SIRDAM4.0: a Support Infrastructure for Reliable Data Acquisition and Management in Industry 4.0

Published Version:
Corradi A., Di Modica G., Foschini L., Patera L., Solimando M. (2022). SIRDAM4.0: a Support Infrastructure for Reliable Data Acquisition and Management in Industry 4.0. IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, 10, 1605-1620 [10.1109/TETC.2021.3111974].

Availability:
This version is available at: https://hdl.handle.net/11585/851000 since: 2022-11-24

Published:
DOI: http://doi.org/10.1109/TETC.2021.3111974

SIRDAM4.0: a Support Infrastructure for Reliable Data Acquisition and Management in Industry 4.0

Antonio Corradi, *Senior Member, IEEE,* Giuseppe Di Modica, *Member, IEEE,* Luca Foschini, *Senior Member, IEEE,* Lorenzo Patera, *Member, IEEE,* and Michele Solimando, *Member, IEEE*

Abstract—The main claim of the Industry 4.0 manifesto (I4.0) is the promise of a significant transformation of production and delivery processes due to the digitization of manufacturing. A mandatory requirement of the I4.0 transition is the pervasive adoption of Information Technologies (IT), such as Industrial Internet of Things (IIoT), Cyber-physical Systems (CPS), and Big Data, starting with business management departments down to production sites. Despite IT has reached a consistent maturity level, its integration with Operation Technology (OT) is still a mostly unresolved challenge. In this paper, we propose the design of a platform that supports reliable data gathering and sharing among OT and IT layers of an industrial manufacturing company, while also putting the basis for straightforward integration of business stakeholders in the data sharing loop. By leveraging widely used open-source tools, we implemented a software prototype of the platform that Small Medium Enterprises can use to face the IT/OT convergence challenge in an affordable way. Results collected from an extensive assessment in real manufacturing settings show that the proposed solution allows meeting both functional and non-functional requirements of a typical data gathering/sharing process in a near-real-time scenario, by granting reliability with very low overhead cost.

Index Terms—Industry 4.0; SME; IT/OT convergence; IIoT; SCADA; resilience; scalability

1 Introduction

The Industry 4.0 (I4.0) smart factory scenario is currently under great consideration by both big companies and central/regional governments, pushed by the expectation of a groundbreaking change of perspective in the manufacturing sector. Major expectations include the delivery of new services, the provisioning of new business opportunities, the creation of new employment, and many more [1, 2, 3]. In the aim of pushing for a coordinated and homogeneous approach towards the smart factory paradigm, standardization groups [4, 5, 6] as well as governments worldwide are proposing new unifying architectures and taxonomies addressing both business and technological issues. The main goal is to provide a common playground to better identify the main critical aspects and easily compare different solutions.

Within manufacturing enterprises, especially SMEs, a big obstacle to the adoption of standard principles and guidelines is the obsolete and rigid separation between technologies that characterize departments devoted to product manufacturing (hosting work machines and production lines) and departments committed to managerial tasks. Differently from traditional sensor and actuator devices that have been used in industrial applications for years, Industrial IoT (IIoT) devices exchange data over the network exploiting a variety of network protocols and make large amounts of new types of data available, contributing to streamlining production [7]. As a result, it has become possible for components and machines to manage production autonomously in a flexible, efficient, and resource-saving manner. Therefore, especially for the manufacturing industries, the automatic collection and management of different data coming from the Operation Technology (OT) floor and Information Technology (IT) support became an essential enabler for new business models and opportunities in terms of servitization of the shop floors, enabling core services for I4.0 environments such as, for instance, predictive maintenance.

Recently, in the industrial and manufacturing world, a growing hype has surrounded the convergence of IT and OT layers [8], along with relative solutions and implications. According to Gartner [9], by 2020 50% of OT service providers will make new deals with IT service providers in order to increase the range of services provided by IoT devices used in industrial production sites, and many companies are moving along that direction. While the research community is very active on the IT/OT convergence topic, to the best of our knowledge state-of-art proposals do not offer SMEs concrete and affordable solutions to attain the convergence goal. In a former work [10], we discussed the design and implementation of a platform for collecting data from manufacturing machines and enabling data exchange between OT and IT layers. In this paper, motivated by a data sharing scenario involving both manufacturing factories and stakeholders of the production chain, we propose our Support Infrastructure for Reliable Data Acquisition and Management in Industry 4.0 (SIRDAM4.0). In particular, this

- *A. Corradi, G. Di Modica, L. Foschini, L. Patera, and M. Solimando were with the Department of Computer Engineering, University of Bologna, Italy.*
  E-mail: {antonio.corradi, giuseppe.dimodica, luca.foschini, lorenzo.patera, michele.solimando}@unibo.it
  CORRESPONDING AUTHOR: L. Foschini (luca.foschini@unibo.it)

paper advances the findings of [10], by providing several elements of novelty that include:

- the implementation and test of a support that grants stakeholders safe and selective access to valuable data reflecting the production dynamics;
- the implementation and test of a support that guarantees service continuity in case of data gathering software components faults;
- stress tests of the platform with respect to heavy and increasing workloads.

The rest of the paper is organized as follows. We review the state of the art in Section 2. In Section 3, we discuss how standardization efforts have been addressing IT/OT convergence, and dwell on convergence opportunities and threats. In Section 4, we present the scenario that motivated this work. We briefly recall the design and the implementation details of our platform in Sections 5 and 6. In Section 7, we show the results of extensive tests conducted on the platform software prototype. Finally, we draw the conclusion and mention our future work in Section 8.

2 Literature review

The most authoritative I4.0 standards are pushing the concept of *smart factory*, where "objects" at any level (machines, products, processes, and even workers) are connected to each other and also reachable from outside of the factory premise [4]. The transition to such a revolutionary perspective is enabled by the progressive adoption of information technologies also at the operation level, toward the full IT/OT convergence. Despite the many potential advantages that convergence may face, it still poses a number of mostly unresolved challenges that researchers and practitioners should solve. Among the most compelling ones, we can cite the need for dependable support for data acquisition and management, and the design of a reliable and safe industrial environment comprising machines, workers, and crucial company information. In our survey, we collected research contributions that address the IT/OT convergence from the data management perspective. Specifically, In Section 2.1 we discuss relevant literature works that propose data acquisition and management solutions in IT/OT converging environments. In Section 2.2, we review works that propose practical solutions to face typical IT/OT integration issues. Finally, in Section 2.3 we suggest some analysis dimensions on which we build a comparative framework for the surveyed literature. At the end of the analysis, we sum up the results in Table 1 to help the reader grasp the advantages offered by our platform compared to those offered by state-of-art proposals.

2.1 Data acquisition and management

In [11], authors investigate the use of Digital Twins (DT) for simulating processes, systems, and equipment in industrial production environments. They address the problem of gathering OT data to feed DT. The proposed solution consists of connecting the Apache Kafka message broker with Kepler, software for the design and execution of scientific workflows. The resulting Digital Twin model runs in the Cloud and manages to simulate real factory objects. Despite authors address an interesting research point, they only deal with issues related to a specific use case, and do not tackle the problem of fast data processing.

Similarly, in [12] authors implement a stateful stream processing tool to feed digital twins 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. Microservices provide the required level of scalability when the load of messages increases, while the Kafka broker provides the possibility of managing the data processing state by synchronizing the local state with the set of intermediate Kafka topics. Solid management of the state, in its turn, proves to be effective in helping the system to recover from faults.

In [13], authors address the gathering and processing of machines real-time data streams. They propose a classical IoT platform consisting of a gateway to which sensor devices are connected. The platform supports the continuous collection of data from the FIWARE IoT Hub component placed between data producers and data analytic tools. This core component supports device communication and data management. Data collected by IoT agents traverse a Complex Event Processing (CEP) broker and are finally delivered to consumers by way of AMQP brokers and web sockets. To test the platform, the authors used the Factory Automation Systems and Technologies Laboratory (FAST Lab), a test bench for modern I4.0 assets.

In [14], authors focus on an electrical utility use case to speculate about the advantages of a novel IT/OT integration framework. The paper schematizes all the tasks and functions operating in the electric facilities and shows how they can potentially converge in a unifying framework for IT and OT domains. The work proposes a new Integrated Outage Management System (OMS) leveraging a common data framework to integrate the call center applications with the SAP, SCADA, Geographic Information System (GIS), and Automated Meter Reading (AMR) tools. The outcome of the integration was the drastic decrease in the time required to restore an outage and the immediate update of the GIS coordinates, operations that require much more time and effort if performed with non-convergent classical infrastructure. A similar work [15] conducts an analysis on an O&G scenario and highlights the great advantages deriving from IT/OT convergence. Authors claim that convergence is not just about integrating the two layers on a common platform. The skills of OT and IT departments professionals must be enhanced as well, since convergence is not only a matter of updating tools but, rather, re-thinking of responsibilities and assignments within the production sites.

In [16], authors claim that the future of the Internet of Things, Service, and People (IoT-SP) for the industrial sector is the IT/OT integration. They draw attention to one O&G production example, proposing their idea of "digital oilfield" to acquire information about what is working best for this domain in order to predict equipment failures, track employees in the firm, train them, and signal hazardous situations in real-time. They set up an advanced test rig to observe the influence of electrical faults on some components used in the O&G field, i.e. the compressors, basic devices for gas transportation. The resulting testbed, called ORKAN, can be used to test the integration of new equipment with industrial IT networking, simulating cloud and local connections, and providing condition monitoring and diagnostic algorithms testing.

