Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

COMPUTER SCIENCE AND ENGINEERING

Ciclo 34

**Settore Concorsuale:** 09/H1 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

**Settore Scientifico Disciplinare:** ING-INF/05 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

# A DISTRIBUTED MIDDLEWARE FOR IT/OT CONVERGENCE IN MODERN INDUSTRIAL ENVIRONMENTS

**Presentata da:** Michele Solimando

**Coordinatore Dottorato**

Davide Sangiorgi

**Supervisore**

Antonio Corradi

**Esame finale anno 2022**

# Abstract

The modern industrial environment is populated by a myriad of intelligent devices that collaborate for the accomplishment of the business processes in place at the production sites. The close collaboration between humans and work machines poses new interesting challenges that industry must overcome in order to implement the new digital policies demanded by the industrial transition.

The last ten years have been crossed by the digitization movement called Industry 4.0. In recent times, and due to recent events that have involved not only the industrial sector but society as a whole, we are witnessing the affirmation of new needs, brought together under the umbrella of the fifth industrial transition. The Industry 5.0 movement is a companion revolution of the previous Industry 4.0, and it is not replacing or upsetting its principles but bringing new ones that will lead industrial development back to a human dimension. Industry 5.0 relies on three characteristics that any industrial sector should have and pursue: human centrality, resilience, and sustainability.

The application of the fifth industrial revolution cannot be completed without moving from the implementation of Industry 4.0-enabled platforms. The common feature found in the development of this kind of platform is the need to integrate the main domains employed in modern enterprises and production realities, namely the Information and Operational layers. The state of the art on the implementation of the Industry 5.0 movement is lacking, moreover many attempts that address the IT/OT convergence challenges seem not to follow some of the basic principles that the integration of these paramount layers requires.

Our thesis focuses on the implementation of a platform addressing all the digitization features foreseen by the fourth industrial revolution, making the IT/OT convergence inside production plants an improvement and not a risk. Furthermore, we added modular features to our platform enabling the Industry 5.0 vision, favoring human centrality in the evolutionary process, without neglecting the robustness of the resources and the continuous optimization of operation in order to increase corporate sustainability.

We used successful IT domain technologies to extend the features of our integration platform. Crowdsensing helps the involvement of humans in business decisions and industrial progress. This thesis provides a use case on mobile crowdsensing technology managing

critical scenarios. We adapted our mobile crowdsensing architecture to handle a pandemic situation, but it can be extended in order to cope with hazardous situations occurring in production sites, where machines and employees collaborate.

Considering the Industry 5.0 objective of improving corporate reliability, our framework has a cloud layer where potentially all corporate information turnout. Among the plethora of pluggable and modular services that can exploit the company knowledge base, in this thesis, we showed two use cases. The first service audits the public cloud providers' resource performance. The second is a simulator to choose and improve the deployment of the network and IT resource in a private corporate data center. The choice of the right service provider that meets the company requirements, as well as the optimum load balancing of enterprise processes aim to improve corporate sustainability, another important goal set by Industry 5.0

Last but not least, we conducted cybersecurity tests on one of the most used protocols on the shop floors employing complex work machines, namely OPC UA. In production plants, information security is closely related to the safety of workers as malfunctions can compromise people's health. Our work encompasses this need and analyzes and tests the behavior of the software used to interconnect the machines to the enterprise ecosystem and pave the way for the black-box analysis of low-level communication protocols, whether open source or proprietary.

The contributions of this thesis is given by the expansion of the main objectives set by the fourth industrial revolution, with the addition of tools that support the implementation of the three pillars of industry 5.0, namely human-centrality, resilience, and sustainability. Our work analyzes and respects the technical specifications required by the Industry 4.0 transition, by enabling IT/OT convergence. In addition, thanks to the enrichment of our platform with crowdsensing tools and cloud computing services, it enables an ecosystem that favors the entire corporate development through the attainment of common intentions for the industry and for society as a whole.

We achieved important and encouraging results in all the domains on which we conducted our experiments. Our IT/OT convergence-enabled platform exhibits the right performance needed to satisfy the strict requirements of production sites. The multi-layer capability of the framework enables the exploitation of data not strictly coming from work machines, allowing a more strict interaction between the company, its employees, and customers.

