



# A Survey on the Use of Lightweight Virtualization in I4.0 Manufacturing Environments

Giuseppe Di Modica<sup>1</sup> · Luca Foschini<sup>1</sup>

Received: 23 June 2022 / Revised: 21 December 2022 / Accepted: 24 January 2023 /  
Published online: 13 February 2023  
© The Author(s) 2023

## Abstract

Over the past decade, in the industrial sector we have witnessed the rise of a revolutionary movement, known as Industry 4.0, that promotes the digital transformation as the key to increase the competitiveness of manufacturing factories. Among the many technologies recognized as “drivers” of such revolutionary transition, microservices stand out as a software development paradigm capable of bringing several benefits to the manufacturing process. Whilst the literature offers many examples of initiatives exploiting microservices in digitally-advanced sectors (e.g., finance, telecommunication, retailing), its potential in the industrial manufacturing is yet to be fully unleashed. We conducted an extensive literature survey in the twofold aim of bringing to the reader’s attention the many benefits that the microservices paradigm may offer in industrial manufacturing settings, and drawing a picture of how light virtualization techniques are actually being exploited to achieve Industry 4.0 digitization goals. In this paper, we propose a structured analysis of the collected literature proposals which combines the benefits sought by authors when approaching to the microservices techniques and the specific scope of application of proposals. We conclude the paper highlighting the research aspects that have not been sufficiently explored in the literature and that would deserve further attention in the near future.

**Keywords** Light virtualization · Microservices · Industry 4.0 · CPS · IIoT · IT/OT convergence · Edge computing · Cloud computing

---

✉ Giuseppe Di Modica  
giuseppe.dimodica@unibo.it

Luca Foschini  
luca.foschini@unibo.it

<sup>1</sup> Department of Computer Science Engineering, University of Bologna, Bologna, Italy

## 1 Introduction

The fourth industrial revolution, in short *Industry 4.0* (I4.0), promises to bring a significant transformation of industrial manufacturing processes through extensive digitization of factories. According to Industry 4.0 promoters, a prerequisite for the realization of the digital transition is the tear down of historical, cultural and technological barriers that separate the shop floor, where Operational Technologies (OT) are in force, from the business departments, which make extensive use of Information Technologies (IT). Among others, the achievement of the factory digitization goal relies on the capability of IT to penetrate OT departments, while complying with the safety and security constraints set by the latter.

There is a wealth of information technologies that the research community recognizes as “enablers” of the aforementioned revolution. Among those, this paper investigates the *microservices technology* with respect to the contribution it brings to the I4.0 revolution. The overall benefits of microservices are clear from the software engineering perspective, independently of the vertical domains of application. In this paper, we revisit the main microservices features (modularity, scalability, etc.) in light of (i) the extensive use of microservices made by researchers and of (ii) the expected concrete benefits across all facets of the factory digitization process. The proposed perspective will help the reader to get a grasp on the current research trend of the application of microservices in the manufacturing sector as well as on future research directions.

To the best of our knowledge, the body of literature is lacking such a vertical perspective. In [34], authors provide an overview of core technologies commonly recognized as enablers of I4.0 and dwell on the countries/regions that are contributing most to the industrial revolution. In the paper, four specific aspects of the I4.0 are addressed: interconnectivity, data, integration, and innovation. This work presents a very high-level perspective of technologies and the respective target of application. A similar effort to depicting a general overview of I4.0 enabling technologies is proposed in [39]. Authors bring to the attention of the community both functional and non-functional aspects such as the management of data explosion, modeling and simulation of physical systems, and enforcement of safety and security. [54, 64] discuss more in details the key role that Service Oriented Architectures (SOA) will play as a technology enabler in the fourth revolution. They identify SOA as the right architectural choice to support the flexibility demanded by I4.0 systems. Finally, a comprehensive and exhaustive review of the adoption of service computing technologies in I4.0 settings is made in [55]. In here, authors surveyed most notable I4.0 architectural solutions and found out that microservices are more and more gaining ground as a software architectural pattern. They list all surveyed microservice-based proposals and discuss the wide range of employment options in the industrial field, like e.g. software encapsulation of industrial control devices, support for data communication, orchestration of production processes.

The research methodology adopted in our study includes the following steps. We surveyed the literature to seek frameworks, platforms, software prototypes and

prospective proposals that have addressed the use of microservices in the industrial manufacturing domain. The sources taken into consideration in this survey are the following: Scopus,<sup>1</sup> ACM Digital Library,<sup>2</sup> IEEE Xplore Digital Library,<sup>3</sup> Elsevier ScienceDirect,<sup>4</sup> and SpringerLink.<sup>5</sup> We also filtered out research items that are dated earlier than the last decade. Afterwards, we categorized the collected works according to the microservices feature(s) that motivated the authors to embrace the microservice paradigm to pursue their research goals. Then, we mapped each work's scope of action on a layered representation of the I4.0 factory (the latter being inspired by RAMI 4.0 [42], a standardised and authoritative industrial reference architecture). We produced a synthetic view of the collected works, by collapsing all literature contributions in a table that puts in correlation exploited microservices features with scopes of actions. Finally, based on that view, we provide the reader with a full picture of both current research trends and unexplored paths towards adoption of microservices in I4.0 scenarios.

The paper structure is the following. In Sect. 2, we introduce the I4.0 smart factory and dwell on the role of IT technologies in achieving the digitization goal. In Sect. 3, after briefly touching on some of the main features and benefits of microservices, we thoroughly discuss the surveyed literature works. In Sect. 4, we comment the synoptic table that summarises the results of our analysis. Conclusive remarks on the contribution provided by the present work are drawn in Sect. 5.

## 2 Towards the Factory of the Future: an IT-Driven Perspective

In the last decade, we have been witnessing to a revolution in the industrial manufacturing sector that practitioners and researchers recognize under the name of *Industry 4.0* (I4.0). I4.0 is expected to push a groundbreaking change of perspective in the manufacturing sector that will revolve around the *smart factory* paradigm. Main expected benefits of the I4.0 transition are (i) the enhancement of the capability of enterprises to deliver new products/services, (ii) cost reduction, (iii) provisioning of new business opportunities, and iv) increase in employment [13]. Standardization bodies operating in several countries worldwide [14, 42] are proposing new taxonomies and unifying architectures addressing both business and technological issues this revolution will uncover. One of the big challenges this revolutionary movement will have to face is tearing down the cultural and technological barriers that still keep production and managerial departments apart from one another. I4.0 is paving the way for a progressive convergence of *Operational Technologies* (OT), i.e., technologies adopted in operational department such as the shop floor, and *Information*

---

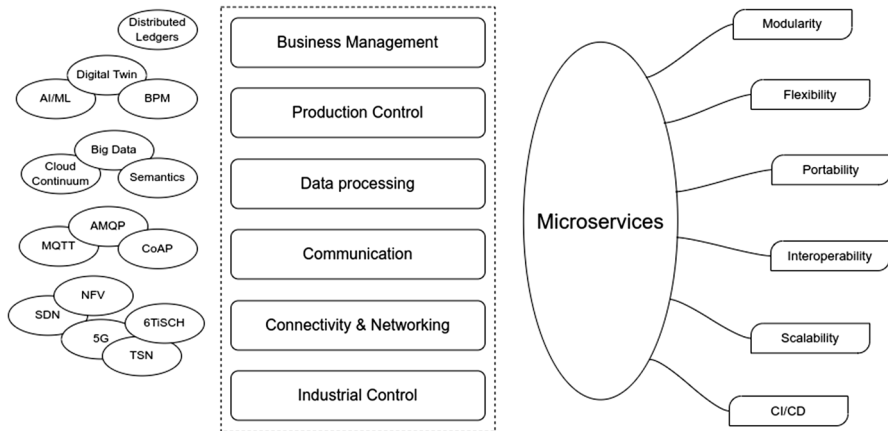
<sup>1</sup> <http://www.scopus.com/>.

<sup>2</sup> <https://dl.acm.org/>.

<sup>3</sup> <http://ieeexplore.ieee.org/>.

<sup>4</sup> <https://www.sciencedirect.com/>.

<sup>5</sup> <https://link.springer.com/>.



**Fig. 1** A layered view of I4.0 factory and IT enablers

*Technologies* (IT), which are typically used in managerial departments for business control purposes.

The enforcement of a tight IT/OT integration is the key to realize a profound technological transformation known as factory "digitization", i.e., a process that will transform factory assets (machines, production chains, production processes, etc.) into digital objects that communicate and collaborate with each other for the achievement of the business goals. Not only will digitized assets form a network of interconnected objects operating within the factory boundary, they will also establish cross-boundary interactions with third-party services to actively support B2B operations (e.g., supply-chain management and product servitization once a product has been delivered [14, 42]).

