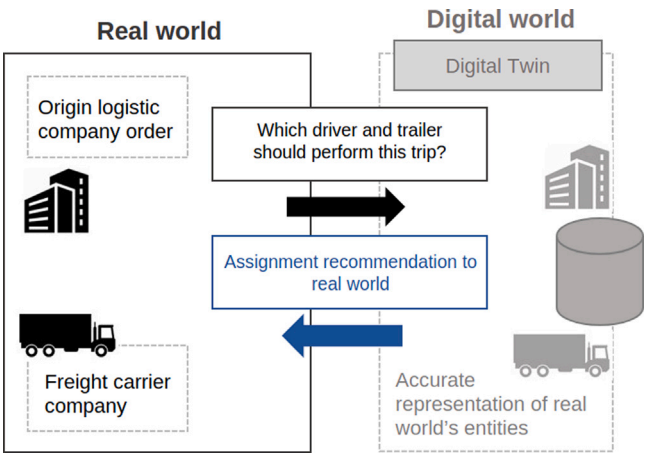

Fig. 2. Overall explanation of the scenario.

All the described logic of the DT allows the simplification of the processes and activities of these companies. This implies having a record of the most important steps in the logistics process, having greater transparency and integrity of the data generated, minimising supply chain times, reducing expenses and polluting emissions from the vehicle fleet, etc.

3.3. Data collection procedure

All information sources used to feed the DT can be considered as IoT components of the overall system architecture because they are data sources of different types and origins that constantly send information flows that are useful for the intended applications. In addition, the developed platform also uses other IoT components, such as a context broker, entities to store information, the phones of the drivers to capture distributed data in real-time, etc. The aforementioned data sources are the following:

- **Real-time positioning**: trucks of the fleet by the haulier company are equipped with IoT GPS devices that continuously provide data to the DT (frequency: 10 s). These data allow them to know the routes of the vehicles, the rest times of the drivers, the start and end time of the trips, etc.
- **IS of the logistics hub**: Information System of a logistic hub requiring the processing of the transport order (trip). In the scenario deployed in this work, this materialises in a Port Community System (PCS) of a container maritime port, which is an electronic platform that allows information to be exchanged securely between entities of a port community.
- **Transport carrier database**: an updated database owned by the freight transport service provider including the latest data about drivers, vehicle fleet availability, points of interest and other related information.
- **Driver's app**: the drivers in the scenario have an app installed on their smartphones through which they can report any incidence occurred during a trip, the places where they stop and continue the journey, etc.
- **Surrounding data**: data related to the environment, such as weather conditions, public traffic congestion situation and various Application Programming Interfaces (APIs), such as vehicle plate number related queries.

In order to store all the data coming from the different sources of information and manage all the logic behind the processes and actions performed in the DT, it was deemed relevant to develop and entity-relationship model. The business logic exposed through Fig. 4 represents different entities and relationships that help express the current state of the system and that are leveraged to recommend trips to drivers. This class diagram is fully expanded in Appendix B where the fields that complete the entity-relationship model prepared for case study generation can be found.

3.4. Layer-based architecture of the proposed solution

This subsection introduces the layer-based architecture adopted as a model to address the case study. The various layers defined are detailed below and are shown in Fig. 5.

- **Physical space**: Contains a set of sensors that provide data from the real world to the upper layers.
- **Information sources**: contains the different data sources (i.e., local databases) that can provide the necessary data to generate DTs.
- **Communication Layer**: Acts as gateway between the data sources and the upper layers, selecting the data from the lower layers and sending it to the Data Access Layer.
- **Data Access Layer**: Collects the data, extracts the relevant information for the DTs, adds context to this data, and converts it into the proper format and semantics.
- **Digital Twin Layer**: Collects and stores the information sent by the Data Access Layer and based on this information, generates and stores DTs.
- **Application Layer**: Provides services and applications based on the DTs, including composite indicators and data-driven predictive models.
- **DLT Layer**: The DLT Layer is responsible for storing essential information from the DTs (as well as applications) in the ledger. The ledger maintains a record of only the most crucial data, ensuring traceability and transparency among various stakeholders.
- **Security and Privacy**: Provides the means for ensuring data security and privacy protection across all layers of the architecture.

3.5. Analysis and technical design

As discussed earlier, a comprehensive monitoring of the entire transaction of goods addresses the current limitations in logistics operations by pinpointing delays and issues while also preventing fraud. In this paper, the introduction of DT and blockchain technologies aims to show the possible improvements in the logistical sector. This subsection presents the technical design of the solution (Fig. 6), associating the components with the corresponding layers of the architecture (Fig. 5), and describes its operation, considering the different types of input data, using the flow diagram seen in Fig. 7.

Following the architecture (Fig. 5) from top to bottom, the first layer is the Application Layer, where components such as the Dashboard and the REST API services (Fig. 6) are located. The second is the Digital Twin Layer, where the data must be synchronised between the real and the digital worlds, for this purpose, the Rules Engine and the Composite Modeller are necessary, as well as the Orion Context Broker and Canis Major using Ethereum. The last two components provide the vertical **DLT Layer** of the architecture. The third is the Data Access Layer, where all the components that handle data and redirect traffic are located, such as the KrakenD API Gateway, the Orion Context Broker, Data IoT Agent and the App Gateway. Following this the fourth one is the Communication Layer, where the multiple IoT communication protocols are located. The fifth are the Physical Space and the Information sources that send the data into the communication layer and they consist of the Data Sources and the Sensors. The last two layers are vertical and comprise the entire architecture, the Security and Privacy layers ensure only verified petitions can be made to the system and they consist of both the Keycloak IDM and the KrakenD API Gateway, which is also present here.

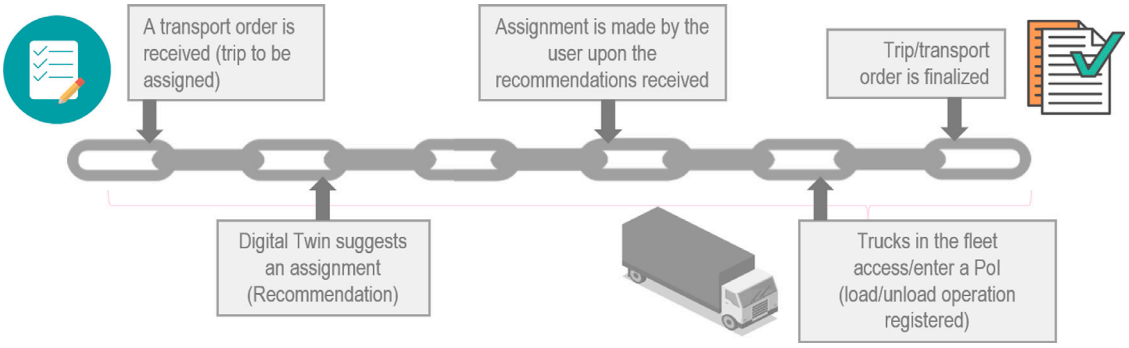

Fig. 3. Blockchain registers selected in the scenario.

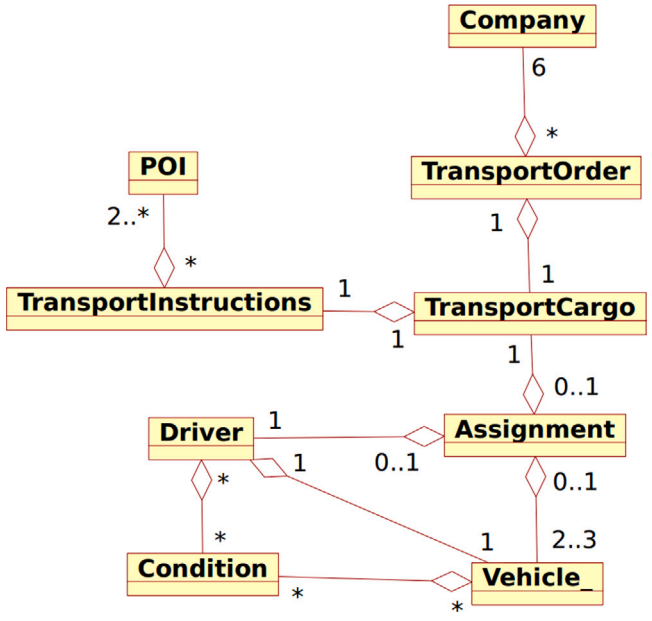

Fig. 4. Class diagram.