# Table of contents

# List of figures

# List of tables

# Listings

# Introduction

The planetary events occurred in recent years and the requirements that have emerged pushed the industry to evolve beyond the Industry 4.0 (I4.0) paradigm, even though the digital transformation is still in evolution as a general context. The new movement goes outside the walls of the enterprises and sets its goals and challenges by considering an entire society in evolution. The main concept that the term Industry 5.0 (I5.0) wants to settle is the contribution that most industrial sectors can and must make to society as a whole. Industry 5.0 is a concept born not only from the technology advancements but mainly from the current events that society faces. The devaluation of the world of work, the indiscriminate use of resources that increases global warming, the occurrence of pandemic events on a global scale, are just some factors contributing to the adoption of specific measures in the industrial sector.

On the border between Industry 4.0 and 5.0 movements, our work deal with one of the main requirements of these transitions, namely the convergence of IT and OT layers. We will introduce the main standardization initiatives that explicitly report the need for a convergent industrial environment, and that have inspired our work. Throughout our work, we will often refer to the term *community*, which encompasses different actors, such as research and workgroups, practitioners, and standardization bodies working on the IT/OT convergence topic and more generally on the various aspects relating to technological transitions. The Cyber-Physical Systems (CPS) alongside the Industrial IoT (IIoT) are only two of the supporting technologies in the industrial transition, and they are not the only communities that address the issue of IT/OT convergence. Various organizations, such as International Electrotechnical Commission (IEC) [1] and International Society of Automation (ISA) [2], through the publication of RFCs and new standards, try to address the different challenges of IT/OT convergence, like cybersecurity and time-sensitive communication.

Established the needs of the fourth industrial revolution, which we have partially covered by enabling I4.0 innovations, we will analyze the modern Industry 5.0 European movement. This companion revolution collects the legacy of what we learned from the ongoing Industry

[1] https://www.iec.ch/homepage
[2] https://www.isa.org/

4.0 to add new objectives that put the industry at the service of humankind, society, and the environment, for synergistic progress in all sectors.

The main motivation that pushed us in the development of our platform was the convergence of the Information Technology (IT) and Operational Technology (OT) layers within companies that are facing the Industry 4.0 transition. The main feature of modern business companies is the sharing of data between the technical and managerial levels. For industrial and manufacturing sectors, it means to be ready for the merger of two historically separate levels, i.e., the IT and the OT layers. Companies facing the I4.0 transition must rely on platforms that meet the performance needs of production sites and that have the same reliability and security features of the IT domain. The continuous monitor of the work machines, their consequent servitization, the possibility of activating remote commands from the managerial departments down to the operational levels, and the possibility of having comparisons and aggregations of the data from all the distributed assets are just some of the functions that the industry of the future requires to enable the vision predicted by the I5.0 transition.

In the Emilia Romagna region, mainly in the Packaging Valley, the vast majority of companies are Small and Medium Enterprises (SME) [1] for which the digital transition can be an insurmountable obstacle. In the long run, these realities may face a big gap with the innovations introduced by competitors. In this sense, the innovation in digitalization becomes a mandatory process, to fulfill by following guidelines and principles pursued by Industry 4.0 and Industry 5.0 transitions. Especially for SMEs operating in the manufacturing field, to think of being able to replace all existing assets to meet the needs dictated by the implementation of the I4.0 and I5.0 services is not a viable way. At the same time, many large companies do not trust this enormous technological evolution, due to a lack of confidence in a fast ROI or due to the uncertainty in sharing and amalgamating their data with other departments or even third-party players.

Our work targets precisely the SME and all the companies that want to embrace the IT/OT convergence in order to prepare the field for the next industrial revolutions. These realities, despite having a strong innovative drive, nevertheless fail to implement the architecture necessary to gather and extract valuable knowledge from the data produced on their shop floors. Implementing an IT/OT convergent architecture, we want to prove that the industrial transition is a profitable solution for all the enterprises facing the new challenges of the continuously evolving market.

Furthermore, in order to fulfill one of the main requirements of the Industry 5.0 movement, human centrality, we conducted experiments on Mobile Crowdsensing applied to critical scenarios, to evaluate the feasibility of this strategy for improving the quality of life of workers and facilitating their integration with new technologies. Crowdsensing campaigns could include a broad range of participants, making the society a central player in industrial advancement. We believe that crowdsensing is a key tool to overcome social differences and bring business progress back to a human dimension. Its employment could enhance technology acceptance and trustiness in new technologies, helping the training of people and workers in using them.