The rigid automation system architecture based on the ISA95 pyramid [6] must evolve to accommodate the need for interoperability between departments responsible of controlling the automation and the ones in charge of leading the business. Indeed, we are already assisting to a progressive penetration of IT into factory's operational layers. Figure 1 shows an IT-polarized view of the factory of the future inspired by the Reference Architecture Model Industrie 4.0 (RAMI 4.0) [42], which consists of a three-dimensional coordinate system that contains the essential aspects of I4.0. In particular, the layered view shown in the center of Fig. 1 retraces the "Layers" axis of the RAMI 4.0 specification and depicts the *scope of actions* of the factory digitization process along with the main technological "enablers" that support such a process.

As remarked throughout the paper, the microservices technology has a broad scope of adoption that spans all layers. Microservice-based implementation techniques allow to build services out of sensors/actuators, Programmable Logic Controllers (PLCs), line productions, or even an entire production process. Servitized assets can be shared and accessed from anywhere, both within the factory premises and outside. Counting on features such as modularity, flexibility, interoperability, scalability, microservices empower manufacturing factories to implement effective,

adaptive, robust and scalable production strategies capable of coping with the unpredictable dynamics of the global market. A further proof of the large adoption of such technology in the industrial sector is the fact that services offered by most prominent Industrial IOT (IIoT) platforms in the market [5, 28, 31, 41, 45] are built on top of ecosystems of composable microservices.

Hereafter, we give an insight of each layer and, where applicable, of the corresponding enabling technologies. In Sect. 4, we shall use those layers as an analysis dimension of the surveyed literature works.

## 2.1 Industrial Control

Industrial Control Systems (ICSs) carry out tasks of automation and control of the shop floor machinery. Old generation ICS [12], characterized by monolithic deployments of SCADA/PLC/RTU devices, are being superseded by more flexible, composable, highly programmable control schemes, mostly "softwarized" (e.g., Virtual PLC [7, 16], Remotely-implemented PLC [25]), that can promptly and easily adapt to changes of the production schedule.

## 2.2 Communication

The collapse of IT/OT boundaries fostered by I4.0 urges the need to establish secure, robust and reliable communication channels between operational and business departments. From the connectivity perspective, emerging networks/protocols such as 5 G, 6TISCH and TSN will support data exchanging (both upward and downward) while meeting the requirements of most demanding applications in terms of, e.g., end-to-end packet delay. On top of that, SDN and VNF guarantee flexibility and scalability when setting up and maintaining virtual network paths and correlated services. Finally, MQTT, AMQP and CoAP represent consolidated, message-based communication protocols that provides applications with lightweight and asynchronous mechanisms to exchange information in a reliable and scalable way.

## 2.3 Data Processing

A breed of solid Big Data computing methodologies and software tools will provide powerful means to extract knowledge out of tons of raw data, and will help business departments take more informed decisions. From an infrastructural point of view, advanced distributed computing paradigms such as the Edge/Fog and the Cloud promise to offer support to (i) implement powerful and safe control routines of industrial processes and (ii) off-line post-processing of production data collected from the shop floor.

## 2.4 Production Control

To control production processes and assure high-quality standard in product delivery, factories will benefit methodologies and technologies developed by the Business

Process Management (BPM) research community. Also, the “Digital Twins” [46] is gaining momentum as a paradigm to implement distributed, flexible and more effective production control strategies (like, e.g., predictive/prescriptive maintenance), also leveraging on well-consolidated AI/ML techniques that enable the implementation of data-driven control routines.

## 2.5 Business Management

The all-connected concept promoted by I4.0 span far beyond the factory premises. Smart factories will open their servitized business processes to support automated B2B routines in the context of what is best known as *collaborative manufacturing* [61]. One relevant example of such trend of openness in B2B interactions is the collaborative management of the supply-chain, for which Distributed Ledgers promise to play a key role as a technology enabler.

The purpose of this work is to explore the literature in order to discover how research communities are harnessing microservices features to achieve the factory digitization goal set by I4.0. To this end, we adopt an approach that walks the reader through each of the many advantages offered by microservices and discloses how researchers benefit them in the industrial manufacturing field.

## 3 Adoption of Microservices in Industrial Manufacturing Environments

Among the many specific definitions given for microservices and the related software development paradigm, a popular and authoritative one is provided by Lewis and Fowler, who defined the microservice architectural style as “...an approach to developing a single application as a suite of small services each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API” [24].

In the remainder of the paper, we will interchangeably use the term *microservice* and *lightweight container*. Among the many benefits and features offered by the lightweight containerization paradigm, we briefly recall some of the most relevant ones that drove our investigation.

*Modularity* offered by lightweight containerization technologies is the capability of encapsulating functionalities into modules with customizable granularity. In contrast to monolithic architectures, the lightweight containerization paradigm fosters the adoption of finer-grained modularized components, each specialized in a specific task, enforcing the loose coupling/high cohesion programming principles.

Lightweight containers intrinsically provide *Flexibility*, in that they allow an easy and quick adaptation of the implemented system to unexpected changes of requirements or, more generally, of the context’s boundary conditions. Indeed, when a system’s provided services are distributed over multiple modules, a modification of initial requirements has limited impact (usually, on one or a few modules) in terms of re-coding efforts.

A software system that complies with the lightweight containerization paradigm gains *Portability*. Containers are autonomous entities embedding all libraries/artifacts necessary to their operation. They can easily be undeployed from their original run-time environment and re-deployed in other container-enabled environments, without having to worry about compatibility issues.

*Interoperability* is the capability of software entities to exchange information and data independently of the language used to code their functions. Lightweight containers may expose services through REST-full endpoints, which enable communication through standardized internet protocols such as IP and HTTP.

The microservices paradigm allows an ease implementation of reliable *CI/CD* pipelines. Microservice-based software architectures enable typical DevOps practices that ensure quick response to market needs, agile development of features, continuous integration of functionalities.

In the following, we present a survey of recent literature works that promote the use of microservices to support smart manufacturing companies in achieving the I4.0 goals. Noted that the majority of proposals adopts the microservice paradigm to take advantage of the multiple benefits it offers, we categorized the surveyed works according to the microservice feature that best supports their research objective.

For the reader's convenience, in Table 1 we provide a synoptic view of the surveyed literature works that helps us depict the current interest of industrial research communities towards the microservice technology, and spot potential areas of applications for microservices that have not yet been fully explored. In the table, a two-fold characterisation of the works is proposed in regards to the harnessed *microservice feature(s)* and the *scope of action* (as discussed in Sect. 2) respectively. Specifically, when a microservice feature is claimed to be the main one harnessed by authors in their work, in the corresponding cell we have reported a synthetic note on the work's main contribution; on the same row, further (minor) features are instead marked with ✓. The categorization of each work's scope of action can be found in the first column of the table. We will further discuss and comment the table results in Sect. 4.

### 3.1 Modularity: CPSs as an Ecosystem of Composable Modules

The diversity and functional heterogeneity of OT assets constitute a huge obstacle towards the achievement of the I4.0 ambitious goal of *all-connected factories*. Cyber-physical systems (CPSs) require a mechanism that turns assets into entities capable of interacting with each other and with the factory ecosystem. The modularity feature of microservices meets this need as it provides a way to build a software module out of any factory asset at the desired level of granularity, be it a sensor, a controller, or even an entire production chain. In the example in Fig. 2, we depict the digitization of some physical entities populating the shop floor. Sensors, machines, PLCs, etc., can be *containerized* in microservices which will form an ecosystem of modules that, in turn, can be composed to build industrial control applications.

