Journal of Network and Computer Applications 225 (2024) 103857

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Research paper

The convergence of Digital Twins and Distributed Ledger Technologies: A systematic literature review and an architectural proposal

Alessandra Somma [a,*], Alessandra De Benedictis [a], Christiancarmine Esposito [b], Nicola Mazzocca [a]

[a] *University of Naples Federico II, Via Claudio, 21, Naples, 80125, Italy*
[b] *University of Salerno, Via Giovanni Paolo II, 132, Fisciano (SA), 84084, Italy*

ARTICLE INFO

Keywords:
Digital Twins
Data security
Data trustworthiness
Distributed Ledger Technologies
Blockchain
Reference architecture

ABSTRACT

In recent years, the emerging Digital Twin (DT) technology is playing a key role in fostering the transition towards the Industry 4.0. DTs, representing virtual replicas of physical objects, products or processes established thanks to a bidirectional continuous flow of information between the physical and the virtual world, are currently adopted in multiple domains such as manufacturing, aerospace, automotive, energy, construction, smart cities and smart mobility, etc.. DTs live together with the physical system they replicate, receiving the same data and often triggering specific control actions that directly impact on the real system status.

When dealing with DTs of complex Cyber-Physical Systems (CPSs), the data sources may be heterogeneous and untrustworthy, which requires the DT to be able to address security issues related to data in transit from/to the physical twin and data at rest. A possible way to address these data security issues consists in adopting Distributed Ledger Technologies (DLTs) which provide several security guarantees on data through cryptographic hashing techniques.

In this work, we present a three-fold contribution: (i) we discuss the results of a Systematic Literature Review (SLR) on the state-of-the-art of the research related to the integration between DLT and Digital Twins, aimed at clarifying relevant aspects such as, among others, what is the favourite DLT choice in existing proposals or what is the type of DT-related information to store on-chain; (ii) leveraging the SLR results and other related research, we propose an architectural framework for the integration of DT and DLTs (more specifically, blockchains); (iii) we validate the proposed architecture by means of two proof-of-concept implementations leveraging different technological stacks and taking into account two different operational scenarios. Finally, we conduct a requirements coverage analysis and compare the PoCs through a coverage matrix.

1. Introduction

In the era of Industry 4.0 (I4.0), several industrial sectors are experiencing a digital transformation that is making production processes smarter and more efficient. The **Digital Twin** (DT) is one of the I4.0 emerging technologies, represented by a virtual replica of a physical asset/process characterized by the seamless bi-directional communication between the digital and real worlds enabled by the real-time data exchange (Pires et al., 2019; van der Valk et al., 2022; Singh et al., 2021). The DT incorporates multidisciplinary knowledge from different models such as geometrical, physical, behavioural and rule-based ones and upon the DT technology several services, like monitoring, prediction, optimization, control and so on, can be implemented (Tao et al., 2022).

The deployment of digital twins involves the cooperation between different technologies, *e.g.*, virtualization, Internet of Things (IoT), Artificial Intelligence (AI), and heavily relies on a huge amount of data coming from the physical world or on the information provided for example by domain experts or human operators in order to faithfully replicate the behaviour of the physical twin (Singh et al., 2021; Farahani et al., 2021). For this reason, the DT paradigm still suffers from several security problems which require further investigation (Alcaraz and Lopez, 2022), such as data tampering that arises when an attacker damages data consistency.

Leveraging on data obtained from heterogeneous and often untrustworthy data sources, simulation and other services built upon DTs can lead to ineffective and potentially harmful decision-making, particularly when the PT receives malicious changes previously made

[*] Corresponding author.
E-mail addresses: alessandra.somma@unina.it (A. Somma), alessandra.debenedictis@unina.it (A.D. Benedictis), esposito@unisa.it (C. Esposito), nimazzoc@unina.it (N. Mazzocca).

https://doi.org/10.1016/j.jnca.2024.103857
Received 11 September 2023; Received in revised form 9 February 2024; Accepted 28 February 2024
Available online 1 March 2024

to the DT. Thanks to properties such as immutability, transparency and auditability, **Distributed Ledger Technologies** (DLTs) play an important role in securing the data used by a DT when both in transit and at rest, besides ensuring the trustworthiness of sources (Vilas-Boas et al., 2022; Yaqoob et al., 2020; Farahani et al., 2021). In fact, through cryptographic mechanisms and in particular hashing techniques, DLTs build a tamper-proof distributed ledger in which DT data, such as models and/or critical information, can be stored.

Despite the growing interest in the possible integration of DTs and DLTs, there is still a lack of understanding about how to implement the combination of these two technologies. In fact, in their review, Li and Kassem (2021) pointed out the need for more researches on DLT and smart contract integration with DTs because of the reduced number of studies that focus their attention on this topic. In addition to the issues related to the identification of the best DLT solution to use based on existing quality requirements (in terms of unmodifiability properties for example), to the need for an (expensive) infrastructure to build and manage DLTs, and to the impact on performance, the lack of a clear DT software architecture (as pointed out by Ferko et al., 2022; De Benedictis et al., 2022) where DLTs can be inserted makes the integration not straightforward.

In light of the above, the contribution of this paper is three-fold:

- we conduct a Systematic Literature Review (SLR) on the adoption of DLTs in DTs. In contrast to existing SLRs, we answer to 7 research questions whose aim is to understand not only the benefits and challenges of including DLTs in a DTs environment, but also what are the available implementations, their functional and non-functional requirements, architectures and evaluation metrics. Moreover, we investigate different kinds of DLTs, such as blockchain and tangle, and thus determine which is currently the best choice with respect to DTs requirements;
- as we find out that there are only few implementations of DTs and DLTs integration and none of these solutions specify an architectural framework, we propose a generalized architecture that can be used as a reference for the realization of virtual replicas that leverage on DLTs to cope with DT data security issues;
- in order to demonstrate the validity of our architecture proposal for the integration of DTs and blockchains, we provide two Proof-of-Concepts (PoCs) implementations of the proposed architecture related to two different operation scenarios and leveraging two different technological stacks. Finally, we evaluate the two PoCs with respect to the functional and non-functional requirements identified in the conducted SLR. Through a coverage matrix, we analyse their strengths and their weaknesses and discuss our future work.

The paper is organized as follows. Section 2 provides an overview of the main concepts behind the DT and its data security issues and DLTs. Section 3 provides the systematic review of the state-of-the-art of the DT and DLT literature. Section 4 discusses a general reference architecture for DTs systems that integrate DLTs. Finally, Section 5 illustrates two PoCs implementation based on the proposed architecture to prove its validity and Section 6 discusses the two PoCs from a qualitative point of view with respect to functional and non-functional requirements and our future work. To conclude, Section 7 draws our conclusions.

2. Background

2.1. Digital Twins and data security issues

Although the name "*Digital Twin*" was firstly introduced in 2010 by NASA's John Vickers, DT technology raised in the early '60s when NASA used basic notions of twinning for space programming. Later on, in 2002 Grieves conceived the early DT concept as a support for the activity of Product Lifecycle Management (PLM), which included elements as real space, virtual space and the information flow between real and virtual space.

Even if there is no yet consensus on the DT definition because of the tight coupling between the DT and its application domain, Qi et al. (2021) formalized the DT as the following quintuple:

$$M_{DT} = (PS, VS, Ss, DD, CN) \tag{1}$$

where (i) real entities, and their internal and external interactions characterize the *Physical Space* (PS) (Tao et al., 2022); (ii) the *Virtual Space* (VS) consists of digital replicas, fed with real-time data obtained from the physical world and historical data; (iii) the *DT Data* (DD) are the so-called "Digital Twin fuel" and can be data obtained from the PS, the VS, but also data generated from DT-based services and/or domain expert knowledge; (iv) the *Services* (Ss) that leverage upon DT technology are simulation, real-time monitoring, control, optimization, prediction of future states. Moreover, DT has been proposed as a solution for the enhancement of Cyber-Physical Systems (CPSs) security as it can provide security testing and/or intrusion detection capabilities (Eckhart and Ekelhart, 2019), but can also allow system maintainers to evaluate critical infrastructures' operational behaviour and security without affecting live systems up-time (Suhail et al., 2022b). (v) Finally, there are *connections* (CN) that are responsible for enabling the communication between the four parties.

The deployment of a DT involves the composition of technologies such as virtualization, IoT, AI, Big Data analytics and CPSs. Together with the continuous interaction established with the physical twin, the confluence of these technologies leads to several security problems which still require further investigation, as argued in Alcaraz and Lopez (2022). The same paper proposes a taxonomy of the security threats to which a DT system is exposed which comprises *data tampering*, *knowledge tampering* and *representation tampering*. The *data tampering* security issue arises when potential adversaries damage data consistency in terms of fidelity and granularity due to poorly specified access control mechanisms. Similarly, the *knowledge tampering* involves the unauthorized modification of data providing a more detailed understanding of reality which can result in imprecise predictions. *Representation tampering* is a direct consequence of the two previous ones that inevitably affect the final data representation to end users. Finally, DTs activities leverage on data coming from heterogeneous untrustworthy data sources and this may lead to ineffective and potentially harmful decision-making, especially when malicious changes made to the DT are propagated to the physical system (Eckhart and Ekelhart, 2019).

As demonstrated by the existing research work analysed in this paper, DLTs play an important role in securing the data used by a DT to perform its elaborations when both in transit and at rest, besides ensuring the trustworthiness of sources. In the following subsection we will provide some background on DLTs before going into the details of our literature review.

2.2. Distributed Ledger Technologies

Distributed Ledger Technology is the enabler of the realization and operation of distributed ledgers, i.e., a type of distributed database that assumes the presence of malicious nodes and in which data can only be appended or read. Through a shared consensus mechanism, benign nodes agree on an immutable record of transactions and maintain a full copy of the ledger (Sunyaev, 2020). To ensure security, a cryptography algorithm is used to link multiple copies of each record together with a cryptography algorithm (Sunyaev, 2020; Soltani et al., 2022). DLT can be divided in five sub-technologies, i.e., blockchain, tangle, hashgraph, side-chains and holochain that differ in the manner in which they retain data and the consensus algorithm they use. In this work, we will focus on blockchain and tangle technologies.

The *blockchain* technology is a DLT in which the distributed ledger is structured as a chain of numbered blocks, that are made of an array of transactions and are connected by means of cryptographic hashes: each block maintains the hash code of the previous one; in this way, if a block attached to the ledger is modified after it has been added to the chain, the unlawful change can be immediately detected by calculating the hash code of the modified block and by comparing it with the hash code stored in its next block (Sunyaev, 2020). This technology has three main properties: *transparency*, since blockchain transactions can be tracked; *immutability*, because data are recorded in blockchains in different encrypted copies and linked to previous data, and *decentralization* (Soltani et al., 2022). However, the inherent topology of the blockchain allows to append only one block per time, thus strongly affecting the *scalability* of the ledger and resulting in a fewer number of confirmed transactions per second and high transaction latency when the number of transactions increases (Zhou et al., 2020; Xie et al., 2019).

Moreover, the blockchain has many security issues, most of which are related to the power held by miners (i.e., nodes that participate in the mining process, namely the generation of blocks for a reward such as the cryptocurrency Bitcoin) (Sunyaev, 2020; Soltani et al., 2022). Finally, since the blockchain leverages on public–private key cryptography and cryptographic hashing, quantum computing could endanger the integrity and security of blockchain systems since it makes many NP problems easier to solve (Schärer and Comuzzi, 2023). To solve these problems, in particular to tackle blockchain scalability issues, the *tangle* technology has been proposed, which is a decentralized encrypted network for recording data in a scalable and secure manner (Soltani et al., 2022). In this network, the transactions are directly connected with one another in a scheme structured as a *Directed Acyclic Graph* (DAG). When a new transaction arrives, it is in charge of validating other two transactions; if these two transactions are judged invalid, the new transaction will also be considered invalid. The great advantage of this solution is that more than one transaction can be appended to the network in parallel resulting in higher scalability and throughput improvement.

3. DTs and DLTs: a systematic literature review

In this section, we present the systematic literature review that we conducted according to Kitchenham guidelines (Kitchenham, 2004) to investigate the overall vision of the state-of-the-art research about DTs, the potential application of DLTs to secure DTs. The methodology consists in three high-level phases: (i) *planning* the review to determine if such a review is required answering the question "*is there really a need for a literature review in this area?*"; (ii) *conducting* the review made up with five sub-phases that will be explained in the reminder of this section; (iii) *reporting* the review that consists in communicating the analysis process' results.

3.1. Planning the review

The first SLR's phase requires to firstly identify whether the study is required and, if yes, to conduct the review according to a precise protocol, namely defining the search strategy including search terms, used resources (*e.g.*, digital libraries), inclusion and exclusion criteria based on the pre-defined research questions and the study quality assessment.

3.1.1. Identification of the need for a review

Li and Kassem (2021) presented a comprehensive view of the body of literature available on DLT and smart contracts in the construction sector. However, the authors pointed out the need for more researches on DLT and smart contract integration with other technological systems such as DT, since only few studies focus their attention on this topic. Vilas-Boas et al. (2022) argued that the DTs and DLTs integration has the potential to improve several processes' efficiency. The authors supported the idea of DLT as a tool for DT data management, providing secure data storage and sharing, avoiding man-in-the-middle attacks and tampering. Thus, it is worth to better investigate the adoption of the DLT in DT-based systems, facing the challenges that emerge from this integration. Instead, Yaqoob et al. (2020) focused on how blockchain can be exploited to make DTs more effective in addressing industrial problems. However, they stated that there are still challenging issues that hinder the successful implementation and deployment of blockchain technology in DTs in the industrial context, such as scalability, data privacy, interoperability, energy consumption and integration with legacy systems. Therefore, they suggest carrying out further researches to address these issues. Finally, apart from underlying the key benefits of using blockchain-based DT, Suhail et al. (2022a) reviewed the state-of-the-art trying to understand the position of AI in a possible DT and blockchain integration and proposed a 7D model for trustworthy DT empowered with blockchain without providing any implementation.

Despite there are different DLTs and DTs integration state-of-the-art studies, literature main focus is on blockchain technology and its potential for securing DTs, not the complete spectrum of Distributed Ledger Technologies. Moreover, as previously said, the puropose of these reviews is to determine if blockchain usage in DTs is worthwhile or not, and occasionally to analyse blockchain shortcomings in DT environments. In our literature analysis, we consider all DLTs, investigating not only if and when it makes sense integrate a DLT in a DT ecosystem, but also taking into account aspects such as requirements, software architectures and evaluation metrics. Furthermore, leveraging on the results obtained from our SLR, we propose an high-level *domain-agnostic* architectural model that combines DLTs and DTs. Finally, to prove the feasability of the proposal, we instantiate the architecture in two PoCs.

3.1.2. Research questions

The Research Questions (RQs) that we aim to answer through our literature review are the following:

1. **RQ1**: is there any solution which integrates DLTs into a DTs? If yes, is any implementation provided?
2. **RQ2** (*where*): which are the application domains and the use cases in which these technologies are jointly used?
3. **RQ3** (*why*): what are the benefits and thus the reasons that justify this integration?
4. **RQ4** (*which*): which is, so far, the preferred choice between blockchain and tangle?
    - **RQ4.a**: if blockchain is considered the best choice, does the manuscript deal with blockchain scalability and quantum immunity issues? If yes, how?
5. **RQ5** (*what*): what kind of data or information is stored in a DLT when used in the context of a Digital Twin solution?
6. **RQ6** (*how*): which are requirements, software architectures (and if mentioned, design patterns) and evaluation metrics of a Digital Twin that uses a DLT?
7. **RQ7**: which are the main challenges to address for implementing the DTs and DLTs integration?

3.1.3. Data sources and search strategy

Search terms choice consists in deciding the set of keywords to be used for researching papers on digital libraries. Since it is the backbone phase of the process, choosing incorrect terms can compromise the whole literature review because of the risk to not include all the relevant studies. Even if the integration of DLTs and DTs is a hot topic in the literature, there are still few publications, that is why in our study we considered all possible combinations of terms and structured the research query as follows:

(*Digital* OR *Virtual*) AND (*Twin* OR *Replica*) AND

(*Blockchain* OR *Block-chain* OR *DLT* OR (*Distributed*
AND *Ledger* AND *Technologies*))

The actual search was performed in the period between December 2022 and March 2023 and was conducted in different databases in order to not polarize the review, *i.e.*, Scopus, Web of Science (WoS) and IEEE Digital Library.