From here to the end of the section said data flow will be explored and explained, using the design seen in Fig. 6 and the flow diagram seen in Fig. 7 as reference, as well as two sequence diagrams (Figs. 8 and 9). In line with the described processes and the flow diagram, the traffic has two main ways of starting:

1. The **sensors** and the **data sources**. The sensors cover both the physical space layer (in sensors focused on obtaining data directly from the physical environment) and the information sources layer (in sensors that manage events occurring in real-time) seen in Fig. 5. The collection of raw data from the sensors is the first phase of any IoT environment. The collection of raw data from the sensors is the first phase of any IoT environment. Once the data has been obtained through the sensors, it is sent to an element that acts as **App Gateway**. This includes the identification of each sensor (or data source), along with the data obtained itself. For data exchange between the sources and the application to happen, IoT communication protocols are leveraged. This is part of the Communication Layer. The data extracted from the sensors is, then, pre-processed. For this purpose, different **FIWARE IoT Agents** are implemented to send data (from data sources) to and be managed from a Context Broker using native protocols, this is the beginning of the Data Access Layer. In effect, this brings a standard interface to all IoT interactions at the context information management level. Each group of data-provisioning elements (e.g., IoT devices) are able to use their own proprietary protocols and disparate communication mechanisms under the hood whilst the associated IoT Agent offers a facade pattern to handle this complexity. This Agent reads the incoming data stream and not only enforces data integrity, but also, if necessary, transforms the data.
2. The **dashboard** or the **Rest API**. These components are critical in ensuring that all changes in both ends of the spectrum are reflected immediately on the other end, thus achieving a DT system. Firstly, it relies in the usage of a **Composite Modeller**, which in combination with a **Rules Engine** does (i) mirror the behaviour of the system by processing information and applying grey-box models, (ii) emulate the optimised output of an assignment procedure by proposing a recommended driver-trailer tandem, and (iii) ensure that all modifications and decisions made by the user are properly persisted. For direct usability, this module presents a simple, user-friendly web interface for visualisation and decision making by authorised clients. Following the microservices-based approach of the architecture proposed, it exposes a REST API endpoint ready for all interactions with the component. These components are part of the Application Layer.

The data, regardless of, now enters the **Powered by FIWARE** zone of the platform as seen in Fig. 6. The entry point materialises with the **KrakenD API gateway** and can be seen in Fig. 7 when the HTTP Request reaches KrakenD, a critical component of the Data Access Layer and the Security and Privacy layers seen in Fig. 5. KrakenD is an open-source API Gateway that also implements the back-end for front-end and Micro-front-ends patterns to eliminate the necessity of dealing with multiple REST services. KrakenD receives all connections coming from the clients and verifies access to the data through the Identity Manager user tokens sent by the IoT Agent. This is the beginning of the process shown in Fig. 9. In addition, KrakenD receives the formatted data from

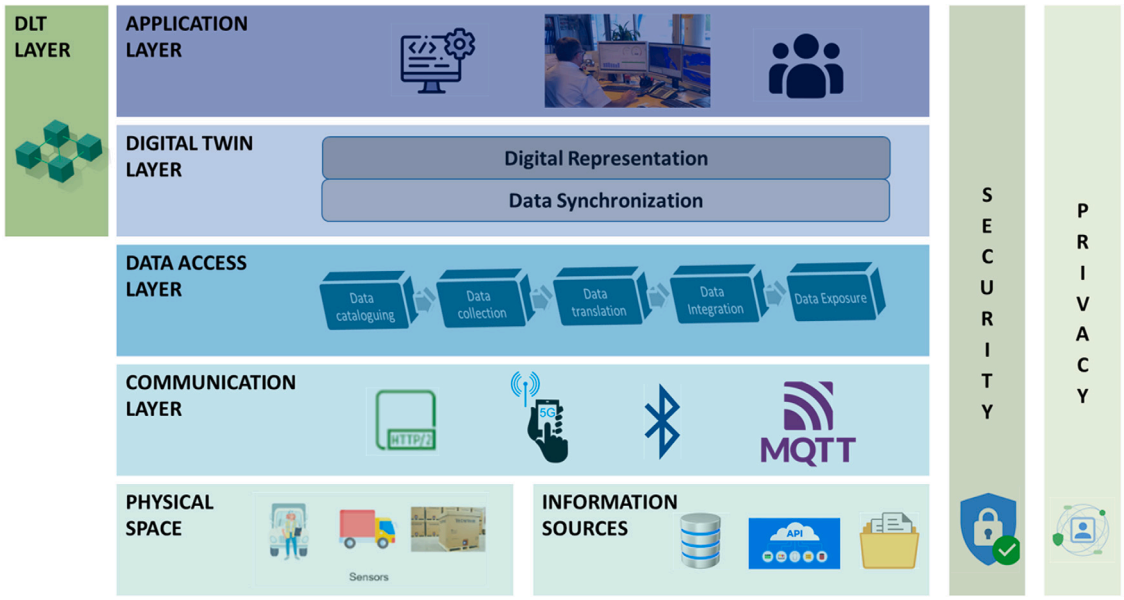

Fig. 5. Layer-based architecture of the proposed solution.

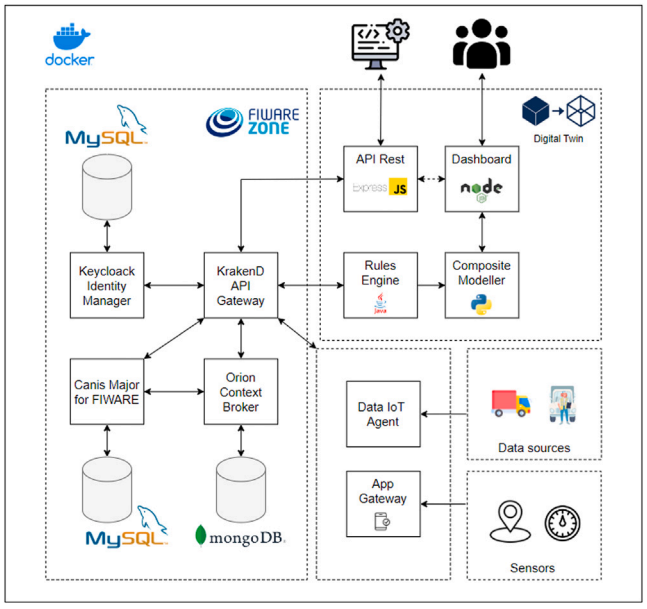

Fig. 6. Component-based design of the solution.

the gateway side, which should include the aforementioned identification token of the registered sensor together with the DLT token in case the data should be included in the Blockchain as shown in Figs. 8 and 7. This is done specifically to ensure further authentication and interaction capabilities with the Blockchain. Both the DLT token and the authentication token are imperative, as they serve unique functions within the system, interacting with diverse components in varying ways, thus requiring their concurrent application for this operations:

- The Keycloak Identity Manager token carries authentication and authorisation information regarding the network.
- The DLT token carries authentication and authorisation information relevant to the blockchain itself.

In order to access the Data Access Layer via KrakenD, all petitions must use a token extracted from the **Keycloak Identity Manager**, which is a critical part of the Security and Privacy Layers. Keycloak was chosen because it not only provides authentication to all actors trying to enter the platform via user tokens but it also provides sign on and login services, meaning that applications do not have to deal with login forms and storing user credentials. It manages user roles and verifies tokens from all the petitions coming into the network, ensuring no unauthenticated users can enter the network from the outside. Other access control models were evaluated such as a Self-Sovereign Identity Based Access Control (SSIBAC) [27], which while being a very innovative new model to access the data did not completely fit into the proposed mould, where a more centralised Identity Management option such as Keycloak was more than enough for the architecture necessities.