Authors of [48] refer to Industrial Internet of things (IIoTs) as systems that rely on different and heterogeneous devices working together, gathering and sharing

**Table 1** Characterisation of literature works according to microservices features and scope of action

Scope of action	Year	Microservices features					CI/CD
		Modularity	Flexibility	Portability	Interoperability	Scalability	
Pontarolli et al. [44]	2020	Industrial Control	Modular and composable control loop ✓				✓
Bigheti et al. [9]	2019		Flexible control loop ✓				
Dai et al. [17]	2019			Interoperable industrial control functions			
Goldschmidt et al. [26]	2018		Flexible PLC				✓
Tasci et al. [57]	2018		✓	Portable control applications			
Rufino et al. [48]	2017		Modular industrial control functions				
Gonzalez-Nalda et al. [27]	2017				Interoperable robot control functions		✓
Schneider et al. [51]	2019	Networking	Flexible NVF for traffic isolation				
Tseng et al. [59]	2018		✓			Scalable agent-based networking architecture ✓	
Corradi et al. [15]	2021	Communication	Modular data gathering architecture				
Strijje et al. [56]	2019					Scalable message-based communication framework	
Mena et al. [40]	2019				Interoperable Digital Twins for data shipment		
Alaassam et al. [3]	2019						Scalable Digital Twins for data streaming
Um et al. [60]	2019			✓	Interoperable M2M communication		

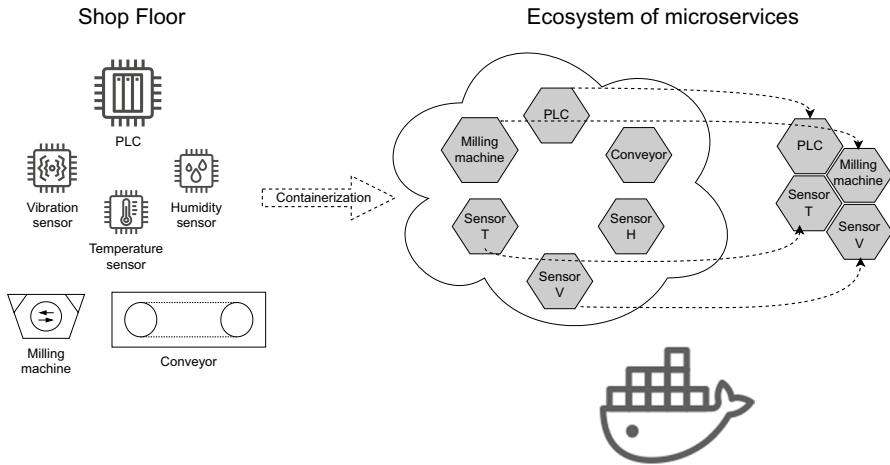


**Table 1** (continued)

Scope of action	Year	Microservices features					
		Modularity	Flexibility	Portability	Interoperability	Scalability	
Hohr et al. [30]	2019	Modular architecture for data mining				Scalable data management facility for battery production	
Zhou et al. [66]	2019					Scalable architecture for industrial analytics in robotic environments	
Dinh-Tuan et al. [18]	2019						
Wang et al. [63]	2019		Flexible big data processing architecture				
Ahasam et al. [2]	2018			Portable Digital Twins for scientific workflows		✓	
Borghesi et al. [10]	2021	Production Control Business Management		Portable Digital Twins for product quality control			✓
Kühn et al. [36]	2020					Scalable tracking functions for automotive systems	
Omar et al. [43]	2019					Orchestration of interoperable assets of the production system	✓
Kousiouris et al. [35]	2019						
Thramboulidis et al. [58]	2018	Composable cyber-physical assets of a production plant					CI/CD of a supply-chain management system
Yin et al. [65]	2018		✓	Scheduling of portable production tasks			
Li et al. [37]	2018		✓				Life-cycle management of industrial software

**Table 1** (continued)

Scope of action	Year	Microservices features					CI/CD
		Modularity	Flexibility	Portability	Interoperability	Scalability	
Dobaj et al. [19]	2018		✓			✓	CI/CD and design patterns to implement software architecture in IIoT environments
Senington et al. [52]	2018		✓				Devops to manage the production in a cyber-physical environment
Adambarage et al. [1]	2019					Scalable and highly available enterprise system	
Schaffer et al. [49]	2018				✓		Knowledge and process workflows sharing across factory departments
Innerbichler et al. [32]	2017					Collaborative and scalable supply-chain management for wooden buildings	



**Fig. 2** Exploiting modularity to build composite cyber-physical ecosystems

data. To face the heterogeneity of IIoTs, they propose to use a modular architecture based on microservices. Within the proposed framework, industrial control functions are distributed over the physical layer, where Docker containers hosted on Raspberry-Pi nodes carry out sensing tasks and IIoT gateways run simple data analytics. The proposed system was tested in a controlled environment, and proved able to fulfil real-time requirements typical of operational settings.

In [58], authors leverage model-driven engineering (MDE) and microservices to face the complexity of the software development process within the industrial manufacturing domain. They exploit the microservices paradigm to encapsulate physical units of the manufacturing plant into smart software entities called *cyber-physical microservices* (CPMSs). Furthermore, they propose a layered architecture where the manufacturing cyber-physical system is defined in the cyber-physical microservice layer as a composition of well-defined CPMSs, which are either Primitive or Composite. A Primitive cyber-physical microservice encapsulates a physical artefact and transforms it into a smart entity able to process, transport and store energy, material, parts, sub-products and products. A Composite cyber-physical microservice is the microservice that offers its functionality by using the functionality of at least one Primitive CPMS. Typical examples of Composite CPMSs are plant processes.

In [30], authors argue that the success of data analytics solutions in manufacturing environments depends on how tightly IT services are integrated into the manufacturing system, as well as on the ability of machine operators of using such services in collaboration with data scientists. The foundation of their work is CRISP-DM [53], a standard model that describes the necessary activities for the implementation of data analytics as a cross-industry standard process for data mining. Accordingly, authors propose an ecosystem of loosely-coupled microservices implementing simple data mining functions, capable of communicating with each other by means of a Kafka message broker. As a proof of concept, authors provide a use case example aimed at

analyzing general causes of bad product quality, analyzing the specific causes of a faulty product, and predicting the quality of a product during manufacturing.

Authors of [15] propose the design and implementation of a platform that supports data gathering and sharing among OT and IT layers of an industrial manufacturing factory. The architectural design grounds on four building functions, namely: data extraction and transformation, data shipment, data visualization, and data processing. With the exception of data extraction and transformation (which is prerogative of OT), all functions can be offered at each layer to serve different purposes. The platform implementation revolves around the realization of fine-grained software modules, implemented as Docker containers, that can be composed to build the specific functions according to each layer's need.

[44] leverages the modularity of microservices to build fine-grained, softwarized industrial control services and implement on top of them an automated orchestration of the overall process control loop. Authors proved that the adoption of microservices as building blocks of the control loop greatly simplifies the service composition task. They also implemented a prototype of the designed microservice-based architecture and implemented a pipe pressure control loop in a pilot industrial plant. Besides the ease of implementation and the gain in flexibility/scalability with respect to legacy process control loop approaches, authors claim to have measured significant response time for what concern service communication in the orchestration.

### 3.2 Flexibility: Softwarization of the Industrial Control Loop

In [26], authors address the need of advanced flexibility in production for future control systems. In particular, they identify the PLC as one of the main building block of automation systems that need to become more flexible. To fulfil such need, they propose a *multi-purpose* controller that inspires to lightweight container solutions such as Docker. According to the approach, controllers encapsulated into Docker containers can take advantage of containerization benefits. The proposed system aims to support scenarios of flexible and automated controller deployment and enables support for the emulation of legacy equipment too, such as control systems and the respective controllers which have been designed up to 30 years ago and are still in use in many installations. Finally, authors tested the real-time capability of the system on a test bench called *CyclicTest*<sup>6</sup>. The benchmark was executed on Raspberry-Pi 2 nodes equipped with a Linux OS and Docker runtime. Test results showed that the system, despite the overhead introduced by the containerization, was capable to manage control cycles being executed in the range of 100ms to 1 s.

[9] proposes a control-as-a-service approach to handle closed process control loops. In the context of the proposed solution, the Data Acquisition function and the PLC layer are implemented by the DAQ and the SoftPLC microservices respectively. DAQ transparently offers higher level applications a data acquisition service that uses underlying hardware modules allocated in the process. As a result,

<sup>6</sup> <https://rt.wiki.kernel.org/index.php/Cyclictest>.

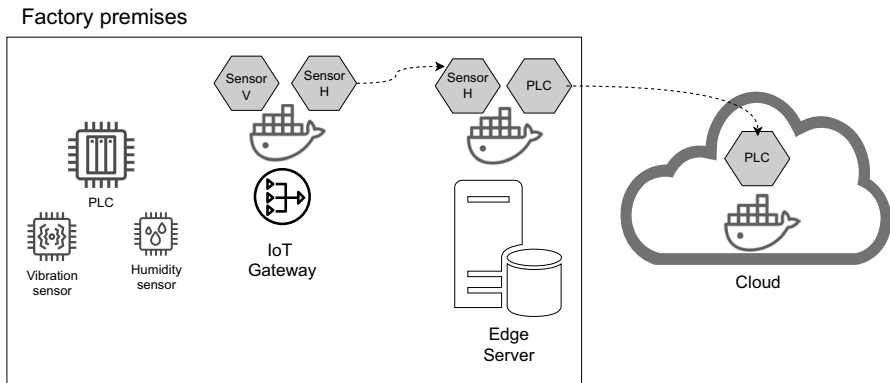
applications do not need to comply with I/O hardware specification neither with the code (software) necessary for obtaining the required variables. SoftPLC combines the features of a PLC with the benefits of an open hardware and software architecture. This microservice allows running PLC programs (and logical sequencing of tasks), as well as monitoring IO data. The core of this microservice is the OpenPLC project [4]. They tested the control-as-a-service process using a direct current motor control in a didactic plant and collected results showed that the proposed solution manage to control and maintain the plant stability, and that the response time of control execution complies with that required in real industrial plants.