[17] proposes a novel mechanism for the auto-configuration of an OPC UA server, considering the joining (or the failure) of PLCs and industrial devices in the production plant network. The authors, after introducing the importance of autoconfiguration in modern industrial environments, show an interesting industrial ecosystem as a testbed, composed of OPC UA server/clients and an Arduino One, and an M-DUINO PLC as data generators. The tests provided show different industrial scenarios and demonstrate how, thanks to the autoconfiguration capabilities, it is possible to continuously run an OPC UA server, also subsequently to the adding/failure of managed industrial devices, reducing deployment and maintenance costs.

Also, for what concerns data gathering at the OT level, our proposal guarantees high performance in terms of timeliness of data access, which in turn strives for high security within CPS systems, with machines operating in strict collaboration with human employees.

Let us conclude by noting that the above works represent authoritative points of view on the implementation of the IT/OT convergence paradigm. They helped us to identify requirements and a potential approach to enforce a data-wise convergence of IT and OT. With respect to the reviewed works, the data gathering support presented in this paper exhibits both significant capabilities to fairly scale up with the increase of data volume and to tolerate unexpected software faults.

2.2 IT/OT integration experiences

In [18], authors propose the introduction of new IT paradigms into the OT domain to accomplish the convergence. They present an intermediate layer of integration between IT systems and OT devices that blends seamlessly and in a cost-effective manner the legacy systems in modern deployments. They suggest a model based on the ISA 95 industrial standard [19] and promise minimal modification to the OT layer. Outlining the impracticality of connecting legacy devices with external domains, they argue that the convergence with the IT world must occur at the data level (raw data produced by machines, components, and internal registers). In this respect, they use the semantic model formalized by the ISA 95 standard family as a reference model to describe industrial equipment and its relationships. In the paper, three use cases are also discussed in which authors measure the dispatch time of the components performing the integration of legacy hardware through the semantic model.

In [20], authors propose an interesting and practical solution to IT/OT convergence leveraging the semantic technology. They try to take down the IT/OT boundaries by defining a new modeling language that shapes all OT, IT, and business entities. Interesting features offered by the language are the possibility to represent existing OT standards, building a model from an existing one, and adapting the general description to specific use cases. The proposed solution extends ArchiMate [20], a well-known language for describing complex enterprise structures, involving IT systems, business, and organizational processes, and information flows. They propose an extension of the language that specifically models the oil and gas (O&G) domain, and validate it with the collaboration of 10 industry experts from 5 different O&G companies.

In [21], authors adopt the *asset health* approach and test it in Smart Grid (SG) use cases involving American Electric Power (AEP), a big energy transmission system operator (TSO) in the USA. They leverage IT/OT integration to enable the prescriptive maintenance on SG assets. The asset health approach is a combination of tools, strategies, and processes that enterprises can use to predict the future health of their asset. They outline the direct correlation between the asset health application, the reduction of unplanned maintenance, and the lowering of operating and capital expenditures (OpEx and CapEx).

In [22], authors present an original approach that promotes the self-configuration of real-time networks as an enabler of IT/OT convergence. They call upon software-defined network (SDN) and time-sensitive networking (TSN) to achieve their objective. The TSN paradigm enables the use of Ethernet protocols in fields such as industrial automation and automotive and the SDN, separating data and control plane. It offers many freedom degrees in configuring the network deployment. Authors claim that the proposed system can be adopted in critical environments such as corporate production sites in the manufacturing sector.

Authors of [23] build a practical laboratory to demonstrate the possibility of connecting the services implemented in the operative environment with tools available at the IT layer. In the proposed industry 4.0-compliant laboratory, the OT layer is a service-oriented shop floor, while the IT layer includes three main management systems: an ERP system, a database, and a tool for simulating predictive maintenance. This work put forward the stochastically handling of unpredictable events as the main result of convergence between OT devices and IT capabilities. The work proposes the laboratory as a useful benchmark for testing algorithms devised by the research community and as well as functions developed by companies, tearing down a barrier that prevents small and medium enterprises (SMEs) from joining the transition to industry 4.0.

Although analyzed works present interesting perspectives and solutions addressing the IT/OT convergence, with respect to our work they only partially follow the principles and good practices found in literature and learned during field testing. [18] and [20] mostly deal with the problem of semantic convergence of shop floors entities. [21] and [23] address the management of the asset, neglecting other entities of the complex industrial system; they also lack a very practical use case and a comparison to other test cases. The TSN analysis proposed in [22] is a helpful point of view, but it does not take care of integrating legacy assets, which, instead, is one of the targets specifically addressed by our work. Finally, we remark that our platform leverages off-the-shelf equipment and open-source tools. This accommodates the needs of SMEs of embracing the I4.0 revolution at a very limited cost. However, load-intensive tests run in our benchmark proved that the platform can fit both medium and big enterprises.

2.3 Comparative analysis

From all sources, it emerges that IT/OT convergence is a hot topic attracting the interest of both the manufacturing industry and research communities. We found a number of solutions that propose approaches to implement the integration of shop floors and IT departments at the data layer. We also analyzed the technological barriers and research gaps that hinder the full adoption of IT/OT integration solutions by SMEs. The viability of an integration approach must take into account the following main aspects:

- *Support for legacy protocols*. In the near future, most manufacturing companies are not willing to invest money to renovate their equipment. In that respect, integration with legacy protocols may not be disregarded.
- *Scalability to the increase of workload*. Production needs to adapt to the market volatility. An increase in the market demand may trigger more intense data workloads that the system needs to cope with.
- *Tolerance to unexpected faults*. Resiliency is a must-have feature of any data gathering and management solution, as interruptions in operation may severely impact production.
- *Secure and safe data/asset management*. Data security and the safety of manufacturing working environments are among the main concerns of companies. A strong defense against data leakage as well as the intrusion of malicious actors must be deployed.

In Table 1, we report a comparative scheme that highlights which of the above-listed aspects have been addressed by the recent literature. For comparison purpose, we selected the works that propose a platform or a framework for data management in production plants. Among those, some address specific sectors ([16], [20], [21]), while others specifically focus on the implementation of more general-purpose Digital Twins ([11], [12]). Depending on the level of support offered by the proposal, aspects may be marked as fully addressed (✓), partially addressed ((✓)), or not addressed (x) respectively.

As regards the support of existing communication protocols, the works [12] and [16] do not provide any compliance with the legacy protocols adopted in the production sites. Other works envisage some form of integration, i.e. [11] through the use of Kafka, [22] suggesting to connect TSN and OPC UA, and [23] proposing different configurations of the OPC UA protocol. From this point of view, the ones proposing a full support for legacy protocols are [13], [17], [18], and [20]. Specifically, [13]provides an IoT hub, the architecture introduced in [17] supports the pluggability between OPC UA and other Modbus-like protocols, while [18] and [20] provide, respectively, a layer of interoperability between different protocols and semantic modeling of resources at the OT level.

The platforms introduced in the works [11], [12], and [17] address scalability. [11] and [12] use natively scalable tools such as Kafka and [17] provides for the automatic configuration of OPC UA systems based on the number of controlled PLCs. Some scalability aspects are covered in [13], with the support of the Cloud, and in [18].

TABLE 1: Literature Review Comparison

	Legacy Support	**Scalability**	**Fault Tolerance**	**Safety/ Security**
SIRDAM4.0	✓	✓	✓	(✓)
[11]	(✓)	✓	x	x
[12]	x	✓	✓	x
[13]	✓	(✓)	x	x
[16]	x	x	(✓)	✓
[17]	✓	✓	x	x
[18]	✓	(✓)	x	x
[20]	✓	x	x	x
[22]	(✓)	x	x	x
[23]	(✓)	x	x	x

Robust fault tolerance is implemented by [12] through the use of intermediate topics in Kafka Stream. However, [16] provides a certain degree of fault tolerance in its ORKAN test architecture, although the implementation is not provided.

[16] takes into account the safety of the production environment and proposes secure mechanisms for information exchange in the production sites, such as the defense-in-depth approach. The rest of the works lack dedicated care on the security issue, so no implementation is provided at all or the authors simply use the security mechanisms bundled in the tools used to implement the architectures.

We claim that, compared to other works, SIRDAM4.0 addresses all aspects, although overall security is currently not fully supported.

3 IT/OT Convergence

In the modern I4.0 perspective, the *connected factory* concept is a key technology enabler for manufacturing operators, to drive the production environments up to an expecting development impacting both IT and OT layers. The technological improvement of Industrial IoT (IIoT) brings new highly reliable devices with advanced built-in communication capabilities, since new state-of-the-art devices, together with the work machines operating in production sites, generate a huge amount of data. Any division of the organization, from design departments to shop floors, provided with adequate access permission, can take advantage of this unprecedented data depot.

The attainment of most goals of the I4.0 transition is bound to the gradual integration of algorithms used by manufacturing machines with information about surrounding infrastructure. In the literature, this trend is referred to as *IT/OT layers convergence*, which denotes the decrease in the gap between the manufacturing processes in the shop floor, on the one hand, and IT department resources (storage, networking, computing facility) on the other hand.

3.1 Standardization efforts

Convergence is a topic covered by most authoritative standardization bodies such as the International Society of Automation (ISA) (https://www.isa.org/) and the International Electrotechnical Commission (IEC) (https://www.iec.ch/), with a lot of specifications covering aspects dealing with the convergence.

The US Industrial Internet Consortium (IIC) (https://www.iiconsortium.org/) proposes the *Industrial*

Internet Reference Architecture (IIRA) [5] as guidance in developing smart IIoT architectures. The specification strongly stresses the concept of *convergence of the IT and OT layers* at various levels, underlining the common requirements for tackling the transition from an industrial perspective with proper computer architectural patterns.