As can be seen in Fig. 7, as well as both Figs. 8 and 9, after receiving data, KrakenD sends a verification request to the Identity Manager in order to check if the user's petition is valid. All petitions, after being authenticated, are redirected to **Orion Context Broker** and Canis Major as necessary. This is possible thanks to KrakenD's packet modifiers and its integrated compatibility with LUA scripts, allowing modification of incoming and outgoing data, as well as the creation of new messages under determined circumstances. These are some of the main reasons why KrakenD was selected for this implementation. The petition is allowed into the context broker, and into **Canis Major** if the petition has critical data that is considered important enough to be stored in the Blockchain. The combined use of the Keycloak Identity Manager and the KrakenD API Gateway enables **vertical security** through the entire solution:

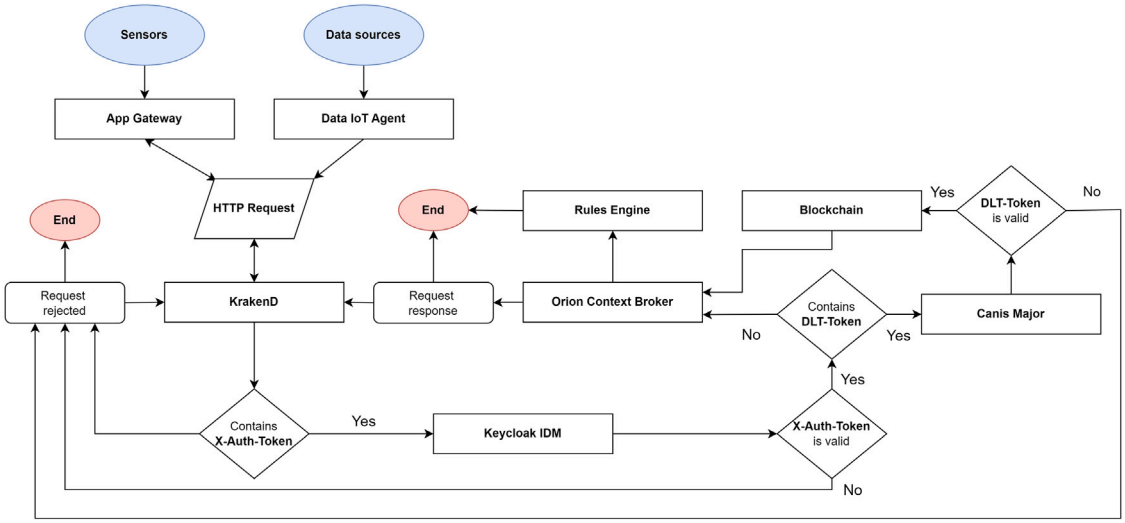

Fig. 7. Data flows in the proposed solution.

- On the client side, upon successful authentication by the physical client or the exposed API, it will be possible to view the processed data based on the viewing permissions that are granted.
- On the gateway side, data will be created and modified in the Context Broker and Blockchain based on the events received.

Depending on the type of message and the data it contains the flow will take one of two possible directions, seen in Fig. 7. The first of the possible destinations of the petitions redirected by KrakenD is the **Orion Context Broker**. Orion is an implementation of the NGSIv2 REST API binding developed as a part of the FIWARE platform. Orion Context Broker allows the management of the entire life-cycle of context information, including updates, queries, registrations and subscriptions. It is an NGSIv2 server implementation to manage context information and its availability. In addition, it provides subscription to context information so when some condition occurs (e.g., the context elements have changed) a notification is sent. The FIWARE zone represents the true back-end of the solution, with multiple databases in charge of storing everything from user authentication data to blockchain relevant information, as well as the API gateway, identity manager, context broker and DLT component. Furthermore, the proposed solution is a platform "powered by FIWARE", emphasising its integration with a technology promoted by the European Commission, among others [28].

The other possible destination of the data after passing through the API Gateway is **Canis Major** [29]. Canis Major stands out as an innovative blockchain software that provides a layer of data security through veracity and non-repudiation. It acts as an adapter that allows connecting an Ethereum Main Net Blockchain into a FIWARE-compliant ecosystem. Canis Major works using Ethereum at its core, which offers certain advantages, such as decentralisation and transparency, although it can also present some efficiency concerns for a logistics application due to scalability issues and high transaction costs associated with its public blockchain. The deliberate choice of Canis Major as the underlying technology with its usage of Ethereum was made after careful consideration of the trade-offs and requirements associated with permissioned networks. Ethereum's robust and proven framework provides a solid foundation for Canis Major's role as a secure and interoperable adapter. Emphasising interoperability, Canis Major serves as a bridge, facilitating the seamless integration of Ethereum-based technologies into our FIWARE-compliant environment. This strategic choice not only ensures the reliability and security of data transactions but also leverages Canis Major's adaptability and versatility within the context of emerging technologies. When used, the Canis Major component retrieves data in a standard format and records it in the designated wallet of the blockchain, thus acting as an intermediary. What distinguishes Canis Major is its great adaptability to the IoT ecosystem along with and ease of use through access tokens.

Once the entire deployment is in place, the environment is ready to process incoming communications from all data sources. The entire flow described in this section can be seen reflected in the data flow diagram seen in Fig. 7:

1. The flow starts at either the sensors or the data sources, sending a petition to the App Gateway and the Data IoT Agent respectively.
2. An HTTPS request is made to the KrakenD API Gateway.
3. KrakenD sends the token to Keycloak to verify it, if its invalid or does not have a token the request is discarded.
4. If the token is valid, KrakenD verifies if the petition has a DLT Token, if it does not have one, the request is sent to the Orion CB.
5. If the petition has a DLT Token it is sent to Canis Major in order to validate it. If the DLT Token received is valid the payload is added to the blockchain and then the request can continue the to Orion CB. If the DLT Token is invalid the petition is discarded.
6. Orion processes the petition, applying the changes to the Rules Engine, before sending the response back to KrakenD.

Alternatively, a more in-depth depiction of both cases can be seen below, reflecting the contents of Figs. 8 and 9.

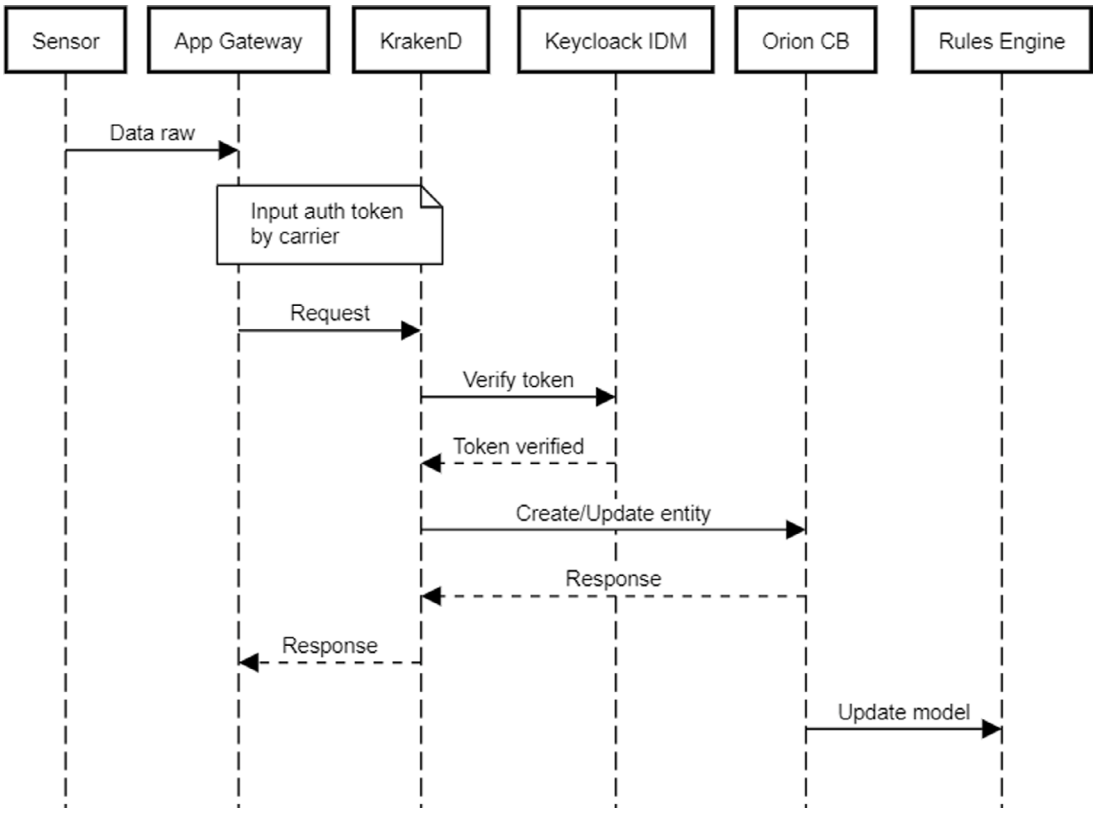

Fig. 8. Sensors sequence diagram (without DLT token).

