IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 22, 2022, accepted April 29, 2022, date of publication May 5, 2022, date of current version May 13, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3172964*

# Architecting Digital Twins

**ENXHI FERKO, ALESSIO BUCAIONI, AND MORIS BEHNAM**
Division of Product Realisation, Mälardalen University, 631 05 Eskilstuna, Sweden

Corresponding author: Enxhi Ferko (enxhi.ferko@mdu.se)

This work was supported in part by the Swedish Knowledge Foundation [Stiftelsen för kunskaps- och kompetensutveckling (KKS)] through the Assured Cloud Platforms for Industrial Cyber-physical Systems (ACICS) and Model-Based DevOps for Cyber-Physical System Product Lines (Modev) projects, in part by the Sweden's Innovation Agency [Verket för innovationssystem (VINNOVA)] through the PANORAMA Project, and in part by the Excellence in Production Research [Excellence in Production Research (XPRES)] Framework.

**ABSTRACT** In 2002, Grieves defined the concept of the digital twin as a virtual instance of physical assets capable of continuously mirroring them. Ever since then, driven by remarkable industrial attention, digital twins flourished and ripened in several sectors. The notable industrial adoption has been sided by a growing interest from the software engineering community in general and the software architecture community in particular as demonstrated by the growing number of published peer-reviewed publications and proposed software architectural solutions for digital twins. In this paper, we report on the planning, execution, and results of a systematic mapping study on architecting digital twins. The study captures crucial aspects of software architectures for digital twins as types of architectural solutions, quality attributes, and architectural patterns. It supports practitioners in creating digital twins tailored to their specific needs and researchers in identifying trends and open challenges. Starting from an initial set of potentially relevant 1630 peer-reviewed publications, we selected 140 primary studies. We analysed the set of primary studies using thorough data extraction, analysis, and synthesis process. To compensate for single method limitations and reduce possible threats to conclusion validity, we discussed the results of our study with experts in the software architecture community. Based on our results, the field of software architecture for digital twins is lively and an increasing number of architectural solutions are being proposed. Although there is a lack of widely accepted reference architectural solutions for digital twins, most of them are built using a combination of the layered and service-oriented patterns and address maintainability, performance efficiency, and compatibility quality attributes.

**INDEX TERMS** Architectural patterns, digital twin, software architectures, quality attributes.

## I. INTRODUCTION

After the launch of the National Aeronautics and Space Administration (NASA) Apollo 13 in April 1970, an explosion in the oxygen tanks critically damaged the spacecraft. A key success factor of the rescue mission was that NASA created physically duplicated systems at ground level to match the spacecraft in orbit and used these physical twins to train astronauts and mission controllers in every mission scenario, including multiple failure ones. That was the dawn of the concept of Digital Twin (DT). Since then, twins have been becoming predominantly virtual rather than physical. During a talk at the Florida Institute of Technology in 2002, Grieves proposed the first definition of DTs as information-mirroring models of physical assets. Accordingly, a physical system has a virtual representation and the information flow between the two systems enables a continuous synchronisation of the virtual to the physical part and vice versa. Advances in information technology brought the ideas of Grieves to life and enabled a comprehensive implementation of DTs.

While DTs flourished and ripened within the production domain, in recent years, it has been applied to several additional domains from smart cities to aerospace through web services and automotive [1]. In 2019, the global research and advisory firm Gartner rated DTs among the top ten strategic technology trends capable of a profound impact on modern society.[1] Following the remarkable industrial adoption, the software engineering research community developed a growing interest in DTs. In 2021, the International Conference on Digital Twins Technologies (ICDTT)[2] was held for the first

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

[1] https://www.gartner.com/en/documents/3904569/top-10-strategic-technology-trends-for-2019-digital-twin

[2] https://dtwin.ca

time while the International Conference on Model Driven Engineering Languages and Systems (MODELS) hosted the first International Workshop on Model Driven Engineering of Digital Twins (ModDiT).[3] Considering these premises, it is safe to position DT in the early majority phase of its adoption life cycle [2].

Despite their strategic relevance, the engineering of DTs is still an utterly complex process for organisations from any type of industry. How to create realistic models of DTs and how to ensure quick, flexible, and cost-effective adoption of this technology are only a few of the engineering challenges associated with DTs and their adoption [3], [4]. Software architectures (SAs) are cornerstones for the engineering of software-intensive systems like DTs. SAs represent holistic views of software systems comprising of software components, relationships among them, and properties of both [5]. SAs heavily influence the ability of software systems to fulfill both functional and quality requirements through design decisions enabling or inhibiting quality attributes(QAs) [6]. Besides, several studies found SAs to be crucial to several aspects of the engineering and development process including: understanding, reuse, construction, evolution, analysis, management, and communication [7]. As a matter of fact, only in 2021, the research community published 687 studies on architecting DTs. Given the growing interest of the SAs community in DTs, we believe that the time is ripe to create a structured and detailed snapshot of existing software architectural solutions for DTs. Such a snapshot would capture both commonalities and unique features of architectural solutions for DTs and support practitioners in creating DTs tailored to their specific needs and researchers in identifying trends and open challenges when architecting DTs.

In this work, we report on the planning, execution, and results of the first systematic mapping study (SMS) on architecting DTs. To the best of our knowledge, this work represents the first empirical study investigating DT architectures. According to Petersen, a SMS allows discovering research gaps and trends by providing a structure of the results published within a research area using visual summaries and mappings [8]. Starting from an initial set of 1630 peer-reviewed publications, we identified a final set of *140 primary studies*, which we analysed thoroughly following a meticulous data extraction, analysis, and synthesis process. To compensate for single method limitations and reduce possible threats to conclusion validity, we discussed the results of our study with experts in the SAs community using online, semi-structured, in-depth interviews [9]. A summary of the highlights resulting from our study is the following:

- The first peer-reviewed publications on SAs for DTs appeared in 2016. The majority of studies in this field are published as conference or journal papers.
- We identified 14 quality attributes that are relevant for DTs. We started from the ISO 25010 standard on software products quality and complemented it with additional attributes found in the primary studies or deemed relevant by the experts. Performance efficiency, maintainability, and compatibility are the top-mentioned attributes.
- 56.42% of the architectural solutions for digital twins are simple divisions of functionalities with data flow between them.
- Architectural solutions for DTs use a catalogue of 10 patterns either in isolation or in combination with other patterns. The layered and the service-oriented patterns are the most used ones.

The remainder of this paper is structured as follows. Section II compares our work with the related literature. Section III describes the research method in all its steps. Section IV describes the main threats to validity and related mitigation strategies. Section V presents the results of the vertical and orthogonal analyses. Section VI discusses the findings of our study together with the results of the expert interviews. Section VII concludes the paper and proposes possible future work.

## II. RELATED WORK

To the best of our knowledge, the work in this paper represents the first SMS that focuses on architecting DTs. Nevertheless, there exist some studies that shed light on specific DTs architectural solutions or aspects. Hereafter, we discuss some of these studies together with other existing secondary studies focusing on complementary aspects of engineering DTs and discuss them to motivate the need for our research.

In their paper, Harper *et al.* specified DTs as a key component in an Industrial Internet of Things (IIoT) ecosystem and discussed DTs architectures and standards [10]. They identified six architectural interactions (features) within the IIoT ecosystem as, e.g., interoperability, information model, and synchronization. Moreover, they proposed 9 architectural evaluation criteria. When discussing DT standards they pointed out ISO 23247 [11] as a standard for a Digital Twin manufacturing network. The ISO 23247 standard comprises of four parts including (i) requirements for developing DTs in manufacturing, (ii) reference architecture with functional views, (iii) list of attributes for manufacturing elements (iv) requirements for information exchange between elements. The scope of our work is broader as we classify and analyse architectural solutions with different levels of abstraction starting from reference model to SAs through reference architectures and patterns for the realisation of DTs.

Another study on DTs in the Internet of Things (IoT) context is proposed by Minerva *et al.* [12]. In their work, they surveyed main DTs definitions, specifications, and implementations. Besides, they attempted to consolidate main DT features and surveyed DT applicability in relevant IoT application scenarios. Both studies do not follow a systematic process and they only focus on a specific context, which is IoT.

Tao *et al.* reported on the state of the art and practice of DT applications and development [13]. They focused on the

[3] https://gemoc.org/events/moddit2021

review of 50 papers (from 2003 to 2018), eight patents, and the outcomes of some multinational corporations. Their paper outlined the key enabling technologies for DT modelling, simulation, data fusion, and service. Besides, it investigated possible DT applications in a product life-cycle and concluded that DT was mostly used for prognosis and health management (PHM). Tao *et al.* pointed out that one of the main practical challenges was the creation of a unified modelling method for DTs. The main difference between our work and the work by Tao *et al.* is the goal of the research as we focused on SAs for DTs. Besides, in contrast to Tao *et al.*, we used a meticulous process based on the guidelines for performing secondary studies in software engineering by Kichenham [14].

Barricelli *et al.* shed some light on DTs using a survey-based research [15]. They focused on collecting existing DTs definitions, investigating the main DTs characteristic, identifying the domains where DTs have been developed, and DTs open issues and challenges. The main characteristics of DT reported in their research are self-adaptation, self-parametrization, predictive analytic, and high-dimensional data coding techniques. Our work can be seen as complementary to the work by Barricelli *et al.* as some of the characteristics identified in their work, notably self-adaptation, may have a (direct) impact on the quality attributes driving the creation of SAs solutions for DT.

Fuller *et al.* provided a literature review that categorised existing studies by research areas [1]. Similar to Barricelli *et al.* they provided an assessment of the enabling technologies, challenges and open research for DT. In our work, we analysed publication trends with respect to publication years and venues. However, we only considered studies focusing on software architectural aspects of DT.

Perno *et al.* made a systematic literature review focusing on DT enablers and barriers in the process industry [3]. They used the identified barriers for building a model that relates them to the DT enabling technologies. Some of the issues identified in their study are related to security, system integration, development, and performance. Our research complements the one by Perno *et al.* as some of the challenges reported by them, security, interoperability, and performance, can be successfully addressed at the SA level using specific architectural solutions, e.g., patterns.

Rashed *et al.*provided a review on methodologies and techniques related to the realization of DT from a modeling perspective [16]. Besides, they collected challenges too. Some of the most reported modelling challenges are related to data management, real-time communication, and real-time modeling.

Rathore *et al.* emphasized the role of big data and artificial intelligence in the creation of DT-based systems using a systematic review on top of multidisciplinary electronic bibliographic databases [17]. The work by Rathore *et al.* is the first of its kind in terms of reviewing artificial intelligence and big data analytic techniques for DTs. Besides, they identified development tools that can facilitate DTs and designed a big data-driven and artificial intelligence enriched reference architecture. Our work differs from the research by Rathore *et al.* as we reviewed architectural aspects of DT-based systems.