Worldwide, other authoritative standards like, e.g., the European *Reference Architectural Model Industrie 4.0 (RAMI 4.0)* [4] and the Chinese *Intelligent Manufacturing System Architecture (IMSA)* [24], push towards tighter integration of IT and OT. RAMI 4.0 and IMSA are comprehensive models proposing high-level reference architectures addressing all Industry 4.0 concepts. The *smart objects* became leading actors in the production facilities, representing the industrial equipment outfitted with the right level of intelligence to allow secure control from the highest tiers of architecture, i.e. the managerial ones.

The OPC UA [6] produced by the OPC foundation [25] collocates in the panorama of I4.0 standard initiatives as a concrete effort to propose low-level and technical specifications. OPC UA defines a standard set of objects and interfaces facilitating the *interoperability among control processes and manufacturing automation applications* [6]. The many challenges of the I4.0 transition forced the OPC foundation to think about the heterogeneity of software and hardware components by proposing a unifying architecture tackling these diversities, namely the Unified Architecture (UA) specification [25], released in 2006 and become the IEC-62541 standard [26].

The work presented in this paper draws inspiration from OPC UA specification to implement a practical convergence of IT and OT layers in industrial manufacturing scenarios and Section 5 provides references to specific items of the standard that guided our work.

3.2 Opportunities and threats

Notwithstanding the several ongoing standardization efforts, a clear definition of a convergence layer to share OT data generated by different work machines at the IT layer, and consequently, the design of new services taking advantage of this larger and richer information set, is still missing.

Within the organization premise, a clear advantage coming from sharing of information between machines and neighboring IT infrastructures is *improved machine utilization*, with better focus on safety and efficiency, via remote control or the execution of safe remote operations. An infrastructure where OT and IT converge allows organizations to design more accurate Key Performance Indicators (KPIs) about their facilities. The integration of data produced by work machines is useful also to align the views provided by business software systems such as enterprise resource planning (ERP) tools, manufacturing execution systems (MES), and manufacturing information systems (MIS).

The availability of production data at the IT layer can bring further advantages also in B2B scenarios. Disclosing portions of such data to business partners will enable a new breed of services that could yield opportunities to all stakeholders. For instance, if machine vendors had given access to real-time production data of all their customers, on the one hand, they could implement more effective support services in the perspective of *preventive maintenance*; on the other hand, they would benefit from loads of real-time production data to be exploited for the business purpose toward product improvement.

Unfortunately, data integration and sharing outside the company premises will raise issues to be tackled, where security is the most relevant one. *Security* is surely the most relevant one. OT layer used to be disconnected from the network, so attacks in this area were limited to physical or near proximity-based attacks. The convergence with the IT layer, a domain where a lot of cyber attacks can occur, increases the Industrial Automation and Control Systems (IACS) attack surface. We need to enable monitoring and secure data flows also among work machines because potential flaws in IACS systems can lead to serious damages, such as the leakage of sensitive company data and concrete hazards to humans who work close to machines. Many safety aspects have been addressed by the ISA/IEC 62443 standard developed by the ISA99 committee [27], whose objective is to develop and share standards and best practices for designing, implementing, managing manufacturing and control systems in a secure way. *Integration of legacy systems and protocols* already employed in the shop floors is a very sought feature in IT/OT convergence and in I4.0 transition in general [10, 18, 28]. Most OT assets were not designed to interact with modern communication protocols largely used in IT environments, because the software they use is obsolete or not pursuing the same purposes as the software present at the IT level. Therefore, OT does not natively fit remote monitoring and control. The integration layer has to take into account the problem of legacy assets, in order to avoid that the I4.0 transition becomes infeasible for SMEs that do not have economic resources to upgrade their fleet. A further drawback is a need to *cross-train employees* of the two different departments since convergence inevitably leads to the mix of heterogeneous technologies. Therefore, it is paramount that the two departments interact to reach a common knowledge base that allows in-depth and shared knowledge of the convergent system, that is no longer confined in separate areas of the company.

The reader will find a thorough discussion of other threats deriving from the enforcement of IT/OT convergence in Section 2.

4 MOTIVATING SCENARIO AND CONTRIBUTION

The present work proposes advancements to a research line undertaken in collaboration with many manufacturing companies based in the "Packaging Valley" district located in Emilia Romagna, Italy (http://thepackagingvalley.com). Grounding on requirements elicited from the district manufacturing companies, we aim to develop tools to concretely support manufacturing companies in the transition to Industry 4.0.

During the first steps of collaboration, we drafted a guideline for the implementation of the convergence of the factory OT and the IT layers as a way to enable the Industry 4.0 transition [28]. In the following scientific contribution, we discussed main technological requirements at the basis of a smart factory scenario and a first draft architectural view of a data gathering and integration platform [10]. The platform aims to address some of the challenges raised by IIRA and OPC-UA. At the current stage, the platform provisions data gathering and structuring at the OT layer, and transmission of such data to upper layers. In particular, it implements a communication pattern between the company departments that is based on asynchronous messages exchange carried out by a Message Oriented Middleware (MOM) that adopts the publish-subscribe model. The MOM enforces IT/OT integration at the data level that acts as a data conveyor from bottom to top layers, but at the moment inhibits the flow of control commands from top layers downwards, so to avoid by design the possible threats due to exposing the OT to the direct control of an external entity.

In Figure 1, we have depicted a typical representation of the technological layers of a manufacturing factory. At the bottom layer, machines and workers strictly interact for production purposes. Production data generated by machines are used by workers to correctly operate the production line. Here, relevant data streams are characterized by great speed and variety, since a high volume of data is generated by the many working machines. Furthermore, a strict data access mechanism is required to prevent malicious intruders from stealing confidential information or injecting data that could eventually bring the operational layer to an unsafe state.

Data Gathering in the OT layer represents the process of real-time collecting data produced by machines at the CPS layer. OT layer consumes data gathered for operational purposes transferring them to upper layers for business purposes. Data flows generated by this function are usually filtered before reaching the IT layer (not all production data are useful at upper layers). Once here, they feed a number of IT tools to help business experts to control the production and coordinate other managerial tasks. The *Data Gathering* system shall act as a *separation layer* for the underlying layer so as not to expose machine-production data but, at the same time, to shield the shop floor from any attempt to make direct access to machines.

Let us now consider a typical industrial scenario involving many actors of the manufacturing sector, and focus our attention on the data gathering and management issues that this scenario will raise. In the prospected scenario, two competing manufacturing companies *Company A* and *Company B* run production lines in Imola and in Milan respectively. Both company production lines are operated by machinery manufactured by a *Machine Manufacturer Company (MMC)*. Let us also assume that Company A needs to increase the production in order to catch a new business opportunity, so as to establish a new production line in Bologna. Due to this expansion, both *Company A* premises will deploy a LOCAL IT layer equipped with some data filtering and aggregation tools, while only Imola premise will also deploy a CENTRAL IT layer responsible for aggregating data coming from the two LOCAL IT layers before feeding them to business-level ERP software. Finally, we can assume that other companies deploying MMC machinery may join the scenario (e.g., Company N).

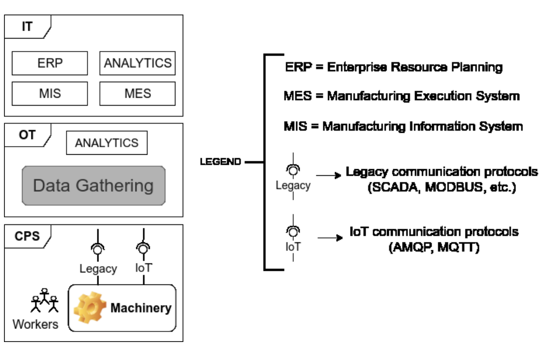


Fig. 1: Technological layers in a manufacturing factory

To defend and grow its market share, MMC aims at continuously improving its product so to take advantage of gaining information on production data of machines running at customer premises: such data, if timely analyzed and processed, will help MMC to spot machine misbehavior and detect potential causes (to cite a few: misconfiguration of machine parameters, machine design defects, assembly defects). Unfortunately, disclosing customers' production data to MMC poses a huge security problem. Customers, for obvious reasons, refrain from disclosing anyone (not even the machine manufacturer) their data. Yet, with the opportunity of receiving by MMC a timed and more effective technical support, customers may be willing to disclose agreed portions of their production data. MMC by accessing these data could promptly detect and diagnose run-time anomalies, suggest more effective machine parameters' configuration/setting, advise machine part replacement, etc.

To enforce the described scenario and meet the need of all stakeholders in terms of availability and accessibility to relevant data, we considered the following requirements:

- R1. *Timely access to data*. At the shop floor level, data must be made timely available and accessible. The purpose is twofold: complying with the near-real time constraints of the targeted production process and promptly undertaking countermeasures in case of potential hazards.
- R2. *Handling of heavy workloads*. Depending on the market demand, to deal with requests for increasing production, new machines might have to be added to production lines. The data gathering system must be able to absorb spikes in data generation and guarantee a timely data delivery also in case of heavy workloads.
- R3. *Controlled access to data*. Access to data needs to be regulated. Data must be carefully partitioned and made available to the intended recipient (be it the shop floor, the company business department, or the machine manufacturer) only for specific use.
- R4. *Tolerance to faults*. In order to maximize the company profit, close to 100% machine operational continuity has to be granted. Faults occurring at any level, and in particular, at the OT layer, have to be solved in a time that is compatible with the criticality of the data that could potentially get compromised by a shutdown.