During our studies, we investigated other fields that are strongly integrated with industrial environments, such as cloud, and edge computing. We think the cloud play a central role in the transmission, storage, analysis, and sharing of enterprise data and information, features that Industry 5.0 needs for enabling its forecasting. At the same time, it is important to ensure its correct use, to minimize costs in case of external outsourcing of resources and emissions in case of private cloud environments. For this reason, we focused part of our research on cloud computing, developing services that can be associated with our integrated platform. These features include the analysis of the performance of the major cloud providers on the market and the timely simulation of private cloud deployments based on customer needs. These two services in an enterprise integration architecture are only two use cases to prove that the cloud layer enables pluggable functionalities which could improve the reliability and ecological impact of the entire company.

In our thesis, we could not neglect one of the most stringent requirements of both industrial revolutions, namely the cybersecurity of communication. Nowadays, the OT environment is no longer detached from external networking. We have built a tool to analyze, without any assumptions but the protocol knowledge, the implementation correctness of the OPC UA protocol, widely used in workshops to communicate with machines and collect data. The need for securing IT-OT communication guided our research in providing mechanisms for the offline analysis of the behavior of communication protocols under not standard stimuli. Although this last part of the work is still under development, due to the need to perform long test sessions, the results obtained so far open up new scenarios to protect the communication between the industrial asset.

# Thesis Contribution

In this section we want to outline the main contributions of the thesis. The general aim of this thesis is to provide a platform enabling the ongoing industrial transitions, and investigate the challenges and opportunities offered by two of the most commonly used technologies in the IT field, namely the cloud computing and the crowdsensing, in order to enable the I5.0 features.

The proposed framework aims to enable IT/OT convergence in SMEs, with reduced costs and performing results. Our work is based on the ability to integrate all the data sources encompassed in the workshops and production sites. This thesis has a twofold objective, covering the Industry 4.0 requirements and putting the focus on the Industry 5.0 directions. First, our framework wants to enable the IT/OT convergence, allowing the enrichment of the companies' renewed knowledge base and the cooperation of several source/sink connectors. Secondly, our broad ecosystem takes into account not only the changes in technical and managerial departments inside the company but also the impact produced by the innovation process on society and the environment, as Industry 5.0 movement suggests.

Although our work represents small steps in different directions, it can be the basis for the construction of a modern ecosystem that takes into account not only the technological innovations of Industry 4.0 but also the needs arisen in recent years, covered by Industry 5.0 transition. How we will deeply see in the next sections, we chose to investigate the Mobile Crowdsensing, the Cloud Computing, and the cybersecurity enhancement to enable our vision, in which the transitions to Industry 4.0 and Industry 5.0 are closely related and addressed by the same framework. In particular, the contributions of this thesis are as follows

- The thesis presents a novel architectural model to enable the modern industrial transition in SME. Our multi-layer middleware tackles the Industry 4.0 requirements and paves the way for the incoming **Industry 5.0 transition**, straightforward integrating of business stakeholders and citizens in the data sharing loop. Our attempt at integration is mainly due to three macro areas through which we want to meet the three main concepts on which the Industry 5.0 is based. To promote *human centrality* we exploited crowdsensing technology, to increase *reliability* we relied on cloud computing and a robust data ingestion and gathering. Finally, to provide for corporate *sustainability*, the combination of crowdsensing and cloud services can be the keystone in optimizing product design, goods path from production to distribution, and services for society. The kind of sustainability we can support from the IT point of view takes the form of the optimization of software processes (which can be implemented through network simulations) and the careful use of remote resources (which can be estimated through auditing techniques). At the OT level, regardless of the mechanics, renewable energy, and innovative materials, beyond the scope of our thesis, as a future development of our platform, we hypothesized an emissions analysis obtainable from the data collected by the work machines and from the feedback of workers and citizens. These techniques can lead to a smaller footprint of company production processes.