3.1.4. Study selection

Usually the amount of studies obtained from databases search is too big to be included in a review process and since many of them are not relevant to the purpose of the literature study (*e.g.*, some studies can be included simply because they mention the research query's keyword), a filtering step is required to obtain a set of strictly appropriate included studies. In fact, we formulated the following set of inclusion/exclusion criteria, thus the studies obtained from the first research are thereafter filtered by inspecting the content of their titles and abstracts:

- Exclusion Criterion 1 (**EC1**): exclude all the studies not available in full text.
- Exclusion criterion 2 (**EC2**): exclude all the studies that are not written in English.
- Exclusion criterion 3 (**EC3**): exclude all the studies that are not primary.
- Exclusion criterion 4 (**EC4**): exclude all the studies that do not contain at least one of the following keywords, *i.e.*, Digital Twin; blockchain technology; blockchain; block-chain; Distributed Ledger Technology.
- Exclusion criterion 5 (**EC5**): exclude all the studies published before 2017, as DT concept began to catch on that year (Qi et al., 2021).
- Inclusion criterion 1 (**IC1**): include only studies published in journal and/or conferences related to computer science and engineering.

3.1.5. Study quality assessment

The process of study quality assessment aims at setting a minimum quality threshold for primary studies to be included in the review, in terms of level of evidence brought by the study. As stated in Kitchenham (2004), both primary studies with high and low quality scores will be accepted. The only viable quality-based exclusion criteria would consist in excluding primary studies in which evidences are obtained from expert opinion based on theory or consensus, when the number of high quality studies exceeds them. This is not our case because all the selected primary studies bring evidences which are not exclusively based on expert opinion, thus this phase is skipped.

3.1.6. Data extraction

This phase enables the gathering of information and insights from primary studies in order to answer the research questions by means of data extraction forms. For this review, we designed and developed an extraction format answering research questions in order to obtain useful information from the selected studies regarding, for instance, the type of DLT employed in the study or if any implementation is provided.

3.1.7. Data synthesis

Finally, collected data were summarized and evidences obtained from the selected studies are presented by means of tables and also in a qualitative manner, by using narrative and semantic synthesis.

3.2. Conducting the review

By feeding the aforementioned databases with the search query, Scopus returned 3943 studies, WoS returned 252 studies and IEEE Digital Library returned 119 studies as shown in Fig. 1. By applying the first exclusion criteria to the full list of studies returned by Scopus, the amount significantly decreases to 1863 papers, instead the EC2 application does not change the number of studies returned by Scopus. EC3 decreases paper numbers to 1466 and finally the number of manuscripts is greatly reduced to 436 as result of EC4. EC5 has no effect on Scopus results, while IC1 brings to 380 the number of studies to be taken into account. When it comes to Web Of Science, the total number of studies which complies to all the exclusion and inclusion criteria is equal to 62. As for IEEE Digital Library, the application of all filters returned 13 studies. The studies obtained from different databases are merged into one list being careful to include the studies which appear in multiple lists only once and then another filtering process based on titles and abstracts is performed. Finally, 61 are the manuscripts that we analysed in order to extract the needed information to answer the RQs. The document that lists, classifies and comments the selected papers is publicly available at the link https://docs.google.com/spreadsheets/d/1poURtE2jkeMo_ZlTPDO_0mIxOVsdCuwo/edit?usp=sharing&ouid=110130922453565990010&rtpof=true&sd=true.

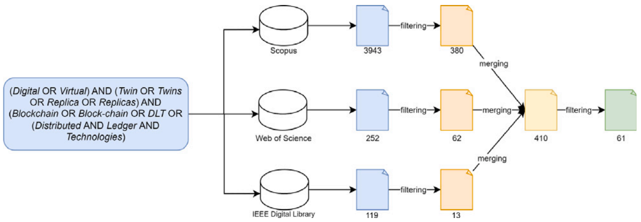

(Digital OR Virtual) AND (Twin OR Twins OR Replica OR Replicas) AND (Blockchain OR Block-chain OR DLT OR (Distributed AND Ledger AND Technologies)
Scopus
Web of Science
IEEE Digital Library
3943
252
119
filtering
380
62
13
merging
410
filtering
61

Fig. 1. Conducting the review according to planning' sub-phases.

3.3. Reporting the review

In this section we present the results of the literature review in a discursive form and their comprehension is supported by tables.

3.3.1. RQ1: is there any solution which integrates DLTs into a DTs? if yes, is any implementation provided?

All the selected studies deal with DLTs (and mostly blockchain) and DTs integration and multiple solutions at different levels of complexity and in different contexts are proposed. For instance, Song and Hong (2021) propose a more efficient city management leveraging the set-up of a virtual city model powered by real operational data and, in order to ensure data integrity, confidentiality and availability, authors propose a blockchain-based authentication mechanism. Celik et al. (2021) present the possibility to combine blockchain with Building Information Modelling (BIM) methods in order to develop a DT for construction projects: the main idea is to store every life-cycle step of a construction project in a blockchain through smart contracts establishing a secure environment among participants. In the healthcare context, Lu et al. (2021) propose an inter-hospital resilient network for pandemic response based on blockchain and "Dynamic Digital Twins" of several independent hospitals. Since the patient data are strictly confidential, the blockchain is proposed as a possible solution to address the challenge of establishing a secure communication among hospitals, integrating patients information and medical resource.

However, only 57% of the selected studies (35 over 61) provides an implementation and the majority of proposed solutions is not complete and ranges from simple PoCs to holistic prototypes. For example, Pittaras et al. (2022) integrate W3C's Web of Things (WoT) standards with DLTs enabling the creation of physical devices and/or CPS virtual replicas through smart contracts, but the proposed solution for evaluation implements a proof-of-concept which emulates two IoT devices, one smart lamp and one smart coffee machine. Instead, Li et al. (2021) propose a blockchain-based DT sharing platform in order to support social manufacturing, supported by a prototypical implementation of DTs of three different 3D printers and two smart warehousing units that shows how heterogeneous social manufacturing resources are rapidly integrated through this sharing solution.

3.3.2. RQ2 (where): which are the application domains and the use cases in which these technologies are jointly used?

Among the selected studies, 18 manuscripts apply the DLTs and DTs integration in I4.0 and Smart Manufacturing in order to track the production process (i.e., supply chain), thus allowing consumers to trace back products' origin proving their authenticity and identity (Li et al., 2021). Usually, information generated during maintenance and operation phases are stored in blockchain for purposes of anomaly/intrusion detection, predictive maintenance, failure prediction, and optimization of the production process. For instance, Huang et al. (2020) face Digital Twin data management (data storage, access, sharing, authenticity and tracking) problems with blockchain: they propose to store every change of life-cycle product data through smart contracts mechanism. Moreover, the public-key encryption allows to establish an ownership of the product and its life-cycle data that will be owned by every contributor to the product life-cycle. Other works that belong to I4.0 application field are (Putz et al., 2021; Zhang et al., 2020; Manoharan et al., 2022; Cohen et al., 2021).

Nine studies implement different use cases in Smart City/Smart Mobility such as traffic monitoring and prediction, simulation of disruption events and subsequent rapid maintenance, real-time control. For instance, Chai et al. (2021) develop a secure and privacy-preserving authentication scheme to enable intelligent traffic service, while Salim et al. (2022) aim at achieving early botnet detection in the IIoT environment for Smart Cities. Another example, proposed by Song and Hong (2021), is the layered smart city model based on blockchain to store all the energy-related information.

Application domains are also the construction industry, healthcare and economy sector, while less often DTs and DLTs are combined in others context, such as smart grids, smart society, networks and so on. The distribution of papers in these categories can be found in Table 1. Finally, twelve studies do not concern to any specific domain. For example, Raj (2021) raises the problems of data breaching, leakage of confidential information, denial of access to the DT and corruption of DT data and proposes to address them simultaneously by creating a DT into a blockchain, thus all the changes are recorded immutably, turning the DT into a digital certificate of the physical product.

3.3.3. RQ3 (why): what are the benefits and thus the reasons that justify this integration?

Despite DLTs and DTs integration is applicable to different application domains, some common motivations that justify their joint use and a set of benefits can be identified in the selected papers. Clearly, reminding that DLTs are a sort of distributed storages, it would not make sense to employ a DLT in a state-less system or, more in general, when there is not the *need to store information*. In fact, every study included in the review involves the development of models, frameworks and prototypical solutions that *require to store a state*.

Moreover, even if the majority of trustability problems in the selected papers can be solved introducing a Trusted Third Part (TTP) in the DT ecosystem and thus regulating the interactions among untrusthworthy participants, another reason that drives the adoption of DLTs is the *need for disintermediation* from a TTP to further restrict the trusted base of the system as well as to achieve distributed governance. An example is Putz et al. work (Putz et al., 2021), which builds the EtherTwin DApp that "implements the complex DT sharing requirements of the Industry 4.0 landscape without the need for a TTP".

DTs heavily exploit data-driven models and AI to accurately mirror the behaviour of their physical counterpart and are powered by big data coming from an huge amount of heterogeneous IoT devices located in different places, especially when dealing with complex systems (Tao et al., 2022). All DT data sources can be considered as *multiple writers* in a distributed system that are usually *untrustworthy* and this is one *one of the main conditions in which the application of DLTs is mostly appropriate*. In fact, as proven by 53 papers, DLTs brilliantly support *the secure sharing of DT data* because they can preserve the integrity of stored information through cryptography techniques solving multiple untrustworthy participants that interact with the DT and among each other (e.g., Lu et al., 2021; Dietz et al., 2019; Salim et al., 2022; Huang et al., 2020). Moreover, 51 papers state that data sources can be made reliable through authentication mechanisms based on the DLTs (e.g., Akash and Ferdous, 2022; Cohen et al., 2021), and argue that DLTs can help into fulfil the traceability requirement that usually arises in a DT ecosystem (e.g., Dietz et al., 2019; Lopez et al., 2021).

Furthermore, almost all the studies state that introducing DLTs into their DTs allows to benefit from a *decentralized* and *distributed architecture*: for instance, Putz et al. (2021) achieve a fully decentralized application logic with front end code running in the users' browser and the back end running as a smart contract executed by all the blockchain nodes. To further enhance decentralization and improve robustness by reducing single points of failure, some papers leverage decentralized cloud storage and distributed file-storage systems (e.g., Teisserenc and Sepasgozar, 2022).

3.3.4. RQ4 (which): which is, so far, the preferred choice between blockchain and tangle? RQ4.a: if blockchain is considered the best choice, does the manuscript deal with blockchain scalability and quantum immunity issues? if yes, how?

Most of the selected studies (58 over 63) adopt blockchain in their solutions while DAG-structured DLTs are still poorly chosen. Among all, only Altun and Tavli (2019) study considers both Blockchains and DAG-chains as promising solutions for implementing their model which aims at establishing a secure communication between DTs for purposes of predictive maintenance in the manufacturing industry. In particular, the authors propose a new reference model which integrates DTs, DLTs and Fog Computing to address the issues of heterogeneity, scalability, constrained resources, mobility, security and privacy in the context of Smart Manufacturing.

Since nowadays blockchain seems to be the most common choice in a DT environment, we wondered if blockchain typical issues are taken into account. From our analysis, it has emerged that about the 60% of studies deals with the blockchain scalability issue, either by simply acknowledging its impact on the system or by providing less or more complex solutions that tackle its effects. For example, many works propose the adoption of an off-chain storage solution, like InterPlanetary File System (IPFS)[1] or Swarm,[2] or of Distributed Hash Tables (DHTs) that can be defined as distributed systems implementing a lookup service like hash tables. Only 5 works propose to use a scalable DLT as a solution to blockchain scalability issue, and it is the IOTA[3] one (Suhail et al., 2021; Stanke et al., 2020; Sahal et al., 2021; Sun et al., 2020; Altun and Tavli, 2019). Futhermore, 2 works (Suhail et al., 2021; Altun and Tavli, 2019) take into account the blockchain quantum immunity issue arguing that quantum computers will put blockchain security at risk by easily breaking the cryptographic primitives that blockchain heavily relies on. That is why they adopt a blockchain scheme that inherently supports postquantum cryptography, such as IOTA. Finally, 23 studies do not deal neither with the scalability issue nor with the quantum immunity issue.

[1] https://ipfs.tech/

[2] https://www.ethswarm.org/

[3] https://www.iota.org/

Table 1
Use cases of DTs and DLTs integration classified for application domains.

Application domains	Manusc.	Use cases
Industry 4.0 and Smart Manufacturing	Li et al. (2021), Putz et al. (2021), Zhang et al. (2020), Manoharan et al. (2022), Cohen et al. (2021), Suhail et al. (2021), Hemdan and Mahmoud (2021), Hasan et al. (2020), Shen et al. (2021), Shukla et al. (2021), Li et al. (2022), Lopez-Arevalo et al. (2021), Stanke et al. (2020), Sun et al. (2020), Nielsen et al. (2020), Altun and Tavli (2019), Pincheira et al. (2023), Huang et al. (2020)	Planning, monitoring, optimization control, predictive maintenance, anomaly detection, documenting assembly process, risk management, prediction of failures.
Smart City and Smart Mobility	Song and Hong (2021), Chai et al. (2021), Salim et al. (2022), Sahal et al. (2021), Liao et al. (2021), Liu et al. (2022), Nguyen et al. (2023), He et al. (2022), Takahashi et al. (2022)	Accurate situation prediction, real-time control, botnet detection, rapid/predictive maintenance, traffic monitoring and prediction.
Construction Industry	Celik et al. (2021), Hunhevicz et al. (2022), Celik et al. (2023), Teisserenc and Sepasgozar (2022), Yang et al. (2022), Teisserenc and Sepasgozar (2021a,b)	Building monitoring, predictive maintenance, information sharing, build management optimization, information management and secure payments.
Healthcare	Lu et al. (2021), Sahal et al. (2022a), Akash and Ferdous (2022), Sahal et al. (2022b), Firouzi et al. (2021), Asadi (2021)	Inter-hospital communication, prevention care, personalized care, rapid diagnosis, remote care, fitness and well-being monitoring, pandemic/medical alert, prediction, epidemic monitoring and control.
Economy	Obushnyi et al. (2019), Nabeeh et al. (2022), Huang et al. (2022)	Monitoring of economic systems, detection of deviations in operating modes, secure transfer of value.
Others (e.g., networking, power grids, smart society, etc.)	Khan et al. (2022), Kumar et al. (2022), Lu et al. (2020), Chen et al. (2022), Lopez et al. (2021), Dai et al. (2022)	Failure prediction, real-time cyber-security issues detection, control, remote testing, proactive maintenance, prediction, performance optimization of real devices, decision-making.
Not specified	Raj (2021), Wang et al. (2022), Suhail et al. (2022b), Qiao et al. (2022), Son et al. (2022), Qu et al. (2021), Kanak et al. (2019), Pittaras et al. (2022), Pittaras and Polyzos (2022), Kuruppuarachchi et al. (2023), Qu et al. (2022), Dietz et al. (2019)	No application domain and thus no use cases are discussed.

3.3.5. RQ5 (what): what kind of data is stored in a DLT-based Digital Twin solution?

The study that we conducted allowed us to identify 12 major categories in which it is possible to classify the data that can be stored in a solution that integrates DT and DLTs, i.e., DT model, DT state based on sensor data, parameters/configuration of the physical device, historical data, critical data, writers' identities/authentication data, Safety & Security (S&S) rules/access control policies, decisions/reports/simulation results, hash values, link to an off-chain storage, shared data and changes made to the DT.