- Fig. 8 shows the sequence of the data transmitted by the sensors through the platform to the DT zone, where data is processed and displayed through both communication channels shown in Fig. 6. The flow begins in the sensors themselves that send the raw data to the App Gateway, which adds the authentication token to the petition before sending it to the KrakenD API Gateway. KrakenD now verifies the token with the Keycloak IDM and sends the petition to the Orion CB that replies with a confirmation of the changes and updates the model in the Rules Engine.
- Fig. 9 shows what happens if the flow begins in one of the Data Sources themselves, where the information obtained generates events that are not based on time series. This means that after the petition that reached KrakenD has been validated by Keycloak the information not only goes to Orion to be added and modify the Rules Engine, it also goes into Canis Major, which submits the data transaction into the blockchain if the DLT token is successfully validated.

With this section, the design and implementation of the presented solution are explained in detail. The next steps are showcasing the results of committing to an actual deployment of the case study.

4. Results

In this section, the application of Canis Major is illustrated to understand the problems faced in logistics related immutability and trust processes. The focus is on the Digital Twin process that is part of the overall value chain process of creating an event called *Allocation*. The analysis of how the proposed Blockchain-based DT has been useful for the Logistics and Transportation case revolves around how: (i) the DT performs certain processes upon the generated Allocations and (ii) the Blockchain (through Canis Major component) integrates the relevant parts of the event and, (iii) the system itself performs in terms of response times and functionality coverage. The section is structured in three subsections, each focused in answering one of the three research questions presented in 3.1. Section 4.1 focuses on the Digital Twin and how it can help in the proposed use case, Section 4.2 focuses on blockchain and how it improves traceability and security in DT and finally Section 4.3 explains how scalable the combination of these technologies actually is.

4.1. Digital twin process

As mentioned, changes on either side of a DT are reflected on the other almost instantaneously. In this subsection, through the process of describing the entire DT workflow, the first research question will be answered.

RQ1: How can DT help in logistic and transportation processes?

For the purpose of the generated information to be displayed, authorised users first authenticate to the web interface using the KrakenD API Gateway. Once the user gains access, the interface makes a request for information through the Composite Modeller and the Rules Engine to Orion Context Broker to obtain all the data of the necessary entities in order to complete the tables and the map with the specified information. Before displaying the requested data, grey-box models are applied to process it and adjust it to the interface representation format. When physical sensors or other data sources send data to be stored in the system through Orion Context Broker, some of this information is forwarded through the Rules Engine and Composite Modeller to the

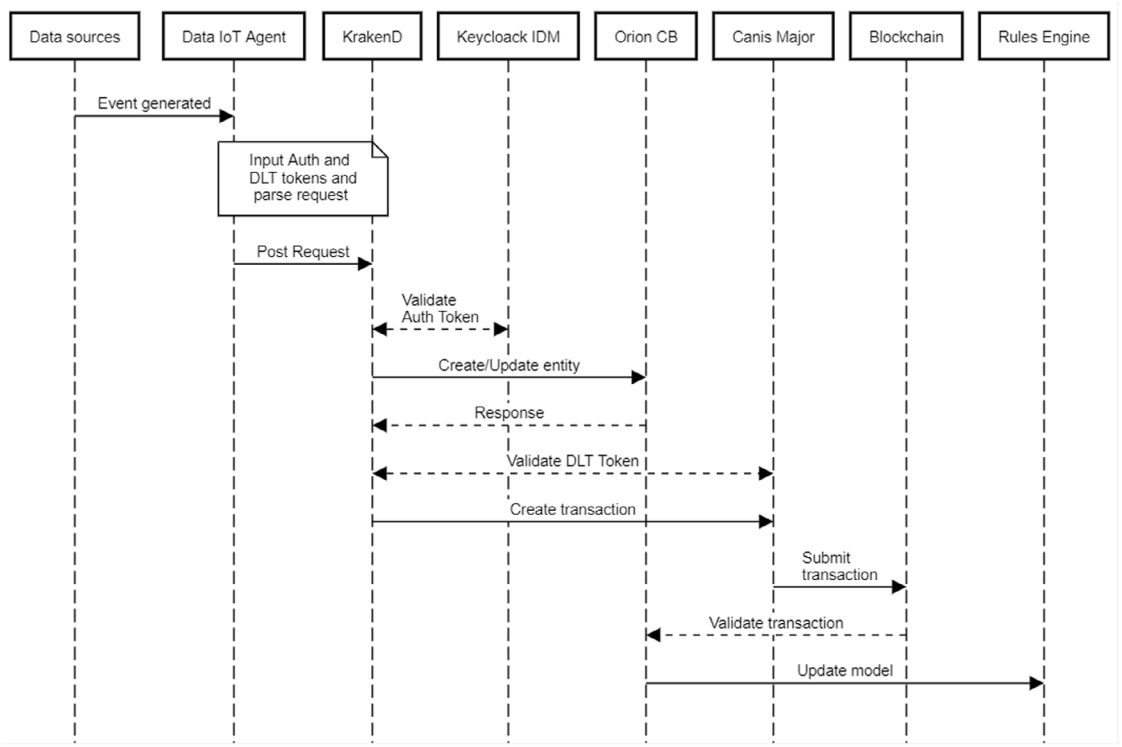

Fig. 9. Data sources sequence diagram (with DLT token).

dashboard for display. In this way it is possible to always have an updated and real-time vision of the environment controlled by the DT.

To obtain an assignment recommendation, the assignment manager indicates in the second tab of the dashboard the identifier of the Transport Order and the identifier of the Transport Instructions associated with the Transport Cargo to which the trip must be assigned. The control panel sends the request to the Rules Engine through the Composite Modeller to determine the best driver-trailer tandem for that trip. The Rules Engine, obtaining the access token through the KrakenD API Gateway, accesses all the vehicle and driver information stored in the Orion Context Broker database. Here, the proper, optimal selection happens. In this paper, a grey-box behaviour model (based on composite indices and a dynamic rules engine) has been used. However, it must be noted that more advanced solutions could be used. For instance, KPI-oriented optimisers, or ML-model based resolution, etc. might be envisaged to provide the best assignment selection. Specific AI methods to perform such optimisation were not the main focus of this work, thus further investigation will be needed to deepen this part.

As mentioned, in this work, once the necessary information is obtained, the Rules Engine composes the entities and analyses the established parameters to determine the most suitable tandem to make the trip. To choose the best tandem, the Rules Engine executes two well-differentiated steps. The first step is to eliminate those drivers and vehicles that do not meet the necessary requirements to make the trip. Then a list of possible tandems that can make the trip is obtained. In the second step, a series of KPIs are analysed, which are determined through the normalisation of the values of the attributes of each tandem. These normalised values are compared to obtain the optimal tandem. Examples of these attributes are: vehicle fuel consumption, type of engine for each truck, available driving hours, kilometres travelled that month, etc. Finally, the Rules Engine generates an Assignment entity following the data model (Fig. 4) and sends it to Orion Context Broker to be stored in its database. At the same time, the assignment recommendation is sent to the Composite Modeller to be properly formatted and visible to users through the dashboard.

At this point, the assignment manager can either accept the assignment recommendation or reject it and manually enter a new one. At the moment the assignment manager makes a final assignment, an **event** is generated in the platform. With this, the integration of DT technologies have improved the logistical transportation process by helping the manager and the employees of said logistical company have a clearer real-time vision of the entire fleet, as well as by providing the best available route for any given task.