In [51], authors propose a concrete use case of smart manufacturing building on relevant technologies like 5 G and Network Function Virtualization (NFV) [21]. Specifically, they address a smart manufacturing scenario in which manufacturing machines need to be interconnected in order to collect and analyze their relevant data, such as configuration parameters or values produced by sensors. They propose the implementation of network services consisting of multiple virtual network functions (VNFs), where each VNF is implemented in one or more Docker containers. The objective is to provide a flexible mechanism to segment the network in order to adequately isolate data traffic, detect malicious intrusion and counteract immediately.

Authors of [63] proposes a new industrial big data processing architecture implementing batch data processing, stream data processing, and real-time control in a scenario of aviation manufacturing. The proposed architecture follows the microservices pattern, which improves the flexibility and stability of the architecture, and makes it independently operated in multi-scenarios such as state monitoring of workshop, adaptive data acquisition, feedback control, and user-oriented information classification. As a proof of concept, the architecture was tested in a simulation digital manufacturing workshop. According to the authors claim, microservices greatly improved the efficiency, adaptability, and extensibility of the manufacturing process.

### 3.3 Portability: Seamless Deployment of Control Loop Over the Cloud Continuum

One of the revolutionary concepts introduced by I4.0 is the opportunity of including IT resources owned by third parties (i.e., residing outside of companies boundaries) to make the execution of business processes more efficient. In such a distributed environment, there will be the need of a technology that enables a seamless deployment/migration of services on potentially heterogeneous IT resources provided in the IoT-Edge-Cloud continuum. Microservices are services designed to run in an execution runtime supported by the majority of hardware architectures and operating systems. By virtue of such portability feature, they can be run, stopped, moved around and resumed anywhere in the mentioned computing continuum. In the Fig. 3, an example is shown where containers are initially deployed on an IoT Gateway (*SensorV*, *SensorH*) and on an Edge Server (*PLC*). Due to some change of the boundary conditions—e.g., an overload at the IoT Gateway—*SensorH* is seamlessly



**Fig. 3** Exploiting portability to move microservice in the cloud continuum

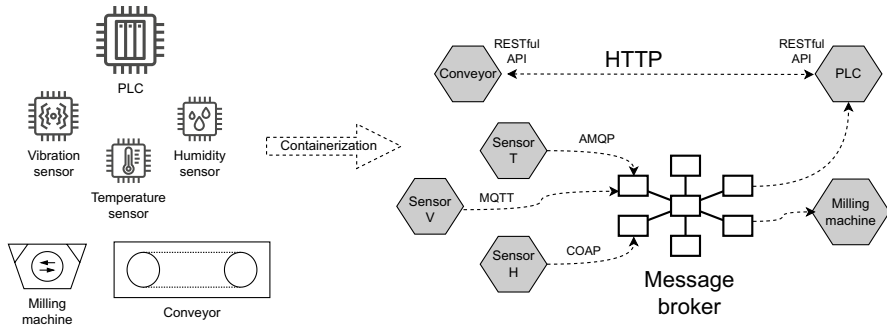
shifted from the IoT Gateway to the Edge Server and *PLC* is moved from the Edge Server to the Cloud.

In [2], authors propose an event-driven architecture to implement Digital Twins that combines the power of scientific workflows, the flexibility of containers technology, and robustness of the distributed streaming approach. The approach consists in re-designing the scientific workflow (SWF) into smaller, loosely-coupled sets of SWFs called Micro-Workflows (MWF). MWFs allow independent deployment and increase the potential of horizontal scaling. For example, if low-latency response is a requirement, MWFs can be deployed near the IIoT equipment, while if intensive processing is requested, they can be deployed in the public cloud. The reference implementation makes use of Kepler [33] as SWF management system, and Kafka as a distributed streaming platform supporting horizontal scalability.

Grounding on the assumption that current monolithic PLCs are unfit to accommodate market dynamics, [57] proposes the use of lightweight containers to modularize real-time control applications, and relies on a hardware abstraction layer to improve portability (separating domain logic from platform or hardware-specific code) and flexibility. Along with the mentioned features, the proposed system also exhibits the ability to cope with real-time constraints of typical industrial environments thanks to the employment of Linux machines with some additional kernel configurations<sup>7</sup> (namely, a kernel patched with PREEMPT\_RT [22] and Cobalt [20]).

In [10], authors propose an open framework to implement distributed Digital Twins (DTs) for I4.0 manufacturing scenarios. In particular, DTs are devised that enforce predictive maintenance and product quality control strategies. Authors employ portable and reusable Docker containers to implement the building blocks of service pipelines that represent the DTs of shop-floor machines. The roll-out of a DT is carried out by an orchestrator that harnesses the portability feature of

<sup>7</sup> In the considered environment, non-preemptible kernel routines could delay real-time tasks.



**Fig. 4** Interoperable ecosystem of microservices

containers to seamlessly deploy service pipelines over a continuum of computing resources provided by the factory’s Edge machines and by a Cloud provider.

Authors of [65] harness the Fog computing paradigm to flexibly and efficiently implement smart manufacturing tasks. According to them, tasks supporting the management of the production, such as fault detection and state analysis of devices deployed in assembly lines, can be carried out in a *middle layer* between the industrial cloud and the terminal device itself. As tasks may expose different requirements, they propose to use a scheduler that allocates tasks to computing nodes available in the Fog according to such constraints. The proposed approach makes use of the lightweight containers technology to encapsulate tasks into microservices that can execute in both Fog nodes and the Cloud. The task scheduler is in charge of deciding whether a task has to be allocated to a Fog node or to the Cloud depending on the delay-sensitiveness of the task.

### 3.4 Interoperability: Need for Open Approaches and Standard API in Production Management

Production environments are characterized by a strong presence of heterogeneous devices as well as a high number of (often proprietary) machine-to-machine (M2M) communication protocols. In the typical all-connected-factory scenario fostered by I4.0, where communication is not bound to just “close” factory layers, it is mandatory to increase the interoperability among all factory assets by ironing out the above mentioned heterogeneity. With Rest-based interfaces, Microservices technology provides an open and standard way to build an ecosystem of interoperable factory assets. In the Fig. 4, a scenario of interoperable microservices is depicted.

In [27], authors put the focus on mobile robotics in production systems and industrial automation environments. Authors claim that robotics software has been chronically facing problems in industry and academy due to the lack of standardization, interoperability and reuse of software libraries [11]. In this panorama, the Robot Operating System (ROS) [47] stands out among the few open-source initiatives proposing an operating system that provides functionalities and tools to operate robots. Basically, it defines different entities including nodes, message topics and services.

Nodes are software modules that can communicate with one another by message passing. Authors propose to encapsulate ROS nodes into Docker containers. In brief, Docker isolates ROS nodes with the twofold objective of improving security (the attack surface is reduced) and easing system administration. The overall control function is realized by a distributed application consisting of Docker nodes deployed on top of a single-board powerful computer and a microcontroller with low/limited processing capabilities but high amount of hardware interfaces. As a proof of concept, they implemented a scenario of Automated Guided Vehicles (AGV) that makes use of off-the-shelf hardware such as PCs, Arduino [38] and Raspberry-Pi [23] nodes.

In [60], authors claim that the increase in the market of specific software solutions tailored to vertical manufacturing domains is the reason for the proliferation of heterogeneous IoT devices and machine-to-machine (M2M) communication protocols. They argue that, in order to achieve an effective and low cost management of the manufacturing environment, a standardization effort of IoT/M2M systems is necessary. They introduce *oneM2M*, an M2M common service layer that focuses on interoperability of message processing among different IoT solutions. The *oneM2M* model has a hierarchical resource structure, and all services are provided through resources that are accessible via the create, retrieve, update, delete (CRUD) commands commonly used in RESTful API architectures. Each resource is referenced through a URI and communication among resources is implemented via MQTT, HTTP, and CoAP protocols. They call on containerization to implement tasks that are typically common to IoT nodes and the management infrastructure. Common tasks are then encapsulated and published on a Docker registry for re-use for different purpose.

Authors of [17] propose a knowledge-driven microservice-based architecture to enable plug-and-play software components for industrial edges. Authors innovative approach revolves around the intuition of wrapping IEC 61499 [62] function block instances (used to implement industrial control functions at shop floor) in microservices, and turning IEC 61499 resource managers into service repositories for register and publish services. In the wrapping step, all IEC 61499 actions of the management commands (CREATE, DELETE, START, STOP, KILL, RESET, QUERY, READ) were mapped to standard HTTP commands (GET, POST, PUT, DELETE). Furthermore, harnessing the interoperability feature offered by the microservice-based architecture, authors implement a service orchestrator capable of dynamically composing the services to achieve the the desired control flow.