Semerano *et al.* recently conducted a literature review providing an up-to-date and multi-perspective view of DTs [18]. They have defined and investigated four main topics covering DTs functions, technologies, life cycle and architecture. Among other things, they discussed DT architecture on a functional and component level taking in consideration reference models and reference architectures. Our work complements the results of their study as we dig deeper on the software perspective of DTs. We collect important aspects such as software architecture patterns and QAs. We provide a catalog of architectural patterns leveraged on the proposed solutions of the primary studies and map them to QAs identified for DTs.

It is worth mentioning that there are other secondary studies that review reference architecture of related technologies and initiatives such as Industry 4.0 [19]. Although some of the reference architectures for Industry 4.0 have specific modules or asset administration shells for managing the DTs, we consider Industry 4.0 and DTs as different and complementary concepts [20].Hence, we do not focus on architectural solutions for Industry 4.0.

Given the current body of literature on DTs, we consider our work to be complementary to the existing researches and helping researchers and practitioners in understanding the architectural aspects of DT-based systems.

## III. RESEARCH METHOD

We designed and conducted this study by following the guidelines for performing secondary studies in software engineering by Kitchenham [14] and Petersen *et al.* [8]. To compensate for single method limitations and reduce possible validity threats due to a lack of expert evaluation, we performed expert interviews using guidelines by Molleri *et al.* for performing surveys in software engineering [21]. Our research method consisted of three main phases: *planning*, *conducting* and *reporting*.

The main goals of the planning phase were (i) the identification of the needs for this systematic mapping study, (ii) the definition of the research goal and questions, and (iii) the definition of the protocol to be followed by the authors for carrying out the study.

During the conducting phase, we executed all the steps defined in the research protocol, which were *search and selection*, *definition of the classification framework*, *data extraction*, *data analysis* and *expert interviews*. In the search and selection step, we used the identified search string and performed an automatic search to the set of selected scientific databases and indexing systems. Through a rigorous selection process, we filtered the candidate studies to obtain the final set of primary studies. We complemented the automatic search with fully recursive forward and backward snowballing activities [22]. In the second step, we used the standard keywording process [23] for comparing and evaluating the primary studies as well as for defining a classification framework. We used the classification framework for analysing each primary study and extracting relevant information. This was an iterative process and we refined the framework every time we could not categorize the extracted information into the predefined categories. Eventually, we analysed the data extracted in the previous step to provide answers to the elicited research questions. We performed both quantitative and qualitative analyses using vertical and orthogonal analysis techniques. Besides, we conducted four online interviews with experts in the SAs and DTs domains for validating and discussing the main findings of our analyses.

In the documenting phase, we reported on possible threats to validity affecting this research and related mitigation strategies. Besides, we wrote this paper, which reports on the performed study and its results. We provide a public and complete replication package for supporting independent replication and verification of this study.[4]

### A. RESEARCH GOAL AND QUESTIONS

We used the Goal-Question-Metric perspectives [24] for defining the research goal of this study, which is: *identify, classify and analyse software architectural solutions for digital twins*. Starting from the above goal, we have identified the following research questions (RQs), which contribute to different and unique objectives of digital twin architectures.

*RQ1 – Which are the publication trends in software architectures for DTs?* DT is a research area with contributions from both industrial and academic research groups coming from different research domains and focusing on specific issues with different levels of detail and methodologies. By answering this research question, we (i) quantify the number of publications across the years to analyze the interest in SAs for DTs from researchers and practitioners, (ii) identify the publication venues where ideas and solutions are discussed, (iii) identify the type of research documented in the literature and (iv) identify means of validation. As a systematic mapping study aims at classifying and structuring a field of interest by categorizing and analyzing publication trends, this research question is pivotal for this study.

*RQ2 – Which are the most addressed quality attributes when architecting DTs ?* QAs are measurable properties that are used to assess the extent to which a software system satisfies the needs of its stakeholders [5]. These properties go beyond the system functionality and often represent the main causes for system redesigns: *"systems are frequently redesigned because they are difficult to maintain, port, or scale; or they are too slow, or they have been compromised by hackers"* [5]. In addition, QAs are the main carriers of those business goals that have an architectural relevance [5]. Complex software-intensive systems, such as DTs, exhibit a broad set of QAs that are hard to achieve in isolation. The SA of a system is profoundly influenced by QAs as the mapping of the system functionality into architectural elements determines satisfactory QA responses. By answering this research question, we identify and classify the set of QAs that have been treated in documented research when architecting DTs.

*RQ3 – Which are the software architecture solutions for DTs?* In recent years, SA solutions for DTs have been introduced by different organisations and at different levels of detail. This research question focused on characterising the architectural solutions proposed in the studies using the taxonomy proposed by Bass *et al.* [5]. This taxonomy classifies the solutions based on their completeness. Accordingly, architectural solutions may be categorised as reference models, reference architectures, architectural patterns, and software architecture. According to Bass *et al.* a reference model is the simplest representation of an architecture that illustrates, in box and line sketches, division of functionality together with data flow between the components of an architecture [5]. A reference architecture maps the functionalities defined in the reference model onto software elements [5]. An architectural pattern is a description of elements, their relation types together with a set of constraints [5]. A SA represents holistic views of a software system comprising of software components, relationships among them, and properties of both [5] By answering this research question, we capture and classify the current SAs solutions for DTs for helping researchers and practitioners to navigate the states of the art and practice to select, reuse and customise existing solutions according to their needs.

*RQ4 – Which is the catalogue of architectural patterns for DTs?* Architectural patterns [25], also known as architectural styles [5], emerged as important architecture solutions as they provide packaged strategies for solving known system problems. To this end, patterns define the types of architectural elements and their interaction. According to Bass *et al.*:"*quality attributes should be achieved using well-known architectural*" [5]. Hence patterns are crucial when archtiecting digital twins. Common, well-known architectural patterns are the layered pattern, the client-server pattern, the multi-tier pattern, etc. By answering this research question, we provide a comprehensive catalogue of patterns used by current DTs architectural solutions. Such catalogue can support researchers and practitioners in building DTs.

### B. SEARCH AND SELECTION PROCESS

In this phase, we identified the set of primary studies following the steps illustrated in Figure 1.

We started with an automatic search of four of the largest scientific databases and indexing systems in software engineering [26]: *IEEE Xplore Digital Library*, *ACM Digital Library*, *SCOPUS* and *Web of Science*. We selected the above sources based on their reputation as being effective means in supporting systematic studies in software engineering [27].

[4] The replication package is available at: https://anonymous.4open.science/r/architectingDTs_replicationPackage

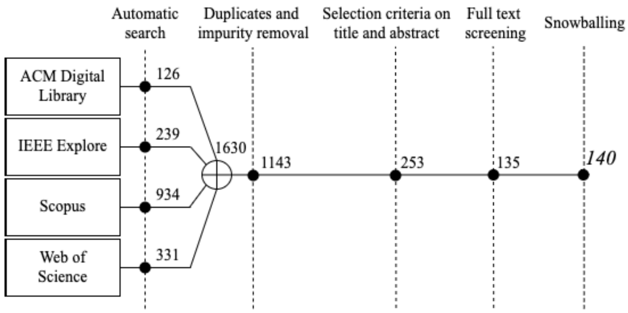

**FIGURE 1.** **Overview of the search and selection process.**

We queried the above sources using the following search string:

("Digital Twin" AND Architect*)

that we created starting from the elicited research questions. We decided to use a simple search string that helped us in collecting as many potential studies as possible and filter them manually. This helped us reducing possible threats to construct validity. The automatic search provided an initial set of 1630 studies, from which we removed studies not being research papers (impurities) and merged duplicates and obtained a new set of 1143 studies. At this point, we applied the following inclusion (IC) and exclusion (EC) criteria on the title, abstract and keywords of the studies:

- Inclusion criteria
  1) Studies describing at least one architectural solution for DTs.
  2) Studies evaluating their main contribution.
  3) Peer-reviewed studies [28].
  4) Studies written in English.
  5) Studies available as full-text.
- Exclusion criteria
  1) Secondary or tertiary studies.
  2) Studies published as tutorial papers, short papers (less than 4 pages), poster papers, editorials and manuals.

We followed the selection process proposed by Ali and Petersen [29] and selected only those studies satisfying all the ICs and no EC. We obtained a new set of 253 studies. We performed a second round of selection on the full text and selected 135 primary studies. Staring from these studies, we carried out closed recursive snowballing activities [22] for reducing possible threats to construct validity. As a result, we obtained the final set of *140* primary studies, which is attached as appendix of this paper.

### C. CLASSIFICATION FRAMEWORK AND DATA EXTRACTION

We constructed a well-defined classification framework (Table 1) to extract and classify the relevant information from the primary studies. The classification framework comprises of 4 facets, one for each RQ. For RQ1, we included standard information on the publication details. Besides, we included the type of research and validation according to the taxonomies by Wieringa *et al.* [30] and Shaw *et al.* [31], respectively. For RQ2, we classified the QAs starting from the taxonomy proposed in the ISO 25010 standard [32], which is the ISO standard defining the product quality model. We extended the ISO 25010 taxonomy with additional QAs found in the primary studies (if they were mentioned in more than 10 studies), which are: scalability, extendability and reconfigurability. Eventually, we discussed with the experts the extended taxonomy with the aim of identifying any missing QAs. For RQ3, we used the characterisation proposed by Bass *et al.* [5]. For RQ4, we carried out a *keywording* systematic process. Hence, we collected related keywords and concepts. Then we arranged these into groups similar to the sorting phase of the grounded theory methodology [33]. During the extraction phase, we collected further information that was deemed relevant, but that the classification framework did not capture yet. We reviewed the additional information and refined the framework if needed. We analysed again the primary studies according to the refined framework and extracted data.

### D. DATA ANALYSIS AND SYNTHESIS

We analysed and synthesised the extracted data according to the recommendations by Cruzes *et al.* [34]. We performed both vertical and orthogonal analyses. The former allows discovering information on each category of the classification framework. The later allows identifying possible relationships among different categories of the framework. For the vertical analysis, we analysed each primary study individually so to classify its features according to the classification framework. Later, we looked at the whole set of primary studies to reason about potential patterns following the process known as the line of argument synthesis [28]. For the orthogonal analysis, we grouped and cross-tabulated extracted data, and compared pairs of categories of the classification framework. We extracted and evaluated significant pair-wise correlations using contingency tables. We used a combination of content analysis [35] and narrative synthesis [36] for both vertical and orthogonal analyses.