4.2. Blockchain workflow

After discussing and explaining everything that happens on the Digital Twin side, thus subsection will focus on the Blockchain side of the network, answering the second research question in the process:

RQ2: How can blockchain provide traceability and security to DT technologies?

```
{
  "id": "assignment:007",
  "type": "Assignment",
  "owner": "UPV",
  "cargoId": [
    "transportCargo:123"
  ],
  "driverId": "driver:088",
  "trailerId": [
    "trailer:198"
  ],
  "assignmentDateTime":
    ↪ "2023-06-26T12:20:00Z",
  "status": "Sent"
}
```

Listing 1: JSON body in event generation

Every time that a relevant event occurs (see Section 3.2), the mechanism for trusted, immutable storage of the occurrence of such event is triggered. Once it happens, an *HTTPS* request is sent through the dashboard containing the tokens as well as the message body in *JSON* format. Following the workflow seen in Fig. 9, KrakenD validates the authentication of the request in conjunction with the Keycloack Identity Manager and sends the payload to the Orion Context Broker. An example of a payload in *JSON* format can be seen in Listing 1. When the content is successfully saved, KrakenD notifies Canis Major to validate the 'DLT-Token' containing the information regarding the user holding the corresponding *wallet* where to store the data on the Blockchain. This is where the blockchain begins adding value.

Canis Major is able to validate the 'DLT-Token' by generating a connection to a compatible Blockchain via Smart Contracts. For this use case, a Blockchain based on Ethereum has been used, where it is recommended to use the AEI contract model in order to describe Smart Contracts. The AEI contract model (Asset, Event and Identity) is written in Solidity14 language using the ERC72115 standard. The design of an AEI contract is shown in Fig. 10, where an Entity (or Asset) has a unique identity by performing a 1:1 mapping (one Entity to one Identity). At the same time, an Entity has a mapping with a 1..n relationship to Events (or Metadata). In addition, an Asset has a 1..n relationship with any other Entity.

When Canis Major receives the token validation request, it obtains the public/private key contained in the token, validates the information using the AEI Smart Contracts with the associated Blockchain and returns the validation status to KrakenD, which, if successful, immediately sends the corresponding data (described in Listing 1) back to Canis Major to create the corresponding transaction. Finally, Canis Major sends the transaction status information to Orion Context Broker, which in turn, through the publisher-subscriber model, sends this information to the DT zone, completing the workflow. With this increased traceability and security has been provided to the Digital Twin, thus fulfilling one of the main objectives of the proposed use case.

Finally, following the path in Fig. 6, in the same way that Canis Major receives events through the Digital Twin zone, it also receives a series of events related to events that occur through different data sources. These events are sent through a Data IoT Agent, which is in charge of entering the credentials to generate the access and DLT tokens, as well as formatting the request in JSON.

4.3. Performance testing

In this subsection it will be showcased how the designed solution does not produce an excessive increase in resource consumption. With this, the last research question will be also be answered:

RQ3: How scalable is a combination of these technologies to a real-world implementation?

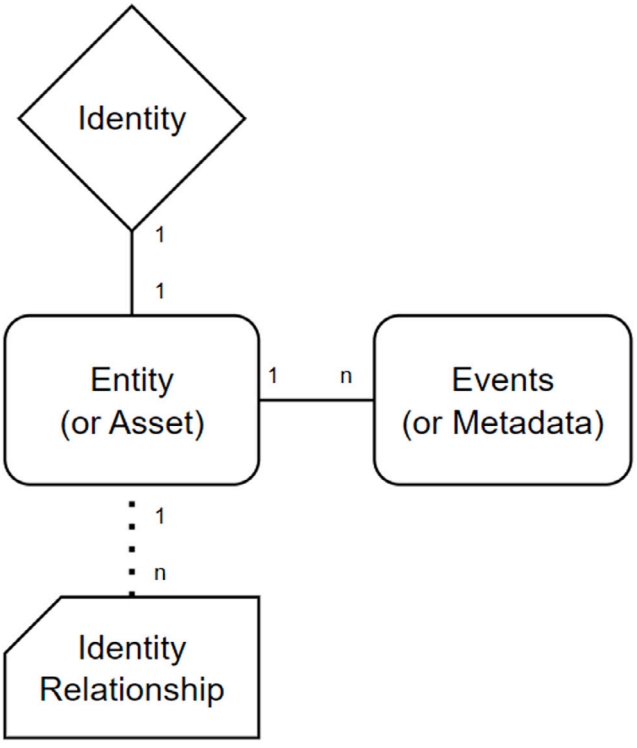

Fig. 10. AEI Contract Entity-relationship model.

Table 1
Simulation requirements.

Docker component	Specification	
	RAM	CPU
Orion Context Broker	6 GB	1
Mongo-DB	16 GB	4

The current scenario would only allow to push the system to a certain limit, given that both the sensors and the data sources were constrained. As delineated in Section 3.5, the validation use case employed in this work reached up to 65 trucks, with a number of assignments in the order of half-a-dozen per day. As explained, these figures correspond to a regional-level haulier company, near the geographic area of the authors. Therefore, in order to learn to which extent could the designed system perform effectively, an environment was prepared to simulate/generate between 1000 and 100 000 sensor updates and between 100 and 1000 per minute. After applying this scalability performance test approach, the following results were found. In total, Canis Major receives between 100 and 1000 transaction creation/update requests on the Blockchain and Orion receives between 1000 and 100 000 sensor entity update requests and between 100 and 1000 event creation/update requests (80% from Data IoT Agent and the remaining 20% from manual assignments in the Digital Twin Zone) plus 1 extra transaction response from the Blockchain in each request in both cases. In addition, the performance of the data update and processing in the Digital Twin Zone has been measured to generate a recommendation after all new data has been entered.

For these tests, an environment was simulated using the FIWARE load-tests [30] tool. This tool allows to simulate sensorisation devices

Table 2
Performance of the Orion Context Broker component in updating registered sensor data.

Request	Executions			Response time (ms)			Total time (s)
	Total	Error (%)	Counts/s	Min	Max	Mean	
Update entity (Real Time Positioning)	1000	0.00	1589	1	3	1	0.62
	5000	0.00	1512	1	18	2	3.31
	10 000	0.00	1355	1	176	17	7.39
	50 000	0.00	1152	1	1204	125	43.4
	100 000	0.00	920	1	1844	164	108.69

Table 3
Performance of the proposed solution in processing generated events.

Request	Executions			Response time (ms)			Total time (s)
	Total	Error (%)	Counts/s	Min	Max	Mean	
Events generated	100	0.00	217	2	12	4	0.46
	500	0.00	40	2	52	25	12.55
	1000	0.00	17	2	89	61	61.23

Table 4
Performance of digital twin assignment recommendation time.

Assignment	Number of executions		Response time (s)		
	Total	Error (%)	Min	Max	Mean
Get recommendation	10	0.00	2.56	2.97	2.7
	50	0.00	2.53	2.95	2.66
	100	0.00	2.53	2.99	2.61
	200	0.00	2.52	3.52	2.63

and to generate entity update requests. For the generation of events, in the same way, devices that generate events automatically and also manually from the dashboard have been simulated. This environment has, among others, the minimum characteristics of some of the components described in the Table 1. Comparing with the recommended test environments for the tool, this environment corresponds with the *tiny* environment.

Table 2 shows the performance of the Orion Context Broker in updating registered sensor data. In this case, GPS positioning sensors from several vehicles are updated simultaneously. The table presents information on the total number of updates executed in a short period of time, the error rate and the number of updates per second recorded. In addition, the table provides data on the response time, including the minimum, maximum and mean. Finally, the total execution time of each test is indicated. It is important to note that the mean execution time of each test in the entity update is as follows: for 1000 concurrent updates it is 1 ms, for 5000 it is 2 ms, for 10 000 it is 17 ms, for 50 000 it is 125 ms, and for 100 000 it is 164 ms. No processing errors have been observed.