Authors of [43] propose a Fog-based approach to improve production systems efficiency and reliability. The use of a Fog layer is justified by the need of meeting typical Industry 4.0 requirements in terms of real time responses, data security, computing scalability and distribution. Authors combine the use of Fog Computing with the containerization and orchestration features offered by Docker and Kubernetes respectively. Rest-based API exposed by Docker containers helped improving the interoperability among the multiple heterogeneous entities that their production systems are composed of, namely, manual and automatic workstations, mobile robots, robotic arms and additive manufacturing machines.

To face the integration and usability issues deriving from the multitude of IoT devices and protocols used to communicate with such devices in CPS environments,



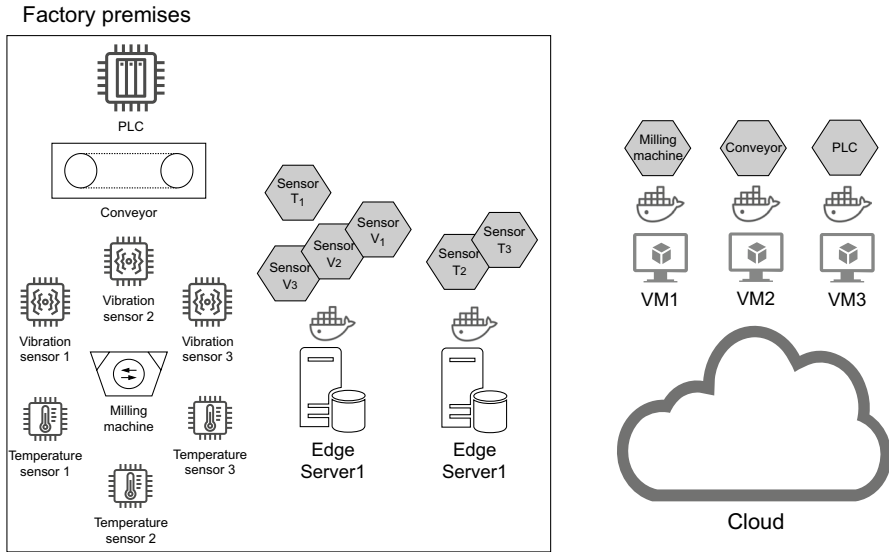


Fig. 5 Scalable ecosystem of microservices

[40] proposes a DT-inspired approach that abstracts devices’ features and implements software counterparts (“Digital Dices”—DDs) capable of interacting with each other through a common communication protocol. DDs are microservices that may play several roles on behalf of the device, such as Controller, Data handler, Event handler, etc. Towards the users, DDs expose standard W3C communication mechanisms, thus hiding from them the specific protocol used by the physical device.

### 3.5 Scalability: Scaling the Production Environment

I4.0 fosters an acceleration of the factories digitization process that will affect both business departments and shop floors. Future manufacturing factories will need to put in place an ecosystem of interconnected digitized assets capable of promptly reacting to market demands (e.g., increase, decrease, diversify production) while keeping the overall delivered quality to highly competitive levels. The microservices paradigm fulfills such a need as it offers an easy and effective way to implement the above mentioned ecosystem in a scalable way, i.e., in a way that allows the system to expand/contract depending on the business requirements, with the guarantee that the target quality of the final product remains unchanged. In the Fig. 5, a typical scale-out scenario is depicted. When more physical objects join the production chain (new sensors, in the depicted case), more computing resources can be activated in either the on-premise Edge farm or in the Cloud and, on top of those, the objects’ digital counterparts can be easily and transparently deployed.

In [59], authors present a fog platform for deploying industrial applications based on the VNFs. Although this work has a more general scope, it can fit the networking

needs of a typical manufacturing scenario. Authors propose a framework where a top-level orchestrator coordinates a number of Dockerized network agents which are responsible for implementing network functions and services. Docker network agents are responsible for the allocation of different IP segments, network address translation, and port forwarding service. Several agents can also establish respective virtual private network (VPN) tunnels, so that data can be exchanged via the VPN tunnel without additional network settings. Dockerized management agents are also deployed to monitor the status of resources and feed that information to the orchestrator which, in its turn, enforces scaling up and down of the deployed system in order to guarantee a high quality of the network performance. Results of a large scale test showed that the use of containers did not add much overhead to the network function management.

In [18], authors propose a decentralized software architecture that addresses new challenges in industrial analytics. They focus on a production environment where mobile autonomous robots capture videos with their on-board camera and off-loads object recognition tasks to fog-located servers found on their path. A microservice-based sensor data analytics component can predict robot movement path by analyzing location data from the robots in real-time. In the proposed architecture, authors introduce a Life-cycle management component responsible to monitor the health of each microservices. Specifically, it scales up or down a specific microservice according to load parameters such as memory consumption and length of incoming message queue.

In [66], authors advocate that existing MES solutions do not fulfill the need of business agility that is requested by manufacturing companies to compete in today market. They propose to extend the monolithic MES with a supplementary microservice-based architecture to meet the needs of battery production data management. Other than providing a unified authentication and access service, the supplementary architecture implements a scalable and agile production data management facility that offers data integration, processing, calculation, and mining.

[56] proposes a message-based communication framework that simplifies the connection among domain-specific technologies both within the shop floor and between shop floor and office floor. The framework makes use of the ZeroMQ tool and the Google Protocol Buffers data format to achieve platform independence and a high degree of efficiency. Overall, the tangible output of the paper is a scalable and distributed architecture that provides a distributed registry for the usage and provision of microservice components.

Authors of [3] propose to employ microservices, which are known to be stateless by nature, to implement a stateful stream processing tool that feeds DTs in smart manufacturing environments. Particularly, they study the possibility to use Apache Kafka Stream API (Kafka stream DSL) to build stateful microservices for real-time manufacturing data analysis. The idea put forward by authors is to combine the potential of microservices and the Apache Kafka Stream DSL to build a stateful streaming service. On the one hand, microservices provide the required level of scalability when the load of messages to stream increase; on the other hand, Kafka provides the possibility of managing the data processing state by synchronizing the local state with the set of intermediate Kafka topics.

Authors of [36] address the production of autonomous driving vehicles, and advocate the need of implementing a traceability scheme for automotive electrical systems that will provide support to align the production with the enhanced requirements of autonomous driving. Such a scheme comprehends product configuration, batch structure, process history, quality tests, resource data, and machine parameters. The proposed system is an event-based distributed architecture where, on the one hand, trace objects (orders, configurations, resources, processes) publish data to a message broker, on the other one tracking functions (quality management, order tracking, etc.) consume the provided data. As the number use cases to monitor increases exponentially with the number of traceable objects, authors opted for implementing tracking functions as microservices in order to mitigate scalability issues.

In [32], authors present an architectural approach to build a collaborative Industry 4.0 platform that enables IoT-based real-time monitoring, optimization and negotiation in supply chains. The proposal revolves around a practical use case concerning the collaborative planning and tracking of assets in a manufacturing system for wooden buildings. In particular, the collaborative platform supports the manufacturer and their suppliers in activities like establishing logistic chains, supporting negotiation processes, monitoring transport and on-site construction, etc. Platform core business activities, as well as functions such as users registration and service discovery, are implemented as ad-hoc microservices. Other than scalability, the microservice-based approach ensures extensibility with more specific value-added business services from third parties.

In [1], authors address the issues concerned with the migration of existing enterprise systems towards microservices. According to them, containerization can bring desirable performance characteristics such as high scalability, high availability, high cohesion and low coupling. Grounding on the consideration that manually restructuring enterprise systems into microservices is a tough and error-prone task, they propose a discovery technique that supports the identification of suitable parts of enterprise systems as good candidates to be re-engineered as microservices based on the knowledge gained through the analysis of business object relationships and their execution patterns. The provided system is capable of recommending consumer-oriented objects that, if containerized, would make the enterprise system gain better availability and scalability. The system was validated against two open source customer management systems, namely SugarCRM<sup>8</sup> and ChurchCRM.<sup>9</sup>

### 3.6 CI/CD-DevOps: Managing the Life-cycle of Digital and Physical Assets

The factory's digitization concept will realize through a progressive penetration of information technologies into business departments and shop-floors. Similarly to their physical counterparts, digitized assets have a life-cycle that includes operations

---

<sup>8</sup> <https://www.sugarcrm.com/>.