The definition of a framework for gathering data at OT layer and making them safely accessible to stakeholders is an important step that poses the basis for the OT/IT

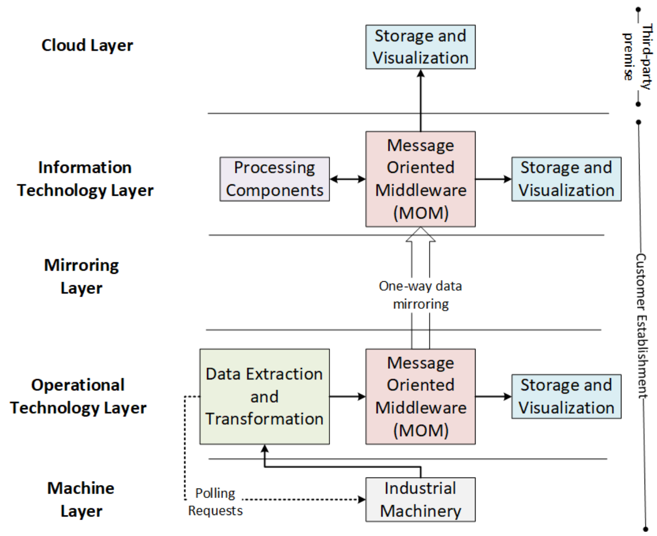


Fig. 2: Overall architecture schema

convergence in industrial settings. In [10] we presented the design and implementation of a platform that gathers the OT data and guarantees real-time access to data consumers, thus addressing *R1*. The aim of the present paper is to prove that the platform also meets other requirements listed above.

5 Platform Design

In this section, we recall the architectural model presented in [10] that addresses the IT/OT convergence issues. Furthermore, we discuss some relevant details of the software prototype of the data gathering platform we implemented and employed during the tests.

Experience gained in past collaborations with important manufacturing enterprises and corroborated by an in-depth study of the main revolutionary industrial specifications, helped us draft some guidelines [28] that has driven the design and implementation of the platform *SIRDAM4.0*. The platform supplies the following services:

- gathering of data produced by manufacturing machines;
- long-term storage, fast processing and user-friendly presentation of such data at OT layer;
- secure mirroring of operational data flows for use by IT departments;
- secure and selective provision of operational data to third party stakeholders.

SIRDAM4.0 design principles inspire mainstream I4.0 standardization activities such as RAMI 4.0, IIRA, and OPC UA. A feature SIRDAM4.0 borrowed from all specifications is secure and selective access to information guaranteed to both the stakeholders acting inside a modern I4.0 manufacturing company and those operating outside (e.g., the machine vendor).

SIRDAM4.0 architecture depicted in Figure 2 reflects the physical separation of OT and IT layers enforced in most production sites and addresses their *integration at data level*.

Despite SIRDAM4.0 does not exhibit strict compliance for what concerns the implementation of entities and classes, its architecture largely adheres to OPC UA specifications. Indeed, SIRDAM4.0 follows the interoperability principle reiterated in part 1 and in part 14 of the OPC UA specification, which advises the use of Pub/Sub communication pattern. Along with that goal, we placed a message-oriented middleware (MOM) between machines and the tools charged with data elaboration and storage tasks. We also decoupled the data type of the machine registers from the specific communication protocol by ”flattening” the data type according to a unifying scheme, following the prescription reported in part 14 of the OPC UA specification.

Stemming from the fact that most SMEs have little or no economical resources to undertake the I4.0 transformation, SIRDAM4.0 also supports the gathering of data produced by existing legacy assets and its integration with modern MOMs and IoT protocols. Specifically, protocols from the SCADA family, still widely used in manufacturing realities [29, 30, 31], are fully supported by the platform.

The data streams generated at **Machine layer** are characterized by high speed, large varieties, and big volumes, due to the number of different machines operating on the shop floor. As often reiterated in both IIRA and RAMI 4.0 standards, companies need proper solutions to manage data in a secure and reliable way, avoiding damages to surrounding people and to the machines themselves during remote operations. We take the OPC UA advice (part 2) to restrict access to this layer in order to achieve the right level of security and safety. As a mandatory practice of I4.0 specifications, the **OT layer** needs to provision very low data latency, good bandwidth, enhanced security mechanisms, and resilience. In this layer, we placed a component to collect data from sources (*Data Extraction and Transformation*) and a *MOM* capable of delivering such data to consumers in a Pub-Sub fashion, and of guaranteeing a data latency compatible with near-real-time constraints. Keeping latency low allows the software of this layer (*Storage and Visualization*) to align with the update frequencies of the machines and carry out fast data processing.

On top of the OT layer, the **Mirror layer** offers support to implement a fine-grained control of the convergence. In line with the vision of the Chinese IMSA standard that recommends flexibility of management policy in relation to the requirements of the considered industrial application domain, different data storing policies (what, when, and how data have to be exchanged between OT and IT layers) can be enforced at this layer. The Mirror layer may also serve as a backup of OT-generated data, thus ensuring the whole platform a good degree of robustness with respect to potential faults of the Machine layer.

Technical and maintenance departments, as well as management and logistics, are consumers of information: the formers consume raw information, while the latter are interested in aggregated information. At **IT Layer**, multiple stakeholders need to consume different portions of the available data set that is fed with data coming from underlying layers. Then, we decided to replicate here the OT layer component scheme, which provides for a MOM distributing data according to Pub-Sub, and a set of tools (to be used by data consumers) devoted to the processing, storage, and visualization of data. The actual distinction between OT and IT at the design level lies in the relaxation of the requirements at the IT level, where no work machines operate and the risk of jeopardizing workers' safety is much lower.

Finally, SIRDAM4.0 opens to the involvement of third-party stakeholders (TPS), i.e., potential partners, sharing common goals with the company, that could generate value from production data. Being TPS outside of the manufacturing establishment, in the architectural view we collocated them in the **Cloud Layer**. This tier collects selected data coming from production sites and runs analytics over it. As a data consumer, the *Storage and Visualization component* is allowed to subscribe only to specific topics published by the IT layer MOM. This mechanism aims at avoiding any leakage of private and confidential company data that does not serve TPS purposes.

To conclude the discussion on architectural aspects, we would like to remark some of the platform features that are also strongly accounted for in the I4.0 vision of RAMI 4.0 standard:

- Isolation of the Machine layer;
- High availability of production data and real-time data processing at OT layer;
- Secure and selective access to production data by process stakeholders, be they company IT departments or external partners.

6 Platform Implementation

This section provides some implementation details of the software prototype of the SIRDAM4.0 platform. Our prototype makes use of state-of-art and open software tools with the goal not to build an enterprise commercial product, but of implementing a proof-of-concept for all our claims and proving that the first step towards I4.0 transition can also be taken by small-medium enterprises (SMEs) while keeping the transition cost low. For a more complete list of implementation details go to [10].

After surveying a list of candidate software tools that might fit our needs, we decided to use Apache Kafka Broker to implement the MOM component of our architecture and some event streaming tools offered by the Confluent suite (https://www.confluent.io). A Kafka Broker instance can handle a data ingestion rate as high as 420K messages/second [32] while guaranteeing an almost constant performance in terms of end-to-end message delay. Should the user need to handle a higher throughput, multiple Kafka Broker instances can be clustered and run as a more powerful broker. Clustered brokers also implement a data replication scheme that provides the system with high system resiliency against sudden and unexpected software faults.

6.1 Machine and OT layer

Figure 3 depicts a schematic view of machinery populating the *Machine layer* and the software components implementing the functionality of the OT layer discussed in Section 5. Since the fourth industrial revolution encompasses the interconnection of the machines, achieved during the third industrial revolution, the transition requires incorporating all legacy patterns and communication protocols. We made the choice of supporting SCADA protocols, which in most cases

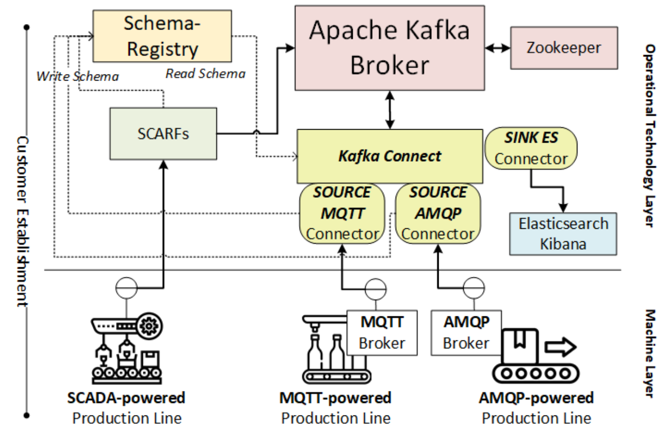


Fig. 3: Operation Technology Layer schema

are hard-coded in the firmware of manufacturing machines. Although we tested MODBUS TCP [1] (see section 7), other protocols such as Profibus [2], CANOpen [3], and DeviceNet [4] must be supported as well. Of course, modern IIoT protocols like Message Queuing Telemetry Transport (MQTT) and Advanced Message Queuing Protocol (AMQP) should be supported too, as explicitly advised in part 14 of OPC UA.

OT layer interfaces with the underlying work machines. In the following, we provide a description of the software tools implementing the functionality offered by each architecture component that populates this layer.

6.1.1 Data Extraction and Transformation

Software components devoted to provisioning the service of data extraction and transformation are listed below:

- SchemaRegistry. OPC UA specification, in parts 3 and 5, addresses the standardization of components (objects and servers) and registries inside shop floors via "AddressSpace" and "Information Model". In compliance with the OPC UA specification, SchemaRegistry implements the repository of the schema describing the format of production data, which are useful for carrying out operations on production data. As suggested in Figure 3, schema can be uploaded to or retrieved from SchemaRegistry through simple read and write operations.
- SCADA Reader and Forwarder (SCARF). As mentioned before, many machines are powered with SCADA capabilities, i.e., can interoperate via a protocol of the SCADA family. The SCARF component, implemented on top of the pymodbus tool v2.1.0[5], interfaces with SCADA-powered machines and carries out the following tasks: data retrieval, data validation, data serialization, and data forwarding to Kafka Broker. Most SCADA protocols do not provide spontaneous sending of their production data; SCARF can poll machine registry at predefined and

1. https://modbus.org/
2. https://www.profibus.com/
3. https://www.canopensolutions.com/
4. https://www.rtautomation.com/technologies/devicenet/
5. https://pymodbus.readthedocs.io/en/latest/index.html

configurable time intervals. As a general principle, a SCARF instance is instructed to read from a machine registry. In a production chain, usually populated by many work machines producing a not-negligible load of big data, we carefully adopted and tailored a lightweight format for data compression, namely AVRO (https://avro.apache.org/), a data serialization framework arranging information in a compact binary format. SchemaRegistry stores the AVRO schemes for serialization and deserialization purposes. Eventually, SCARF instances send serialized data to Kafka broker.