Once the environment is considered to be able to persist with a large number of simultaneous requests, the next performance test determines the processing time when between 100 and 1000 events generated by different data sources and manual assignments are received (approximately 80% and 20%, respectively). For this test, the processing time of all the components involved in the process has been measured, highlighting Orion Context Broker, Blockchain and Canis Major (for this example, the authentication and authorisation time of each request by KrakenD and Keycloack Identity manager is considered negligible). Table 3 shows the result of the minimum, maximum, mean and total time from when an event request is received until it is entered into the Blockchain and validated in Orion Context Broker. In this case, no processing errors have been observed either. The mean time obtained for 100 events is 4 ms, for 500 is 25 ms and for 1000 events generated is 61 ms.

Once the sensor update and event generation processing times are known, the last performance test consists of obtaining the Digital Twin processing time from the moment a user requests an assignment recommendation. Table 4 shows the total recommended processing time of a manual assignment based on several simultaneous requests. This process has been performed several times with different executions and the minimum, maximum, average and total time for each test has been obtained. In this case it can be seen that the average processing time in all cases is around 2–3 s.

Based on the results of the scalability tests presented in Tables 2, 3 and 4, it is empirically evident that the software not only meets the defined requirements but also exhibits an impressive traffic scalability capacity exceeding 1000%. These tests support the robustness and efficiency of the system, demonstrating its suitability to handle significantly higher workloads than previously anticipated.

At this point all three research questions have been successfully answered: (i) Digital Twin has been proven to help in logistic and transportation cases by reducing the time it takes to calculate the most optimal route, (ii) Blockchain technologies can add an extra layer of security as well as traceability over said DT technologies and (iii) an implementation of these technologies in an already existing real-world environment does not cause a severe load increase.

5. Limitations and discussion

The structure of this section is as follows. Section 5.1 presents in detail the hardware and software environment used for the experiments of the case study and their limitations. Section 5.2 describes the tests that would have been performed and the results that would have been expected if a more powerful hardware environment had been available.

5.1. Limitations

It is important to highlight that the experiments conducted in this case study present some limitations. First, it should be noted that this solution was proposed and evaluated in the specific context of transportation and logistics, resulting in a limited scope of the study in an environment with previously known entities and low variability of data. Also, as mentioned in the previous sections, the volume of sensors and the size of the whole experiment were constrained. That is why a larger (simulation) environment was created to push the system to validate the performance of it. Thus, the performance tests were conducted in a controlled environment, making use of simulated software. As a result, some important considerations, such as connectivity issues or performance in environments with third-party software, whenever envisioning large-scale demonstration, were not fully addressed.

A virtual machine with a configuration adapted to the use case was used to perform the experiments described above. This virtual machine was installed on a development and test server of the research team. The configuration used in this virtual machine is oriented to offer sufficient hardware resources to the proposed solution, but with resource limitations due to the development server itself. The operating system used was the latest available Long Term Support (LTS) version of Ubuntu, namely version 22.04.1. In order to run all the software components of the case study, Docker Engine version 20.10.21 and Docker Compose version 1.25.0 have been used. In order to support the required workload, the virtual machine was configured with 8 processor cores of the Intel(R) Xeon(R) Gold 6230R processor, 64 GB of RAM and 64 GB of internal HDD storage.

The hardware and software environment used for the use case experiments is limited by the development server itself, namely by the maximum storage capacity and the amount of resources available for the virtual machine. These restrictions and limitations are due the research team's need to use the development server for other projects, services and requirements.

5.2. Discussion

As discussed above, the validation use case employed in this work reached up to 65 trucks, with a number of assignments in the order of half-a-dozen per day. Due to the limitations of the development server used as the test environment, the scenario developed only allows the system to be pushed to a certain limit.

If a development server with more powerful hardware had been available for the test environment to provide a virtual machine with more resources, load testing of the software components could have been performed by simulating truck fleets of up to 480 or 500 vehicles as validation for the use case. The number of assignments performed could be as high as 45 to 50 per day, taking into account the increase in the vehicle fleet. These values would correspond to a nationwide transportation company.

On the other hand, the *mid* test environment of the FIWARE load-tests tool could have been used for performance testing. That way, up to 12.000 updates per second of the vehicle positions could have been achieved, in the best case scenario. This would demonstrate the capability of the system to be used by national, as well as regional, companies.

6. Conclusions and future work

Throughout the case study paper, the main topics discussed have been the Digital Twin model and Blockchain technologies and how they can be used in conjunction to improve a Smart Logistics use case. First, an investigative work was done to ascertain the areas in which DT is used and how it operates, particularly in the industrial sector (including transportation) and then following it up with established Blockchain technologies. Here, an unexplored and very interesting synergy can be seen between both technologies. It was then confirmed that a DLT-empowered Digital Twin results in a proper solution for monitoring and managing IoT deployments in logistics and transportation. In the Smart Logistics scenario, where variables change in real time and must be correlated with information-based decisions that have an effect in the real world, DTs are the preferred innovative tool. The goal is to propose a flexible, open-source Blockchain-based DT architecture to allow road transport logistic companies to make better decisions leveraging IoT concepts and technologies. For this purpose, additional technologies have been used in support, such as Canis Major for FIWARE accompanied with an Ethereum deployment as the Blockchain element of the solution, Keycloak as the Identity Manager, KrakenD as the API Gateway and Orion as a context broker. Additionally, a Rules Engine and Composite Modeller have been used in conjunction with a Dashboard to represent the real-world side of the Digital Twin.

The focus of this study was on evaluating the solution in a limited scale, limited scope environment while prospecting the feasibility and compatibility of the solution integrating Digital Twin together with Blockchain-based technologies. The main results of the study show that the synergy between both technologies has allowed greater data veracity, integrity and immutability, while providing real-time information and personalised recommendations to the platform managers. The solution presented could easily work in an actual logistics company, as has been discussed previously, although the implementation of the solution in a large, full-fledged, real-world environment may require further consideration of additional aspects and practical challenges that were not explored in depth during these controlled experiments.

Due to the limitations of the software explained in the Limitations section, this case study was approached from a theoretical perspective with a limited number of daily actions, the next step would be to try and formalise the proposed solution in a real-world use case scenario. More can yet be discovered by continuing with this line of investigation from the Blockchain standpoint. Also, an ecologic use case study could be conducted to see just how much energy is being consumed and how other alternatives could prove more efficient. Looking at Digital Twin's research perspective, the permeation of AI services and mechanisms seems to be a very promising prospect that authors will want to further explore.

CRediT authorship contribution statement

Salvador Cuñat Negueroles: Conceived and designed the analysis, Contributed data or analysis tools, Performed the analysis, Wrote the paper. **Raúl Reinosa Simón:** Conceived and designed the analysis, Contributed data or analysis tools, Performed the analysis, Wrote the paper. **Matilde Julián:** Conceived and designed the analysis, Contributed data or analysis tools, Performed the analysis, Wrote the paper. **Andreu Belsa:** Conceived and designed the analysis, Contributed data or analysis tools, Performed the analysis, Wrote the paper. **Ignacio Lacalle:** Conceived and designed the analysis, Contributed data or analysis tools, Performed the analysis, Wrote the paper, Associated funding. **Raúl S-Julián:** Contributed data or analysis tools, Performed the analysis, Wrote the paper. **Carlos E. Palau:** Conceived and designed the analysis, Performed the analysis, Wrote the paper, Supervision and associated funding.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research has been partially funded by the European Commission, under the Horizon 2020 project OPTIMAI, grant number 958264, and the Spanish Regional Program Agencia Valenciana Innovaciò with grant number INNEST/2021/164.

Appendix A. Comparative with existing works

See Table A.5.

Appendix B. Data model

See Fig. B.11.