- We analyzed the implications of the **IT/OT convergence** in industrial environments, so we built a platform for data gathering and ingestion from work machines, namely the

SIRDAM framework. Despite the IT domain has reached a consistent maturity level, its integration with Operation Technology (OT) is still a mostly unresolved challenge. We propose the design of a platform that supports *reliable data gathering and sharing* among OT and IT layers of an industrial manufacturing company. Our integration middleware employs modern and performant IT technologies and solves the main issues arising from the integration of IT and OT layers. It provides a comprehensive view of the work machines' operation, even in the case of distributed production sites. Furthermore, thanks to the data set separation, our platform includes all the stakeholders in the industrial progress, achieved through differentiated accesses to the information and aggregation layers on top of the operational ones. Our solution meets both functional and non-functional requirements of a typical data gathering/sharing process in a near-real-time scenario, by granting reliability at a very low overhead cost.

- We investigated the possibilities that **crowdsensing** can offer to promote *human centrality* in industrial environments, a target feature for the Industry 5.0 movement. In this scenario, we tested a platform for collecting contributions from users used to identify very crowded urban areas and suggest safe routes. This use case, considered in reaction to the recent health emergency, is extendible to all critical scenarios, such as industrial ones. The mobile crowdsensing applied to the employees in the production plants could achieve the human centrality foreseen by Industry 5.0. Furthermore, crowdsensing can improve corporate sustainability thanks to the feedback provided by contributions from distribution and supply chains. Experimentation regarding these new functions is still in an evolutionary phase with respect to the data acquisition platform. We created a common data set where the data gathered from the crowdsensing campaigns and from the work machines flow. Possible inferential intersections and processing algorithms are future developments of our platform aimed at including the pillars of the fifth industrial revolution among functional requirements.

- In the way of improving corporate *reliability*, we experienced how **Cloud Computing** services can contribute to sustainability in modern industries. We have investigated two different areas by creating and testing dedicated platforms. On the one hand, we have provided companies with an *auditing tool* that can help in choosing the best provider according to their needs, namely Audit4Cloud platform. The second cloud solution concerns the development of a *network simulator*, namely DCNs-2, that includes all the entities engaged in modern company data centers, including virtual ones. In this way, the network architects can optimize the company deployment in order to avoid under-utilized resources and to contain costs and emissions.

- In conclusion, we worked on the **cybersecurity** topic, even if it is not the central core of this thesis. The state of the art about industrial systems and the international standardization initiatives address the issue of security in the exchange of Machine-to-Machine and Business-to-Machine information. In the business and high layers of our framework, we are using *cryptographic* and *access control* mechanisms provided by modern IT tools. At lower layers, we contributed to securing the machine communication through the analysis of network protocols. We developed a fuzzer to check the implementation correctness of the OPC UA protocol. We used a *fuzz testing* technique that considers the target system as a black box, with no knowledge of its internal details except for the protocol specification. We adopted this approach to overcome the problem of the many private industrial protocol implementations and the lack of industrial communication datasets. This solution increases the investors' and users' trustiness in new protocols employed in production sites.

From the point of view of functional requirements and performance indicators, we achieved our results thanks to the collaboration with local enterprises, from which we obtained valuable Key Performance Indicators (KPIs) that helped us to define guidelines [2] for the development of a framework that would address the IT/OT convergence topic. The implementation of the transition to industry 4.0 led us to investigate different application areas in parallel, especially in terms of infrastructure, such as cloud and edge computing. The multiplicity of services executed on off-premise resources allows companies and researchers to have greater freedom of action with respect to business processes strictly performed on private Commercial Off-The-Shelf (COTS) resources. On the side, in the context of international collaboration projects [3] and educational activities, we were able to propose innovative solutions in the field of cwordsensing, a topic that proved to be of great interest as regards the digital and societal transformations described by Industry 5.0.

We obtained encouraging results from the test of each component of the platform. First, we have a very fast time in information gathering and storage and in their presentation to the departments out of operational one. This is the core of our comprehensive platform and reflects the idea that the new IT/OT convergence era is enabled mainly by the correct information management produced on the shop floors by working machines. We gained good performance and functional results also in the Mobile Crowdsensing development, thinking of a project aimed to warn people of dangerous situations and drive them to a safe zone. Finally, our experiments on Cloud Computing proved the possibility of improving business efficiency using cloud services, by choosing the most suitable solutions provided by public cloud providers, or by correctly setting up a private data center through network simulations.

The novelty of our framework resides in the high customizability capable of covering many industrial use cases, most notable it is suitable for the SMEs that usually exhibit difficulty in technological advancements, due to lack of resources and costs to be incurred.