Most of the reviewed studies do not store only one category of information on-chain, instead they usually store different information types. In fact, there is still a trend to store heavy information in blockchain like DT models (17 studies), DT state (19 studies) and static parameters and configuration of the physical device (13 studies) as well as a combination of them. Eleven studies store in a blockchain all the changes made to the DT, which can be useful when there is the need to maintain a trace of the DT history. For instance, Hasan et al. (2020) present a solution in which DTs are created and managed through a smart contract. Consequently, every interaction with the DT is stored on-chain, while the detailed information of the DT model and data are stored in a decentralized way using a Distributed File System (DFS). Instead, in 8 studies it is preferred to store critical data in blockchain, such as the tool wearing condition of Computer Numerical Control (CNC) machines in Shukla et al. (2021) for predictive maintenance purpose. Besides the aforementioned work, seven other studies choose to store identities/authentication data on-chain e.g., Salim et al. (2022), Chai et al. (2021).

Based on the fact that storing a great amount of data in a blockchain is not efficient at all, 8 studies decide to lighten the load to which blockchain is subject by storing only hash values of big data on-chain and the big data themselves in cloud or in a distributed storage. This is the case of Shen et al. (2021), where only hash codes of Big Digital Twin Data (BDTD) are stored on-chain while the original BDTD are encrypted and then stored in cloud. Finally, in some scenarios it is important to secure not only input data of the DT but also the outcomes of simulations, reports and decisions made. In this regard, the authors of Sahal et al. (2022b) propose to store reports and decisions made by government, health organization and hospitals regarding the restrictions to apply to contain the COVID-19 pandemic.

3.3.6. RQ6 (how): which are the requirements, software architectures (and if mentioned, design patterns) and evaluation metrics of a Digital Twin that uses a DLT?

Our literature analysis has brought out several functional and non-functional requirements for systems integrating DLT and DT, as shown in Tables 2 and 3. More in detail, Table 2 reports the main functional requirements that we identified with a brief description. *Data-driven synchronization*, treated as a macro requirement that includes data acquisition and monitoring and/or controlling activities based on the acquired information (e.g., Akash and Ferdous, 2022; Qiao et al., 2022; Pittaras et al., 2022; Dietz et al., 2019), and *data storing* (e.g., Asadi, 2021; Akash and Ferdous, 2022; Kumar et al., 2022) *and sharing* (e.g., Salim et al., 2022; Sahal et al., 2022a; Chai et al., 2021; Son et al., 2022; Lu et al., 2021) requirements are the most common ones. In fact, the properties and states of the phyical asset/process have to be congruous with the ones exposed by its virtual twin to ensure high consistency between real and digital replicas (Akash and Ferdous, 2022; Qiao et al., 2022). This can be achieved by enabling a nearly seamless data flow from the physical to the virtual world (Pittaras et al., 2022), minimizing the overhead that the DLT and in particular blockchain introduces in the systems.

Indeed, when it comes to data storing, as we found out in RQ4 results 3.3.4, the most used DLT solution is the blockchain one. However, due to the fact that the access to the blockchain is time-consuming, it is impractical to store data (such as DT models, states, results of elaborations) on-chain, also because these data are usually characterized by high velocity and variety (Dietz et al., 2019). That is why many studies (e.g., Asadi, 2021; Akash and Ferdous, 2022; Kumar et al., 2022) propose to store data in an off-chain storage, for instance distributed file systems as IPFS or Swarm, and store only hash data proofs on-chain. Finally, a requirement that can be considered optional

Table 2
Functional requirements.

Functional Requirements (FRs)	Manuscripts	Description
FR1: data-driven synchronization	Song and Hong (2021), Shukla et al. (2021), Hunhevicz et al. (2022), Wang et al. (2022), Suhail et al. (2022b), Teisserenc and Sepasgozar (2022), Salim et al. (2022), Sahal et al. (2022a), Huang et al. (2022), Akash and Ferdous (2022), Manoharan et al. (2022), Sahal et al. (2022b), Qu et al. (2021), Firouzi et al. (2021), Cohen et al. (2021), Chen et al. (2022), Altun and Tavli (2019), Liao et al. (2021), Kanak et al. (2019), Pittaras et al. (2022), Dai et al. (2022), Nguyen et al. (2023), He et al. (2022), Kuruppuarachchi et al. (2023), Hemdan and Mahmoud (2021), Dietz et al. (2019), Qiao et al. (2022)	Data acquisition, monitor and/or control of the physical twin (if possible, in real-time or near real-time).
FR2: data storing	Huang et al. (2020), Hasan et al. (2020), Teisserenc and Sepasgozar (2022), Sahal et al. (2022a), Akash and Ferdous (2022), Sahal et al. (2021), Lopez-Arevalo et al. (2021), Lu et al. (2020), Raj (2021), Teisserenc and Sepasgozar (2021b), Liao et al. (2021), Asadi (2021), Kumar et al. (2022)	Data memorization, such as physical asset/process states and other related information.
FR3: data sharing	Dietz et al. (2019), Huang et al. (2020), Hasan et al. (2020), Shen et al. (2021), Putz et al. (2021), Celik et al. (2023), Suhail et al. (2022b), Teisserenc and Sepasgozar (2022), Li et al. (2022), Salim et al. (2022), Sahal et al. (2022a), Chai et al. (2021), Son et al. (2022), Lu et al. (2021), Sun et al. (2020), Yang et al. (2022), Teisserenc and Sepasgozar (2021b), Altun and Tavli (2019), Dai et al. (2022), Liu et al. (2022), Pittaras and Polyzos (2022), Takahashi et al. (2022), Pincheira et al. (2023), Li et al. (2021)	Communication among multiple untrustworthy parties to share data.
FR4: interacting with the DT	Obushnyi et al. (2019), Putz et al. (2021), Suhail et al. (2022b), Khan et al. (2022), Kumar et al. (2022), Lu et al. (2020, 2021), Nielsen et al. (2020), Pittaras and Polyzos (2022), He et al. (2022), Qu et al. (2022), Takahashi et al. (2022), Li et al. (2021)	Multiple untrustworthy parties communication with the DT.
FR5: processes automation	Suhail et al. (2021), Hasan et al. (2020), Hunhevicz et al. (2022), Zhang et al. (2020), Qiao et al. (2022), Celik et al. (2021), Asadi (2021), Stanke et al. (2020), Yang et al. (2022), Teisserenc and Sepasgozar (2021a), Lopez et al. (2021)	Automation of physical processes and procedures with multiple untrustworthy parties involved.
(Optional) Payments	Obushnyi et al. (2019), Nabeeh et al. (2022), Huang et al. (2022), Stanke et al. (2020), Teisserenc and Sepasgozar (2021b), Dai et al. (2022)	Participants can exchange monetary value.

because strictly related to the application is the *payment* one which enables systems' participants to exchange monetary value (Stanke et al., 2020; Teisserenc and Sepasgozar, 2021b; Dai et al., 2022). For instance, Stanke et al. (2020) implement a finite element simulation as a service, therefore the need for a monetary transaction technology embedded in the system architecture arises to enable the user to pay for the service.

Instead, Table 3 lists the non-functional requirements that we found out with our study, such as *scalability* (Asadi, 2021), *interoperability* (Suhail et al., 2021; Teisserenc and Sepasgozar, 2021a,b), *data variety and velocity* (Dietz et al., 2019; Song and Hong, 2021). For instance, Asadi (2021) work deals with the realization of an architecture that enables the Cognitive Digital Twins (CDTs) creation, i.e., human skills and knowledge virtual replication. To this aim, the author chose the blockchain technology as a secure ledger for users' activities, while he adopted virtualization in order to prevent scalability issues related to the fact that each node of the blockchain has to maintain a full copy of the ledger.

Moreover, we conducted some analyses aimed at identifying the most common evaluation metrics used to validate DT ecosystems integrating DLTs. The works in Cohen et al. (2021), Chai et al. (2021), Salim et al. (2022), Hasan et al. (2020), Son et al. (2022) perform a *security analysis* on their final systems to validate their conformity with the security requirements. The works in Hasan et al. (2020), Hunhevicz et al. (2022), Suhail et al. (2022b) evaluate the adherence of their solutions to the *DT requirements* but all of them support this analysis with other quantitative evaluations such as by measuring throughput and latency. More specifically, the studies in Li et al. (2021), Zhang et al. (2020), Takahashi et al. (2022) evaluate blockchain performance in terms of number of confirmed transactions per second (TPS), namely *throughput*, while the *latency of transaction confirmation* is by far the most popular metric, as 15 studies employ it. Finally, when it comes to the prediction of future events such as faults, the *accuracy*, *precision*, *recall* as well as the *F1 score* become particularly relevant metrics because they allow to quantify how much that prediction is near to reality. Manuscripts (Shukla et al., 2021; Salim et al., 2022; Yang et al., 2022; Dai et al., 2022; Qu et al., 2022) deal with prediction and forecasting and adopt these metrics to evaluate the quality of their solutions.

Regarding the architectural point of view, apart from some papers such as Stanke et al. (2020) in which authors presented a serverless system architecture to increase scalability and efficiency and Suhail

Table 3
Non-functional requirements.

Non-Functional Requirements	Manuscripts	Description
NFR1: scalability	Suhail et al. (2021), Nabeeh et al. (2022), Huang et al. (2022), Qu et al. (2021), Liao et al. (2021), Asadi (2021), Altun and Tavli (2019)	Increasing the number of involved IoT devices should not impact the whole system performances.
NFR2: interoperability between heterogeneous devices and blockchains	Suhail et al. (2021), Shen et al. (2021), Teisserenc and Sepasgozar (2022), Khan et al. (2022), Kumar et al. (2022), Manoharan et al. (2022), Qu et al. (2021), Lopez-Arevalo et al. (2021), Sun et al. (2020), Teisserenc and Sepasgozar (2021a), Chen et al. (2022), Liao et al. (2021), Nguyen et al. (2023), He et al. (2022), Kuruppuarachchi et al. (2023), Li et al. (2021), Teisserenc and Sepasgozar (2021b)	Heterogeneous devices have to interact seamlessly. When multiple blockchains are employed, data exchanging among them should be possible.
NFR3: data volume, variety and velocity	Dietz et al. (2019), Shen et al. (2021), Song and Hong (2021), Putz et al. (2021), Wang et al. (2022)	The whole system should be able to cope with big data typical features, i.e., various types and big amounts of data from heterogeneous devices, high frequency acquisition.
NFR4: performance	Hunhevicz et al. (2022), Putz et al. (2021), Zhang et al. (2020), Teisserenc and Sepasgozar (2022), Chai et al. (2021), Kumar et al. (2022), Akash and Ferdous (2022), Lopez-Arevalo et al. (2021), Lu et al. (2020), Cohen et al. (2021), Stanke et al. (2020), Kanak et al. (2019), Pittaras et al. (2022), Liu et al. (2022), Pittaras and Polyzos (2022), Takahashi et al. (2022), Li et al. (2021), Takahashi et al. (2022)	Maximize throughput, minimize latency and optimize the energy usage of resource-constrained devices.
NFR5: trustworthy sources	Suhail et al. (2021), Shen et al. (2021), Obushnyi et al. (2019), Celik et al. (2023), Suhail et al. (2022b), Celik et al. (2021), Altun and Tavli (2019), Pincheira et al. (2023)	Heterogeneous untrustworthy data sources should be made trusted to avoid ineffective and potentially harmful decision-making.
NFR6: data confidentiality and traceability	Suhail et al. (2021), Dietz et al. (2019), Hasan et al. (2020), Shen et al. (2021), Obushnyi et al. (2019), Qiao et al. (2022), Salim et al. (2022), Sahal et al. (2022a, 2021), Asadi (2021), Nielsen et al. (2020), Raj (2021), Teisserenc and Sepasgozar (2021b), Lopez et al. (2021), Pittaras et al. (2022), Nguyen et al. (2023), Cohen et al. (2021), Chai et al. (2021), Son et al. (2022), Dietz et al. (2019), Lopez et al. (2021)	Confidential data should not be accessible from unauthorized users.
NFR7: data integrity	Suhail et al. (2021), Dietz et al. (2019), Huang et al. (2020), Hasan et al. (2020), Suhail et al. (2022b), Qiao et al. (2022), Salim et al. (2022), Sahal et al. (2021), Asadi (2021), Nielsen et al. (2020), Raj (2021), Yang et al. (2022), Lopez et al. (2021), Pittaras et al. (2022), Dai et al. (2022), Nguyen et al. (2023), Hemdan and Mahmoud (2021), Cohen et al. (2021), Chai et al. (2021), Son et al. (2022), Altun and Tavli (2019)	An unauthorized modification or destruction of data or operations while being processed, in transit or at rest has to be avoided.
NFR8: data availability	Suhail et al. (2021), Hasan et al. (2020), Suhail et al. (2022b), Salim et al. (2022), Son et al. (2022), Asadi (2021), Raj (2021), Lopez et al. (2021), Pittaras et al. (2022), Nguyen et al. (2023), Hemdan and Mahmoud (2021), Cohen et al. (2021), Chai et al. (2021)	When requested, stored data should be available in a timely manner.
NFR9: privacy	Celik et al. (2023), Li et al. (2022), Salim et al. (2022), Sahal et al. (2022a), Chai et al. (2021), Nabeeh et al. (2022), Son et al. (2022), Sahal et al. (2022b), Firouzi et al. (2021), Lu et al. (2021), Stanke et al. (2020), Teisserenc and Sepasgozar (2021a), He et al. (2022), Qu et al. (2022), Altun and Tavli (2019)	Participants' information (e.g., their identities) should remain hidden.

et al. (2022b) in which a blockchain-based DT framework is proposed to guarantee data trustworthiness by storing safety and security (S&S) rules in the blockchain, there are not so many evidences. Moreover, the few cases that discuss their framework proposal are domain specific. For instance, in the layered proposal of Song and Hong (2021) the architecture does not include digital twins' representative features, and blockchain is applied to ensure authentication in a City DT model. Another example is the hybrid blockchain-based six-layered DT architecture for CNC machines discussed in Shukla et al. (2021), in which there is no implementation and a simulation is carried out to evaluate the model-based prediction. Authors (Celik et al., 2023) present a blockchain architecture in a BIM environment alongside a prototype in the construction industry. Therefore, it can be noted that it is quite difficult to find out a generic solution for the implementation of DTs

Table 4
Challenges and solutions to address when integrating DLTs in a Digital Twin ecosystem.

Challenges	Solutions
Choosing a suitable DLT	(i) Requirements analysis; (ii) "Do you need a Blockchain?" methodology (Wüst and Gervais, 2018).
Keeping overhead low when low latency is a DT requirement to work properly	(i) *Technology*: choosing a blockchain with low finality or optimal block size; (ii) *Storage*: Minimize the information stored on-chain; (iii) *Data update frequency*: reduce the update frequency of data stored on-chain; (iv) *Consensus*: Choose a consensus algorithm that allows participants to achieve agreement in low time.
DLT requires specific IT infrastructure	Blockchain-as-a-Service (BaaS)
Lack of standards, reference architectures and evaluation metrics	Unsolved

and DLTs integration, mainly due to the fact the DT itself has no standardized architecture (De Benedictis et al., 2022), thus also the DLTs and DTs combination is affected by this problem.

3.3.7. RQ7: which are the main challenges to address for implementing the DTs and DLTs integration?

The process of integrating DLTs into a DT ecosystem is not straightforward. Indeed, as shown in Table 4, we identified four main challenges and the corresponding proposed solutions in the primary studies. The first challenge is the *selection of a suitable DLT* and clearly, this requires a careful analysis of application requirements that need to be addressed. Moreover, since there are some cases in which the use of DLTs and in particular of blockchain technology is not appropriate (e.g., there is a single writer in the system), the "Do you need a Blockchain?" methodology (Wüst and Gervais, 2018) can be useful to help developers into establishing if they really need blockchain for their purposes.

The second challenge is keeping *overhead low when low latency is one of the DT requirements*: since DTs have to be synchronized with their physical counterparts in (near) real-time, it is important to not let DLTs compromise their activities. Several solutions have been proposed in literature in order to address this problem. For instance, Li et al. (2021) carried out an analysis of blockchain impact in terms of throughput and latency with respect to the *block size* parameter. In fact, as pointed out in Wilhelmi et al. (2022), a small block size will inevitably decrease the probability of fork, since the propagation time is reduced, but would also lead to an increase of the overall transaction confirmation latency. By contrast, a bigger block size would decrease the transaction confirmation latency because more transactions would fit in a single block, thus remaining less time in the pool of unconfirmed transactions. However, after creating a block, it has to be broadcasted to all the nodes in the network in order to achieve consensus which leads to a non negligible communication overhead when dealing with big blocks. Therefore, it is important to evaluate a fair trade-off for the size of a block which guarantees both low transaction confirmation latency and low communication overhead. Authors (Li et al., 2021) propose to establish the optimal value for BlockSize based on a sensitivity analysis performed between latency and throughput.