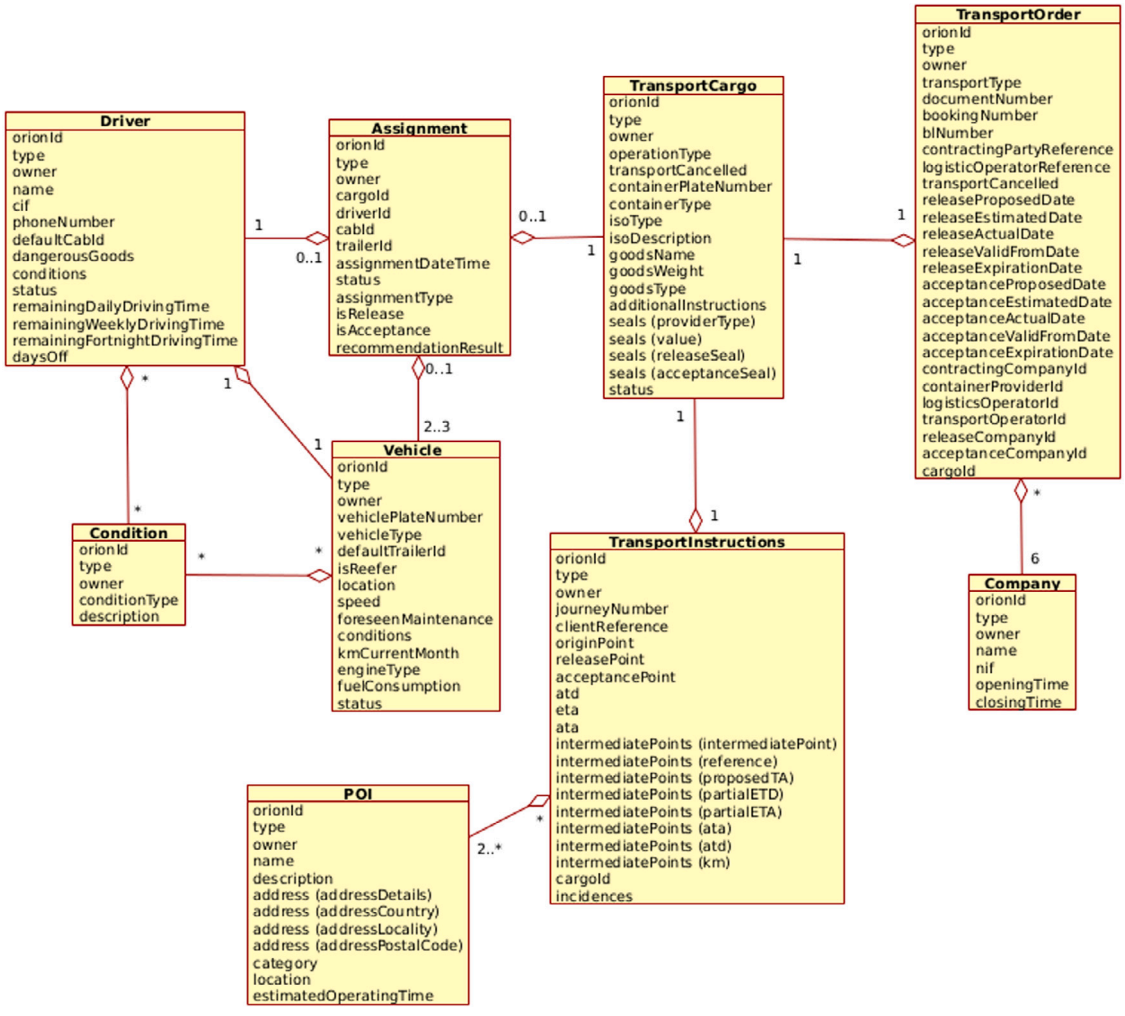

Fig. B.11. Class diagram.

Table A.5
Comparison of the proposed solution with the existing related works.

Publication (year)	Industry	Implemented	Blockchain technology	Storage	Security model	Method to link physical object to digital model	Aim of the DLT integration
Dietz, Putz and Pernul [11] (2019)	Manufacturing	No	Permissioned (not specified)	Sensor data stored off-chain. Hash stored on-chain. DT specifications stored on-chain.	Blockchain identities. Smart contracts for role-based authorisation (RBAC). Access control lists (ACL) stored with each key–value pair.	A device agent coordinates the devices with the system. Smart contracts for data synchronisation and interaction with physical devices.	Secure DT data sharing
ManuChain [12] (2020)	Manufacturing	Prototype	Permissioned (Hyperledger Fabric)	DT data stored off-chain. Events stored on-chain.	Blockchain identities. RBAC	IIoT infrastructure combined with Blockchain. Each physical entity has multiple virtual copies.	Decentralised self-organisation
Makerchain [13] (2019)	Manufacturing	No	Permissioned (not specified)	DT data stored off-chain. Hash stored on-chain. Events stored on-chain.	Blockchain identities.	Descriptive models for smart machines and services. Smart contracts for synchronisation.	Decentralisation, security, and trust
MBCoT [14] (2020)	Manufacturing	Prototype	Permissioned (Hyperledger Fabric)	States of objects and transactions stored on-chain. Current state stored off-chain.	Blockchain identities. RBAC	Smart contracts	Decentralisation, security, and traceability
Huang et al. [15] (2020)	Manufacturing	No	Consortium (not specified)	Product status updates stored on-chain	Each participant has a Blockchain node. Access control and encryption.	DT data shared using Blockchain. Sensor data and smart contracts for synchronisation.	Product life cycle management
Hasan et al. [16] (2020)	Manufacturing	Open-source prototype of the smart contracts	Permissioned Ethereum (Hyperledger Besu).	DT data stored off-chain. Hash stored on-chain.	Ethereum identities. Access control. Transactions signed.	Steps in the process registered on-chain. Smart contracts to interact with on-chain resources.	Decentralisation, security, and traceability
Hemdan and Mahmoud [17] (2021)	Manufacturing	No	Not specified.	DT activity stored on-chain	Not specified.	Activity of the DT logged as transactions	Protect the link between physical and virtual assets
EtherTwin [18] (2021)	Manufacturing	Open-source prototype	Ethereum.	Sensor data stored off-chain. DT metadata and status stored on-chain.	Blockchain identities. Hybrid access control combining RBAC and ABAC.	Bi-directional communication interface for synchronisation. Distributed application to enable data sharing.	Secure and decentralised data sharing
Salim et al. [19] (2022)	Manufacturing	Prototype	Custom private Blockchain.	Relevant data stored on-chain. Temporary off-chain DT data storage.	DTs and Packet Auditor registered in the Blockchain. Smart contracts for authentication. Block validation by security vendor	Synchronisation through smart contracts. Packet Auditor to facilitate data synchronisation.	Security and data integrity

(*continued on next page*)

Table A.5 (*continued*).

Lee et al. [20] (2021)	Construction	Prototype	Quorum (private Ethereum Blockchain).	On-chain	Participants log into Azure Active Directory. More than 51% of the nodes must accept the new block. Nodes sharing incorrect blocks are excluded.	DT combines building information modelling and sensor data. DT generates compliance statement and shares it on Blockchain.	Secure DT data sharing and traceability
Hunhevicz, Motie and Hall [21] (2022)	Construction	Open-source prototype	Ethereum Prototype tested using Ganache and Truffle.	DT data stored off-chain. Relevant data stored on-chain.	Blockchain identities. RBAC Oracles connect DTs and stakeholders with the Blockchain. DTs' transactions signed by oracle.	Siemens building twin platform.	Automatic performance evaluation and digital payments
Sahal et al. [22] (2021)	Transportation	No	Not specified	On-chain	Blockchain security mechanisms.	Blockchain synchronises DTs' status within the system. Transactions generated when data is transferred between physical and digital parts.	Decentralised DT collaboration
Sahal et al. [23] (2022)	Medicine	No	Not specified	On-chain	Blockchain security mechanisms.	Blockchain synchronises the DT's status within the system.	Decentralised DT collaboration
Cuñat et al.	Transportation (Smart Logistics)	Prototype implementation using open-source software	Ethereum (private or public). Canis Major	DT data stored off-chain. Relevant events stored on-chain.	User tokens and RBAC, implemented using Keycloak and KrakenD.	Synchronisation via Orion Context Broker, Composite Modeller and Rules Engine.	Traceability and verifiability of relevant events

References

[1] M. Grieves, J. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems, Transdiscip. Perspect. Complex Syst.: New Find. Approaches (2016) 85–113.