Furthermore, there are still few attempts to integrate the principles proposed by I5.0. Our framework not only addresses the I4.0 digitization, providing the tools to tackle the IT/OT convergence but also spans many complementary topics touching the pillars of the I5.0 movement, such as mobile crowdsensing. In our opinion, this integration is essential to link the progress of society to that of technology.

The servitization of the corporate ecosystem used to manage the company's resources could provide additional value to the systems usually employed for industrial control. Thanks to our solution, we provided modularity in pluggable services that a company can add to its deployment. The cloud computing cooperation, the cybersecurity features, and the mobile crowdsensing expansion make our framework a milestone in the field of corporate control. It breaks down the barriers of the shop floors to achieve the company departments and the whole society, through the possibility of providing inclusive services that entice final customers, distributors, and all interested stakeholders to contribute to the process of corporate development and progress.

Many challenges are still open about the implementation of I4.0 and I5.0 environments. Taking into account the necessary IT/OT convergence, it is necessary to smooth out the different perspectives of the various departments that collaborate in the company's technological progress, starting from the managerial ones to the technical ones. This integration must be pursued not only from the point of view of personnel but also from that of technologies. There must be greater synergy between the evolution of the IT domain, usually very fast and dynamic, and that of the technologies used at the OT level, which usually follow a less frequent upgrading philosophy.

Further challenges, but on different meta-levels, come from the implementation of the changes suggested by the I5.0 movement. The first fundamental coordination for enabling the I5.0 vision must take place within society, and it provides for the acceptance by people of new technologies, which coincides with training people on how to use them, through upskilling and reskilling employees and citizens. The socio-centricity concept implied by I5.0 movement should complement and integrate the needs of individuals and employees with the needs of the entire workforce and the society as a whole. Furthermore, the technological process must take into account the planetary boundaries and, keeping the ecological perspective in mind, it should comprise CO2 neutrality and all the measures aimed at limiting waste and pollution. in conclusion, this big transition is a challenge to be addressed as humanity as a whole, by arranging interdisciplinary approaches that bring together engineers, and experts in life sciences and environmental changes. An agile government is not exempt from the evolutionary process, the ruling and political class should be ready to quickly absorb technological innovations, participatory towards citizens, and adaptable to the different needs not to leave anyone behind.

## Thesis Outline

This section introduces the remainder of the thesis, organized as follow:

- in Chapter 1 we provide the background about the modern industrial and societal revolutions, with a focus on IT/OT convergence, its enabling factors, and the main technologies,
- Chapter 2 analyzes the state-of-the-art on modern manufacturing, focusing on IT/OT convergent related concepts and platforms, and ending with the recent works tackling the Industry 5.0 and Society 5.0 topics,
- in Chapter 3 we introduce our framework addressing the current concerns about industrial revolutions and IT/OT convergence in production environments. We show the main additional features of our platform, relying on crowdsensing and cloud computing to enable some of the Industry 5.0 concepts,
- Chapter 4 offers an in-depth view of the development of a multi-layer industrial platform, focused on the IT/OT convergence topic. After the design of the architecture, we show the implementation insights and the results coming from the extensive testing,
- in Chapter 5, we introduce the crowdsensing platform used to prove the effectiveness of edge computing and blockchain to manage critical situations. Our extension is capable of alerting people of too crowded areas and suggesting safer routes,
- Chapter 6 shows the two use cases based on cloud computing that could improve the company's reliability. The first part of the chapter will deal with the network simulations and will present a novel data center network simulator. The second use case proposes the auditing of public cloud providers through a platform giving the latency and bandwidth performance of remote virtual resources,
- Chapter 7 focuses on the cybersecurity extension of the platform, proposing the industrial control systems security through the guarantee of the communication protocols implementations correctness at lower layers. The Chapter provides an introduction to fuzz testing and our tool to fuzz any OPC UA protocol server implementation,

- in the end, the *Conclusion* Chapter outlines the thesis contributions and the future directions our work paves the way to. We summarize the novelty of our middleware and we plan the next steps in order to give our platform a central role in the next future industrial transition.

# Chapter 1

# The Industrial Sector in Evolving Society

In this section, we want to show a panoramic view of the evolution of the industrial scenario in relation to the benefits that the whole of society could derive from it. We will make an overview of the ongoing Industry 4.0 transformation and of its central challenge, the convergence of the IT and OT layers inside production plants. The key concept of integration between these two layers is a starting point to introduce the enabling technologies and the main international standardization attempts driving the Industry 4.0 movement and focused on IT/OT convergence. In conclusion, we relate industrial progress with societal one by presenting two paradigms that describe the current evolutionary scenario, namely the Industry 5.0, companion of Industry 4.0, and the Society 5.0 movement, focused on the birth of super-smart communities.