<sup>9</sup> <http://churchcrm.io/>.

like implementation, configuration, deployment, maintenance, upgrading, disposal, etc. In the highly dynamic manufacturing scenarios depicted by I4.0, the management of assets life-cycle needs to be as lean as effective, i.e., should require low human efforts and minimize potential errors rate. Lightweight containerization, along with flourishing orchestration technologies [29], allows a fine-grained and straightforward control of the (micro)services life-cycle, thus showing itself a promising technology to manage operations across the entire DevOps chain [8] also in manufacturing contexts.

In [26], authors experiment the usage of a container orchestrator to speed-up the life-cycle management of dockerized controllers employed at production level. A deployment coordinator leverages the services provided by the powerful Kubernetes tool [29] to carry out typical deployment activities (configure, install, run, stop, re-configure, migrate) according to the status of the controllers, which is constantly monitored at run-time.

[35] proposes a microservice-based architecture to support typical tasks of supply chain management such as monitoring, goods location analysis and delays. In the paper, authors makes use of Docker to softwarize the supply management services, and call upon Docker Swarm<sup>10</sup> to enable quick and seamless deployment of containerized services and their automated discovery, configuration and wiring. Taking the supply chain management as an exemplary challenging scenario in the industrial setting, authors stress out that microservices can be the leverage to build an integrated system of complex and heterogeneous services and to quickly design and enforce go-to-production routines.

In [49], authors advocate that the enhancement of enterprises *knowledge and process management* is one of the key enabler of the transition to I4.0. They analyze the challenges that prevent the introduction of an efficient knowledge and process management within internationally operating enterprises. According to that study, one of the big barrier is the establishment of complex, inflexible monolithic software solutions that prevent the correct management and sharing of enterprises accumulated knowledge and process workflows. In order to build a common framework for knowledge sharing, authors propose to define communication interfaces between departments and to adopt “Configurators” (web based tools) which support each department in the containerization of its services and tools, and make them available to other departments. The most beneficial aspect of using microservice-based Configurators is to ensure digital evidence for process releases and to distribute purposefully the knowledge to affected employees of the organization. In [50], same authors discusses the result of a test implementation along a Configurator for the planning of robot-based automation solutions.

In [37], authors criticize the industrial software development practices of the past, such as top-down design and long release cycles, arguing that those can no longer fulfill the requirements of industrial software delivery. Leveraging their experience in developing and delivering industrial software for IoT and edge computing, they advocate the need of calling upon the microservice development approach and

---

<sup>10</sup> <https://docs.docker.com/engine/swarm/>.

propose *architecture patterns* that cover four phases in the life cycle of industrial software: deployment, monitoring, adaptation, and testing. Within the mentioned cycles, interesting proposed patterns are: “downshifting”, which consists in moving the microservice from the cloud to the edge to improve the response time to the shop floor and to meet privacy concerns of customers; “monitoring connectors”, which oversee properties of microservices at the edge, such as performance and resource usage; “edge-to-cloud load balancing”, consisting in dynamically routing client requests to either cloud or edge in order to ensure optimal runtime performance of microservices.

[19] addresses the interoperability challenges between devices and systems that industry is facing up. Authors suggest the use of design patterns to develop a software architecture that fits the needs of IIoT environments. Specifically, they present a microservices-based architecture that is inspired to state-of-art development techniques (continuous integration, continuous deployment) and design patterns (Decompose, Mediator, Publisher–Observer, Discovery). The proposed architecture is suitable for deployment in multiple computing environment such as Cloud, Fog and Edge, and is able to guarantee non-functional features like scalability, dependability and flexibility.

[52] focuses on the issues of software management, deployment, configuration and integration in smart factories. Leveraging the requirement of flexibility in manufacturing systems, authors propose the use of Docker to distribute, deploy and manage the configuration of multiple software modules by multiple teams in an environment where humans and robots are in very close contact. Streamlining the deployment process helped to reduce the work needed to activate the system and lowered the chance of human error in the installation processes. Also, as part of the system often required customisation for specific deployments or use cases, authors set up an encapsulated environment with capabilities for both compilation and execution of the current build of the tool. Eventually, enabling static code optimization within the execution environment allowed improved performance.

## 4 Final Discussion

In the Table 1, we have depicted the result of our literature survey. In this section, we draw some synthetic considerations and spot potential lines of research that deserve further investigation.

Researchers have harnessed the *Modularity* feature of microservices to mostly propose solutions addressing the lowest layers of the factory view proposed in Fig. 1. As modularity enables composability, some [44, 48, 57] have devised systems that combine services provided by lightweight components to implement tailored industrial control functions. A few [15, 30] have proposed modular architectures to support data management facilities. Microservices *Flexibility* is extensively exploited by quite a number of authors who have proposed control loop systems [9, 17, 26, 44], virtualized networking systems [51, 59] and DevOps approaches to the management of production assets [19, 37, 52, 65] respectively. The *Portability* feature is poorly exploited to serve Industrial Control and Communication purposes.

A few researchers [2, 10, 65] have proposed to exploit the portability of software containers to schedule the execution of production control tasks on the Cloud or on the Edge, according to specific production needs. The REST-based communication natively offered by containers is harnessed in the literature to implement interoperable control functions [17, 27] and more in general to support interaction among heterogeneous industrial assets [40, 43, 60]. With the exception of Industrial Control, *Scalability* is exploited across all scopes of action. In particular, researchers have addressed the compelling need of handling the increasing amount of data produced in the factory by proposing scalable communication frameworks [3, 15, 56, 59] and scalable data management facilities [18, 66]. Finally, a handful of works have called upon microservices motivated by the fact that this paradigm enables agile approaches to the development, deployment and maintenance of software components [19, 35, 37, 52]. The *CI/CD* methodology was exploited to mainly implement software solutions devoted to the production control, whilst it was poorly addressed for other purposes.

Let us analyze the table from the perspective of the scopes of action. First, a good number of papers [9, 17, 26, 27, 44, 48, 57] have proposed the use of microservices to implement more flexible and interoperable control functions at the the shop-floor as an alternative to legacy control devices such as the PLCs. Software encapsulation and isolation properties, along with the maturity reached by well-known and widespread container frameworks, offer adequate guarantees for adopting such technology at the shop-floor, where strict safety and security constraints are placed. Still, microservices fall short when deployed in mission critical applications, as they fail to fulfill hard real-time requirements. This is surely a line of research which deserves further investigation.

It is no surprise that research communities push to deploy microservices in office-floor departments (which is historically a fertile ground for information technologies) to implement typical tasks of Production Control [10, 19, 35–37, 43, 52, 58, 65] or, more generally, Data Processing [2, 18, 30, 63, 66]. In here, according to researchers, the most appreciated quality of microservices is the strong support they provide to develop, maintain, update, deploy and scale industrial software solutions. Also, leveraging on features like portability and interoperability, microservices also proved to be effective in unleashing the power of computing paradigms like the Digital Twins and the Cloud continuum in industrial scenarios.

Unfortunately, the experimentation of microservice-based approaches to cater for networking needs is still lagging behind. The poor number of efforts addressing networking issues [51, 59] is probably due to the cultural scepticism in adopting software-based (and thus, potentially more vulnerable) approaches to networking that still pervades operational departments. In brief, there is much room here for further investigation that will involve both the networking and the cybersec communities.

Finally, the use of microservices to implement functionalities and services at the business management layer is poorly addressed in the literature. There is plenty of software applications in the market already covering much of the business needs (e.g., ERP or CRM systems). We found just one paper [1] that invokes the need of more flexible, modular and scalable enterprise systems, showcasing how microservices can be useful to the cause. We advocate that, in the *all-connected* landscape

envisaged by I4.0, factories should harness interoperability offered by microservices to implement B2B services that will make collaboration with stakeholders more agile.

## 5 Conclusion

Industrial manufacturing factories need to start a *transition-to-digital* process in order to achieve the goals set by I4.0. To this end, they may leverage a number of mature information technologies which have proven successful in other application fields. Among those, the microservices paradigm looks a promising technology due to the multi-fold benefits it may bring at the multiple stages of the manufacturing production process. To shed the light on the actual feeling of industrial research communities about such disruptive technology, we surveyed the body of recent literature. We explored several digital libraries seeking proposals (frameworks, architectures, software prototypes, or even arguable opinions) addressing the use of microservices in the production process. We analyzed the collected works and categorized them according to the microservices feature(s) that motivated authors to investigate on the use of this technology to face I4.0 challenges. The results of our analysis have been summarized in a synoptic table that highlights both the current trends of microservices adoption and potential fields of application that require further exploration.

**Author Contributions** All authors contributed equally to the study conception, design, material preparation, data collection, analysis, preparation of drafts, read and approval of the final manuscript.