### E. EXPERT INTERVIEWS

We conducted online, semi-structured, in-depth interviews for validating and discussing the main finding of our analyses. We chose semi-structured interviews for minimising possible threats to conclusion validity and for gathering more qualitative insights. We selected the experts based on their reputation in the SAs community and their experience with DT-based systems. The pool of experts included the following profiles:

- A professor in computer science with more than 20 years of experience in SA and autonomous systems.
- An associate professor in computer science with more than 25 years of experience in i) software quality requirements modeling, analysis and verification, ii) formal methods for (self-)adaptive dependable IT systems and iii) model-driven software engineering.

**TABLE 1.** Classification framework.

| Facet | Cluster | Category | Description | Value |
|---|---|---|---|---|
| RQ1 | Publication details | Authors | Identifies the list of authors | String |
| | | Publication title | Identifies the title of the study | String |
| | | Venue name | Identifies the name of the venue | String |
| | | Venue type | Identifies the type of venue | Workshop, conference, journal, book chapter |
| | | Year | Identifies the year of publication | Numeric value (e.g., 2020) |
| | Publication analysis | Research type | Identifies the type of the research according to the taxonomy in [30] | Solution proposal, validation research, evaluation research, philosophical paper, opinion papers, experience paper |
| | | Validation type | identifies the type of validation of the study according to the taxonomy in [31] | Analysis, experience, example, evaluation, persuasion, blatant assertion |
| RQ2 | – | Quality attributes | Identifies the quality attributes according to the ISO 25010 standard [32] and further QAs as listed in the studies | Functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, portability, extensibility, scalability, reconfigurability |
| RQ3 | – | Architectural solutions | Identifies the architectural solutions according to the classification in [5] | Architectural pattern, reference model, reference architecture, software architecture |
| RQ4 | – | Architectural patterns | Identifies the architectural patterns as listed in the studies | String |

- An assistant professor in computer science with more than 15 years of experience in performance and dependability modeling, analysis and verification. The expert is one of the organizers of the ModDiT workshop on DT.
- A software architect working for a Nordic manufacturer among the world leaders in the heavy lorries, trucks and buses production.

It is worth remarking that all the experts held, at least, a doctorate degree in computer science. Prior to the interview, each participant received the interview preparation sheet, where we outlined the main focus of the interview, its expected duration and the measures we took for ensuring privacy and confidentiality. All the interviews were carried out online using the Zoom platform[5] and performed by all researchers. We audio-recorded the interviews. The interviews contained four sections: one section per research question two to four, plus an additional section for final remarks and feedback from the expert. The interviews lasted between 30 and 45 minutes. Section VI discusses the results of the interviews.

## IV. THREATS TO VALIDITY

We conducted this research following well-established guidelines for systematic studies. These guidelines helped us in minimising possible threats to validity, in particular threats to internal and construct validity.

In systematic reviews, one of the main threats to external validity is that primary studies may be unrepresentative of the state of the art and practice. To mitigate such a threat, we targeted four different scientific databases and indexing systems. Besides, we complemented the automatic search with closed recursive backward and forward snowballing. These operations helped us in mitigating threats to construct validity, too. We excluded studies not written in English. However, we consider the validity threat of excluding relevant studies not written in English to be minimal as English is the *de-facto* standard language for scientific research in computer science and software engineering.

We mitigated internal validity threats concerning data analysis and synthesis using descriptive statistics. In addition, we cross-analysed different facets of the classification framework and performed sanity checks on the extracted data. These tasks helped us identify and solve potential issues on the consistency of the extracted data.

A possible threat to construct validity may be using a badly designed search string. We used a simple search string whose construction did not require particular attention.

To mitigate possible threats to conclusion validity, we provided a complete and public replication package, which allows reproducing each step of our study. The classification framework was defined using well-established taxonomies and collecting values from the set of primary studies. The framework was designed by all the authors who participated in the data extraction, analysis, and synthesis steps, too. Eventually, we discussed the results of our study with four experts in the fields of SAs and DT.

## V. RESULTS

In this section, we report on the results from the vertical and orthogonal analyses. It is worth mentioning that, for specific features, we extracted multiple values or no value at all from the studies. Hence, the total number of occurrences in the plots may differ from the total number of primary studies.

### A. PUBLICATION TRENDS (RQ1)

With respect to RQ1, we identified publication trends in terms of the trend over time and venue type. In addition, we identified the most common research types and validation types for studies in SAs for DTs. Figure 2 illustrates the distribution of the number of primary studies and their venue type over the years. The bars show the number of primary studies, while the lines the venue type.

According to the synthesised data, research on SAs for DTs is developing. The earliest study on SAs for DTs was published in 2016 (P74) and it focused on data management using web services. It is worth remarking, that our study

[5] https://zoom.us

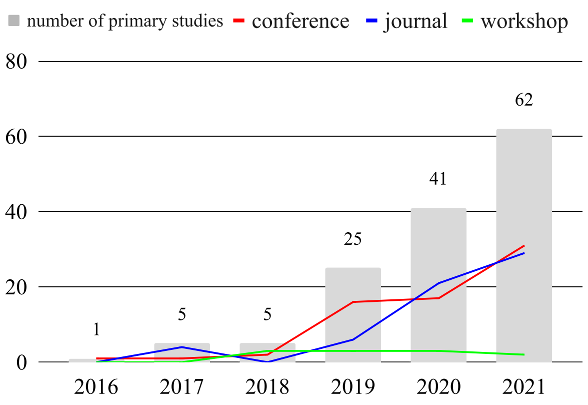

**FIGURE 2. Distribution of primary studies by year and by type of publication.**

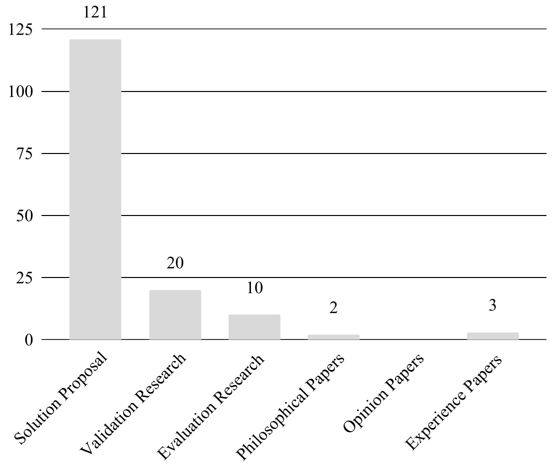

**FIGURE 3. Research type.**

focused on DTs architectures. Hence previous studies on related topics were outside the scope of our work. The number of primary studies in 2017 and 2018 remained reasonably low, with interest in data-driven (e.g., P9, P36, P79) and microservice-based architectures (e.g., P15) for DTs. Starting from 2019, the interest of the research community peeked. This is evident from the 5-, 9- and 12-fold increase in the number of primary studies in 2019, 2020 and 2021 as compared to 2018.[6] Considering the steep increment of primary studies in the last three years, we expect the number of publications to continue growing in the next few years. Concerning the venue type, the great majority of primary studies (91.42%) were published in conferences or journals. Conference is the most common venue type holding 48.57% of the primary studies, while workshop is the least common venue type with only 7.85% of the studies. In terms of venues, the primary studies have an extremely fragmented distribution and they are scattered across 85 different venues. The top three venues are the journal of IEEE Access, the Journal of Manufacturing Systems, and the Journal of CIRP Annals holding 5%, 4.2%, and 4.2% of the primary studies, respectively. We noticed that most of the top mentioned domains are manufacturing and production.

As shown in Figure 3 The most common type of research among the primary studies is solution proposal (86%). These types of studies contribute with (domain-specific) architectural solutions for the realisation of DTs. For instance, P5 proposes an architecture for logistic process planning in the automotive domain. P7 discussed a DTs approach for smart manufacturing focusing on energy consumption. The second and third most common types of research are validation research and evaluation research, which account for 14% and 7% of the primary studies, respectively. Studies in these categories are either a novel solution using experiments and prototypes (validation research) or using a real-world case study (evaluation research). For instance, P26 demonstrated the efficiency of the proposed DTs reference model for cloud-based cyber-physical systems implementing a telematics-based prototype for driving assistance. P30 used experiments for evaluating a 4-layer DTs architecture for industrial robots. P41 used an industrial use case from the logistic domain for demonstrating and analyzing the effectiveness of the proposed DTs. Eventually, experience (2%) and philosophical (1.4%) papers are the least reported types of research.

By referring to Figure 4 the top-ranked validation type is example holding 50% of the primary studies. Here, studies tend to be validated using simplified use cases or prototypes rather than real industrial systems. Other types of validation are experience (20.71%), analysis (15%), evaluation (11,4%), and persuasion (10%). The low percentage of persuasion validation type and the absence of blatant assertion suggest that studies on SAs for DTs tend to prefer rigorous validation methods. The research and validation type appears to be in line with the venue type as conferences and journals, which are commonly the best fitting venues for research efforts introducing and evaluating novel architectural solutions on DTs.

*Highlights – RQ1 Publication trends*

- Earliest published works on software architectural solutions for DT appeared in 2016.
- The 12-fold increase in peer-reviewed primary studies in 2021 with respect 2018 suggests that the interest of the community is growing.
- The main validation type for primary studies is validation by example.

## B. QUALITY ATTRIBUTES (RQ2)

This research question focused on identifying the catalogues of relevant QAs for DTs. These were either mentioned directly by the studies or could be inferred by the context.

[6] The final set of primary studies includes 1 study from 2022. We did not include this study in Figure 2 to avoid noise when interpreting the data, since the search and selection process was performed in February 2022.