Another possibility is the adoption of *blockchains with low finality* (Hasan et al., 2020), i.e., the assurance that transactions cannot be altered, reversed, or canceled after they are completed (Anon, 2023). Furthermore, as we have found out in RQ5 3.3.5, the blockchain overhead can be faced reducing the number of information stored on-chain (e.g., Akash and Ferdous, 2022; Celik et al., 2023; Teisserenc and Sepasgozar, 2022) or decreasing the update frequency of data stored on-chain (e.g., batching the updates). For instance, Suhail et al. (2022b) separate static from real-time and dynamic data in order to reduce the frequently time-consuming access to blockchain; instead, in Salim et al. (2022) authors manage cluster of IoT devices through their DTs that are in charge of their authentication, facing the problem of accessing often to blockchain for authentication purposes. In EtherTwin platform (Putz et al., 2021), sensor feeds are updated once per second with batched sensor updates, in this way there is no data losing and failure data are shared in a timely fashion like the authors argue. Finally, this blockchain issue can be avoided adopting a consensus algorithm that allows participants to achieve a fast agreement and that is scalable, thus performances do not decrease when the number of participants increase. An example of these kind of consensus algorithms are Practical Byzantine Fault Tolerance (PBFT) and Proof of Authority (PoA).

The third challenge is building a *specific IT infrastructure to use a DTL-based solution*; in fact, besides being an expensive activity, often enterprises do not have the know-how to build this kind of infrastructures. In the last years, a new paradigm called *Blockchain-as-a-Service* (BaaS) refers to a third-party cloud-based infrastructure for companies and it operates following the model of Software-as-a-Service (SaaS). Popular BaaS are Hyperledger Cello, the Ethereum BaaS hosted on Microsoft Azure and Amazon Managed Blockchain. For example, in Lopez et al. (2021) authors point out the role of Digital Twins in predicting failures and detecting cyber-security issues in real-time, as well as automating processes in Smart Grids. Moreover, they advise the usage of BaaS to benefit from reduced costs and higher scalability of resources, even if this means that enterprises must trust the blockchain provider.

Finally, when integrating DLT in a DT, as pointed out in RQ6 3.3.6, it should also be taken into account that the research on digital twins still suffers many issues that will influence also the DT-based solution, such as the *absence of standards and reference architectures* to use as a starting point in the development of these systems and *of specific evaluation metrics* to evaluate and validate the overall ecosystem.

3.4. Review summary

The main topics addressed by our systematic literature review along with related findings are summarized in the taxonomy in Fig. 2. Our study revealed that several solutions (mainly in the form of proof-of-concepts and prototypes) have been recently proposed in the literature which leverage on the convergence of DT and DLT technologies, therefore the topic is timely and relevant (RQ1). Most common application domains include Industry 4.0 and manufacturing, smart mobility and smart city, healthcare and finance (RQ2), where DTs are used for tasks like planning, monitoring, optimization, prediction, anomaly detection, while DLTs are employed to manage and share configuration and operational data exchanged between a physical system and its digital counterpart, for the (inter-twin) communication among different DTs in a DT ecosystem, or even to help automate processes involving different untrustworthy parties (RQ3, RQ6). Among DLT technologies, blockchain is the preferred choice in DT-related literature (RQ4). Common information stored on-chain include DT models, model changes and/or model states, device data and/or states, configuration and/or authentication data, or hash values of relevant data to reduce overhead (RQ5). Our study highlighted that, despite the prevalence of blockchain-based solutions, only 60% of analysed papers explicitly cope with well-known blockchain issues such as scalability and quantum immunity. As a general criterion, for scalability reasons it is highly recommended to store on-chain only the most critical information (e.g., DT model or model changes, security policies, etc.) while adopting a complementary off-chain storage solution (such as a cloud-based or a distributed file system) for other data. IOTA, a DAG-based DLT, is a promising alternative to blockchain in the DT domain, as it has been designed to overcome blockchain scalability issues and is able to achieve faster transactions, particularly suited for the high volume of data that are exchanged between entities in a DT

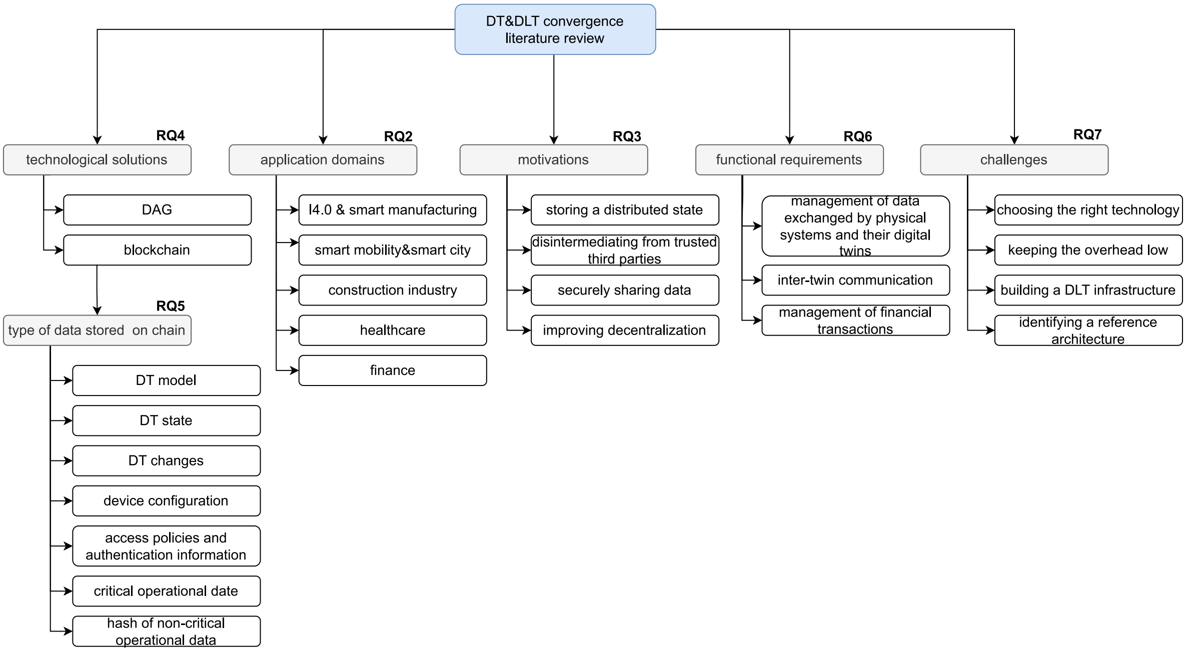

DT&DLT convergence literature review
RQ4
technological solutions
DAG
blockchain
RQ5
type of data stored on chain
DT model
DT state
DT changes
device configuration
access policies and authentication information
critical operational date
hash of non-critical operational data
RQ2
application domains
I4.0 & smart manufacturing
smart mobility&smart city
construction industry
healthcare
finance
RQ3
motivations
storing a distributed state
disintermediating from trusted third parties
securely sharing data
improving decentralization
RQ6
functional requirements
management of data exchanged by physical systems and their digital twins
inter-twin communication
management of financial transactions
RQ7
challenges
choosing the right technology
keeping the overhead low
building a DLT infrastructure
identifying a reference architecture

Fig. 2. A Taxonomy of the addressed research topics and related findings.

environment. Moreover, IOTA is considered quantum-resistant because it uses a combination of cryptographic algorithms that are believed to be resistant to quantum computers, therefore it would be a better solution compared to blockchain even with respect to security. As a final remark, beside the well-known challenges related to the choice of the right DLT technology to adopt based on scalability, efficiency and cost criteria, our study outlined that one big challenge is related to the identification of a reference DT framework where DLT can be integrated (RQ7). In the following of this paper, we will address this issue and propose a reference architecture for DT and DLT integration.

4. An architecture proposal for the integration of DTs and DLTs

Despite the recent hype around the DT concept and its wide diffusion in academic and industrial contexts, a clear understanding of how a DT (intended as a complex software system) is built has still not been reached. This is mainly due to the lack of a standardized and commonly accepted architectural framework for DT implementation. As pointed out by Ferko et al. (2022), the majority of existing scientific papers on DTs that tackle the architectural issue proposes a layered architecture, as the layered pattern helps to cope with the complexity of software intensive systems, such as DTs. However, most of existing proposals are domain-dependent and fail to identify the core functionalities that each DT should offer, regardless of the specific application domain. Similarly, the analysed literature on DT and DLT convergence lacks any integrated architecture proposal clearly identifying DT core functionalities and how these interact with DLT-based capabilities.

In the perspective of integrating DLTs in a domain-agnostic DT architectural model, we considered the recent De Benedictis et al. (2022) high-level 6-layer architecture shown in Fig. 3, in which each layer identifies the core DT functionalities. Briefly summarizing their proposal for the sake of completeness, the authors consider the following layers:

1. the *Physical Twin Layer* comprises the physical system, its subsystems and the components enabling the functionalities of sensing, actuating and control&configuration.
2. the *Data Layer* is responsible for the persistence of data that come from heterogeneous sources, their elaboration and presentation;

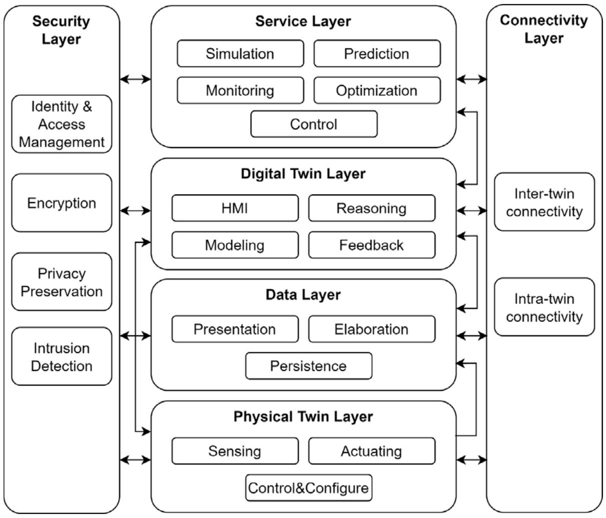


Fig. 3. Digital Twin high-level conceptual architecture proposed by De Benedictis et al. (2022).

3. the *Digital Twin Layer* constitutes the core of the architecture and includes the digital replica of the physical system implemented thorough modelling functionalities, reasoning for knowledge extraction and feedback generation capabilities as well as a Human Machine Interface (HMI) to allow direct interaction with end-users.
4. the *Service layer* is introduced to take into account the specific goals of the DT (*e.g.*, monitoring, prediction, simulation and so on).
5. the *Connectivity Layer* enables a real-time bidirectional communication between the digital twin and the physical system through the functionality of intra-twin communication for keeping consistent DT and PT states. It also provides inter-twin communication capabilities enabling the communication among virtually replicated physical entities.

6. the *Security Layer* aims at mitigating the vulnerabilities that the Digital Twin itself, with its interaction with the live system under observation, shows at different levels. It provides identity and access management, encryption, intrusion detection etc.

To cope with DT data security issues, we incorporated typical DLT architectural layers (Anthony Jnr, 2023; Zhu et al., 2019) in De Benedictis et al.' proposal. However, as we found out that there are many data types that can be stored in a DLT (or in a DFS and their hash stored in a DLT) answering RQ5 (Section 3.3.5), the architecture cannot be simply considered as it is for DLT integration. In fact, even if authors state that the Data Layer and the whole architecture should be capable of handling "Big Digital Twin Data", they do not really take into account this problem in their model. Thus, the 6-layer architecture has been firstly modified to put a major focus on data and their management, then has been extended integrating DLT-related functionalities needed to set-up the whole DLT infrastructure addressing (all or some of) data security issues in a DT-based application. Indeed, the original Data Layer has been splitted into four levels, devoted to data ingestion, storage, processing and visualization; the previous Digital Twin Layer has been distributed across a specific layer dedicated to digital replication and functionalities included in already existing layers; finally, consensus and peer-to-peer network levels for DLT development has been added.

More in detail, the **Physical Layer** has the same functionalities of Physical Twin Layer apart from the sensing capability. In fact, sensors data represent only one of the data types leveraged by a DT-based solution. For this reason, we consider `general data sources` such as structured (*e.g.*, CSV files), semi-structured (*e.g.*, JSON, XML), unstructured (*e.g.*, textual, audio, images), time-series data sources and so on. Moreover, data feeding the DT can be dynamic/real-time streams (*i.e.*, operational data (De Benedictis et al., 2022)) and/or static/historical ones (*e.g.*, information provided by domain-knowledge experts (De Benedictis et al., 2022)). For this reason, the data flow from the physical to the digital world has been divided in real-time streams and batches to ensure data management with respect to "Lambda" architecture (Debauche et al., 2022). In fact, handling data as they arrive, without any delay, and performing immediate actions or analyses in response to events or data changes, typically within milliseconds or seconds of data arrival, require real-time and thus low latency data processing. Instead, large volumes of static or historical data can be processed periodically (e.g., hourly, daily, weekly). This explains the two arrows labelled with "`streams`" and "`batches`" from the Physical Layer data sources.

Consequently, we added a **Communication Layer** with `network system` that represents the general network infrastructure that allows the connection between physical and digital worlds, `protocol translation` and `data transmission` functionalities to cope with protocol compatibility issues and batches data transfer. On top of this level, there is the **Data Ingestion Layer** responsible for `collecting` and processing raw data from various sources. As these data usually come in different formats and structures, this layer is also in charge of possibly performing basic `data transforming` operations, e.g., data parsing, data type conversions, data cleansing. `Data validating` is the functionality executed to ensure accuracy and integrity of incoming data, applying validation rules to check for anomalies, missing values and other data quality issues. Moreover, it can be useful to carry out `data aggregating` activities to group data into larger datasets for further meaningful analysis (*e.g.*, combining data from various sources or time periods). The upper level is the **Data Storage Layer**, whose goal is to securely and efficiently store the ingested and processed data so that they can be easily retrieved, analysed, and managed, thus it offers one or more `data storage` solutions. In particular, we distinguished `not-distributed` (*e.g.*, local, single node database, traditional file systems) and `distributed storage` (*e.g.*, cloud, distributed database) including Distributed File Systems such as IPFS or Hadoop.

In this level, `blockchain` and `DAG-based data structures` are included: blockchains and DAG-based DLTs are not traditional distributed storage solutions, that is why are not depicted in the distributed storage box. However, since they allow the management and recording of transactions in a decentralized and distributed manner, as shown in general DLT architectures (Zhu et al., 2019; Anthony Jnr, 2023), the underlying data structures (*e.g.*, transactions, blocks, genesis block, Merkle tree, DAG) are put into the Data Storage Layer, while the consensus mechanism and the peer-to-peer network populate two other layers. The **Consensus Layer** contains the three main classes of consensus algorithms: (i) `proof-based consensus` (*e.g.*, Proof-of-Work), `BFT-based consensus` (*e.g.*, PBFT) and `DAG-based consensus` (lightweight Proof-of-Work). The **Network Infrastructure Layer** is characterized by the `peer-to-peer model`, `node type` (*e.g.* full/lightweight node) according to the chosen DLT, `permissionless/permissioned network` and `virtual machines` to support smart contracts execution. Please, note that these two layers are depicted in vertical to manage both situations in which DLTs are simply used as security enforced storage solutions and the ones in which DLTs are used also to guarantee security in the communication between the physical and the digital layer. In the latter case, the network system of Communication Layer and the Network Infrastructure Layer represent the same infrastructure (*i.e.*, the peer-to-peer network).