[2] E.H. Glaessgen, D.S. Stargel, The digital twin paradigm for future NASA and U.S. Air force vehicles, in: Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 2012, http://dx.doi.org/10.2514/6.2012-1818, URL https://arc.aiaa.org/doi/10.2514/6.2012-1818.

[3] S. Evans, C. Savian, A. Burns, C. Cooper, Digital twins for the built environment: An introduction to the opportunities, benefits, challenges and risks, Built Environ. News (2019).

[4] Gemelos digitales en logística: cómo moldearán el futuro de la industria? - SKU logistics, 2023, URL https://skulogistics.com/gemelos-digitales-logistica/, Accessed on 25.07.2023.

[5] Digital transformation in transportation and logistics, 2022, URL https://www.i-scoop.eu/digital-transformation-/transportation-logistics-supply-chain-management/.

[6] M. Mashaly, Connecting the Twins: A review on digital twin technology and its networking requirements, Procedia Comput. Sci. 184 (2021) 299–305, http://dx.doi.org/10.1016/j.procs.2021.03.039, URL https://www.sciencedirect.com/science/article/pii/S1877050921006694, The 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[7] S. Suhail, R. Hussain, R. Jurdak, A. Oracevic, K. Salah, C.S. Hong, R. Matulevičius, Blockchain-based digital twins: Research trends, issues, and future challenges, ACM Comput. Surv. 54 (2022) 1–34, http://dx.doi.org/10.1145/3517189, URL https://dl.acm.org/doi/10.1145/3517189.

[8] T. Alam, A reliable communication framework and its use in internet of things (IoT), in: CSEIT1835111 — Received, Vol. 10, 2018, pp. 450–456.

[9] M. Kosacka-Olejnik, M. Kostrzewski, M. Marczewska, B. Mrówczyńska, P. Pawlewski, How digital twin concept supports internal transport systems?—Literature review, Energies 14 (2021) http://dx.doi.org/10.3390/en14164919.

[10] H. Yao, D. Wang, M. Su, Y. Qi, Application of digital twins in port system, J. Phys. Conf. Ser. 1846 (2021) 012008, http://dx.doi.org/10.1088/1742-6596/1846/1/012008, URL https://iopscience.iop.org/article/10.1088/1742-6596/1846/1/012008.

[11] M. Dietz, B. Putz, G. Pernul, A distributed ledger approach to digital twin secure data sharing, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), in: LNCS, vol. 11559 (2019) 281–300.

[12] J. Leng, D. Yan, Q. Liu, K. Xu, J.L. Zhao, R. Shi, L. Wei, D. Zhang, X. Chen, ManuChain: Combining permissioned blockchain with a Holistic Optimization model as Bi-level intelligence for smart manufacturing, IEEE Trans. Syst. Man Cybern.: Syst. 50 (2020) 182–192, http://dx.doi.org/10.1109/TSMC.2019.2930418.

[13] J. Leng, P. Jiang, K. Xu, Q. Liu, J.L. Zhao, Y. Bian, R. Shi, Makerchain: A blockchain with chemical signature for self-organizing process in social manufacturing, J. Clean. Prod. 234 (2019) 767–778, http://dx.doi.org/10.1016/J.JCLEPRO.2019.06.265.

[14] C. Zhang, G. Zhou, H. Li, Y. Cao, Manufacturing blockchain of things for the configuration of a data- and knowledge-driven digital twin manufacturing cell, IEEE Internet Things J. 7 (2020) 11884–11894, http://dx.doi.org/10.1109/JIOT.2020.3005729.

[15] S. Huang, G. Wang, Y. Yan, X. Fang, Blockchain-based data management for digital twin of product, J. Manuf. Syst. 54 (2020) 361–371, http://dx.doi.org/10.1016/J.JMSY.2020.01.009.

[16] H.R. Hasan, K. Salah, R. Jayaraman, M. Omar, I. Yaqoob, S. Pesic, T. Taylor, D. Boscovic, A blockchain-based approach for the creation of digital twins, IEEE Access 8 (2020) 34113–34126, http://dx.doi.org/10.1109/ACCESS.2020.2974810.

[17] E.E.D. Hemdan, A.S.A. Mahmoud, BlockTwins: A blockchain-based digital twins framework, in: EAI/Springer Innovations in Communication and Computing, Springer Science and Business Media Deutschland GmbH, 2021, pp. 177–186.

[18] B. Putz, M. Dietz, P. Empl, G. Pernul, EtherTwin: Blockchain-based secure digital twin information management, Inf. Process. Manage. 58 (2021) 102425, http://dx.doi.org/10.1016/J.IPM.2020.102425.

[19] M.M. Salim, A.K. Comivi, T. Nurbek, H. Park, J.H. Park, A blockchain-enabled secure digital twin framework for early botnet detection in IIoT environment, Sensors 22 (2022) 6133, http://dx.doi.org/10.3390/S22166133, 2022, Vol. 22, Page 6133, URL https://www.mdpi.com/1424-8220/22/16/6133/htmhttps://www.mdpi.com/1424-8220/22/16/6133.

[20] D. Lee, S.H. Lee, N. Masoud, M.S. Krishnan, V.C. Li, Integrated digital twin and blockchain framework to support accountable information sharing in construction projects, Autom. Constr. 127 (2021) 103688, http://dx.doi.org/10.1016/J.AUTCON.2021.103688.

[21] J.J. Hunhevicz, M. Motie, D.M. Hall, Digital building twins and blockchain for performance-based (smart) contracts, Autom. Constr. 133 (2022) 103981, http://dx.doi.org/10.1016/J.AUTCON.2021.103981.

[22] R. Sahal, S.H. Alsamhi, K.N. Brown, D. O'shea, C. McCarthy, M. Guizani, Blockchain-empowered digital twins collaboration: Smart transportation use case, Machines 9 (2021) http://dx.doi.org/10.3390/MACHINES9090193.

[23] R. Sahal, S.H. Alsamhi, K.N. Brown, D. O'Shea, B. Alouffi, Blockchain-based digital twins collaboration for smart pandemic alerting: Decentralized COVID-19 pandemic alerting use case, Comput. Intell. Neurosci. 2022 (2022) http://dx.doi.org/10.1155/2022/7786441.

[24] R. Bambura, M. Šolc, M. Dado, L. Kotek, Implementation of digital twin for engine block manufacturing processes, Appl. Sci. 10 (2020) 6578, http://dx.doi.org/10.3390/APP10186578, 2020, Vol. 10, Page 6578, URL https://www.mdpi.com/2076-3417/10/18/6578/htmhttps://www.mdpi.com/2076-3417/10/18/6578.

[25] UN/CEFACT, White Paper Smart Containers Real-time Smart Container data for supply chain excellence, Tech. Rep., 2020, www.unece.org/cefact.

[26] J.S. Sara Saberi, L. Shen, Blockchain technology and its relationships to sustainable supply chain management, Int. J. Prod. Res. 57 (2019) 2117–2135, http://dx.doi.org/10.1080/00207543.2018.1533261, URL https://www.tandfonline.com/doi/full/10.1080/00207543.2018.1533261.

[27] R. Belchior, B. Putz, G. Pernul, M. Correia, A. Vasconcelos, S. Guerreiro, SSIBAC: Self-sovereign identity based access control, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2020, pp. 1935–1943, http://dx.doi.org/10.1109/TrustCom50675.2020.00264.

[28] FIWARE context broker launches as a CEF building block – FIWARE, 2018, URL https://www.fiware.org/2018/08/08/fiware-context--broker-launches-as-a-cef-building-block/, Accessed on 25.07.2023.

[29] S. Loss, H.P. Singh, N. Cacho, F. Lopes, Using FIWARE and blockchain in smart cities solutions, Cluster Comput. (2022) 1–14, http://dx.doi.org/10.1007/S10586-022-03732-X/FIGURES/9, URL https://link.springer.com/article/10.1007-/s10586-022-03732-x.

[30] FIWARE, Loadtest for FIWARE components, 2021, URL https://github.com/FIWARE/load-tests, Accessed on 30.06.2023.

Salvador Cuñat Negueroles

Raúl Reinosa Simón

Matilde Julián

Andreu Belsa

Ignacio Lacalle

Carlos E. Palau