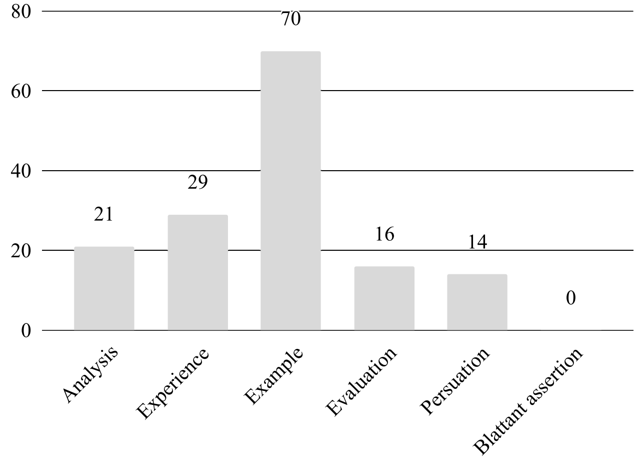

**FIGURE 4.** Validation type.

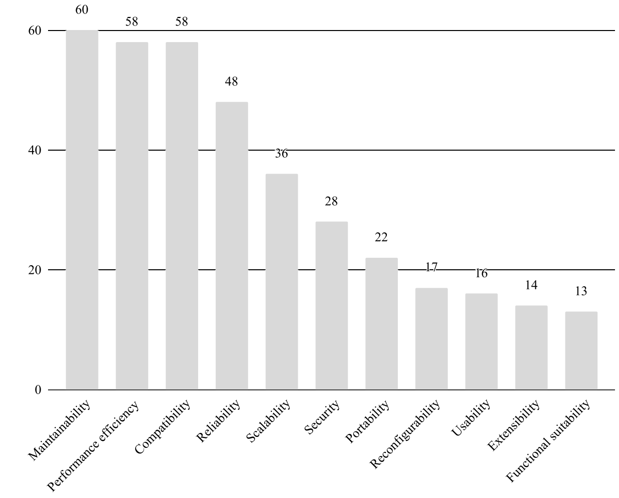

**FIGURE 5.** Quality attributes of DTs.

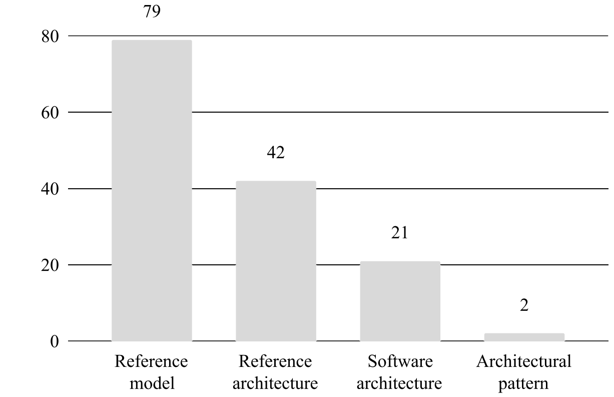

**FIGURE 6.** Architectural solutions for DTs.

We extracted the QAs starting for the taxonomy proposed by the ISO 25010 [37] standard on the software products quality. The ISO 25010 standard is composed of 8 attributes (characteristics), grouping 31 sub-attributes. As discussed in Section III-C, we considered three additional attributes, which were mentioned in more than 10 studies being: scalability, reconfigurability and extensibility. Figure 5 shows the occurrences of the attributes as mentioned in the 140 primary studies. For the sake of clarity, Figure 5 shows only the main attribute of ISO 25010 standard and omits the related sub-attributes.

The most mentioned quality attribute is maintainability (16.2%) followed by performance efficiency (15.6%), compatibility (15.6%), and reliability (12.9%). The authors of P34 conducted experiments to evaluate the performance efficiency of the proposed DTs approach supporting Industry 4.0 and showed that it could be scaled up for multiple applications with minimal impact on performance. P47 investigated the relationships between performance efficiency and maintainability. In particular, the authors claimed that improved resource utilisation in simulation activities would result in better maintainability of the system, as potential errors could be identified earlier and resolved before the system is deployed. Among the studies that discussed maintainability, 71% of them focused on the modularity sub-quality attribute. P74 and P75 focused on achieving compatibility using the service-oriented architecture (SOA) pattern. P34, P42, P46, P49, P68 explored the use of machine-to-machine communication technologies (e.g., OPC UA, MQTT), ontology languages (RDF, OWL), and semantic models for achieving compatibility. Other mentioned QAs were scalability (9.7%), security (7.5%), portability (5.9%), reconfigurability (4.7%), usability (4.3%), extensibility (3.7%) and functional suitability (3.5%). It is worth remarking that 25% of the primary studies focused on more than one quality attribute.

*Highlights – RQ2 Quality attributes*

- Maintainability, performance efficiency, and compatibility are the top-mentioned attributes.
- Scalability, reconfigurability and extensibility are important attributes mentioned by many primary studies, but not part of the ISO 25010 standard.

### C. ARCHITECTURAL SOLUTIONS (RQ3)

Figure 6 shows that the majority of the primary studies (56.42%) proposed a reference model. A reference model represents a division of functionality together with data flow between the components of a system. This result confirms that the interest in DTs is alive, although architectural solutions for DT-based systems are still evolving and are just presented as simplified models. P21 proposed a reference model of an intelligent DT in a cyber-physical production system that enabled the development of an autonomous system with decision-making capabilities based on artificial intelligence. Authors in P27 proposed a reference model for a DT of the human heart, called cardio twin. The model focused on monitoring and detecting problems via the cardio twin and accordingly operating on the heart. 30% of the primary studies investigated reference architectures for DTs. A reference architecture extends a reference model by mapping its functionalities onto software elements and the data flows between them. P24 discussed a reference architecture of DTs for smart manufacturing using open source technologies. P40 proposed a novel reference architecture for DTs and used the architecture for introducing the concept of DT as a service. This is achieved by integrating ThingWorx, industrial internet of things platform, and Vuforia, an augmented reality software development kit. 21 primary studies (15%) discussed SAs for DT-based systems. For example, P32 proposed a six-layer digital twin software architecture for power systems based on ontological modelling. P10 and P118 are the only primary studies that propose an architectural pattern for realising DTs. It is worth remarking that we categorised as architectural patterns those architectural solutions that only put constraints on system elements and their interactions. Hence, we did not classify into the pattern categories those patterns used for building more complex reference models, architectures and SAs.

*Highlights – RQ3 Architectural solutions*

- The majority of the solutions are proposed as reference models hence as a division of functionalities and data flow among the components.
- The increasing number of solutions proposed as reference and software architectures may indicate an evolution of architectural solutions for digital twins.

### D. ARCHITECTURAL PATTERNS (RQ4)

By answering this research question, we provide a catalog of architectural patterns for DTs. All the identified patterns were explicitly mentioned in the primary studies or could be inferred by the context. It is worth remarking that for this research question we considered all the patterns mentioned by any architectural solution hence even those patterns used to build reference models, reference architectures, and SAs. We identified a total of 10 unique architectural patterns (Figure 7).

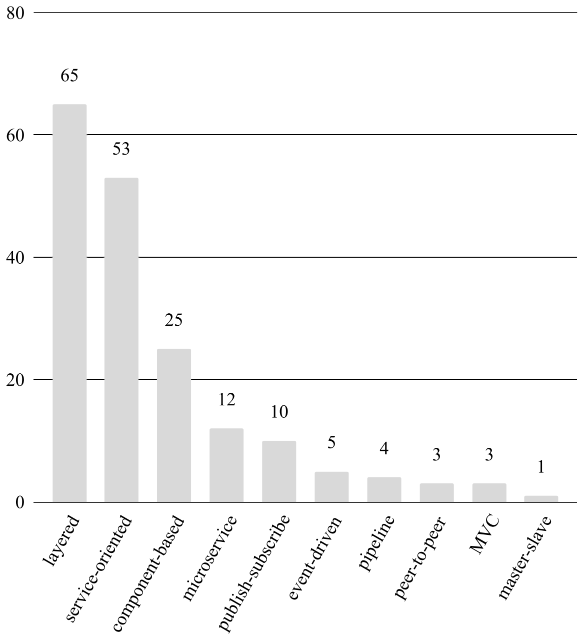

**FIGURE 7.** Architectural patterns for DTs.

The most recurring pattern is the layered pattern (35.9%), followed by SOA (29.3%), component-based (13.8%), and microservices (6.6%) patterns. P74, P75 motivated the use of layer pattern, as it can play the role of an interface between physical and virtual space, consolidating syntax and semantics of different perspectives. Authors in P74 realised SOA for DTs using web services to increase maintainability and interoperability, motivated by the fact that web services can be seen as modular units that provide functionalities over the network, facilitating data exchange and machine-to-machine interaction.

P24, P76, P77, and P78 investigated the use of microservices for implementing DT-based systems. The studies argued that the microservices pattern offers more flexibility in building data-intensive applications and enables cloud computing along with its benefits such as high availability, scalability, and reliability. In particular, P78 proposed a novel concept of DT fully based on microservices called digital dice. The remaining patterns are mentioned in a few studies. Publish-subscribe (5.5%), event-driven (2.7%), pipeline (2.2%), while peer-to-peer and model-view-controller (MVC) patterns are mentioned in three papers each (1.7%). There is only one study (P28) that uses the master-slave pattern. P3 and P33 investigated the use of the publish-subscribe pattern for managing the communication and ensuring deployability and scalability to DTs. P6 and P26 proposed the use of a peer-to-peer network for sharing and collecting data on cloud-based DTs.

We noticed that architectural patterns were not mentioned in isolation, but together with other patterns in 55 studies (38.19%). We found 17 different pattern combinations, 9 of them use the layered pattern in combination with, e.g., SOA, component-based. The layered and SOA combination is, by far, the most used combination and accounts for 27 studies. It is followed by the combination of layered and component-based (6 studies) and the combination of SOA and publish-subscribe (4 studies). The combination of layered and microservice is mentioned in 3 studies, while component-based and event-driven pattern combination is used in 2 studies. The remaining 8 combinations are each mentioned in one study, only.

*Highlights – RQ4 Architectural patterns*

- Architectural solutions for DTs use a catalogue of 10 patterns and layered and SOA are by far the most used ones.
- In half of the primary studies, patterns are used in combination with other patterns.

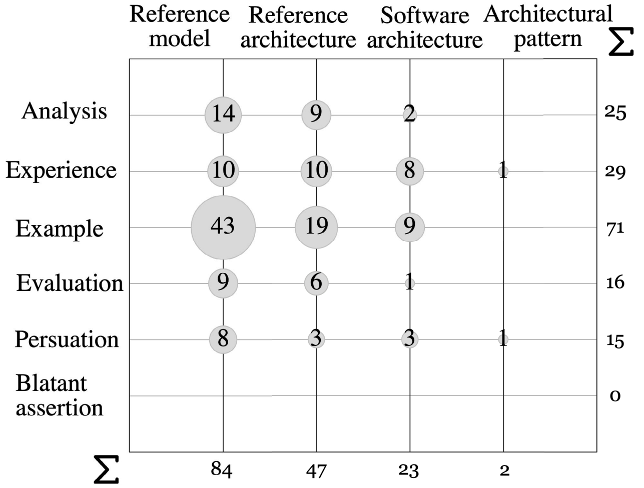

**FIGURE 8.** Architectural solutions vs validation type.

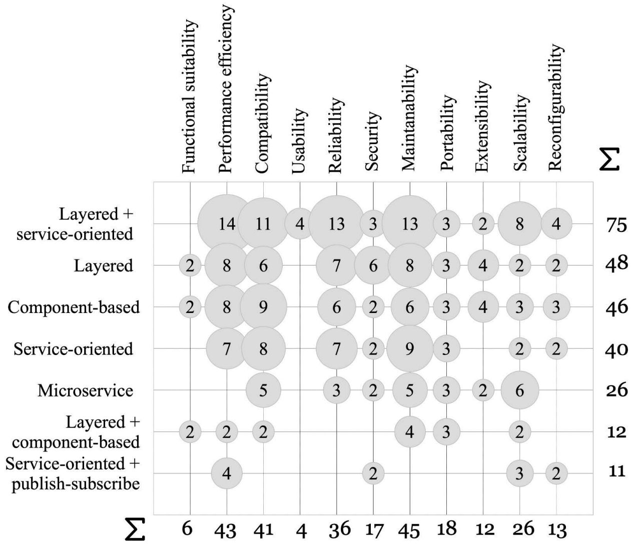

**FIGURE 9.** Quality attributes vs patterns.

### E. ARCHITECTURAL SOLUTIONS VS VALIDATION TYPE

Figure 8 illustrates the correlation between the architectural solutions proposed in the primary studies and the method used for validating them. The last column and the last row of Figure 8 shows the sum of the occurrences for each category. It is worth remarking that Figure 8 shows the absolute values of the occurrences. The interested reader can find the graphs with the normalised data in the replication package.