The **Virtual Layer** contains the living digital replica, *i.e.*, the runtime models executed in their execution environment, thus it is responsible of `model simulation` or `model emulation` or both. In particular, model simulation is the creation of a virtual representation of a physical system using mathematical and computational models that mimic the behaviour of the real-world systems. On the other hand, model emulation is a more advanced approach that involves a high-fidelity model of the physical system for a more accurate reflection of its behaviour at any given moment. This level of fidelity is valuable for scenarios where real-time performance and feedback are critical, such as controlling complex systems like power grids or autonomous vehicles (Qi et al., 2021; Hunhevicz et al., 2022; Ferko et al., 2022; Semeraro et al., 2021). It is also possible to emulate critical parts of the physical system and simulate the remaining ones (Semeraro et al., 2021).

As said before, our Virtual Layer extends the modelling functionality of the Digital Twin Layer in Fig. 3. Furthermore, in model by De Benedictis et al., there is a reasoning capability responsible for the knowledge extraction and DT models feeding with static or dynamic data: the outcomes of this functionality are essentially the outcomes of an additional layer, namely the **Analytics Layer**. In fact, this level contains the `low latency data processing` functionality that leverages on stream processing engines, like Apache Kafka, Apache Flink, Apache Storm, or Apache Spark Streaming that provide the ability to process and analyse data streams in real-time, and real-time analytics tools such as Elasticsearch, Kibana, or Grafana for real-time data analysis and visualization. The `batches data processing` is not time-sensitive and focuses on analysing data in batch to uncover patterns and insights that can be used for future decision-making or predictive purposes. Batch processing engines like Apache Hadoop, Apache Spark, or Apache Beam are useful for this kind of activity since they allow for distributed processing of large datasets.

The **Visualization Layer** extends the general HMI and Presentation functionalities depicted in Fig. 3, since all architectural levels are interested in the included capabilities. In fact, it contains: (i) `data representation` for displaying data obtained from physical and digital replicas, but also information coming from the Service Layer; `3D visualization` that allows users to view the system model in a spatial context, facilitating a better understanding of its physical layout and interactions; `customizable dashboards` to arrange and prioritize the information user want to see; `collaboration and sharing` allowing multiple users to interact with the DT simultaneously.

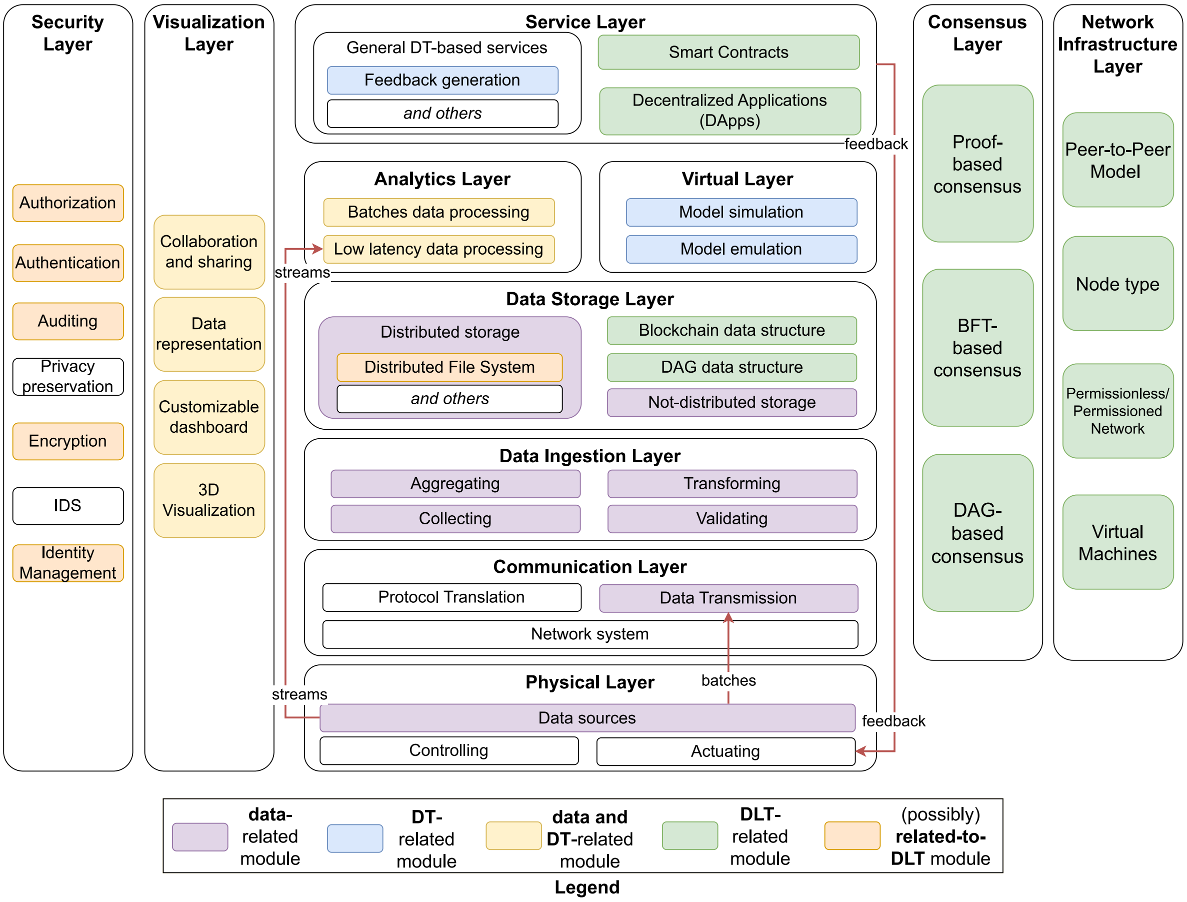


Fig. 4. High-level architectural model for DT and DLTs integration. Boxes in blue represent the layers that expand the DT Layer in De Benedictis et al. (2022), boxes in violet are the Data Layer extension, the ones in yellow are related to both levels. Green boxes are introduced to set-up the whole DLT infrastructure, while orange ones are boxes that possibly can be related to DLTs.

The **Service Layer** contains the DT-based services such as monitoring, prediction and the `feedback generation` capability in charge of alerts/alarms generation depending on the DT-based application purposes, *i.e.*, simply warning operators in the physical world that some actions should be taken or directly executing these commands into the physical world. This justifies the "`feedback`" labelled red-arrow from the Service to the Physical Layer. Since some of the services can be implemented through `Smart Contracts` and/or `Decentralized Applications` (DApps), these have been added in the Service Layer, corresponding to the general application layer in DLT architecture (Anthony Jnr, 2023). Finally, the **Security Layer** is depicted in vertical in Fig. 4 and contains the main functionalities to cope with different levels security issues. Note that some of them are orange coloured to emphasize the possibility to solve these problems through DLT adoption.

In conclusion, the proposed architecture facilitate the integration of DLTs, thus coping with DT data security issues at one and/or more from different levels perspective. Moreover, even if from the outcomes of RQ4, RQ5 and RQ6 (Sections 3.3.4, 3.3.5, 3.3.6) there is still a trend to store heavy information on-chain, in such architecture, regardless of the chosen blockchain, this technology can be easily introduced storing real-time data in distributed file systems or cloud storage once they are treated in the Analytics Layer and their hash on-chain. In this way, it is possible to tackle blockchains scalability and performance issues in DT context, facing one of the challenges identified answering RQ7 (Section 3.3.7), *i.e.*, keeping the overhead low when low latency is one of the DT requirement.

5. Proof-of-Concepts

In order to demonstrate the validity of our architecture proposal for the integration of DTs and DLTs, in this Section we discuss two different PoCs implemented with different technological stacks, which show how the blockchain technology can be concretely used in two different DT operational scenarios. Even if the architecture depicted in Fig. 4 allows the integration of different DLTs, we decided to instantiate it with blockchain since this seems to be the most common solution as pointed out in RQ4 (Section 3.3.4).

5.1. PoC 1: a (DT+blockchain)-enabled distributed intrusion detection and prediction system

The first PoC takes into account a distributed physical system made up of an arbitrary number of physical CPSs which are digitally replicated by their Digital Twins. In the considered scenario, involved CPSs, as well as their respective replicas, are managed by different parties and are not mutually trusted, which is the case in many application domains. Assume that the DT provides, for each physical system, intrusion detection and prediction services, and assume that the different DTs cooperate by sharing intrusion detection results to detect and predict intrusions that may be found only by correlating all available information from the different systems.

In this scenario, data generated by built-in Intrusion Detection Systems (IDS) may be sensitive and critical, and must therefore be securely stored. Rather than adopting a centralized cloud-based service

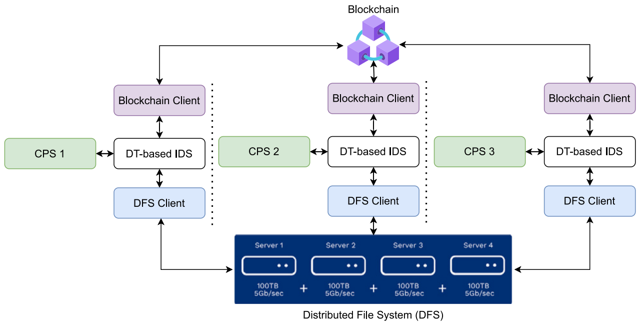


Fig. 5. Distributed DT-based IDS leveraging blockchain.

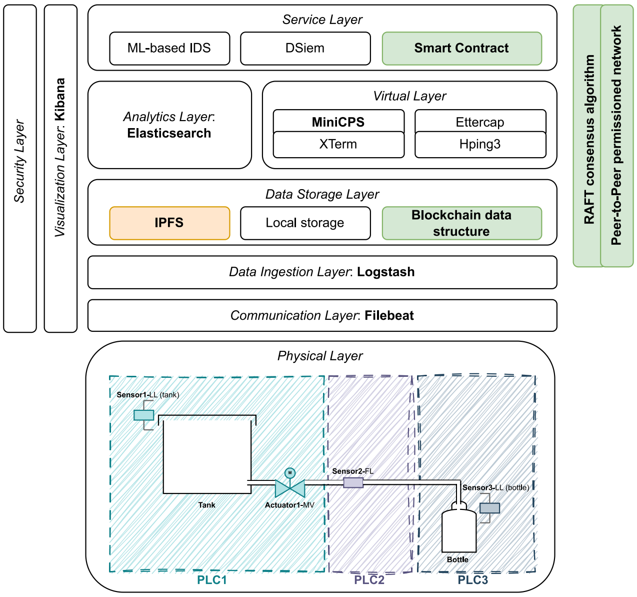


Fig. 6. Mapping of general DT architecture integrating DLTs within the first PoCs. Green boxes are needed to set-up Minifabric blockchain.

to aggregate and analyse data, which would require an agreement with a TTP and that, in any case, would not ensure immutability of data, in our first PoC IDS data are stored into a DFS and, to preserve data integrity and authenticity and provide trustworthiness, their file system references are stored in a blockchain. Later on, these data can be retrieved by one of the DT instances (called *DT-based IDSs* in this PoC) to identify suitable control actions to take depending on the detected intrusions. As depicted in Fig. 5, each *DT-based-IDS* stores the results of its own elaborations in a DFS through a *DFS Client*, while it accesses the blockchain through a *blockchain Client* to store the references of respective file system. This approach not only addresses the inherent Single Point of Failure (SPOF) issue of centralized systems that becomes less efficient, secure, and resilient as the network size and heterogeneity increase (Li et al., 2023), but also achieves a secure collaboration among untrustworthy entities thanks to the blockchain cryptographic properties.

This PoC was built on top of the existing project discussed in Varghese et al. (2022), which leverages the MiniCPS toolkit (Antonioli and Tippenhauer, 2015) for CPS simulation to implement a Digital Twin security framework with intrusion detection capabilities, called *Machine Learning (ML)-based IDS*. In Fig. 5, such a system may represent the ith DT-based-IDS which is integrated with a DFS and a blockchain in order to implement the scenario described above.

As depicted in Fig. 6, the *Physical Layer* contains a CPS that is an industrial filling plant comprising three Programmable Logic Controllers (PLCs): the physical process that drives the evolution of the plant is composed of a liquid tank, a bottle and a pipe connecting them. Three sensors, namely *Sensor1-LL*, *Sensor2-FL* and *Sensor3-LL* read the liquid level in the tank, the flow level in the pipe and the liquid level in the bottle respectively, meanwhile a motor valve actuator controls the liquid flow in the pipe. PLC2 and PLC3 are responsible for gathering sensor readings from *Sensor2-FL* and *Sensor3-LL* respectively and for forwarding them to PLC1, which in turn is in charge of receiving the sensor readings from PLC2 and PLC3. Moreover, PLC1 must also collect sensor readings from *Sensor1-LL* and eventually controls the motor valve actuator depending on all three sensors measurements. Operational data are transmitted through Filebeat,[4] and then ingested and pre-processed by Logstash.[5] The *Data Storage Layer* is composed by: local storage (system logs in csv format) for storing data received from the physical world; (ii) IPFS for intrusion detection results memorization; (iii) blockchain data structure. More in detail, the blockchain adopted in this PoC is *Hyperledger Fabric* one of the most popular permissioned blockchain platforms developed by the Linux foundation, explaining the RAFT consensus algorithm in the *Consensus Layer* and the peer-to-peer permissioned network model of *Network Infrastructure Layer*. However, since Fabric consists of various pluggable components (*e.g.*, endorsers, ordering service, a set of databases and committers), we decided to adopt ***Minifabric***,[6] *i.e.*, an IPFS Client based on Kubo,[7] to enable the interaction with the IPFS aimed at relieving the burden of the blockchain.

The *Virtual Layer* is composed of MiniCPS executed with XTerm emulator and security attack simulator implemented by means of Ettercap and Hping3. The *Analytics Layer* is represented by the analytics and search engine called Elasticsearch.[8] As regards the *Visualization Layer* Kibana[9] has been used for data visualization.

Finally, in the *Service Layer*, we can distinguish the *ML-based IDS* module in charge of classifying the data samples coming from the DT, the *DSiem* correlation engine that performs incident detection based on a rule-based correlation engine that monitors system logs and a *smart contract* written in Go and deployed on peers participating in the blockchain. This contract enables to store and retrieve the hash value of the IDS results file providing three main functionalities: *init* that initializes the blockchain with an empty value for the hash, *update* which allows for updating the hash value and *query* that an end-user can leverage to retrieve the current hash value stored on-chain. Please, note that the *Security Layer* is depicted in vertical for consistency with the general proposed architecture, however the security functionalities such as IDS, authorization, authentication are implemented through the components distributed in the whole framework.

Fig. 7 shows the main interactions among involved modules. Apart from the *IDS* module, responsible for performing intrusion detection based on operational data generated by the *Digital Twin* module, the *IPFS Client* and the *Minifabric Client* that allow the *IDS* and *Digital Twin* modules to interact with the IPFS and the Minifabric blockchain, the diagram also reports two entity types, namely *shared volumes* and *pipes*. The former represents a shared memory that enables communication and file sharing between containers, while the latter represents a named pipe which is a UNIX concept that allows a program to communicate with another program using a pipe. In this work, named pipes are used to forward control commands from a container to another.

[4] https://www.elastic.co/beats/filebeat
[5] https://www.elastic.co/logstash/
[6] https://labs.hyperledger.org/labs/minifabric.html
[7] https://github.com/ipfs/kubo
[8] https://www.elastic.co/
[9] https://www.elastic.co/kibana/

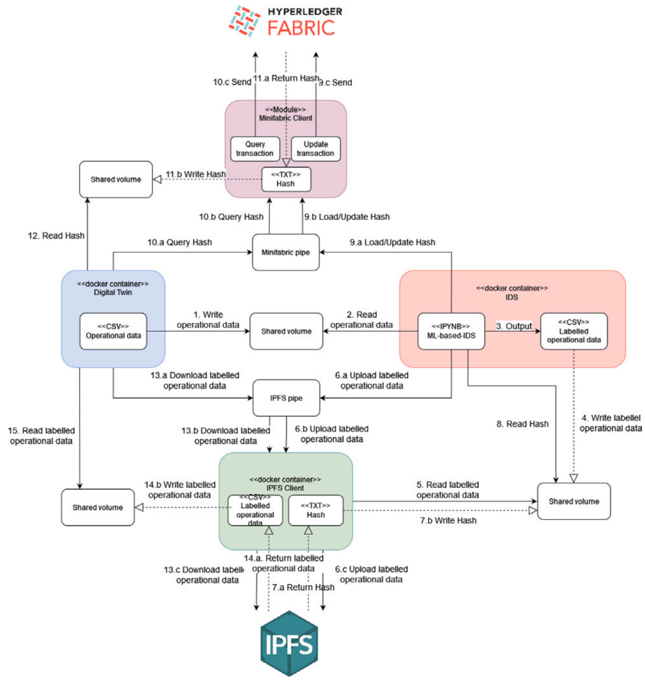


Fig. 7. Diagram showing numbered interactions making up the use-case scenario.