- IoT connectors. Different than the SCADA-powered machines, which require a polling mechanism to implement data gathering, IoT-powered machines interface to message brokers that implement the Pub-Sub mechanism. To gather data produced by such machines, a potential consumer just subscribes to machine topics and gets data while they are published by machines. We decided to support the interaction and communication with machines powered by MQTT and AMQP messaging protocols. Specifically, in the Machine layer, MQTT and AMQP messages are managed by *Eclipse Mosquito* (https://mosquitto.org) and *RabbitMQ* (https://www.rabbitmq.com/) message brokers respectively. In the OT layer, *MQTT connector* and *AMQP connector* of the Confluent suite act as subscribers of messages published by production machines. The Kafka Connect components (https://docs.confluent.io/current/connect/index.html) are open-source Kafka plugins containing converters and connectors to interface the Kafka broker with external platforms, both source and destination of data.

6.1.2 Message Oriented Middleware

The *Apache Kafka Broker* (https://kafka.apache.org) implements the MOM component of the architecture. It is a typical message broker that supports the Pub-Sub mechanism for distributing messages among participants. In this layer, data producers (i.e., the publishers) are SCARFs, MQTT, and AMQP Brokers via the respective connectors, while Elasticsearch is the only data consumer (subscriber). We would like to stress that Kafka Broker represents the software component that physically ”shields” the OT layer from the overlying layers, but at the same time, it is where the first step of IT/OT convergence is taken. It represents a gate through which production data can flow upwards to reach stakeholders (both internal and TPS). To enforce security, no message originated by the IT layer is allowed to transit to the OT layer: all stakeholders of overlying layers can just act as subscribers of OT layer topics.

Finally, Kafka Broker calls upon Zookeeper (https://zookeeper.apache.org/) as a coordinating central point for retrieving services such as naming and distributed synchronization.

6.1.3 Storage and Visualization

We selected *Elasticsearch* (https://www.elastic.co/elasticsearch) as a long-standing storage tool and *Kibana*

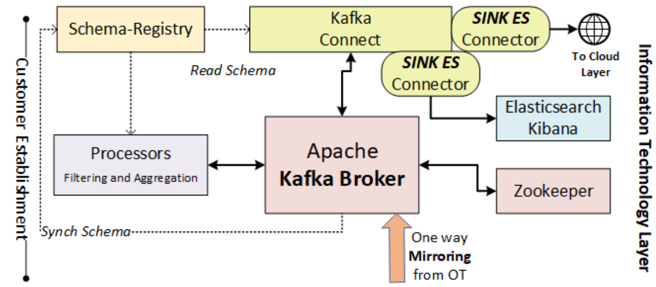


Fig. 4: Information Technology Layer schema

(https://www.elastic.co/kibana) for presenting data to the customers. We chose the document-oriented Elasticsearch storage tool for its speed, scalability, and search options features. Kibana guarantees a high customization level, which allowed us to define a dashboard for each plant (and a view for every machine inside it), for its capabilities of defining users, roles, accesses to data, and for the nice rendering of interactive graphs, tables, and pie charts. The *Sink ES connector* is a subscriber of Kafka Broker topics that consumes the data, deserializes them via a schema retrieved from SchemaRegistry, applies any required transformation, and delivers them to Elasticsearch for storage.

6.2 Mirror, IT, and Cloud Layer

The company can deploy resources for mirroring wherever there is hardware availability. Therefore, the *Mirroring layer* could potentially collapse inside either IT or OT. With the conceptual division proposed in our architecture, we want to remark that the logic of mirroring is customizable and under the control of the company. This feature allows fine-grained control of the company, in defining policies for configuration of the system and in data protection. The Kafka MirrorMaker (https://docs.confluent.io/3.2.2/multi-dc/mirrormaker.html) is a stand-alone component that copies a subset of topics from OT to IT layer (see Figure 4). In practice, MirrorMaker places a consumer at the source broker (the OT layer’s) and a producer at the destination broker (the IT layer). The company can also choose a specific distribution and replication level of the MirrorMaker component. By mirroring the OT Kafka broker, we make the whole system gain availability, avoiding a single point of failure, and enforce the separation principle between OT and IT layers.

The *IT layer* is populated with the same software modules as the OT layer, with the exception of SCARFs, given that the IT layer does not have a connection with physical machines. As mentioned in previous sections, data in the IT layer are a full copy or a subset of data in the OT, depending on the MirrorMaker configuration policy. Due to the absence of humans in contact with production machines, the time constraints are more relaxed with respect to the OT layer: the topics update frequency is lower and it is possible to deploy specific data analytic logic and advanced software modules. As shown in Figure 4, we added data processing modules using Kafka Processors, components able to aggregate and transform data before sending information to the Kafka Broker. The company can add custom business logic (such as lambda functions) for data manipulation. We have placed two connectors consuming messages from the Kafka Broker: one brings data to Elasticsearch storage, the other forwards information to the Cloud layer.

The main objective of the *Cloud layer* is aggregation and presentation of data coming from production sites. In order to reach this layer, data forwarded by the IT layer will have to use a secure communication channel, as they have to traverse the public Internet. State of art solutions guaranteeing integrity and confidentiality (SSL/TLS) will be used to enforce data security. In this layer, TPS will use software that fits their business needs. Recalling the example scenario described in Section 4, machine vendors can deploy diagnostic, predictive maintenance, and other after-sales software.

7 Experiment

In this section, we report the result of a thorough assessment of the platform with respect to the performance indicators previously discussed in Section 4: timely access, scalability, controlled data access, and resilience.

We made use of virtualization techniques to implement the platform software prototype: specifically, virtual machines (VMs) realized the physical separation between all layers (OT, IT, View). To achieve system scalability and resilience, as well as flexibility when adding new platform features, we adopted the *microservices* programming paradigm. The microservice-based approach allowed us to develop a horizontally scalable and robust system to easily adapt to the dynamics of the input workload and to tolerate potential run-time faults.

For the message latency tests, we address a typical manufacturing scenario with near-real-time constraints for what concerns the availability at the IT layer of information collected by the work machines [33], that is the most common use case in medium and small manufacturing realities. In such contexts, as shown by experiments, the message delay shall never overcome 100ms, which is compatible with the classical timing of near-real-time systems [34], [35].

We consider the case of a typical SME owning two production sites (plants) located in two different cities, say Imola and Bologna, that belong to the same productive district. The SME intends to implement a scalable and robust data gathering in both premises. We will prove that this goal is easily achieved by means of our platform.

We arranged a VPN service to connect the two sites and isolate the machines from each other inside each site. We used the Openstack infrastructure manager to deploy VM instances. We containerized all the components discussed in the Section 6 using the Docker tool and called upon Kubernetes and Rancher to orchestrate and control the services. Docker containers run inside VM instances in their turn. Table 2 depicts all VM instances, microservices running within each VM, and the physical location of VMs.

Each production site implements OT, Mirroring, and Local IT layers. We collapsed OT and Mirroring layers in one VM for the sake of simplicity and because usually, in a real deployment, these entities are on the same local network. In the case of Imola, the Central IT Layer is also deployed. Central IT gathers and aggregates data coming from Imola and Bologna Local IT layers, emulating a real-world deployment where the headquarters also act as a data collection points. As mentioned in Section 4, this layer includes IT software serving all company departments, such as those for managing staff or project cycles.

TABLE 2: Testbed deployment: Locations, VMs, and Kubernetes Pods

Bologna Plant	Imola Plant	Remote Cloud
	Central IT layer VM - Zookeeper - Schema-registry - Kafka Broker - Processors - Kafka Connect - Elasticsearch - Kibana	**View Layer VM** - Elasticsearch - Kibana
Local IT layer VM - Zookeeper - Schema-registry - Kafka Broker - Processors - Kafka Connect	**Local IT layer VM** - Zookeeper - Schema-registry - Kafka Broker - Processors - Kafka Connect	
Local OT layer VM - MODBUS Server - SCARF - Zookeeper - Schema-registry - Kafka Broker - Kafka Connect - Elasticsearch - Kibana - MirrorMaker	**Local OT layer VM** - MODBUS Server - SCARF - Zookeeper - Schema-registry - Kafka Broker - Kafka Connect - Elasticsearch - Kibana - MirrorMaker	

Almost all VMs are equipped with Ubuntu 18.04, 16 GB of RAM, 100 GB of HD, 8 logical cores, and a connection up to 1GBps. VMs emulating the IT layer are provided with the same operating system, HD size, connection rate, but are assigned 8 GB of RAM and 6 logical CPUs.

Finally, we remark that for each test discussed below we reported statistical values obtained from multiple reiterations of the experiment.

7.1 Stress test

The stress test aims at assessing the impact of a sudden increase in message load on the platform performance. In real situations, the number of messages to handle can raise due to an increase of the rate at which machine registries are polled or when new machines are deployed on the shop floor. We will show that, in spite in a substantial increase of the number of messages, the delivery of messages is not affected, thus guaranteeing the message consumer good performances in terms of delivery time, so as to comply with the manufacturing sector requirements. Two separate stress tests are carried at OT and IT layers respectively. Results show that the platform meets the requirements $R1$ and $R2$ set out in Section 4.