**Funding** Open access funding provided by Alma Mater Studiorum - Università di Bologna within the CRUI-CARE Agreement.

## Declarations

**Conflict of interest** The authors did not receive support from any organization for the submitted work. The authors have no conflicts of interest to declare that are relevant to the content of this article.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Adambarage, A.C.D.A., Alistair, B., Colin, F., Artem, P.: Availability and Scalability Optimized Microservice Discovery from Enterprise Systems, vol. 11877. Springer International Publishing (2019). <https://doi.org/10.1007/978-3-030-33246-4>. <http://link.springer.com/10.1007/978-3-030-33246-4>
2. Alaasam, A.B.A., Radchenko, G., Tchernykh, A., Borodulin, K., Podkorytov, A.: Scientific Micro-Workflows : Where Event-Driven Approach Meets Workflows to Support Digital Twins. Proceedings of the international conference RuSCDays'18 - Russian Supercomputing Days (September 24–25, 2018, Moscow, Russia), MSU **1**, 489–495 (2018)
3. Alaasam, A.B., Radchenko, G., Tchernykh, A.: Stateful stream processing for digital twins: Microservice-based kafka stream dsl. SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings pp. 804–809 (2019). <https://doi.org/10.1109/SIBIRCON48586.2019.8958367>
4. Alves, T.: The openplc project (2017). [www.openplcproject.com](http://www.openplcproject.com). Accessed May 2020
5. Amazon: Aws iot. <https://aws.amazon.com/iot/>. Accessed Apr 2021
6. ANSI/ISA: Enterprise control system integration part 3 : Activity models of manufacturing operations management (2005). <https://www.isa.org/products/ansi-isa-95-00-03-2013-enterprise-control-system-i>. Accessed Oct 2021
7. Badar, A., Lou, D.Z., Graf, U., Barth, C., Stich, C.: Intelligent edge control with deterministic-ip based industrial communication in process automation. In: 2019 15th International Conference on Network and Service Management (CNSM), pp. 1–7 (2019). <https://doi.org/10.23919/CNSM46954.2019.9012680>
8. Balalaie, A., Heydarnoori, A., Jamshidi, P.: Microservices architecture enables devops: migration to a cloud-native architecture. IEEE Softw. **33**(3), 42–52 (2016). <https://doi.org/10.1109/MS.2016.64>
9. Bigheti, J.A., Fernandes, M.M., Godoy, E.D.P.: Control as a Service: A Microservice Approach to Industry 4.0. 2019 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2019 - Proceedings pp. 438–443 (2019). <https://doi.org/10.1109/METROI4.2019.8792918>
10. Borghesi, A., Di Modica, G., Bellavista, P., Gowtham, V., Willner, A., Nehls, D., Kintzler, F., Cejka, S., Tisbeni, S.R., Costantini, A., Galletti, M., Antonacci, M., Ahouangonou, J.C.: Iotwins: Design and implementation of a platform for the management of digital twins in industrial scenarios. In: 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), pp. 625–633 (2021). <https://doi.org/10.1109/CCGrid51090.2021.00075>
11. Bruyninckx, H.: Robotics software: the future should be open [position]. IEEE Robot. Automat. Mag. **15**(1), 9–11 (2008)
12. Cai, N., Wang, J., Yu, X.: Scada system security: Complexity, history and new developments. In: 2008 6th IEEE International Conference on Industrial Informatics, pp. 569–574 (2008). <https://doi.org/10.1109/INDIN.2008.4618165>
13. Cimini, C., Pinto, R., Pezzotta, G., Gaiardelli, P.: The transition towards industry 4.0: Business opportunities and expected impacts for suppliers and manufacturers. In: H. Lödging, R. Riedel, K.D. Thoben, G. von Cieminski, D. Kiritsis (eds.) Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing, pp. 119–126. Springer International Publishing, Cham (2017)
14. Consortium, I.I.: Industrial internet reference architecture. <https://www.iiconsortium.org/IIRA.htm>. Accessed Apr 2021
15. Corradi, A., Di Modica, G., Foschini, L., Patera, L., Solimando, M.: Sirdam4.0: a support infrastructure for reliable data acquisition and management in industry 4.0. IEEE Transactions on Emerging Topics in Computing pp. 1–1 (2021). <https://doi.org/10.1109/TETC.2021.3111974>
16. Cruz, T., Simões, P., Monteiro, E.: Virtualizing programmable logic controllers: toward a convergent approach. IEEE Embed. Syst. Lett. **8**(4), 69–72 (2016). <https://doi.org/10.1109/LES.2016.2608418>
17. Dai, W., Wang, P., Sun, W., Wu, X., Zhang, H., Vyatkin, V., Yang, G.: Semantic integration of plug-and-play software components for industrial edges based on microservices. IEEE Access **7**, 125882–125892 (2019). <https://doi.org/10.1109/ACCESS.2019.2938565>
18. Dinh-Tuan, H., Beierle, F., Garzon, S.R.: MAIA: A microservices-based architecture for industrial data analytics. Proceedings - 2019 IEEE International Conference on Industrial Cyber Physical Systems, ICPS 2019 pp. 23–30 (2019). <https://doi.org/10.1109/ICPHYS.2019.8780345>



19. Dobaj, J., Krisper, M., Iber, J., Kreiner, C.: A microservice architecture for the industrial internet-of-things. ACM International Conference Proceeding Series (2018). <https://doi.org/10.1145/3282308.3282320>
20. Engineering, D.S.: Running a xenomai application as a regular user (2014). [https://gitlab.denx.de/Xenomai/xenomai/-/wikis/Running\\_As\\_Regular\\_User](https://gitlab.denx.de/Xenomai/xenomai/-/wikis/Running_As_Regular_User). Accessed May 2020
21. ETSI: Network functions virtualisation (nfv) (2012). <https://www.etsi.org/technologies/nfv>. Accessed May 2020
22. Foundation, L.: Real-time linux wiki (2007). <https://rt.wiki.kernel.org/>. Accessed May 2020
23. Foundation, R.P.: The raspberry-pi project (2012). <https://www.raspberrypi.org/>. Accessed May 2020
24. Fowler, M., Lewis, J.: Microservices a definition of this new architectural term (2014). <https://martinfowler.com/articles/microservices.html>. Accessed Dec 2020
25. Givchchi, O., Intiaz, J., Trsek, H., Jasperneite, J.: Control-as-a-service from the cloud: A case study for using virtualized plcs. In: 2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014), pp. 1–4 (2014). <https://doi.org/10.1109/WFCS.2014.6837587>
26. Goldschmidt, T., Hauck-Stattelmann, S., Malakuti, S., Grüner, S.: Container-based architecture for flexible industrial control applications. *J. Syst. Arch.* **84**(March), 28–36 (2018)
27. González-Nalda, P., Etxeberria-Agiriano, I., Calvo, I., Otero, M.C.: A modular CPS architecture design based on ROS and Docker. *Int. J. Interact. Des. Manuf.* **11**(4), 949–955 (2017). <https://doi.org/10.1007/s12008-016-0313-8>
28. Google: Google cloud iot solutions. <https://cloud.google.com/solutions/iot>. Accessed Apr 2021
29. Google: Production-grade container orchestration (2014). <https://kubernetes.io/>. Accessed May 2020
30. Hohr, S., Tasci, T., Verl, A.: Realization of Data Analytics Projects in Manufacturing Using a Microservice-Based Approach. Proceedings - 2019 IEEE International Conference on Mechatronics, ICM 2019 pp. 321–326 (2019). <https://doi.org/10.1109/ICMECH.2019.8722926>
31. IBM: Ibm iot solutions. <https://www.ibm.com/cloud/internet-of-things>. Accessed Apr 2021
32. Innerbichler, J., Gonul, S., Damjanovic-Behrendt, V., Mandler, B., Strohmeier, F.: Nimble collaborative platform: Microservice architectural approach to federated iot. In: 2017 Global Internet of Things Summit (GIoTS), pp. 1–6 (2017). <https://doi.org/10.1109/GIOTS.2017.8016216>
33. Kepler/CORE: The kepler project (2008). <https://kepler-project.org/>. Accessed May 2020
34. Knudsen, M.S., Kaivo-Oja, J., Lauraeus, T.: Enabling technologies of industry 4.0 and their global fore-runners: an empirical study of the web of science database. *Commun. Comput. Inf. Sc.* **1027**, 3–13 (2019). [https://doi.org/10.1007/978-3-030-21451-7\\_1](https://doi.org/10.1007/978-3-030-21451-7_1)
35. Kousiouris, G., Tsarsitalidis, S., Psomakelis, E., Koloniaris, S., Bardaki, C., Tserpes, K., Nikolaidou, M., Anagnostopoulos, D.: A microservice-based framework for integrating IoT management platforms, semantic and AI services for supply chain management. *ICT Express* **5**(2), 141–145 (2019). <https://doi.org/10.1016/j.icte.2019.04.002>
36. Kuhn, M., Franke, J.: Smart manufacturing traceability for automotive e/e systems enabled by event-driven microservice architecture. In: 2020 IEEE 11th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), pp. 142–148 (2020). <https://doi.org/10.1109/ICMIMT49010.2020.9041240>
37. Li, F., Lachenmayr, M., Fröhlich, J., Stückjürgen, C., Buschmann, F., Schall, D., Meixner, S.: Microservice patterns for the life cycle of industrial edge software. ACM International Conference Proceeding Series (2018). <https://doi.org/10.1145/3282308.3282313>
38. LLC, A.: The arduino project (2008). <https://www.arduino.cc/>. Accessed: May, (2020)
39. McKee, D.W., Clement, S.J., Almutairi, J., Xu, J.: Massive-scale automation in cyber-physical systems: Vision amp; challenges. In: 2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS), pp. 5–11 (2017). <https://doi.org/10.1109/ISADS.2017.56>
40. Mena, M., Criado, J., Iribarne, L., Corral, A.: Digital dices: towards the integration of cyber-physical systems merging the web of things and microservices. In: Schewe, K.D., Singh, N.K. (eds.) *Model and Data Engineering*, pp. 195–205. Springer International Publishing, Cham (2019)
41. Microsoft: Azure iot hub. <https://azure.microsoft.com/en-us/services/iot-hub/>. Accessed Apr 2021
42. of Automation, I.S.: Rami 4.0 - isa. <https://www.isa.org/intech-home/2019/march-april/features/rami-4-0-reference-architectural-model-for-industr>. [Accessed: Apr 2021]
43. Omar, A., Imen, B., M’Hammed, S., Bouziane, B., David, B.: Deployment of Fog Computing Platform for Cyber Physical Production System Based on Docker Technology. Proceedings - 2019 3rd International Conference on Applied Automation and Industrial Diagnostics, ICAAID 2019 **1**(September), 1–6 (2019). <https://doi.org/10.1109/ICAAID.2019.8934949>