More in detail, four *shared volumes* are represented to enable file sharing between the *IDS* module and the *Digital Twin* module, the *IDS* module and the *IPFS Client* module, the *Digital Twin* module and the *IPFS Client* module and finally the *Digital Twin* module and the *Minifabric Client* module. The diagram shows two named pipes as well, namely *Minifabric pipe* and *IPFS pipe* that allow the *IDS* module and the *Digital Twin* module to send control commands to be executed from the *Minifabric Client* and the *IPFS Client* respectively.

A detailed description of the operational flow reported in the diagram is provided in the following:

1. **Write operational data**: The *Digital Twin* module writes the simulated operational data (system logs) to a volume shared with the *IDS* module.
2. **Read operational data**: The *IDS* module reads the simulated operational data from the shared volume.
3. **Output**: The *IDS* module performs the ML-based intrusion detection based on the simulated operational data with the aim of identifying potential intrusions in the system. The output of this process consists in operational data file in which each log has been labelled with a tag that states the *normal* functioning of the system or the identified *intrusion*.
4. **Write labelled operational data**: The *IDS* modules writes the labelled operational data in a volume shared with the *IPFS Client* module.
5. **Read labelled operational data**: The *IPFS Client* module retrieves the labelled operational data from the shared volume.
6. **Upload labelled operational data**: The *IDS* module commands the *IPFS Client* module to upload the results file to the IPFS. This step is composed of three sub-steps as it is necessary a named pipe, namely *IPFS pipe*, that forwards the upload command to the *IPFS Client*.
7. **Return Hash + Write Hash**: The IPFS returns the hash code of the just uploaded file to the *IPFS Client* module, which is in charge of sharing it with the *IDS* module through the volume they share. Even this step is divided into two sub-steps: the first one which consists in the response returned by the IPFS, the second one in which the returned hash is written to the shared volume.
8. **Read Hash**: The *IDS* module reads the hash code from the shared volume.
9. **Load/Update Hash**: The *IDS* module instructs the *Minifabric Client* to load or update the hash in blockchain, depending whether it is the first time or not. As for the sixth step, this one is divided into three sub-steps as also in this case it is necessary a named pipe, namely *Minifabric pipe*, that forwards the Load/Update command to the *Minifabric Client*. At this point the Minifabric blockchain acknowledges the success of the operation if it was successful.
10. **Query Hash**: Once the *Digital Twin* module learns that the hash code is on-chain (this can be done in two ways which are a shared memory or a TCP/IP communication, but none of these methods has been implemented in this work), it instructs the *Minifabric Client* to query the hash code by means of the *Minifabric pipe*. As the previous step, this one is divided into three sub-steps for the same reason.
11. **Return Hash + Write Hash**: The *Minifabric Client* module returns the required Hash and writes it into the volume that it shares with the *Digital Twin* module.
12. **Read Hash**: The *Digital Twin module* reads the Hash from the shared volume.
13. **Download labelled operational data**: once the *Digital Twin* module owns the hash of the labelled operational data, it can retrieve the file from the IPFS by instructing the *IPFS Client module* to do so via the *IPFS pipe*.
14. **Return + Write labelled operational data**: In this composite step, firstly the IPFS returns the labelled operational data in response to the command previously sent by the *Digital Twin* module and for second the *IPFS Client* writes these data to the volume it shares with the *Digital Twin* module.
15. **Read labelled operational data**: Finally, the *Digital Twin* module can retrieve the labelled operational data from the volume shared with the *IPFS Client*. Depending on these data, the *Digital Twin* module can take countermeasures to contrast the recognized attacks.

5.2. PoC 2: a (DT+blockchain)-enabled Smart City

The second PoC aims at showing how DLTs can be integrated in a DT environment with respect to the proposed architecture (Fig. 4) in a distributed and complex scenario such as the Smart City one where several threats affect data security. In fact, as depicted in Fig. 8, a *Urban Digital Twin* (*i.e.*, a Digital Twin of an entire city) is realized as a composition of DTs of its entities, *e.g.*, digital twins of smart cars, of devices installed in the physical system, of public transport and so on. However, one of the biggest issues in smart cities is the *data sources untrustworthiness* since the IoT devices deployed in a Smart City are managed by different unreliable entities with potentially conflicting interests. Moreover, an intruder could threaten *Confidentiality*, *Integrity* and *Availability* (CIA) of data at rest.

Even if a TTP could solve these issues by playing the role of intermediary between the devices and the Urban DT, it may introduce weaknesses affecting the reliability of the system such as SPOF, which in turn makes the system vulnerable to security attacks such as Denial of Service (DoS) attacks. Therefore, in this PoC we simulated heterogenous IoT devices, sending data to their virtual counterparts; furthermore, in order to address the aforementioned security issues, every action performed on/by each DT has been tracked and stored in a distributed ledger (Hyperledger Fabric).

More in detail, as depicted in Fig. 9, the *Physical Layer* is composed by multiple IoT devices updating the state of their digital twins through ***Eclipse Hono*** that implements the *Communication* and *Data Ingestion Layer*. In fact, in order to develop this PoC, we leveraged on

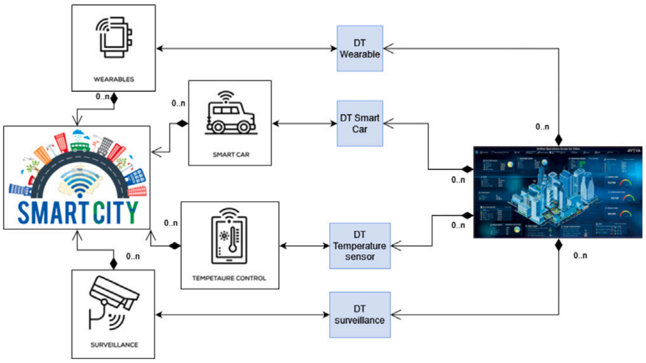


Fig. 8. Application scenario involving a Smart City and its DT.

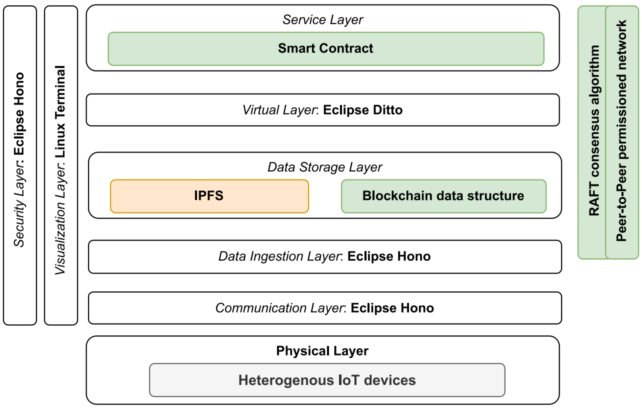
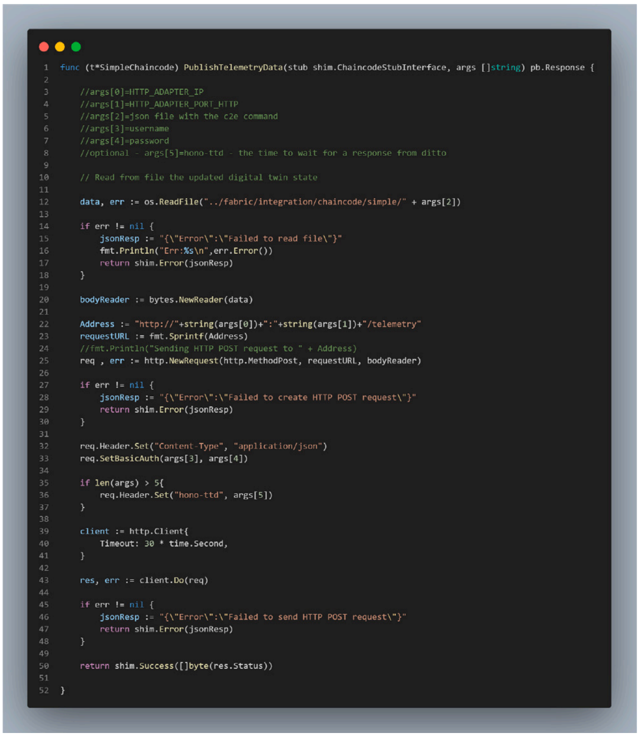

Security Layer: Eclipse Hono
Visualization Layer: Linux Terminal
Service Layer
Smart Contract
Virtual Layer: Eclipse Ditto
Data Storage Layer
IPFS
Blockchain data structure
Data Ingestion Layer: Eclipse Hono
Communication Layer: Eclipse Hono
Physical Layer
Heterogenous IoT devices
RAFT consensus algorithm
Peer-to-Peer permissioned network

Fig. 9. Mapping of general DT architecture integrating DLTs within the second PoC. Green boxes are needed to set-up Hyperledger Fabric blockchain.

Cloud2Edge,[10] *i.e.*, an open-source tool provided by the Eclipse Foundation which integrates two other frameworks, the already mentioned Eclipse Hono and ***Eclipse Ditto*** configuring them to work together:

- *Eclipse Hono* provides interfaces for connecting large number of IoT devices normalizing various IoT protocols to AMQP 1.0 by means of two adapters, namely the *HTTP Adapter* and the *MQTT Adapter*. In this way, upper layers can focus on storage and processing of data. In fact, all the information retrieved from the physical twins and/or related to activities performed on the DT, such as its creation, and on its state (*e.g.*, its update dictated by new sensor readings) are stored in a distributed file system (IPFS) and their hash is put in the Hyperledger Fabric blockchain, both constituting the *Data Storage Layer*. The usage of Fabric implies that the *Consensus* and the *Network Infrastructure* layers are the same of the first PoC.
  Hono is also equipped with a *Device Registry* that is responsible for ensuring that the back-end applications only receive information from properly authenticated and authorized devices. This explains the reason why the *Security Layer* is based on Eclipse Hono alongside with blockchain security enforcement.
- *Eclipse Ditto* is in charge of implementing the digital twins, processing data coming from devices and structuring them by following the DT paradigm to ease access by applications. Thus, Ditto composes the *Virtual Layer*.

Cloud2Edge services can be accessed by means of a RESTful API that allows for *device registration*, *device credentials setting*, *device twin creation*, *telemetry data publication* (that enables to automatically forward telemetry data from a device to its DT in order to update its state), and *commands propagation to devices* based on the received data. Therefore, a *Service Layer* with smart contracts (*e.g.*, telemetry data publication) are developed leveraging on the whole infrastructure. Finally, as regards the *Visualization Layer*, since Ditto is not equipped with a graphical interface as instead happened for MiniCPS, we interacted with the system through a Unix Terminal.



```go
func (t*SimpleChaincode) PublishTelemetryData(stub shim.ChaincodeStubInterface, args []string) pb.Response {

    //args[0]=HTTP_ADAPTER_IP
    //args[1]=HTTP_ADAPTER_PORT_HTTP
    //args[2]=json file with the c2e command
    //args[3]=username
    //args[4]=password
    //optional - args[5]=hono-ttd - the time to wait for a response from ditto

    // Read from file the updated digital twin state

    data, err := os.ReadFile("../fabric/integration/chaincode/simple/" + args[2])

    if err != nil {
        jsonResp := "{\"Error\":\"Failed to read file\"}"
        fmt.Println("Err:%s\n",err.Error())
        return shim.Error(jsonResp)
    }

    bodyReader := bytes.NewReader(data)

    Address := "http://"+string(args[0])+":"+string(args[1])+"/telemetry"
    requestURL := fmt.Sprintf(Address)
    //fmt.Println("Sending HTTP POST request to " + Address)
    req , err := http.NewRequest(http.MethodPost, requestURL, bodyReader)

    if err != nil {
        jsonResp := "{\"Error\":\"Failed to create HTTP POST request\"}"
        return shim.Error(jsonResp)
    }

    req.Header.Set("Content-Type", "application/json")
    req.SetBasicAuth(args[3], args[4])

    if len(args) > 5{
        req.Header.Set("hono-ttd", args[5])
    }

    client := http.Client{
        Timeout: 30 * time.Second,
    }

    res, err := client.Do(req)

    if err != nil {
        jsonResp := "{\"Error\":\"Failed to send HTTP POST request\"}"
        return shim.Error(jsonResp)
    }

    return shim.Success([]byte(res.Status))

}
```

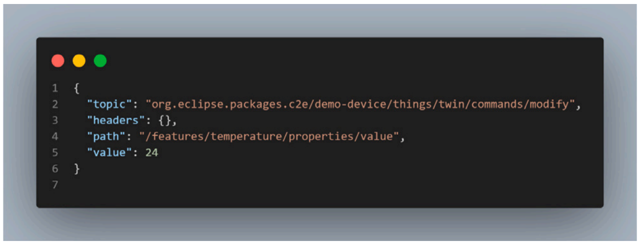
Fig. 10. Example of smart contract code to send telemetry data from the IoT device to its DT.

Concerning the development of the smart contract written in Golang, as we decided to use Hyperledger Fabric (and not the lightweight MiniFabric used in the first PoC), it was essential to find a way to quickly and iteratively test the chaincode without incurring in the overhead due to *chaincode lifecycle commands* required at every modification. Thus, the Fabric network was started in *Development Mode* (DevMode) that allowed for easier deployment, debugging and update of chaincode.

As an example, we reported a snippet of code in Fig. 10, showing how an IoT device can send telemetry data to its Digital Twin through a smart contract: the function `PublishTelemetryData` builds an HTTP POST request by means of the API included in the *net/http* and *net/httputil* packages (imported at the beginning of the source code). Through this HTTP request, telemetry data are then forwarded to the HTTP adapter, that normalizes them to the AMQP 1.0 protocol and eventually sends them to the Virtual Layer. The function requires the following five input parameters in order to be executed:

1. `HTTP_ADAPTER_IP` and `HTTP_ADAPTER_PORT_HTTP` represent the IP address associated to the HTTP Adapter and its port, respectively;
2. a `JSON file` which contains the Cloud2Edge command that allows physical devices to publish telemetry data. For instance, as shown in Fig. 11, the physical device notices its DT that the value for its `temperature` feature has changed to twenty-four;
3. `username` and `password`, firstly set through the function of the smart contract `SetCredentials`, are associated to the physical device representing its identity and thus stored in the `Device Registry`;

[10] https://www.eclipse.org/packages/packages/cloud2edge/



```
{
  "topic": "org.eclipse.packages.c2e/demo-device/things/twin/commands/modify",
  "headers": {},
  "path": "/features/temperature/properties/value",
  "value": 24
}
```

Fig. 11. Cloud2Edge command expressed in JSON format.

4. hono-ttd is the (sixth) optional parameter and enables to specify how much time, in terms of seconds, the device will wait for a response.

In the body of the PublishTelemetryData function, the variable data stores data read from the JSON file, namely the Cloud2Edge command. As this command constitutes the body of the HTTP POST request, it is converted in a new Reader object and then it is stored in the variable bodyReader. In the next lines of code shown in Fig. 10, the URL of the HTTP Adapter is stored in the requestURL variable. The HTTP POST request is built through the http.NewRequest function which requires the HTTP method of the request, the requestURL and the bodyReader respectively as parameters. The HTTP POST request is stored in the req variable. Moreover, the header of the request must indicate that the content of the request is of type application/json. This can be done through the Header.Set function to be called on the variable maintaining the request, namely req. In order to publish telemetry data, IoT devices must specify their identities in the HTTP request which can be done through the SetBasicAuth function to be called on the req variable again. Finally, an HTTP Client is built that allows for control over the HTTP Header and authentication. Eventually, the Client is responsible for forwarding the HTTP POST request to the HTTP adapter through the Do function called on itself.