The orthogonal analysis highlighted a strong relation between architectural solutions in the form of reference models and the example validation technique. In fact, 51% of the studies proposing a reference model are validated using examples. Validation by example is the preferred validation technique for reference architecture and software architecture too, accounting for 40% and 50% of the primary studies, respectively. These results were expected and confirmed that architectural solutions for DTs are still maturing and have been mainly tested on prototypes or simplified use cases. Besides, it is also expected that abstract solutions such as reference models and architectures are not tested on real systems, which may require more detailed solutions. The orthogonal analysis showed that reference models have a strong relation with the analysis techniques, which is the chosen validation technique for 16.6% of them. We believe this is an encouraging result and may be an indication of the ripening of the field of SAs for DTs. Such an observation is also confirmed by the majority of the solutions being evaluated with rigorous techniques (experience, analysis and example) rather than with persuasion or blatant assertion techniques. Considering this, we expect to have a growing number of studies in the near future proposing software architectural solutions for DTs and validating them using experience techniques.

### F. QUALITY ATTRIBUTES VS PATTERNS

In the previous sections, we found out that layered and SOA were the most used patterns (alone or in combination with other patterns) and that performance efficiency and maintainability were the most addressed quality attributes. Using orthogonal analysis, we looked at the correlations between patterns and quality attributes. It is worth remarking that for this analysis, we took into account single patterns as well as combinations of patterns. Figure 9 presents the results of the orthogonal analysis on patterns and quality attributes. For the sake of readability, Figure 9 does not show values lower than 2. It is worth recalling that Figure 9 shows the absolute values and that the charts with the normalised data can be found in the replication package.

Analysing the correlation between patterns and quality attributes, we found out that the combination of layered and SOA are strongly related to performance efficiency (18.6%), reliability(17.3%) and maintainability (17.3%). Other mentioned attributes for the combination of layered and SOA patterns are compatibility, scalability and reliability. P74, P75, P18 and P53 proposed the use of the combination of the layered and SOA patterns as a way of reaching improved compatibility, maintainability and scalability of DTs. The layered pattern helps in managing the complexity by addressing different concerns separately (P75) while the SOA pattern allows to provide platform and device-independent systems supporting interoperable machine-to-machine interaction over a network (P74). P18 presents a six-layers reference architecture for aggregating DTs and providing configuration for DT using the cloud. The authors claimed that the use of the layered pattern would increase modularity and flexibility.

Microservice patterns are often mentioned together with maintainability (19.2%), compatibility (19.2%) and scalability (23%). For instance, P15 proposed a microservice architecture supporting the new generation of DT-based smart-factories where maintainability is achieved by containerizing and deploying microservices as single processes. P24 discussed the use of microservices as a way of reducing complexity and increasing flexibility in the proposed DT. The results in P76 and P77 show that the high reliability, maintainability and on-demand scalability offered by microservices can lead to increased latency, and in this case, a trade-off between performance and other QAs is needed.

The collected data show a clear relation between the component-based pattern and compatibility (19.5%), too. Primary studies P42 and P54 propose mechanisms supporting models and data integration while increasing compatibility. These mechanisms are built using predefined components such as models, data and simulation interfaces.

## VI. DISCUSSION

The synthesised data suggest that architecting digital twins is becoming increasingly important. Years 2020 and 2021 witnessed a 9- and 12-fold increment in the number of primary studies as compared to the year 2018. The steep increment of primary studies has been sided with a proportional growth of researches published as conference papers and journals. The interviewed experts acknowledged the research interest in DT architectural solutions as something positive and as a sign of a ripening research area. However, they suggested that this wealth of research may have a negative impact on the identification of reference architectural solutions. An expert commented that: *"such a lack of consensus and wealth of architectural solutions may bring experts in keeping proposing novel solutions rather than working on consolidating existing ones"*. Our vertical analysis on the architectural solutions and orthogonal analysis on the architectural solutions and validation types partly confirmed the above thought, although we saw an increasing trend in proposing SAs and leveraging more formal validation techniques, e.g., experience. An expert reflected that this could be the result of both research in digital twins shifting from system to software engineering and numerous collaborative research European projects on digital twins being funded in the last few years (e.g., Change2Twin,[7] IoTwins[8]).

According to our data, more than 50% of the primary studies proposed reference models for digital twins. We asked experts to comment on this finding in relation to the maturity and validity of the proposed architectural solutions. One expert reflected that such a result is not unexpected as from an academic point of view it is reasonable to provide more generic and high-level solutions that would benefit a wider community. On the contrary, SAs are usually developed taking into account the business and technical needs of organizations. Hence they are quite specific and tend not to be published as research articles. Another expert commented that reference models are actually desirable solutions from a company's perspective since they are economically cheaper to leverage for multiple business and technical analyses. This is especially true when companies approach new technologies such as digital twins and may use reference models to evaluate the readiness of the company with respect to that technology: *"when performing a preliminary analysis of whether a certain solution would fit our organisation, it is easier to assess certain qualities using reference models or architectures, which require a lower amount of time and money"*. Despite this, all interviewed experts agreed on the need of working towards formalised reference architectures for digital twins. A possible way forward that emerged during the interviews is to bound architectural solutions for digital twins to given contexts. In this respect, we propose to use the results from the work by Broy *et al.* in the automotive domain as guidelines for driving future efforts in architecting digital twins [38]. Broy *et al.* proposed a two-dimension classification of architectures based on the level of abstraction and details. Based on the level of detail, architectures are classified as meta, common, domain-specific, organisation-specific, product line, and product. Based on the abstraction, architectures are classified as functional, logical, and technical. The great majority of the primary studies proposed reference models for DTs that can be classified as functional and common/domain-specific. By narrowing the scope of the solutions (e.g., domain, organisation), researchers and practitioners may be able to propose reference architectures and even SAs spanning through all the levels of abstractions from functional to technical.

Despite the wealth of architectural solutions for digital twins, the data from the vertical analysis highlighted how the majority of them only use a few of the patterns being the layered, SOA, and component-based patterns. On the one hand, this could be interpreted as a sign of an implicit formalization of architectural solutions for DTs. On the other hand, as experts highlighted, this was quite expected especially considering the complexity of a DT system. For instance, the layered pattern is notoriously more efficient for decoupling concepts and tackling the complexity introduced by the many components of a digital twin. Besides, other patterns such as microservices may be less used *"because of the rigidity of some domains that can be safety-critical"* and demanding for certifiable solutions, which are harder to build using given patterns. During the interviews, one expert commented that would have been interesting to check the correlations between patterns. By addressing the expert comments, we found out that 50% of the patterns are used in combination with other patterns. The layered pattern remained the most used one even together with other patterns. Based on this finding, we analysed all the primary studies using the layered pattern (in isolation or in combination) with the aim of identifying a possible reference set of layers for DT SAs. Interestingly, we found out that most of the architectures using the layered pattern are based on four layers being: physical, data and network, virtual, and application layer. The physical layer is where machines, sensors, and actuators reside [P11]. It has connections with the data and network layer, which is responsible for providing connectivity on top of data management utilities (e.g., storage, cleaning) [P23]. The virtual layer builds on top of the data and network layer and provides for the virtual twin of the physical assets. Eventually, the application layer allows for the creation of e.g., configuration

[7] https://www.change2twin.eu

[8] https://www.iotwins.eu

and business-oriented services [P22, P52]. Besides, we found out that specific patterns are used in each of these four levels. For instance, microservices and publish-subscribe patterns [P10, P36] are used in the data and network layer, while SOA and component-based patterns are used within the application and virtual layers [P22, P41].

The data from the vertical and orthogonal analyses on quality attributes showed that there is a demand for digital twins to be efficient, compatible, maintainable, and reliable. During the interviews, all the experts agreed on the importance of extracting the quality attributes starting from the taxonomy defined in the ISO25010 standard of software product quality. However, there was a general consensus on the need of refining such a taxonomy with quality attributes meaningful for digital twins architectures. One expert even argued that the *"standard could be outdated"*. During the data extraction, we refined the taxonomy in ISO25010 with some other attributes being scalability, extensibility, and reconfigurability. In the interviews, we presented the extended taxonomy and asked the experts about our proposed extension and about possible quality attributes omitted in the taxonomy and deemed as relevant for digital twins. All the experts agreed on the importance of having scalability in a possible taxonomy for DTs. We had conflicting opinions on extensibility. One expert argued that extensibility could be captured by maintainability. However, another expert did not agree with such a view. Because of this lack of consensus, we decided to keep extensibility as an additional QA. An expert commented that reconfigurability may stand at the intersection between functional suitability and maintainability categories. However, the expert also found reconfigurability to be tightly coupled with variability and the exported suggested that, in the case of extensive sets of product variants, reconfigurability should be promoted to a separated category. One expert mentioned trustability, *"a user trusts that a virtual model is representing what is happening in the real twin"*, resilience [39] and robustness [40] as important attributes not captured by the extended taxonomy. While resilience may be represented within the reliability category [40], we agreed that trustability and robustness are not captured, but still crucial for digital twin systems. In fact, robustness is defined as the ability of a system to be resilient with respect to external or environmental faults [40]. The same expert mentioned that another important aspect is the degree of alignment between the digital and the physical twins. We found this aspect to be crucial as misalignment among twins may lead to the uselessness of the entire digital twin. However, we think that such concern may be captured by the functional suitability category. Another expert mentioned verifiability as a possible additional quality attribute. The expert recognised that the standard has testability as an attribute, but the expert found testability to be a subcategory of the verifiability as an architecture can be verified before it can be tested.