44. Pontarolli, R.P., Bigheti, J.A., Fernandes, M.M., Domingues, F.O., Risso, S.L., Godoy, E.P.: Microservice orchestration for process control in industry 4.0. In: 2020 IEEE International Workshop on Metrology for Industry 4.0 IoT, pp. 245–249 (2020). <https://doi.org/10.1109/MetroInd4.0IoT48571.2020.9138228>
45. PTC: Thingworx iiot solutions platform. <https://www.ptc.com/en/products/thingworx>. Accessed Apr 2021
46. Qi, Q., Tao, F.: Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. *IEEE Access* **6**, 3585–3593 (2018). <https://doi.org/10.1109/ACCESS.2018.2793265>
47. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.: Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software, vol. 3 (2009)
48. Rufino, J., Alam, M., Ferreira, J., Rehman, A., Tsang, K.F.: Orchestration of containerized microservices for IIoT using Docker. Proceedings of the IEEE International Conference on Industrial Technology pp. 1532–1536 (2017). <https://doi.org/10.1109/ICIT.2017.7915594>
49. Schäffer, E., Leibinger, H., Stamm, A., Brossog, M., Franke, J.: Configuration based process and knowledge management by structuring the software landscape of global operating industrial enterprises with Microservices. *Proc. Manuf.* **24**, 86–93 (2018). <https://doi.org/10.1016/j.promfg.2018.06.013>
50. Schäffer, E., Mayr, A., Fuchs, J., Sjarov, M., Vorndran, J., Franke, J.: Microservice-based architecture for engineering tools enabling a collaborative multi-user configuration of robot-based automation solutions. *Procedia CIRP* **86**, 86–91 (2020). <https://doi.org/10.1016/j.procir.2020.01.017>
51. Schneider, S., Peuster, M., Hannemann, K., Behnke, D., Muller, M., Bök, P., Karl, H.: “producing cloud-native”: Smart manufacturing use cases on kubernetes. In: 2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 1–2 (2019)
52. Senington, R., Pataki, B., Wang, X.V.: Using docker for factory system software management: experience report. *Procedia CIRP* **72**, 659–664 (2018). <https://doi.org/10.1016/j.procir.2018.03.173>
53. Shearer, C.: The crisp-dm model: the new blueprint for data mining. *J. Data Warehousing* **5**(4), 13–22 (2000)
54. Simetinger, F., Zhang, Z.: Deriving secondary traits of industry 4.0: A comparative analysis of significant maturity models. *Syst. Res. Behav. Sci.* **37**(4), 663–678 (2020)
55. Siqueira, F., Davis, J.G.: Service Computing for Industry 4.0: State of the Art, Challenges, and Research Opportunities. *ACM Comput. Surv.* **54** (2021). <https://doi.org/10.1145/3478680>. <https://doi.org/10.1145/3478680>
56. Strljic, M.M., Korb, T., Tasci, T., Tinsel, E.F., Pawlowicz, D., Riedel, O., Lechler, A.: A platform-independent communication framework for the simplified development of shop-floor applications as microservice components. Proceedings of the 2018 IEEE International Conference on Advanced Manufacturing, ICAM 2018 pp. 250–253 (2019). <https://doi.org/10.1109/AMCON.2018.8615044>
57. Tasci, T., Melcher, J., Verl, A.: A Container-based Architecture for Real-Time Control Applications. 2018 IEEE International Conference on Engineering, Technology and Innovation, ICE/ITMC 2018 - Proceedings pp. 1–9 (2018). <https://doi.org/10.1109/ICE.2018.8436369>
58. Thramboulidis, K., Vachtsevanou, D.C., Solanos, A.: Cyber-physical microservices: An IoT-based framework for manufacturing systems. Proceedings - 2018 IEEE Industrial Cyber-Physical Systems, ICPS 2018 pp. 232–239 (2018). <https://doi.org/10.1109/ICPHYS.2018.8387665>
59. Tseng, F.H., Tsai, M.S., Tseng, C.W., Yang, Y.T., Liu, C.C., Chou, L.D.: A lightweight autoscaling mechanism for fog computing in industrial applications. *IEEE Trans. Industr. Inf.* **14**(10), 4529–4537 (2018). <https://doi.org/10.1109/TII.2018.2799230>
60. Um, C., Lee, J., Jeong, J.: Virtualized oneM2M System Architecture in Smart Factory Environments. 2018 28th International Telecommunication Networks and Applications Conference, ITNAC 2018 pp. 1–6 (2019). <https://doi.org/10.1109/ATNAC.2018.8615394>
61. Varela, M., Machado, J., Putnik, G., Manupati, V., Rajyalakshmi, G., Trojanowska, J., et al.: Collaborative manufacturing based on cloud, and on other i4. 0 oriented principles and technologies: a systematic literature review and reflections. *Manag. Prod. Eng. Rev.* (2018)
62. Vyatkin, V.: The iec 61499 standard and its semantics. *IEEE Ind. Electron. Mag.* **3**(4), 40–48 (2009)
63. Wang, W., Fan, L., Huang, P., Li, H.: A new data processing architecture for multi-scenario applications in aviation manufacturing. *IEEE Access* **7**, 83637–83650 (2019). <https://doi.org/10.1109/ACCESS.2019.2925114>
64. Xu, L.D., Xu, E.L., Li, L.: Industry 4.0: State of the art and future trends. *Int. J. Prod. Res.* **56**(8), 2941–2962 (2018). <https://doi.org/10.1080/00207543.2018.1444806>

65. Yin, L., Luo, J., Luo, H.: Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing. *IEEE Trans. Ind. Inf.* **14**(10), 4712–4721 (2018). <https://doi.org/10.1109/TII.2018.2851241>
66. Zhou, J., Li, L., Zhou, N.: Research and application of battery production data management system based on microservice. *ICEIEC 2019 - Proceedings of 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication* pp. 660–664 (2019). <https://doi.org/10.1109/ICEIEC.2019.8784670>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Giuseppe Di Modica** received the Ph.D. in Computer Science and Telecommunication Engineering from the University of Catania, Italy, in 2005. He is currently an assistant professor with the University of Bologna, Italy. His research interests include Cloud Computing and multi Cloud, Edge/Fog computing, Big Data, Internet of Things, Industry 4.0, SOA, Microservices, Business Process Management.

**Luca Foschini** received the Ph.D. degree in computer science engineering from the University of Bologna, Italy, in 2007. He is currently an Associate Professor of computer engineering with the University of Bologna. His interests span from integrated management of distributed systems and services to mobile crowd-sourcing/-sensing, from infrastructures for the deployment of Industry 4.0 solutions to fog/edge cloud systems. Finally, he is serving as secretary of the ComSoc CSIM TC, and as voting member and awards committee chair for the IEEE ComSoc EMEA.