6. Evaluation and future work

The discussed Proof-of-Concepts demonstrate that the proposed architecture integrating DT and DLT is *domain-agnostic* since it has been applied in two different domains. In fact, in contrast with RQ6 results (Section 3.3.6), the architecture has been instantiated to set-up a Digital Twin in a manufacturing and industrial scenario (PoC1 in Section 5.1) and in a simple Smart City example (PoC in Section 5.2). Moreover, in each PoC we leverage on diverse technological stacks proving the *platform-independence* of our architectural proposal. Indeed, in PoC1 we used MiniCPS simulator, Minifabric blockchain, Elasticsearch and command-line tools, while, apart from Fabric lightweight network, in PoC2 we utilized Eclipse Cloud2Edge integrating Eclipse Hono and Eclipse Ditto.

A generic high-level layered architecture offers several benefits such as *modularity* by organizing functional components indifferent layers, *separation of concerns* to isolate different functionalities within specific layers, *scalability* by allowing layers to scale independently based on system's requirements, *reusability* and *flexibility*. Nevertheless, in order to evaluate functional and non-functional requirements, we will take into account the PoCs deployment architectures because some technological choices can and will influence the requirements coverage. For this reason, in the rest of this section, we present a coverage analysis of the requirements, listed in Table 2 and Table 3, of the two PoCs. Such analysis is summarized in Table 5.

Regarding the data-driven synchronization requirement (*FR1*), both PoCs partially cover it: in fact, even if the usage of Filebeat and Logstash in PoC1 and Eclipse Hono in PoC2 ensure data acquisition and monitoring capabilities, the physical twins are merely simulated and no feedbacks are propagated from their digital twins into physical layers. For data storing (*FR2*), in each case study we leverage on the combination of IPFS as distributed file system and Minifabric, the lightweight Fabric network. However, while in Ditto-based PoC the peer-to-peer permissioned network is used for data sharing (*FR3*) among Data Storage, Virtual and Service layers, in MiniCPS PoC Minifabric assure data sharing only between digital twin and intrusion detection modules. This explains the partially vs. complete coverage of FR3 and *FR4* (interacting with DT) in the two case studies.

Even if MiniCPS is a useful tool for research, it does not have the ability to scale (*NFR1*) because adding and/or removing components in the physical twin requires lots of programming effort, thus it cannot be used for setting up different digital twins or simulating different scenarios. In fact, MiniCPS' simulation is computationally-intensive and particularly slow because, as explained by authors (Antonioli and Tippenhauer, 2015), MiniCPS is not a performance simulator (*NFR4*) and not even a tool for optimization. Therefore, independently from the chosen DLT, PoC1 cannot cover the first and the fourth non-functional requirements because of MiniCPS limitations. Moreover, despite Minifabric offers some features promoting interoperability (*NFR2*) such as modular architecture and pluggable consensus mechanism, MiniCPS restrictions make impossible to achieve even NFR2 in PoC1.

On the contrary, PoC2 partially covers NFR1, NFR2 and NFR4 thanks to Eclipse Ditto properties: NFR1 because Ditto is designed to scale horizontally (handling large numbers of digital twins and concurrent requests) and vertically (supporting clustering and distributed deployment architectures); NFR2 thanks to highly interoperability since Eclipse framework supports various protocols (*e.g.*, MQTT, HTTP) and standards for communication and integration with a wide spectrum of IoT devices, platforms, and systems; finally, NFR4 is covered because Eclipse Ditto is optimized for high performance and low-latency interactions between digital twins and IoT devices because of efficient data storing and retrieval mechanisms and asynchronous processing to minimize response times and maximize throughput. Apart from modelling and simulation tools, the chosen permissioned blockchain in our two case studies ensures immutability and integrity of data (*NFR7*), provides transparency and auditability (*NFR6*) and can possibly help to assure participants' privacy (*NFR9*). For this reason, NFR6 and NFR7 are partially covered in PoC1 since Logstash and Filebeat do not provide security mechanisms themselves (even if they can be configured to ensure secure data transport and access); while the aforementioned requirements are totally covered in PoC2 because of Minifabric security capabilities and Eclipse Hono that assures security communication protocols, message encryption, authentication, authorization and auditing. Thus, heterogeneous untrustworthy data sources (*NFR5*) are made trusted in PoC2 and partially in PoC1 (only DT and IDS modules). Finally, big data typical features (*NRF3*) and data availability (*NRF8*) in a timely manner are not taken into account. However, in both case studies, we still tried to integrated a (permissioned) blockchain as happened in the majority of the selected studies (see RQ4 in Section 3.3.4): this choice inevitably introduces complexity and overhead to system architecture and, in spite of the decision to store heavy information off-chain, increases scalability challenges and has impact on performance.

Our analysis shows that PoC2 totally covers 6 over 14 requirements (almost the 43%), while the 28.5% is partially covered and the remaining 28.5% is not covered due to design choices and Minifabric usage. Instead, the PoC1 cannot cover almost the 22% of requirements due to MiniCPS limitations. This clearly justifies why in our future work, we plan to improve mainly PoC2 by replacing Minifabric with *IOTA Tangle* in which academia is gaining attraction and checking for scalability and performance improvements thanks to IOTA' features such as absence of blocks and miners, parallel transaction validation and the localized consensus. Finally, since the Digital Twin is simply executed in "simulation mode", to better prove the applicability and

Table 5
Coverage matrix of functional and non-functional requirements in the two PoCs: ● represents complete coverage, ☾ means that the requirement is partially covered and ○ stands for no coverage at all. The ○* indicates the impossibility to cover the requirement in future work.

	PoC1: MiniCPS+Minifabric	PoC2: Eclipse Ditto+Minifabric
FR1: data-driven synchronization	☾	☾
FR2: data storing	●	●
FR3: data sharing	☾	●
FR4: interacting with the DT	☾	●
FR5: process automation	○	○
NFR1: scalability	○*	☾
NFR2: interoperability	○*	☾
NFR3: data volume, variety and velocity	○	○
NFR4: performance	○*	☾
NFR5: trustworthy sources	☾	●
NFR6: data confidentiality and traceability	☾	●
NFR7: data integrity	☾	●
NFR8: data availability	○	○
NFR9: privacy	○	○

feasibility of the proposal we will connect the DT and the entire Eclipse-based platform to a real physical counterpart. Concluding, the proposed coverage analysis can be also a useful tool for any practitioner to determine the best implementation for their project or context, and this represent an addition added-value of the current paper.

7. Conclusions

In this paper we have investigated the integration among DLTs and DTs. To better understand what is the current state of integration, we conducted a SLR that pointed out a general interest in adopting DLTs (in particular blockchains) to ensure data trustworthiness in DT-enabled applications. Recurring application domains include I4.0, Smart Manufacturing, Smart City and Smart Mobility, where distributed ledgers are used to store a wide spectrum of information, ranging from DT models, DT state and DT change history to device configurations and data. The literature review revealed several challenges for DLT and DT integration, related for example to the identification of the best DLT solution to adopt based on existing requirements, to the need for an expensive infrastructure for DLT implementation, to the need for effective (low finality) and efficient (low overhead) solutions and, above all, to the absence of standards and reference architectures for DTs. This last challenge in particular led us to propose a reference architecture for DTs where all the capabilities related to data management are explicitly identified and where the role of blockchains and distributed file systems is clarified. To support our proposal, we also discussed two different PoCs involving the adoption of different technological stacks, which show how the blockchain technology can be concretely used in two different DT operational scenarios.

As future work, we plan to extend the proposed architecture introducing functionalities to support the bidirectional communication that defines the DT and the usage of the DT and DLT framework (a Secure Cyber Digital Twin De Benedictis et al. (2022)) for the improvement of CPSs security, *e.g.*, attack/intrusion/anomaly detection. Moreover, we further improve the PoC2 by replacing the permissioned blockchain network with a DAG-based DLT such as IOTA Tangle to evaluate scalability and performance improvements in a real case study.

Funding

This work has been partially supported by the Spoke 9 "Digital Society & Smart Cities" of ICSC - Centro Nazionale di Ricerca in High Performance-Computing, Big Data and Quantum Computing, funded by the European Union - NextGenerationEU (PNRR-HPC, CUP: E63C22000980007).

CRediT authorship contribution statement

Alessandra Somma: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Alessandra De Benedictis:** Writing – review & editing, Writing – original draft, Validation, Supervision, Conceptualization. **Christiancarmine Esposito:** Supervision, Conceptualization. **Nicola Mazzocca:** Supervision, Conceptualization.

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.

Data availability

This declaration is "not applicable" as the reported research does not involve the use of any dataset.

References

Akash, S.S., Ferdous, M.S., 2022. A blockchain based system for healthcare digital twin. IEEE Access 10, 50523–50547.

Alcaraz, C., Lopez, J., 2022. Digital twin: A comprehensive survey of security threats. IEEE Commun. Surv. Tutor..

Altun, C., Tavli, B., 2019. Social internet of digital twins via distributed ledger technologies: application of predictive maintenance. In: 2019 27th Telecommunications Forum. TELFOR, IEEE, pp. 1–4.

Anon, 2023. Finality binance academy glossary. https://academy.binance.com/en/glossary/finality. (Accessed 18 February 2023).

Anthony Jnr, B., 2023. A developed distributed ledger technology architectural layer framework for decentralized governance implementation in virtual enterprise. Inf. Syst. e-Bus. Manag. http://dx.doi.org/10.1007/s10257-023-00634-2.

Antonioli, D., Tippenhauer, N.O., 2015. Minicps: A toolkit for security research on cps networks. In: Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/Or Privacy. pp. 91–100.

Asadi, A.R., 2021. Cognitive ledger project: Towards building personal digital twins through cognitive blockchain. In: 2021 2nd International Informatics and Software Engineering Conference. IISEC, IEEE, pp. 1–5.

Celik, Y., Petri, I., Barati, M., 2023. Blockchain supported BIM data provenance for construction projects. Comput. Ind. 144, 103768.

Celik, Y., Petri, I., Rezgui, Y., 2021. Leveraging BIM and blockchain for digital twins. In: 2021 IEEE International Conference on Engineering, Technology and Innovation. ICE/ITMC, IEEE, pp. 1–10.

Chai, H., Leng, S., He, J., Zhang, K., Cheng, B., 2021. CyberChain: Cybertwin empowered blockchain for lightweight and privacy-preserving authentication in internet of vehicles. IEEE Trans. Veh. Technol. 71 (5), 4620–4631.

Chen, X., Tang, X., Xu, X., 2022. Digital technology-driven smart society governance mechanism and practice exploration. Front. Eng. Manag. 1–20.

Cohen, Y., Nabrzyski, J., Taylor, I., 2021. Framework for block-chain deployment in assembly of an air-craft or a SpaceCraft. IFAC-PapersOnLine 54 (1), 988–992.

Dai, M., Wang, T., Li, Y., Wu, Y., Qian, L., Su, Z., 2022. Digital twin envisioned secure air-ground integrated networks: A blockchain-based approach. IEEE Internet Things Mag. 5 (1), 96–103.

De Benedictis, A., Esposito, C., Somma, A., 2022. Toward the adoption of secure cyber digital twins to enhance cyber-physical systems security. In: Vallecillo, A., Visser, J., Pérez-Castillo, R. (Eds.), Quality of Information and Communications Technology. Springer International Publishing, Cham, pp. 307–321.

De Benedictis, A., Mazzocca, N., Somma, A., Strigaro, C., 2022. Digital twins in healthcare: an architectural proposal and its application in a social distancing case study. IEEE J. Biomed. Health Inf. 1–12. http://dx.doi.org/10.1109/JBHI.2022.3205506.

Debauche, O., Mahmoudi, S., Manneback, P., Lebeau, F., 2022. Cloud and distributed architectures for data management in agriculture 4.0 : Review and future trends. J. King Saud Univ. - Comput. Inf. Sci. 34 (9), 7494–7514. http://dx.doi.org/10.1016/j.jksuci.2021.09.015.

Dietz, M., Putz, B., Pernul, G., 2019. A distributed ledger approach to digital twin secure data sharing. In: Data and Applications Security and Privacy XXXIII: 33rd Annual IFIP WG 11.3 Conference, DBSec 2019, Charleston, SC, USA, July 15–17, 2019, Proceedings 33. Springer, pp. 281–300.

Eckhart, M., Ekelhart, A., 2019. Digital twins for cyber-physical systems security: State of the art and outlook. In: Security and Quality in Cyber-Physical Systems Engineering: With Forewords by Robert M. Lee and Tom Gilb. Springer, pp. 383–412.

Farahani, B., Firouzi, F., Luecking, M., 2021. The convergence of IoT and distributed ledger technologies (DLT): Opportunities, challenges, and solutions. J. Netw. Comput. Appl. 177, 102936. http://dx.doi.org/10.1016/j.jnca.2020.102936, URL https://www.sciencedirect.com/science/article/pii/S1084804520303945.

Ferko, E., Bucaioni, A., Behnam, M., 2022. Architecting digital twins. IEEE Access 10, 50335–50350. http://dx.doi.org/10.1109/ACCESS.2022.3172964.

Firouzi, F., Farahani, B., Daneshmand, M., Grise, K., Song, J., Saracco, R., Wang, L.L., Lo, K., Angelov, P., Soares, E., Loh, P.-S., Talebpour, Z., Moradi, R., Goodarzi, M., Ashraf, H., Talebpour, M., Talebpour, A., Romeo, L., Das, R., Heidari, H., Pasquale, D., Moody, J., Woods, C., Huang, E.S., Barnaghi, P., Sarrafzadeh, M., Li, R., Beck, K.L., Isayev, O., Sung, N., Luo, A., 2021. Harnessing the power of smart and connected health to tackle COVID-19: IoT, AI, robotics, and blockchain for a better world. IEEE Internet Things J. 8 (16), 12826–12846. http://dx.doi.org/10.1109/JIOT.2021.3073904.

Hasan, H.R., Salah, K., Jayaraman, R., Omar, M., Yaqoob, I., Pesic, S., Taylor, T., Boscovic, D., 2020. A blockchain-based approach for the creation of digital twins. IEEE Access 8, 34113–34126.

He, C., Luan, T.H., Lu, R., Su, Z., Dong, M., 2022. Security and privacy in vehicular digital twin networks: Challenges and solutions. IEEE Wirel. Commun. 1–8. http://dx.doi.org/10.1109/MWC.002.2200015.

Hemdan, E.E.-D., Mahmoud, A.S.A., 2021. BlockTwins: A blockchain-based digital twins framework. In: Blockchain Applications in IoT Ecosystem. Springer International Publishing, Cham, pp. 177–186. http://dx.doi.org/10.1007/978-3-030-65691-1_12.

Huang, S.-H., Day, J.-D., Shu, M.-H., Huang, H.-C., Huang, J.-C., 2022. Construction of virtual marketing interactive platform for digital twin innovation and entrepreneurship based on blockchain. Sci. Program. 2022, http://dx.doi.org/10.1155/2022/7497323.

Huang, S., Wang, G., Yan, Y., Fang, X., 2020. Blockchain-based data management for digital twin of product. J. Manuf. Syst. 54, 361–371.

Hunhevicz, J.J., Motie, M., Hall, D.M., 2022. Digital building twins and blockchain for performance-based (smart) contracts. Autom. Constr. 133, 103981.

Kanak, A., Ugur, N., Ergun, S., 2019. A visionary model on blockchain-based accountability for secure and collaborative digital twin environments. In: 2019 IEEE International Conference on Systems, Man and Cybernetics. SMC, pp. 3512–3517. http://dx.doi.org/10.1109/SMC.2019.8914304.

Khan, L.U., Han, Z., Saad, W., Hossain, E., Guizani, M., Hong, C.S., 2022. Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities. IEEE Commun. Surv. Tutor. 24 (4), 2230–2254. http://dx.doi.org/10.1109/COMST.2022.3198273.

Kitchenham, B., 2004. Procedures for Performing Systematic Reviews. Vol. 33, (2004), Keele University, Keele, UK, pp. 1–26.

Kumar, P., Kumar, R., Kumar, A., Franklin, A.A., Garg, S., Singh, S., 2022. Blockchain and deep learning for secure communication in digital twin empowered industrial IoT network. IEEE Trans. Netw. Sci. Eng..