Based on the collected data and expert interviews, we elicited the following set of important quality attributes for digital twins: performance efficiency, maintainability, compatibility, reliability, security, portability, usability, functional suitability, scalability, reconfigurability, extensibility, trustability, robustness, and verifiability.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we reported on the planning, execution, and results of a systematic mapping study providing a snapshot of software architectural solutions for DTs. From an initial set of 1630 peer-reviewed publications, we selected 140 primary studies, which we analysed using precise data extraction, analysis, and synthesis process. The main highlights of our results are:

- The first peer-reviewed publication on software architectures for digital twins appeared in 2016. Most of the primary studies in this field are published as conference or journal papers.
- Starting from the ISO 25010 standard on the software products quality, we elicited a catalogue of 14 quality attributes that are relevant for DTs. Three of these attributes were not part of the standard but were mentioned by a high number of studies. The other three attributes were added after the interviews with experts. Among these attributes, performance efficiency, maintainability and compatibility are the top mentioned ones.
- 56.42% of the architectural solutions for digital twins are reference models. One possible direction for increasing the rigour of architectural solutions for digital twins can be to focus on specific domains or product families.
- The analysed architectural solutions used a catalogue of 10 patterns. Half of the primary studies used a combination of patterns. The layered and the SOA patterns are the most used ones. This may be interpreted as an implicit sign of conformance of architectural solutions for the digital twin.

We are considering extending this work into a multi-vocal study for entailing the so-called "grey literature". This would enable incorporating trends coming from other communities (like industry and practitioners in general) which have significant contributions to the field of digital twin and their architectures. In addition, we are considering investigating and proposing software architectures for digital twins for realising Industry 4.0 and beyond.

## REFERENCES

[1] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE Access*, vol. 8, pp. 108952–108971, 2020.

[2] G. M. Beal and J. M. Bohlen, "The diffusion process," Agricult. Exp. Station, Iowa State College, Ames, IA, USA, Tech. Rep. No. 761-2016-51585, 1957, pp. 111–1211956.

[3] M. Perno, L. Hvam, and A. Haug, "Enablers and barriers to the implementation of digital twins in the process industry: A systematic literature review," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2020, pp. 959–964.

[4] Y. Zheng, L. Chen, X. Lu, Y. Sen, and H. Cheng, "Digital twin for geometric feature online inspection system of car body-in-white," *Int. J. Comput. Integr. Manuf.*, vol. 34, nos. 7–8, pp. 752–763, Aug. 2021.

[5] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*. Reading, MA, USA: Addison-Wesley, 2003.

[6] J. Ingeno, *Software Architect's Handbook: Become a Successful Software Architect by Implementing Effective Architecture Concepts*. Birmingham, U.K.: Packt Publishing, 2018.

[7] D. Garlan, ''Software architecture,'' Carnegie Mellon Univ., Forbes Avenue Pittsburgh, Pittsburgh, PA, USA, Tech. Rep. CMU-CS-94-166, 2008.

[8] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, ''Systematic mapping studies in software engineering,'' *Proc. 12th Int. Conf. Eval. Assessment Softw. Eng.*, vol. 17, Jun. 2008, pp. 1–10.

[9] F. Shull, J. Singer, and D. I. Sjøberg, *Guide to Advanced Empirical Software Engineering*. London, U.K.: Springer, 2007.

[10] K. E. Harper, C. Ganz, and S. Malakuti, ''Digital twin architecture and standards,'' *IIC J. Innov.*, vol. 12, pp. 72–83, Nov. 2019.

[11] *Digital twin framework for manufacturing*, Standard ISO 23247-1:2021, 2021. [Online]. Available:https://www.iso.org/standard/75066.html

[12] R. Minerva, G. M. Lee, and N. Crespi, ''Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models,'' *Proc. IEEE*, vol. 108, no. 10, pp. 1785–1824, Oct. 2020.

[13] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, ''Digital twin in industry: State-of-the-art,'' *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[14] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, ''Systematic literature reviews in software engineering—A systematic literature review,'' *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2009.

[15] B. R. Barricelli, E. Casiraghi, and D. Fogli, ''A survey on digital twin: Definitions, characteristics, applications, and design implications,'' *IEEE Access*, vol. 7, pp. 167653–167671, 2019.

[16] A. Rasheed, O. San, and T. Kvamsdal, ''Digital twin: Values, challenges and enablers from a modeling perspective,'' *IEEE Access*, vol. 8, pp. 21980–22012, 2020.

[17] M. M. Rathore, S. A. Shah, D. Shukla, E. Bentafat, and S. Bakiras, ''The role of AI, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities,'' *IEEE Access*, vol. 9, pp. 32030–32052, 2021.

[18] C. Semeraro, M. Lezoche, H. Panetto, and M. Dassisti, ''Digital twin paradigm: A systematic literature review,'' *Comput. Ind.*, vol. 130, Sep. 2021, Art. no. 103469. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03218786

[19] E. Y. Nakagawa, P. O. Antonino, F. Schnicke, R. Capilla, T. Kuhn, and P. Liggesmeyer, ''Industry 4.0 reference architectures: State of the art and future trends,'' *Comput. Ind. Eng.*, vol. 156, Jun. 2021, Art. no. 107241. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835221001455

[20] F. Pires, A. Cachada, J. Barbosa, A. P. Moreira, and P. Leitão, ''Digital twin in industry 4.0: Technologies, applications and challenges,'' in *Proc. IEEE 17th Int. Conf. Ind. Inform. (INDIN)*, vol. 1, Jul. 2019, pp. 721–726.

[21] J. S. Molléri, K. Petersen, and E. Mendes, ''Survey guidelines in software engineering: An annotated review,'' in *Proc. 10th ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas.*, Sep. 2016, pp. 1–6.

[22] C. Wohlin, ''Guidelines for snowballing in systematic literature studies and a replication in software engineering,'' in *Proc. EASE*, 2014, pp. 1–10.

[23] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, ''Systematic mapping studies in software engineering,'' in *Proc. EASE*, Jun. 2008, pp. 1–10.

[24] V. R. Basili, G. Caldiera, and H. D. Rombach, ''The goal question metric approach,'' in *Encyclopedia of Software Engineering*, vol. 1. Hoboken, NJ, USA: Wiley, 1994, pp. 528–532.

[25] M. Richards, *Software Architecture Patterns*, vol. 4. Sebastopol, CA, USA: O'Reilly Media, 2015.

[26] B. Kitchenham and P. Brereton, ''A systematic review of systematic review process research in software engineering,'' *Inf. Softw. Technol.*, vol. 55, no. 12, pp. 2049–2075, Dec. 2013.

[27] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, ''Lessons from applying the systematic literature review process within the software engineering domain,'' *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, Apr. 2007.

[28] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Berlin, Germany: Springer, 2012.

[29] N. B. Ali and K. Petersen, ''Evaluating strategies for study selection in systematic literature studies,'' in *Proc. ESEM*, 2014, pp. 1–4.

[30] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, ''Requirements engineering paper classification and evaluation criteria: A proposal and a discussion,'' *Requirements Eng.*, vol. 11, no. 1, pp. 102–107, Mar. 2006.

[31] M. Shaw, ''What makes good research in software engineering?'' *Int. J. Softw. Tools Technol. Transf.*, vol. 4, pp. 1–7, Oct. 2002.

[32] *Systems and Software Engineering—Systems And Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models*, Standard ISO/IEC 25010:2011, 2011. [Online]. Available:https://iso25000. com/index.php/en/iso-25000-standards/iso-25010

[33] K. Charmaz and L. L. Belgrave, ''Grounded theory,'' in *The Blackwell Encyclopedia of Sociology*. Hoboken, NJ, USA: Wiley, 2007.

[34] D. S. Cruzes and T. Dyba, ''Recommended steps for thematic synthesis in software engineering,'' in *Proc. ESEM*, Sep. 2011, pp. 275–284.

[35] R. Franzosi, *Quantitative Narrative Analysis*. Newbury Park, CA, USA: Sage, 2010.

[36] M. Rodgers, A. Sowden, M. Petticrew, L. Arai, H. Roberts, N. Britten, and J. Popay, ''Testing methodological guidance on the conduct of narrative synthesis in systematic reviews: Effectiveness of interventions to promote smoke alarm ownership and function,'' *Evaluation*, vol. 15, no. 1, pp. 49–73, Jan. 2009.

[37] *IEEE Standard for a Software Quality Metrics Methodology*, IEEE Standard 1061-1992, 1993.

[38] M. Broy, M. Gleirscher, P. Kluge, W. Krenzer, S. Merenda, and D. Wild, *Automotive Architecture Framework: Towards a Holistic and Standardised System Architecture Description*, IBM Corporation, Technische Universität München, München, Germany, Tech. Rep. TUM-I0915, 2009.

[39] A. M. Madni and S. Jackson, ''Towards a conceptual framework for resilience engineering,'' *IEEE Syst. J.*, vol. 3, no. 2, pp. 181–191, Jun. 2009.

[40] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, ''Basic concepts and taxonomy of dependable and secure computing,'' *IEEE Trans. Depend. Sec. Comput.*, vol. 1, no. 1, pp. 11–33, Jan./Mar. 2004.

## PRIMARY STUDIES

[P1] Bazaz *et al.*, ''5-dimensional definition for a manufacturing digital twin,'' *Procedia Manufacturing*, 2019.

[P2] Hasan *et al.*, ''A blockchain-based approach for the creation of digital twins,'' *IEEE Access*, 2020.

[P3] Harrison *et al.*, ''A connective framework to support the lifecycle of cybe-physical production systems,'' *Proceedings of the IEEE*, 2021.

[P4] Erkoyuncu *et al.*, ''A design framework for adaptive digital twins,'' *CIRP Annals*, 2020.

[P5] Guerreiro *et al.*, ''A digital twin for intra-logistics process planning for the automotive sector supported by big data analytics,'' *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, 2019.

[P6] Bottani *et al.*, ''A digital twin model of a pasteurization system for food beverages: Tools and architecture,'' *IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2020.

[P7] Barenji *et al.*, ''A digital twin-driven approach towards smart manufacturing: reduced energy consumption for a robotic cellular,'' *International journal of computer integrated manufacturing*, 2020.

[P8] Lim *et al.*, ''A digital twin-enhanced system for engineering product family design and optimization,'' *Journal of Manufacturing Systems*, 2020.

[P9] Angrish *et al.*, ''A flexible data schema and system architecture for the virtualization of manufacturing machines (vmm),'' *Journal of Manufacturing Systems*, 2017.

[P10] Malakuti *et al.*, ''A four-layer architecture pattern for constructing and managing digital twins,'' *European conference on software architecture (ECSA)*, 2019.