7.1.1 OT layer stress test

The target of the first test is the machine/OT stack. We considered scenarios involving both legacy equipment (SCADA-powered machines) and the modern ones (IIoT-powered machines). This specific test addresses just the OT

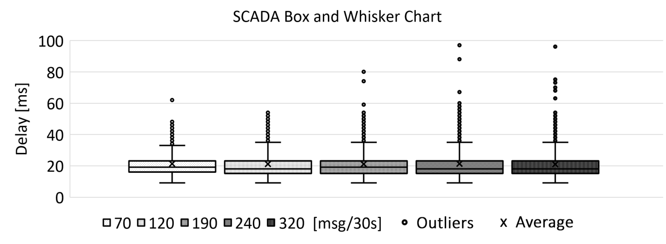


Fig. 5: SCADA delay test

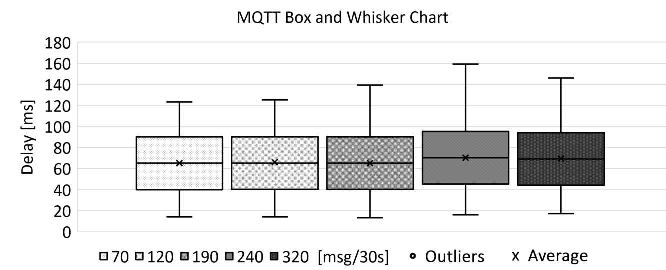


Fig. 6: MQTT delay test

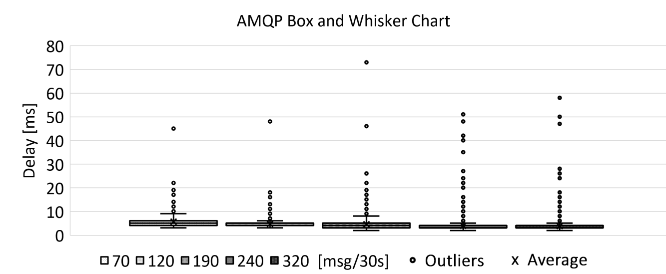


Fig. 7: AMQP delay test

layer of one stack, so we arbitrarily decided to carry it out on the Bologna plant. For tests on SCADA-powered machines, we used ad-hoc MODBUS servers emulating the machine registers as data generators, so as to have the possibility to arbitrarily introduce load spikes and sudden increases. It is worth mentioning that the SCARF component deployed at OT layer supports several communication protocols of the MODBUS family (TCP, UDP, Serial ASCII, Serial RTU, Serial Binary). For the purpose of the experiment, the TCP one is used since machines are equipped with an ethernet interface. This choice will not affect the generality of the experiment outcome because the SCARF component behaves as an adapter or a separation layer between the underlying protocols and the data format our platform deals with. In case of changing the communication protocol used by pymodbus, we just need to change the MODBUS client communication type in the SCARF component. Furthermore, in our tests all the virtualized components are running on machines connected via ethernet TCP/IP protocols, so changing the communication protocol of pymodbus actually does not affect the ”real” underlying communication substrate.

We define *message delay* as the time lapse between the time when a data sample is read from a machine register and the time when the same data is stored to the Kafka broker deployed at the OT layer, i.e., when it becomes available to consumers. We were able to track the delay trend by adding metadata to this sample, a creation timestamp, and a storage timestamp respectively. We then investigated the capability of the platform to handle the message loads produced by different numbers of machines on the shop floor. We assume that every emulated machine is equipped with 7 functional units, each exhibiting exactly one register. Each machine is configured to produces 24 messages every 30 seconds. We observed the performance of the platform handling messages produced by 3, 5, 8, 10, and 13 machines, which corresponds to loads of 70, 120, 190, 240, and 320 msg/30secs respectively. Each experiment assessing the performance of a given load lasted 30 minutes and was repeated 10 times. Figure 5 reports the statistics of each experiment results in the form of Box and whisker plots. By looking at the plots, it appears very clearly that, despite the increase of load, the average message delay is stably set around 20ms. Also, data is very much condensed as evidenced by the short distance between the first and the third quartile, and by the narrow extension of the whiskers. We can conclude that the system is fairly capable of absorbing load fluctuations by providing a constant performance that fits real-time requirements of OT environments.

The same experiment on data delay was conducted on IoT-powered machines equipped with MQTT and AMQP brokers respectively. We remind that, in this case, we used ad-hoc Kafka connectors to extract data from MQTT/AMQP brokers and send it to the Kafka broker, which will eventually publish it. We will focus on the time span from when a data sample is produced by the machine sensor to when it is available to the Kafka broker consumers. We tested the system tolerance with respect to the same increase of message loads employed in the SCADA experiment. As shown in Figure 6, in the case of Mosquitto the message delay is around 70 ms, while for RabbitMQ, depicted in Figure 7, it is about 4ms. In both cases, the increase of load did not impact the performance, thus confirming the good scalability of the proposed solution in realistic industrial settings.

7.1.2 IT layer stress test

The IT-level scalability test aims at assessing the performance of the Central IT Kafka broker when it is loaded with messages coming from multiple Local IT plants. To measure that, we set up a test-bed reproducing the scenario of a company running seven production plants, located in different places geographically distant from each other, that send data to the Central IT layer. The data transfer dynamics vary from plant to plant and are not predictable: to such uncertainty, many indicators contribute the switch off/on of machines determined by the production schedule, the different rates at which machine registries produce data, and the variability of the network bandwidth available during the data transfer. Each plant is emulated via a software message producer with the message rate randomly changed over time throughout the test. This configuration produced an overall message load on the Central IT layer that is variable in time: the objective of the test was to assess the performance of the Central IT broker in response to such variations of the input load.

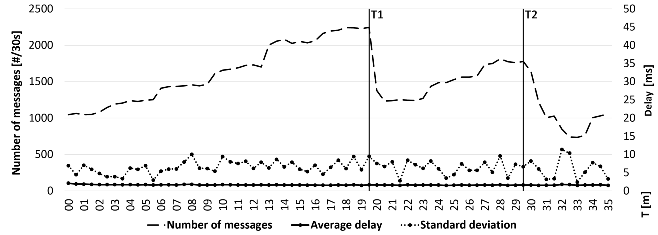


Fig. 8: Central IT Kafka broker stress test

Each Producer produces messages within a time frame of 35 minutes, split into 5 intervals of 7 minutes each, with increasing message throughput. Each interval is characterized by an average message rate plus (or minus) a random delay with a 60% bound of the specific message rate of the interval. The producers wake up at a different time, thus emulating with effectiveness a real scenario in which some production machines are operating, while others are off. Moreover, the highest peak (2240 msg/30s) is reached when all producers are active and generate messages at a rate of 320 msg/30s.

In Figure 8, we report both the overall message load trend and the observed average and standard deviation of the message delay (discrete points curves). Each point represents the average (standard deviation, respectively) delay of messages arrived in a 30s time window. From time 0 to T1 the producers gradually increase the overall system load, sending concurrently up to 2240 msg/30s. At T1, the load suddenly decreases to 1235 msg/30s due to the disconnection of 3 plants. From T1 to T2 the message rate continues to grow, reaching a new local maximum of 1780 msg/30s. At T2, two producers gradually stop the production.

The reader will note that the average delay curve is almost steady (hitting a value of around 1.65ms) independently of the message load fluctuation, while the standard deviation keeps below 10ms. Figure 9 shows the resource consumption of the Docker container that runs the Central IT Kafka broker during the test. We ran the experiment 20 times with different random seeds and changing the start instant of the plants. All experiments showed quite comparable performances in terms of message delay and resource consumption.

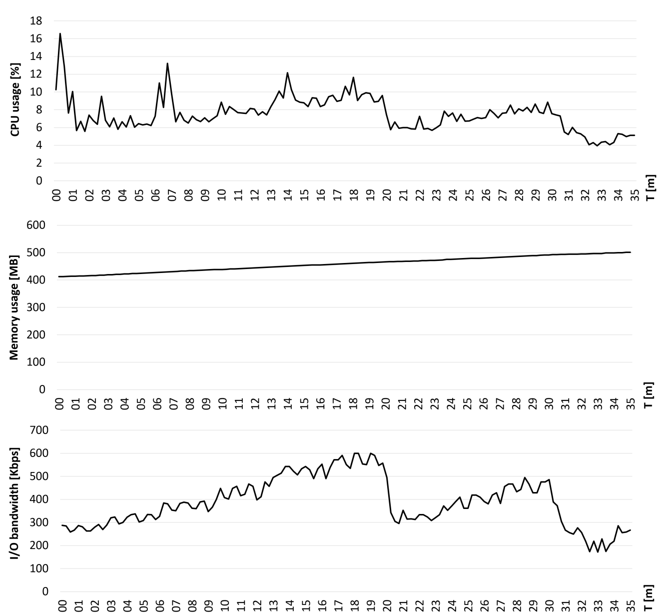


Fig. 9: Central IT Kafka broker resources usage

7.2 Controlled data access test

In Section 4, we thoroughly discussed the advantages of deploying a data gathering and sharing system in a typical industrial manufacturing setting. We then implemented a message broker-based support to gather data from the OT layer and move it to the IT department where it is eventually consumed. Currently, the platform allows data to flow only from OT to IT layers. Should malicious actors gain access to the message brokers, they could steal precious data or to convey tampered information to the management department. To accomplish a full IT/OT integration, in the future we will allow data to flow from the IT downwards to the OT, thus exposing the shop floor to further security and safety issues. In fact, in that case, malicious intruders could exploit the system to inject control data that will eventually cause damage to machines or put the workers' safety at serious risk. In order to face these and future security issues, we have provided the platform with support to prevent malicious or unauthorized access to production data. In the following, we disclose the details of the experiment run to test the implemented access control mechanisms.