Kuruppuarachchi, P.M., Rea, S., McGibney, A., 2023. Trusted and secure composite digital twin architecture for collaborative ecosystems. IET Collab. Intell. Manuf. 5 (1), e12070.

Li, Y., Fan, Y., Zhang, L., Crowcroft, J., 2023. RAFT consensus reliability in wireless networks: Probabilistic analysis. IEEE Internet Things J..

Li, J., Kassem, M., 2021. Applications of distributed ledger technology (DLT) and Blockchain-enabled smart contracts in construction. Autom. Constr. 132, 103955. http://dx.doi.org/10.1016/j.autcon.2021.103955.

Li, M., Li, Z., Huang, X., Qu, T., 2021. Blockchain-based digital twin sharing platform for reconfigurable socialized manufacturing resource integration. Int. J. Prod. Econ. 240, 108223.

Li, T., Wang, H., He, D., Yu, J., 2022. Synchronized provable data possession based on blockchain for digital twin. IEEE Trans. Inf. Forensics Secur. 17, 472–485.

Liao, S., Wu, J., Bashir, A.K., Yang, W., Li, J., Tariq, U., 2021. Digital twin consensus for blockchain-enabled intelligent transportation systems in smart cities. IEEE Trans. Intell. Transp. Syst. 23 (11), 22619–22629.

Liu, J., Zhang, L., Li, C., Bai, J., Lv, H., Lv, Z., 2022. Blockchain-based secure communication of intelligent transportation digital twins system. IEEE Trans. Intell. Transp. Syst. 23 (11), 22630–22640. http://dx.doi.org/10.1109/TITS.2022.3183379.

Lopez, J., Rubio, J.E., Alcaraz, C., 2021. Digital twins for intelligent authorization in the B5G-enabled smart grid. IEEE Wirel. Commun. 28 (2), 48–55.

Lopez-Arevalo, I., Gonzalez-Compean, J.L., Hinojosa-Tijerina, M., Martinez-Rendon, C., Montella, R., Martinez-Rodriguez, J.L., 2021. A WoT-based method for creating digital sentinel twins of IoT devices. Sensors 21 (16), http://dx.doi.org/10.3390/s21165531.

Lu, Y., Huang, X., Zhang, K., Maharjan, S., Zhang, Y., 2020. Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks. IEEE Trans. Ind. Inform. PP, 1. http://dx.doi.org/10.1109/TII.2020.3017668.

Lu, Q., Ye, Z., Fang, Z., Meng, J., Pitt, M., Lin, J., Xie, X., Chen, L., 2021. Creating an inter-hospital resilient network for pandemic response based on blockchain and dynamic digital twins. In: 2021 Winter Simulation Conference. WSC, IEEE, pp. 1–12.

Manoharan, H., Teekaraman, Y., Kuppusamy, R., Kaliyan, N., Thelkar, A.R., 2022. Examining the effect of cyber twin and blockchain technologies for industrial applications using AI. Math. Probl. Eng. 2022.

Nabeeh, N., Abdel-Basset, M., Gamal, A., Chang, V., 2022. Evaluation of production of digital twins based on blockchain technology. Electronics 11, 1268. http://dx.doi.org/10.3390/electronics11081268.

Nguyen, T.A., Kaliappan, V.K., Jeon, S., Jeon, K.-s., Lee, J.-W., Min, D., 2023. Blockchain empowered federated learning with edge computing for digital twin systems in urban air mobility. In: Lee, S., Han, C., Choi, J.-Y., Kim, S., Kim, J.H. (Eds.), The Proceedings of the 2021 Asia-Pacific International Symposium on Aerospace Technology. APISAT 2021, Vol. 2, Springer Nature, Singapore, pp. 935–950.

Nielsen, C.P., da Silva, E.R., Yu, F., 2020. Digital twins and blockchain – proof of concept. Procedia CIRP 93, 251–255. http://dx.doi.org/10.1016/j.procir.2020.04.104, 53rd CIRP Conference on Manufacturing Systems 2020.

Obushnyi, S., Kravchenko, R., Babichenko, Y., 2019. Blockchain as a transaction protocol for guaranteed transfer of values in cluster economic systems with digital twins. In: 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology. PIC S&T, pp. 241–245. http://dx.doi.org/10.1109/PICST47496.2019.9061233.

Pincheira, M., Vecchio, M., Antonelli, F., 2023. SmartTwin: A blockchain-based software framework for digital twins using IoT. In: Blockchain and Applications, 4th International Congress. Springer, pp. 67–77.

Pires, F., Cachada, A., Barbosa, J., Moreira, A.P., Leitão, P., 2019. Digital twin in industry 4.0: Technologies, applications and challenges. In: 2019 IEEE 17th International Conference on Industrial Informatics. INDIN, Vol. 1, IEEE, pp. 721–726.

Pittaras, I., Fotiou, N., Karapapas, C., Siris, V.A., Polyzos, G.C., 2022. Secure, mass web of things actuation using smart contracts-based digital twins. In: 2022 IEEE Symposium on Computers and Communications. ISCC, IEEE, pp. 1–6.

Pittaras, I., Polyzos, G.C., 2022. (POSTER) SmartTwins: Secure and auditable DLT-based digital twins for the WoT. In: 2022 18th International Conference on Distributed Computing in Sensor Systems. DCOSS, pp. 82–84. http://dx.doi.org/10.1109/DCOSS54816.2022.00027.

Putz, B., Dietz, M., Empl, P., Pernul, G., 2021. Ethertwin: Blockchain-based secure digital twin information management. Inf. Process. Manage. 58 (1), 102425.

Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., Wang, L., Nee, A., 2021. Enabling technologies and tools for digital twin. J. Manuf. Syst. 58, 3–21. http://dx.doi.org/10.1016/j.jmsy.2019.10.001.

Qiao, L., Dang, S., Shihada, B., Alouini, M.-S., Nowak, R., Lv, Z., 2022. Can blockchain link the future? Digit. Commun. Netw. 8 (5), 687–694.

Qu, Y., Gao, L., Xiang, Y., Shen, S., Yu, S., 2022. FedTwin: Blockchain-enabled adaptive asynchronous federated learning for digital twin networks. IEEE Netw. 36 (6), 183–190.

Qu, Q., Xu, R., Chen, Y., Blasch, E., Aved, A., 2021. Enable fair proof-of-work (PoW) consensus for blockchains in IoT by miner twins (MinT). Future Internet 13 (11), http://dx.doi.org/10.3390/fi13110291.

Raj, P., 2021. Empowering digital twins with blockchain. In: Advances in Computers. vol. 121, Elsevier, pp. 267–283.

Sahal, R., Alsamhi, S.H., Brown, K.N., 2022a. Personal digital twin: a close look into the present and a step towards the future of personalised healthcare industry. Sensors 22 (15), 5918.

Sahal, R., Alsamhi, S.H., Brown, K.N., O'Shea, D., Alouffi, B., et al., 2022b. Blockchain-based digital twins collaboration for smart pandemic alerting: decentralized COVID-19 pandemic alerting use case. Comput. Intell. Neurosci. 2022.

Sahal, R., Alsamhi, S.H., Brown, K.N., O'shea, D., McCarthy, C., Guizani, M., 2021. Blockchain-empowered digital twins collaboration: smart transportation use case. Machines 9 (9), 193.

Salim, M.M., Comivi, A.K., Nurbek, T., Park, H., Park, J.H., 2022. A blockchain-enabled secure digital twin framework for early botnet detection in IIoT environment. Sensors 22 (16), 6133.

Schärer, K., Comuzzi, M., 2023. The quantum threat to blockchain: summary and timeline analysis. Quantum Mach. Intell. 5, http://dx.doi.org/10.1007/s42484-023-00105-4.

Semeraro, C., Lezoche, M., Panetto, H., Dassisti, M., 2021. Digital twin paradigm: A systematic literature review. Comput. Ind. 130, 103469. http://dx.doi.org/10.1016/j.compind.2021.103469.

Shen, W., Hu, T., Zhang, C., Ma, S., 2021. Secure sharing of big digital twin data for smart manufacturing based on blockchain. J. Manuf. Syst. 61, 338–350.

Shukla, A., Pansuriya, Y., Tanwar, S., Kumar, N., Piran, M.J., 2021. Digital twin-based prediction for CNC machines inspection using blockchain for industry 4.0. In: ICC 2021-IEEE International Conference on Communications. IEEE, pp. 1–6.

Singh, M., Fuenmayor, E., Hinchy, E.P., Qiao, Y., Murray, N., Devine, D., 2021. Digital twin: Origin to future. Appl. Syst. Innov. 4 (2), http://dx.doi.org/10.3390/asi4020036.

Soltani, R., Zaman, M., Joshi, R., Sampalli, S., 2022. Distributed ledger technologies and their applications: A review. Appl. Sci. 12 (15), http://dx.doi.org/10.3390/app12157898.

Son, S., Kwon, D., Lee, J., Yu, S., Jho, N.-S., Park, Y., 2022. On the design of a privacy-preserving communication scheme for cloud-based digital twin environments using blockchain. IEEE Access 10, 75365–75375.

Song, Y., Hong, S., 2021. Build a secure smart city by using blockchain and digital twin. Int. J. Adv. Sci. Converg. 3, 9–13.

Stanke, J., Unterberg, M., Trauth, D., Bergs, T., 2020. Development of a hybrid DLT cloud architecture for the automated use of finite element simulation as a service for fine blanking. Int. J. Adv. Manuf. Technol. 108, 3717–3724.

Suhail, S., Hussain, R., Jurdak, R., Hong, C.S., 2021. Trustworthy digital twins in the industrial internet of things with blockchain. IEEE Internet Comput. 26 (3), 58–67.

Suhail, S., Hussain, R., Jurdak, R., Oracevic, A., Salah, K., Hong, C.S., Matulevičius, R., 2022a. Blockchain-based digital twins: Research trends, issues, and future challenges. ACM Comput. Surv. 54 (11s), http://dx.doi.org/10.1145/3517189.

Suhail, S., Malik, S.U.R., Jurdak, R., Hussain, R., Matulevičius, R., Svetinovic, D., 2022b. Towards situational aware cyber-physical systems: A security-enhancing use case of blockchain-based digital twins. Comput. Ind. 141, 103699.

Sun, S., Zheng, X., Villalba-Díez, J., Ordieres-Meré, J., 2020. Data handling in industry 4.0: Interoperability based on distributed ledger technology. Sensors 20 (11), 3046.

Sunyaev, A., 2020. Distributed ledger technology. In: Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies. Springer International Publishing, Cham, pp. 265–299. http://dx.doi.org/10.1007/978-3-030-34957-8_9.

Takahashi, K., Kanai, K., Nakazato, H., 2022. Performance evaluation of blockchains towards sharing of digital twins. In: 2022 IEEE 4th Global Conference on Life Sciences and Technologies. LifeTech, IEEE, pp. 128–129.

Tao, F., Xiao, B., Qi, Q., Cheng, J., Ji, P., 2022. Digital twin modeling. J. Manuf. Syst. 64, 372–389.

Teisserenc, B., Sepasgozar, S., 2021a. Adoption of blockchain technology through digital twins in the construction industry 4.0: a PESTELS approach. Buildings 11 (12), 670.

Teisserenc, B., Sepasgozar, S., 2021b. Project data categorization, adoption factors, and non-functional requirements for blockchain based digital twins in the construction industry 4.0. Buildings 11 (12), 626.

Teisserenc, B., Sepasgozar, S.M., 2022. Software architecture and non-fungible tokens for digital twin decentralized applications in the built environment. Buildings 12 (9), 1447.

van der Valk, H., Haße, H., Möller, F., Otto, B., 2022. Archetypes of digital twins. Bus. Inf. Syst. Eng. 64 (3), 375–391.

Varghese, S.A., Ghadim, A.D., Balador, A., Alimadadi, Z., Papadimitratos, P., 2022. Digital twin-based intrusion detection for industrial control systems. In: 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events. PerCom Workshops, IEEE, pp. 611–617.

Vilas-Boas, J.L., Rodrigues, J.J., Alberti, A.M., 2022. Convergence of distributed ledger technologies with digital twins, IoT, and AI for fresh food logistics: Challenges and opportunities. J. Ind. Inf. Integr. 100393. http://dx.doi.org/10.1016/j.jii.2022.100393.

Wang, C., Cai, Z., Li, Y., 2022. Sustainable blockchain-based digital twin management architecture for IoT devices. IEEE Internet Things J..

Wilhelmi, F., Barrachina-Muñoz, S., Dini, P., 2022. End-to-end latency analysis and optimal block size of proof-of-work blockchain applications. IEEE Commun. Lett. 26 (10), 2332–2335.

Wüst, K., Gervais, A., 2018. Do you need a blockchain? In: 2018 Crypto Valley Conference on Blockchain Technology. CVCBT, IEEE, pp. 45–54.

Xie, J., Yu, F.R., Huang, T., Xie, R., Liu, J., Liu, Y., 2019. A survey on the scalability of blockchain systems. IEEE Netw. 33 (5), 166–173.

Yang, X., Maiti, A., Jiang, J., Kist, A., 2022. Forecasting and monitoring smart buildings with the internet of things, digital twins and blockchain. In: Online Engineering and Society 4.0: Proceedings of the 18th International Conference on Remote Engineering and Virtual Instrumentation. Springer, pp. 213–224.

Yaqoob, I., Salah, K., Uddin, M., Jayaraman, R., Omar, M., Imran, M., 2020. Blockchain for digital twins: Recent advances and future research challenges. IEEE Netw. 34 (5), 290–298. http://dx.doi.org/10.1109/MNET.001.1900661.

Zhang, C., Zhou, G., Li, H., Cao, Y., 2020. Manufacturing blockchain of things for the configuration of a data-and knowledge-driven digital twin manufacturing cell. IEEE Internet Things J. 7 (12), 11884–11894.

Zhou, Q., Huang, H., Zheng, Z., Bian, J., 2020. Solutions to scalability of blockchain: A survey. IEEE Access 8, 16440–16455.

Zhu, Q., Loke, S.W., Trujillo-Rasua, R., Jiang, F., Xiang, Y., 2019. Applications of distributed ledger technologies to the internet of things: A survey. ACM Comput. Surv. 52 (6), http://dx.doi.org/10.1145/3359982.

Alessandra Somma is a Ph.D. student in Information Technology and Electrical Engineering at the Department of Electrical Engineering and Information Technologies of the University of Naples Federico II, Italy, where she also received her M.Sc. degree in Computer Engineering in 2021. Her research activities concern Digital Twins, their architectural and security issues, their application in IoT/IIoT, healthcare and railway contexts and their usage for the enhancement of Cyber-Physical Systems resilience.

Alessandra De Benedictis is a Tenured Assistant Professor at the Department of Electrical Engineering and Information Technology, University of Naples Federico II, where she got her Ph.D. in Computer and Automation Engineering in 2013. Her research interests include the design and evaluation of secure architectures for the protection of distributed resource-constrained devices, the development of methodologies for the security analysis and assessment of complex applications in IoT and cloud environments, and the analysis and design of Digital Twin-based applications and services.

Christian Esposito received the Ph.D. degree in computer engineering and automation from the University of Naples “Federico II," Naples, Italy in 2009. He is an Associate Professor with the University of Salerno, Fisciano, Italy, and was an Assistant Professor with the University of Naples “Federico II". He has authored more than 50 journal publications and 30 conference papers, and has been involved in several international and national research and industrial projects. His research interests include reliable and secure communications, middleware, distributed systems, multiobjective optimization, and game theory. Dr. Esposito has served as a reviewer or a guest editor for several international journals and conferences, and has been a PC member or a organizer of about 50 international conferences/workshops. He is also a member of three journal editorial boards.

Nicola Mazzocca is a Full Professor of Computer Systems at the Department of Electrical Engineering and Information Technologies of the University of Naples Federico II. Since 1994, he has held numerous university courses and has been involved in several professional training activities on different topics, including high-performance systems, distributed and embedded systems, security, and reliability. His research activities concern computer architecture, distributed systems, high-performance computing, and safety-critical applications. He is author of more than 250 publications in international journals, books, and conference proceedings.