[P11] Zheng *et al.*, ''A generic tri-model-based approach for product-level digital twin development in a smart manufacturing environment,'' *Robotics and Computer-Integrated Manufacturing*, 2020.

[P12] Shangguan *et al.*, ''A hierarchical digital twin model framework for dynamic cyber-physical system design,'' *Proceedings of the 5th International Conference on Mechatronics and Robotics Engineering*, 2019.

[P13] Negri *et al.*, ''A machine state-based digital twin development methodology,'' *Proceedings of the Summer School Francesco Turco*, 2019.

[P14] Schroeder *et al.*, ''A methodology for digital twin modeling and deployment for industry 4.0,'' *Proceedings of the IEEE*, 2020.

[P15] Ciavotta *et al.*, ''A microservice-based middleware for the digital factory,'' *Procedia Manufacturing*, 2017.

[P16] Chevallier *et al.*, "A reference architecture for smart building digital twin," *CEUR Workshop Proceedings*, 2020.

[P17] Moyne *et al.*, "A requirements driven digital twin framework: Specification and opportunities," *IEEE Access*, 2020.

[P18] Redelinghuys *et al.*, "International workshop on service orientation in holonic and multi-agent manufacturing," *IEEE Access*, 2019.

[P19] Redelinghuys. *et al.*, "A six-layer architecture for the digital twin: a manufacturing case study implementation," *Journal of Intelligent Manufacturing*, 2020.

[P20] Qamsane *et al.*, "A unified digital twin framework for real-time monitoring and evaluation of smart manufacturing systems," *IEEE International Conference on Automation Science and Engineering*, 2019.

[P21] Talkhestani *et al.*, "An architecture of an intelligent digital twin in a cyber-physical production system," *At-Automatisierungstechnik*, 2019.

[P22] Mostafa *et al.*, "An effective architecture of digital twin system to support human decision making and ai-driven autonomy," *Concurrency Computation Practice and Experience*, 2020.

[P23] Laamarti *et al.*, "An iso/ieee 11073 standardized digital twin framework for health and well-being in smart cities," *IEEE Access*, 2020.

[P24] Damjanovic-Behrendt *et al.*, "An open source approach to the design and implementation of digital twins for smart manufacturing," *International Journal of Computer Integrated Manufacturing*, 2019.

[P25] Merkle *et al.*, "Architecture of a digital twin for enabling digital services for battery systems," *IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, 2019.

[P26] Alam *et al.*, "C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE Access*, 2017.

[P27] Martinez-Velazquez *et al.*, "Cardio twin: A digital twin of the human heart running on the edge," *Medical Measurements and Applications, (MeMeA)*, 2019.

[P28] Merkle, "Cloud-based battery digital twin middleware using model-based development," *International Symposium on Computer Science and Intelligent Control*, 2019.

[P29] Hoebert *et al.*, "Cloud-based digital twin for industrial robotics," *International Conference, HoloMAS*, 2019.

[P30] Anton *et al.*, "Cloud-based digital twin for robot integration in intelligent manufacturing systems," *International Conference on Robotics in Alpe-Adria Danube Region (RAAD)*, 2020.

[P31] Abburu *et al.*, "Cognitwin - hybrid and cognitive digital twins for the process industry," *IEEE International Conference on Engineering, Technology and Innovation, (ICE/ITMC)*, 2020.

[P32] Andryushkevich *et al.*, "Composition and application of power system digital twins based on ontological modeling," *International Conference on Industrial Informatics (INDIN)*, 2019.

[P33] Lin *et al.*, "Concept design of a system architecture for a manufacturing cyber-physical digital twin system," *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2020.

[P34] Bamunuarachchi *et al.*, "Cyber twins supporting industry 4.0 application development," *International Conference on Advances in Mobile Computing and Multimedia*, 2020.

[P35] Ala-Laurinaho *et al.*, "Data link for the creation of digital twins," *IEEE Access*, 2020.

[P36] Yun *et al.*, "Data-centric middleware based digital twin platform for dependable cyber-physical systems," *International Conference on Ubiquitous and Future Networks (ICUFN)*, 2017.

[P37] Ding *et al.*, "Defining a digital twin-based cyber-physical production system for autonomous manufacturing in smart shop floors," *International Journal of production research*, 2019.

[P38] Eyre *et al.*, "Demonstration of an industrial framework for an implementation of a process digital twin," *International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, 2018.

[P39] Barosan *et al.*, "Development of a virtual simulation environment and a digital twin of an autonomous driving truck for a distribution center," *European Conference on Software Architecture (ECSA)*, 2020.

[P40] Aheleroff *et al.*, "Digital twin as a service (dtaas) in industry 4.0: An architecture reference model," *Advanced Engineering Informatics*, 2021.

[P41] Pan *et al.*, "Digital twin based real-time production logistics synchronization system in a multi-level computing architecture," *Journal of Manufacturing Systems*, 2020.

[P42] Talkhestani *et al.*, "Digital twin of manufacturing systems: A case study on increasing the efficiency of reconfiguration," *At-Automatisierungstechnik*, 2020.

[P43] Liu *et al.*, "Digital twin-based designing of the configuration, motion, control, and optimization model of a flow-type smart manufacturing system," *Journal of Manufacturing Systems*, 2020.

[P44] Wang *et al.*, "Digital twin-based framework for green building maintenance system," *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2020.

[P45] Han *et al.*, "Digital twin-driven 3d visualization monitoring and traceability system for general parts in continuous casting machine," *Journal of Advanced Mechanical Design, Systems and Manufacturing*, 2020.

[P46] Zhang *et al.*, "Dynamic resource allocation optimization for digital twin-driven smart shopfloor," *IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2018.

[P47] Lin *et al.*, "Efficient container virtualization-based digital twin simulation of smart industrial systems," *Journal of Cleaner Production*, 2021.

[P48] Wang *et al.*, "Event-driven online machine state decision for energy-efficient manufacturing system based on digital twin using max-plus algebra," *Journal of Sustainability*, 2019.

[P49] Steindl *et al.*, "Generic digital twin architecture for industrial energy systems," *Journal of Applied Sciences*, 2020.

[P50] Kamath *et al.*, "Industrial iot and digital twins for a smart factory : An open source toolkit for application design and benchmarking," *Proceedings of Global Internet of Things Summit (GIoTS)*, 2020.

[P51] Bolotov *et al.*, "Information model and software architecture for the implementation of the digital twin of the turbine rotor," *Journal of Physics: Conference Series*, 2019.

[P52] Zhang *et al.*, "Information modeling for cyber-physical production system based on digital twin and automationml," *International Journal of Advanced Manufacturing Technology*, 2020.

[P53] Răileanu *et al.*, "Integrating the digital twin of a shop floor conveyor in the manufacturing control system," *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, 2020.

[P54] Steinmetz *et al.*, "Internet of things ontology for digital twin in cyber physical systems," *Brazilian Symposium on Computing System Engineering (SBESC)*, 2018.

[P55] Parri *et al.*, "Jarvis, a hardware/software framework for resilient industry 4.0 systems," *International Workshop on Software Engineering for Resilient Systems*, 2019.

[P56] Altun *et al.*, "Liberalization of digital twins of iot-enabled home appliances via blockchains and absolute ownership rights," *IEEE Communications Magazine*, 2019.

[P57] Min *et al.*, "Machine learning based digital twin framework for production optimization in petrochemical industry," *International Journal of Information Management*, 2019.

[P58] Bibow *et al.*, "Model-driven development of a digital twin for injection molding," *International Conference on Advanced Information Systems Engineering (CAiSE)*, 2020.

[P59] Rivera *et al.*, "On the engineering of iot-intensive digital twin software systems," *International Conference on Software Engineering (ICSE)*, 2020.

[P60] Shahriar *et al.*, "Opendt: A reference framework for service publication and discovery using remote programmable digital twins," *International Conference on Services Computing (SCC)*, 2020.

[P61] Rossini *et al.*, "Replica: A solution for next generation iot and digital twin based fault diagnosis and predictive maintenance," *CEUR Workshop Proceedings*, 2020.

[P62] Xia *et al.*, "Research on construction method of digital twin workshop based on digital twin engine," *International Conference on Advances in Electrical Engineering and Computer Applications, (AEECA)*, 2020.

[P63] Korth *et al.*, "Simulation-ready digital twin for realtime management of logistics systems," *IEEE International Conference on Big Data, Big Data*, 2019.

[P64] Borangiu *et al.*, "Smart manufacturing control with cloud-embedded digital twins," *International Conference on System Theory, Control and Computing (ICSTCC)*, 2020.

[P65] Li *et al.*, "Sustainability assessment of intelligent manufacturing supported by digital twin," *IEEE Access*, 2020.

[P66] Park *et al.*, "Tila: Twin-in-the-loop architecture for cyber-physical production systems," *IEEE International Conference on Computer Design (ICCD)*, 2019.

[P67] Dalibor *et al.*, "Towards a model-driven architecture for interactive digital twin cockpits," *International Conference on Conceptual Modeling*, 2020.

[P68] Lehner *et al.*, "Towards a reference architecture for leveraging model repositories for digital twins," *International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020.

[P69] Eckhart *et al.*, "Towards security-aware virtual environments for digital twins," *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*, 2018.

[P70] Hugues *et al.*, "Twinops - devops meets model-based engineering and digital twins for the engineering of cps," *International Conference on Model Driven Engineering Languages and Systems (MODELS)*, 2020.

[P71] Castelli *et al.*, "Urban intelligence: A modular, fully integrated, and evolving model for cities digital twinning," *International Conference on Smart Cities: Improving Quality of Life using ICT, IoT and AI (HONET-ICT)*, 2019.

[P72] Kaigom *et al.*, "Value-driven robotic digital twins in cyber-physical applications," *IEEE Transactions on Industrial Informatics*, 2020.

[P73] Yildiz *et al.*, "Virtual factory: Digital twin based integrated factory simulations," *Procedia CIRP*, 2020.

[P74] Schroeder *et al.*, "Visualising the digital twin using web services and augmented reality," *IEEE International Conference on Industrial Informatics (INDIN)*, 2016.

[P75] Park *et al.*, "Vredi: virtual representation for a digital twin application in a work-center-level asset administration shell," *Journal of Intelligent Manufacturing*, 2020.

[P76] Alaasam *et al.*, "Stateful stream processing for digital twins: microservice-based kafka stream dsl," *International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, 2019.

[P77] Preuveneers *et al.*, "Robust digital twin compositions for industry 4.0 smart manufacturing systemsr," *IEEE 22nd International Enterprise Distributed Object Computing Workshop (EDOCW)*, 2018.

[P78] Mena *et al.*, "Digital dices: Towards the integration of cyber-physical systems merging the web of things and microservices," *International Conference on Model and Data Engineering*, 2019.