First of all, in our testbed, we leverage a VPN to secure communication among VMs (layers) within a plant, as well as inter-plant communication (in our case, communication between Local IT of Bologna and Central IT of Imola). SSL/TLS is used to guarantee data integrity and confidentiality of communication with TPS premises. In order to reinforce security at the OT layer, we exploited the Kafka Access Control List (ACL) feature to manage access (both in reading and writing mode) to the topics of the OT layer Kafka broker. ACLs can be defined for each topic with a very fine-grained policy.

According to Table 2, the Imola Central IT layer consumes data coming from the underlying OT layer and data coming from the Bologna OT layer. We assume that an intruder managed to gain access to the environment and steal the identity of a producer (e.g. *Cooler*) that is allowed to post messages only to the "bologna_capper-1-cooling" topic. When the tampered publisher Cooler tries to push data to the "bologna_capper-1-data-cycle" topic, which it is not granted to write, the producer is notified about the denial of authorization, while the Kafka authenticator log reports what has happened: timestamp of the failed attempt, the host from which the attempt originated, and the involved topic.

What has been shown is a very simple rule, but through the ACL mechanism, the platform can enforce more complex access control rules at any layer (OT, Local IT, Central IT). That allows preventing the execution of malicious or accidental read/writes operations on topics. In fact, this tool guarantees that all stakeholders, both internal and external to the company, are granted access just to information of

TABLE 3: Resilience test: brokers per topic. Brokers denoted with an asterisk hold a Master replica.

Active Brokers	Replica Brokers	Replicas In Synch
TOPIC bologna_capper_data_cycle		
T1 - [B0,B1,B2]	[B0,B1*]	[B0,B1]
T2 - [B0,B1]	[B0,B1*]	[B0,B1]
T3 - [B0,B1,B2]	[B0,B1*]	[B0,B1]
TOPIC bologna_capper_plasticizer_data		
T1 - [B0,B1,B2]	[B1,B2*]	[B1,B2]
T2 - [B0,B1]	[B1*,B2]	[B1]
T3 - [B0,B1,B2]	[B1,B2*]	[B1,B2]
TOPIC bologna_capper_absolute_totalizers		
T1 - [B0,B1,B2]	[B1*,B2]	[B1,B2]
T2 - [B0,B1]	[B1*,B2]	[1]
T3 - [B0,B1,B2]	[B1*,B2]	[B1,B2]

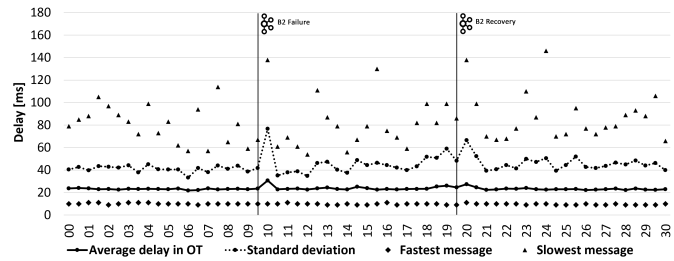


Fig. 10: Resilience test: average and standard deviation of message delay

which they are the intended recipient. Let us conclude by noting that this test demonstrates that the platform meets requirement $R3$ mentioned in Section 4.

7.3 Fault-tolerance test

The objective of our last experiment is testing the platform's capability to react to potential faults. In the implemented data gathering system, we aim to guarantee continuous support to the data ingestion and migration towards the upper layer. Unexpected and sudden interruptions of the mentioned support may cause data blackouts that can severely impact the efficiency and efficacy of processes that need to consume operational data. When a fault occurs, a plan needs to be promptly enforced to recover as quickly as possible and restore the previously provided quality of the service.

Once again, we focus on the OT layer as it represents the data entry point of the platform. Particularly, we intend to preserve the service continuity of the message broker, since faults at this layer may in turn compromise the service continuity of upper layer brokers. We remind that Kafka brokers are implemented as containerized services running inside VMs. Faults can be of many different types (a crash of the container/VM, a hardware failure of the hosting PC). Whatever can go wrong at runtime is a fault the system will have to deal with.

To face faults, we exploit Kafka replication by deploying a redundant number of Kafka brokers in the OT layer. Each topic replication factor is set to 2, meaning that a topic is configured to have 2 replicas (one is the *Master*, the other one is the *Slave*) residing in two of the available brokers respectively. Slave replicas will function as a backup of their respective Masters. Kafka takes care of distributing topic replicas among the brokers, managing faults of brokers, and keeping topics in sync among all replicas.

For the test purpose, we deployed three brokers (*B0*, *B1* and *B2*) and created three topics (*bologna_capper_data_cycle*, *bologna_capper_plasticizer_data* and *bologna_capper_absolute_totalizers* respectively). Then, we configured producers to send messages on the three topics at an overall rate of 192/sec. The experiment consists of getting the whole system up to work at time $T1$, tearing down *B2* at time $T2$ (we simulate the broker fault by killing the broker instance), and getting *B2* back to work at time $T3$. The blackout of B2 lasts for about 5 minutes.

Table 3 shows the dynamics of the system in the course of the experiment. In the "Replica Brokers" column we reported the brokers holding the replica of the considered topic (the one denoted with an asterisk is the broker holding the Master replica). The reader may note that topic *bologna_capper_data_cycle* is not affected by B2 blackout because none of its replicas are held by B2. Topic *bologna_capper_plasticizer_data* has a Master replica in B2 and a Slave replica in B1. At time T2, being the Master replica unreachable due to B2 fault, prosumers are redirected to B1 Slave replica. At time T3, when B2 will be again up and working, the B2 Master replica is synchronized with the Slave, and prosumers are redirected back to it. For what concerns *bologna_capper_absolute_totalizers*, no action is taken at time T2 since B2 is holding a Slave replica. Prosumers will keep using the Master replica held by B1. When B2 recovers, the Slave replica is synchronized with the Master. Of course, with the two replicas configuration, service continuity of a topic is guaranteed as long as at least one replica is available. For highly unstable or overloaded systems, it is advisable to increase the redundancy of the number of brokers and/or topic replicas. In Figure 10 we depicted the average and standard deviation of message delay recorded during the experiment. For each observation, maximum and minimum delay values are also reported. The reader may notice that no message is lost both at time T2 and time T3: small glitches of both the average and the standard deviation curves at the two instants of time prove the robustness of the system, which then is proved to meet the requirement $R4$ set out in Section 4.

We also monitored the trends of CPU usage, memory occupancy, and I/O throughput of the docker containers running the three brokers reported in Figure 11. B2 recovery at T2 causes a glitch in CPU usage and peaks in I/O throughput trends. B2 CPU usage irregular transient (small consecutive peaks) is due to the progressive reactivation of the Docker container's modules. I/O throughput peaks observed at T2, where B2 is the highest, are due to the synchronization of topics among the brokers. As far as B1 and B3 are concerned, except for the I/O throughput, almost no significant resource usage change was observed throughout the experiment. Test results show that thanks to proper management of the underlying message brokering, SIRDAM4.0 is able to absorb sudden and long-lasting faults, guaranteeing the reliability and service continuity of the system with no decrease in the performance level.

8 Conclusion

Within manufacturing factories, IT/OT convergence is considered a key enabler for the transition to I4.0. In this work, we analyzed the benefits and drawbacks of IT/OT convergence within manufacturing factories, with a special focus on SMEs needs to achieve such a goal. In particular, we figure out a scenario where data sharing is extended to multiple stakeholders, both factory internal departments and external suppliers, and highlighted that, from this opportunity, further benefits can be obtained by all involved actors.

Motivated by this scenario, we describe the design and implementation of SIRDAM4.0, a platform that supports large-scale OT data gathering. The platform provides fast, scalable, controlled, and robust access to information. We set up a geographically distributed test-bed that emulates a scenario of SMEs owning production plants in different cities. SMEs deploy machines equipped with both modern (IIOT-based) and legacy (SCADA-based) communication protocols. Our extensive tests showed that the platform can cope with some of the strict constraints imposed by the convergence, i.e., timeliness of data access, secure and selective access to information, and tolerance to unexpected run-time faults. Furthermore, we remark that open-source tools were employed to implement the software prototype of the platform. Other than the obvious advantage of cost containment for adopters, this choice makes the platform easily extensible and reusable.

Encouraged by the achieved result, we will test the platform in a real production environment. Our future research will also focus on scenarios of *smart and fully-connected factory objects* fostered by the RAMI standard. We intend to enhance the current data transfer mechanism by enabling data to traverse the company layers also downward, i.e., from managerial departments to operational ones. The prospected scenario would enable more direct control of machines by the business departments. At the same time, that would expose the shop floor to further security and safety issues, due to direct inputs coming from the IT to the OT.