[P79] Tao *et al.*, "Digital twin-driven product design, manufacturing and service with big data," *International Journal of Advanced Manufacturing Technology*, 2017.

[P80] Mostafa *et al.*, "Digital twin conceptual model within the context of internet of things," *Future Internet Journal*, 2020.

[P81] Pan *et al.*, "A bim-data mining integrated digital twin framework for advanced project management," *Automation in Construction*, 2021.

[P82] Sellitto *et al.*, "A cyber security digital twin for critical infrastructure protection: The intelligent transport system use case," *IFIP Working Conference on The Practice of Enterprise Modeling*, 2021.

[P83] Farsi *et al.*, "A digital twin architecture for effective product lifecycle cost estimation," *Procedia CIRP*, 2021.

[P84] Mourtzis *et al.*, "A digital twin architecture for monitoring and optimization of fused deposition modeling processes," *Procedia CIRP*, 2021.

[P85] Chaux *et al.*, "A digital twin architecture to optimize productivity within controlled environment agriculture," *APPLIED SCIENCES-BASEL*, 2021.

[P86] Ricondo *et al.*, "A digital twin framework for the simulation and optimization of production systems," *Procedia CIRP*, 2021.

[P87] Barenji *et al.*, "A digital twin-driven approach towards smart manufacturing: reduced energy consumption for a robotic cell," *International Journal of Computer Integrated Manufacturing*, 2021.

[P88] Fan *et al.*, "A digital-twin visualized architecture for flexible manufacturing system," *Journal of Manufacturing Systems*, 2021.

[P89] Ouahabi *et al.*, "A distributed digital twin architecture for shop floor monitoring based on edge-cloud collaboration," *International Conference on Transportation and Smart Technologies, TST*, 2021.

[P90] Douthwaite *et al.*, "A modular digital twinning framework for safety assurance of collaborative robotics," *FRONTIERS IN ROBOTICS AND AI*, 2021.

[P91] Barbieri *et al.*, "A virtual commissioning based methodology to integrate digital twins into manufacturing systems," *PRODUCTION ENGINEERING-RESEARCH AND DEVELOPMENT*, 2021.

[P92] Yitmen *et al.*, "An adapted model of cognitive digital twins for building lifecycle management," *Applied Sciences (Switzerland)*, 2021.

[P93] Joseph *et al.*, "An aggregated digital twin solution for human-robot collaboration in industry 4.0 environments," *Studies in Computational Intelligence*, 2021.

[P94] Jacoby *et al.*, "An approach for industrie 4.0-compliant and data-sovereign digital twins realization of the industrie 4.0 asset administration shell with a data-sovereignty extension," *At-Automatisierungstechnik*, 2021.

[P95] Malakuti *et al.*, "An architecture and information meta-model for back-end data access via digital twins," *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021.

[P96] Vrabic *et al.*, "An intelligent agent-based architecture for resilient digital twins in manufacturing," *CIRP Annals*, 2021.

[P97] Rolo *et al.*, "Application of a simulation-based digital twin for predicting distributed manufacturing control system performance," *Applied Sciences (Switzerland)*, 2021.

[P98] Schnicke *et al.*, "Architecture blueprint enabling distributed digital twins," *ACM International Conference Proceeding Series*, 2021.

[P99] Li *et al.*, "Blockchain-based digital twin sharing platform for reconfigurable socialized manufacturing resource integration," *INTERNATIONAL JOURNAL OF PRODUCTION ECONOMICS*, 2021.

[P100] Li.M *et al.*, "Blockchain-enabled digital twin collaboration platform for heterogeneous socialized manufacturing resource management," *INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 2021.

[P101] Vogt *et al.*, "Concept and architecture for information exchange between digital twins of the product (cps) and the production system (cpps)," *Procedia CIRP*, 2021.

[P102] Liu *et al.*, "Construction method of shop-floor digital twin based on mbse," *JOURNAL OF MANUFACTURING SYSTEMS*, 2021.

[P103] Yan *et al.*, "Construction of digital twin ecosystem for coal-fired generating units," *Journal of Physics: Conference Series*, 2021.

[P104] Pires *et al.*, "Decision support based on digital twin simulation: A case study," *Studies in Computational Intelligence*, 2021.

[P105] Kovalyov, "Design and development of a power system digital twin: A model-based approach," *International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, 2021.

[P106] Ghosh, "Developing sensor signal-based digital twins for intelligent machine tools," *JOURNAL OF INDUSTRIAL INFORMATION INTEGRATION*, 2021.

[P107] Hu *et al.*, "Digital twin based reference architecture for petrochemical monitoring and fault diagnosis," *OIL AND GAS SCIENCE AND TECHNOLOGY-REVUE D IFP ENERGIES NOUVELLES*, 2021.

[P108] Pires *et al.*, "Digital twin based what-if simulation for energy management," *IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021.

[P109] Liao *et al.*, "Digital twin consensus for blockchain-enabled intelligent transportation systems in smart cities," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[P110] Human *et al.*, "Digital twin data pipeline using mqtt in sladta," *Studies in Computational Intelligence*, 2021.

[P111] Latsou *et al.*, "Digital twin integration in multi-agent cyber physical manufacturing systems," *IFAC PAPERSONLINE*, 2021.

[P112] Shukla *et al.*, "Digital twin-based prediction for cnc machines inspection using blockchain for industry 4.0," *IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS (ICC 2021)*, 2021.

[P113] Zhang *et al.*, "Digital twin-based smart manufacturing cell: Application case, system architecture and implementation," *Journal of Physics: Conference Series*, 2021.

[P114] Neto *et al.*, "Digital twin-driven decision support system for opportunistic preventive maintenance scheduling in manufacturing," *Procedia Manufacturing*, 2021.

[P115] L. D. Ipiña *et al.*, "Digital twins applied to the implementation of safe-by-design strategies in nano-processes for the reduction of airborne emission and occupational exposure to nano-forms," *Journal of Physics: Conference Series*, 2021.

[P116] Esterle *et al.*, "Digital twins for collaboration and self-integration," *IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, 2021.

[P117] Minerva *et al.*, "Digital twins: Properties, software frameworks, and application scenarios," *IT PROFESSIONAL*, 2021.

[P118] Edrisi *et al.*, "Ea blueprint: An architectural pattern for resilient digital twin of the organization," *Communications in Computer and Information Science*, 2021.

[P119] Huang *et al.*, "Edge computing enhanced digital twins for smart manufacturing," *16th International Manufacturing Science and Engineering Conference, MSEC*, 2021.

[P120] Matei *et al.*, "Engineering a digital twin for manual assembling," *International Symposium on Leveraging Applications of Formal Methods*, 2021.

[P121] Liu *et al.*, "imsestudio: blockchain-enabled secure digital twin platform for service manufacturing," *INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 2021.

[P122] Steinmetz *et al.*, "Key-components for digital twin modeling with granularity: Use case car-as-a-service," *IEEE Transactions on Emerging Topics in Computing*, 2021.

[P123] Amar *et al.*, "Knowledge driven rapid development of white box digital twins for industrial plant systems," *IECON Proceedings (Industrial Electronics Conference)*, 2021.

[P124] Lietaert *et al.*, "Knowledge graphs in digital twins for ai in production," *ADVANCES IN PRODUCTION MANAGEMENT SYSTEMS: ARTIFICIAL INTELLIGENCE FOR SUSTAINABLE AND RESILIENT PRODUCTION SYSTEMS, APMS*, 2021.

[P125] Shahsavari *et al.*, "Mcx - an open-source framework for digital twins," *European Council for Modelling and Simulation, ECMS*, 2021.

[P126] Conde *et al.*, "Modeling digital twin data and architecture: A building guide with fiware as enabling technology," *IEEE Internet Computing*, 2021.

[P127] Cirullies *et al.*, "On-demand shared digital twins - an information architectural model to create transparency in collaborative supply networks," *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2021.

[P128] Lopez *et al.*, "Real-time event-based platform for the development of digital twin applications," *INTERNATIONAL JOURNAL OF ADVANCED MANUFACTURING TECHNOLOGY*, 2021.

[P129] Bolender *et al.*, "Self-adaptive manufacturing with digital twins," *International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2021.

[P130] Steindl *et al.*, "Semantic microservice framework for digital twins," *APPLIED SCIENCES-BASEL*, 2021.

[P131] Ren *et al.*, "Strengthening digital twin applications based on machine learning for complex equipment," *Proceedings -Design, Automation and Test in Europe, DATE*, 2021.

[P132] Tan *et al.*, "Toward a future network architecture for intelligence services: A cyber digital twin-based approach," *IEEE Network*, 2021.

[P133] Lei *et al.*, "Toward a web-based digital twin thermal power plant," *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 2022.

[P134] Redeker *et al.*, "Towards a digital twin platform for industrie 4.0," *4th IEEE International Conference on Industrial Cyber-Physical Systems, ICPS*, 2021.

[P135] Autiosalo *et al.*, "Twinbase: Open-source server software for the digital twin web," *IEEE Access*, 2021.

[P136] Traore *et al.*, "Unifying digital twin framework: Simulation-based proof-of-concept," *IFAC PAPERSONLINE*, 2021.

[P137] Munoz *et al.*, "Using uml and ocl models to realize high-level digital twins," *ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2021.

[P138] Assad *et al.*, "Utilising web-based digital twin to promote assembly line sustainability," *4th IEEE International Conference on Industrial Cyber-Physical Systems, ICPS*, 2021.

[P139] Kherbache *et al.*, "When digital twin meets network softwarization in the industrial iot: Real-time requirements case study," *Sensors*, 2021.

[P140] Picone *et al.*, "Wldt: A general purpose library to build iot digital twins," *SoftwareX*, 2021.

**ENXHI FERKO** received the B.Sc. degree in computer engineering from the Polytechnic University of Tirana, in 2017, and the M.Sc. degree in software engineering from Mälardalen University, in 2020, where she is currently pursuing the Ph.D. degree in computer science. Her research interests include cloud-based cyber-physical systems and industrial Internet of Things.

**ALESSIO BUCAIONI** received the Ph.D. degree from Mälardalen University, in 2018. From 2014 to 2020, he worked for the automotive and industrial automation industries. He is currently an Assistant Professor in computer science at Mälardalen University. His research interest includes development of complex software-intensive systems from software architecture to model-driven development.

**MORIS BEHNAM** received the B.Sc. and M.Sc. degrees in computer and control engineering from the University of Technology, Iraq, in 1995 and 1998, respectively, and the M.Sc., Licentiate, and Ph.D. degrees in computer science and engineering from Mälardalen University, Sweden, in 2005, 2008, and 2010 respectively. He is currently a Professor in computer science with focus on cyber-physical systems.