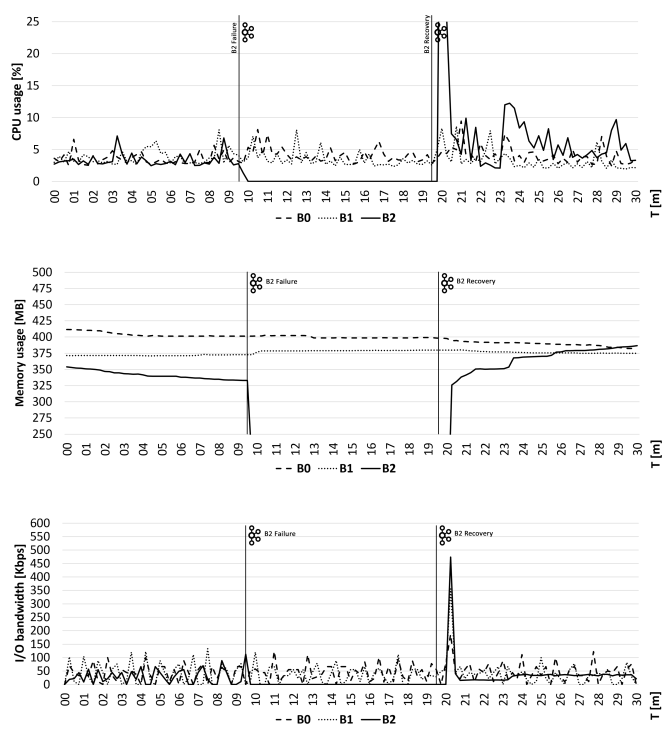


Fig. 11: Resource usage of the Docker containers running the three brokers

References

[1] C. Cimini, R. Pinto, G. Pezzotta, and P. Gaiardelli, "The transition towards industry 4.0: Business opportunities and expected impacts for suppliers and manufacturers," in *Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing*, H. Lödding, R. Riedel, K.-D. Thoben, G. von Cieminski, and D. Kiritsis, Eds. Cham: Springer International Publishing, 2017, pp. 119–126.

[2] A. Corradi, L. Foschini, C. Giannelli, R. Lazzarini, C. Stefanelli, M. Tortonesi, and G. Virgilli, "Smart appliances and rami 4.0: Management and servitization of ice cream machines," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1007–1016, 2019.

[3] U. Dombrowski and S. Fochler, "Servitization as a key driver for digital transformation of manufacturing companies' spare parts service," in *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, 2018, pp. 291–296.

[4] "Reference architecture model industrie 4.0 (rami4.0)," [Accessed: May 2020]. [Online]. Available: https://www.beuth.de/en/technical-rule/din-spec-91345/250940128

[5] "Industrial internet reference architecture," [Accessed: May 2020]. [Online]. Available: https://www.iiconsortium.org/IIRA.htm

[6] "Opc foundation," [Accessed: May 2020]. [Online]. Available: https://opcfoundation.org/

[7] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[8] M. Felser, M. Rentschler, and O. Kleineberg, "Coexistence standardization of operation technology and information technology," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 962–976, 2019.

[9] "When it and operational technology converge," [Accessed: May 2020]. [Online]. Available: https://www.gartner.com/smarterwithgartner/when-it-and-operational-technology-converge/

[10] F. Bosi, A. Corradi, L. Foschini, S. Monti, L. Patera, L. Poli, and M. Solimando, "Cloud-enabled smart data collection in shop floor environments for industry 4.0," in *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2019, pp. 1–8.

[11] G. Radchenko, A. B. A. Alaasam, and A. Tchernykh, "Micro-workflows: Kafka and kepler fusion to support digital twins of industrial processes," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, 2018, pp. 83–88.

[12] A. B. Alaasam, G. Radchenko, and A. Tchernykh, "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.

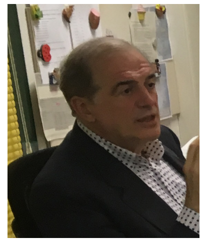
[13] W. M. Mohammed, B. R. Ferrer, U. Iftikhar, J. L. M. Lastra, and J. H. Simarro, "Supporting a cloud platform with streams of factory shop floor data in the context of the intustry 4.0," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, 2018, pp. 786–791.

[14] P. K. Garimella, "It-ot integration challenges in utilities," in *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, 2018, pp. 199–204.

[15] S. Z. Kamal, S. M. A. Mubarak, B. D. Scodova, P. Naik, P. Flichy, and G. Coffin, "IT and OT convergence - opportunities and challenges," in *SPE Intelligent Energy International Conference and Exhibition*. Society of Petroleum Engineers, 2016. [Online]. Available: https://doi.org/10.2118/181087-ms

[16] P. Lipnicki, D. Lewandowski, D. Pareschi, W. Pakos, and E. Ragaini, "Future of iotsp – it and ot integration," in *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2018, pp. 203–207.

[17] J. M. Gutierrez-Guerrero and J. A. Holgado-Terriza, "Automatic configuration of opc ua for industrial internet of things environments," *Electronics*, vol. 8, no. 6, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/6/600
[18] O. Givehchi, K. Landsdorf, P. Simoens, and A. W. Colombo, "Interoperability for industrial cyber-physical systems: An approach for legacy systems," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3370–3378, 2017.
[19] "International society of automation," [Accessed: May 2020]. [Online]. Available: https://www.isa.org/
[20] P. Lara, M. Sánchez, and J. Villalobos, "Ot modeling: The enterprise beyond it," *Business & Information Systems Engineering*, vol. 61, no. 4, pp. 399–411, May 2018. [Online]. Available: https://doi.org/10.1007/s12599-018-0543-3
[21] S. Hagner, "Optimizing transmission asset health with it/ot integration," in *2016 Saudi Arabia Smart Grid (SASG)*, 2016, pp. 1–6.
[22] M. Gutiérrez, A. Ademaj, W. Steiner, R. Dobrin, and S. Punnekkat, "Self-configuration of ieee 802.1 tsn networks," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–8.
[23] M. Zarte, A. Pechmann, J. Wermann, F. Gosewehr, and A. W. Colombo, "Building an industry 4.0-compliant lab environment to demonstrate connectivity between shop floor and it levels of an enterprise," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 6590–6595.
[24] Ministry of Industry and Information Technology, *National Intelligent Manufacturing Standard System Construction Guide*. www.cdti.es, 2015, [Accessed: May 2020]. [Online]. Available: https://www.cdti.es/recursos/doc/Programas/Cooperacion_internacional/Chineka/Documentacion_relacionada/17668_273273201814238.pdf
[25] "Opc unified architecture," [Accessed: May 2020]. [Online]. Available: https://opcfoundation.org/about/opc-technologies/opc-ua/
[26] I. González, A. J. Calderón, J. Figueiredo, and J. M. C. Sousa, "A literature survey on open platform communications (OPC) applied to advanced industrial environments," *Electronics*, vol. 8, no. 5, p. 510, May 2019. [Online]. Available: https://doi.org/10.3390/electronics8050510
[27] "Isa99, industrial automation and control systems security," [Accessed: May 2020]. [Online]. Available: https://www.isa.org/isa99/
[28] P. Bellavista, F. Bosi, A. Corradi, L. Foschini, S. Monti, L. Patera, L. Poli, D. Scotece, and M. Solimando, "Design guidelines for big data gathering in industry 4.0 environments," in *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2019, pp. 1–6.
[29] S. Ghosh and S. Sampalli, "A survey of security in scada networks: Current issues and future challenges," *IEEE Access*, vol. 7, pp. 135 812–135 831, 2019.
[30] J. Reeser, T. Jankowski, and G. M. Kemper, "Maintaining hmi and scada systems through computer virtualization," *IEEE Transactions on Industry Applications*, vol. 51, no. 3, pp. 2558–2564, 2015.
[31] L. I. Minchala, S. Ochoa, E. Velecela, D. F. Astudillo, and J. Gonzalez, "An open source scada system to implement advanced computer integrated manufacturing," *IEEE Latin America Transactions*, vol. 14, no. 12, pp. 4657–4662, 2016.
[32] G. Hesse, C. Matthies, and M. Uflacker, "How fast can we insert? an empirical performance evaluation of apache kafka," in *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, 2020, pp. 641–648.
[33] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," *Procedia CIRP*, vol. 61, pp. 335–340, 2017, the 24th CIRP Conference on Life Cycle Engineering. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827116313129
[34] S. Saxena and S. Gupta, *Practical real-time data processing and analytics: distributed computing and event proc event processing using Apache Spark, Flink, Storm, and Kafka*. Packt Publishing, 2017.
[35] Sep 2019. [Online]. Available: https://blog.cloudera.com/architectural-patterns-for-near-real-time-data-processing-with-apache-hadoop/



Antonio Corradi (Senior Member) graduated from the University of Bologna, Bologna, Italy, and from Cornell University, USA. He is a Full Professor of Distributed Systems at the University of Bologna. His research interests include distributed and parallel middleware, Cloud computing and multi Cloud, microservices and microcontainers, Industry 4.0, IIoT, 5G, IT vs. OT, infrastructures for Industry 4.0, Big Data system support, crowdsourcing and mobile crowdsensing. He is the chair of the Emilia Romagna CLUST-ER for service innovation, very involved in internationalization and technology transfer. Prof. Corradi is member of ACM and IEEE.

Giuseppe Di Modica graduated from the University of Catania, Italy. In 2005 he received the Ph.D. in Computer Science and Telecommunication Engineering from the University of Catania, Italy. He is an assistant professor with the Department of Computer Science Engineering at the University of Bologna, Italy. He has participated in many regional, national and European R&D projects. 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 (Senior Member) 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 board.

Lorenzo Patera received the M.Sc. degree in Computer Science Engineering cum laude from the University of Bologna, Italy. He is currently a Ph.D. student in Computer Science and Engineering at the University of Bologna, Italy. His research interests include Cloud and Fog computing, middleware for Industry 4.0 applications, Internet of Things, IT/OT Convergence, and Cyber-Physical Systems.

Michele Solimando received the master's degree in computer science engineering from the University of Bologna, Italy, in 2016. He is currently a Ph.D. student at the University of Bologna. His research activity focuses on Industry 4.0, IT/OT convergence, Cloud and Fog Computing scenarios, Multi-access Edge Computing, IoT protocols, Crowdsensing, Big Data.