## 1.1 The convergence in the Industry 4.0 ecosystem

This section discusses the *IT/OT convergence* process that is taking place in the industrial sector by companies that want to face the Industry 4.0 transition. Firstly, we will provide an insight into OT and IT layers in companies having production facilities. Besides the definitions, we will introduce the meaning of IT/OT convergence, the opportunities brought to the industry, and the challenges it faces. Then, we will introduce the main research and standardization communities driving the IT/OT convergence movement.

### 1.1.1 Information and Operational Technology layers integration

One of the most authoritative definition of *Operational Technologies* (**OT**) is provided by Gartner [1]: *OT is hardware and software that detects or causes a change, through the direct monitoring and/or control of industrial equipment, assets, processes and events*. OT mainly concerns with mechanical assets within production facilities involving the lower layers of the ISA95 automation pyramid [4]. In OT layer, closed-loop control of working machines is implemented by means of vertical HW/SW systems that leverage sensors/actuators deployed in the proximity of machines. Both sensed data and control signals travel on protected and reliable networks that guarantee high throughput and very low communication latency as requested by the computation-intensive tasks that are in charge of controlling the production process. Furthermore, as human workers operate in close contact with the production machines, machine faults that might compromise employees safety must be timely predicted and corrected effectively. The task-specific nature of OT systems and their employment in mission-critical domains have historically overshadowed the innovation and interoperability in favor of reliability, robustness and human safety.

Historically, at OT level ad hoc systems have been used for the purpose of the workflow control. These include old generations of Supervisory Control And Data Acquisition (SCADA) and Distributed Control System (DCS) systems, isolated from external access for the above-mentioned safety and security reasons. Because of their poor flexibility and to their not general-purpose nature, nowadays, ICS are evolving towards more flexible and open architectures (e.g., Industrial Control Systems (ICS) and DCS), that promote the convergence between the physical and the digital world. Modern SCADA architectures used in industrial environments employ several components entrusted with distinct responsibilities. Remote Terminal Unit (RTU) interfaces and Programmable Logic Controller (PLC) devices monitor the production environment and convert electronic signals into valuable information. The supervisory LAN, VLAN, and firewall control data flow and convey data to the Man-Machine Interface/Human Machine Interface (MMI/HMI), that are in turn responsible for presenting data to the operators in a human-comprehensible way (usually in the form of graphs, schematics, trending diagnostics, and animations).

On top of OT, the **IT** layer embraces applications, frameworks, and telecommunication assets that support the management of enterprise resources/data and the business activities. The IT layer deals with gathering, storing, and analyzing the data produced by work machines at the shop floor. In order to assure a fast and reliable data analysis and processing, the IT world encompasses well-known and standard protocols, and is characterized by a

[1] https : / / www . gartner . com / smarterwithgartner / when-it-and-operational-technology-converge/

high dynamism and reactivity in quickly adapting to new innovations in the technological field. The faster evolution of the technologies employed at the IT layer, with respect to those used at the OT level, is directly connected with the evolution of the protocols and software used in non-industrial sectors. In the industrial IT domain, the network end-point managing information-intensive tasks usually is a high-level tool running on computing devices controlled by human operators. Historically, in IT layer service requirements have been more relaxed than in OT. Data processing occurs offline and is usually accomplished by means of software tools like Manufacturing Execution Systems (MES)/Manufacturing Operations Management (MOM) and Enterprise Resource Planning (ERPs). MES systems enable the monitoring of raw materials and production processes, improving plant performance and containing management costs. ERPs aim to optimize the business performance of the company by providing support to the plant scheduling, the supply chain management, the inventory maintenance, and the customer services provisioning.

Nowadays, MOM and ERP tends to be more interlaced [5] and encompass new analysis tools equipped with AI capabilities in order to face the lack of flexibility and boost the adaption to new business needs and models. Machine learning and modern deep learning techniques, sometimes leveraging semantic and digital twins representation of plant resources, directly exploit information coming from MES tools to control business processes. The faster evolution of technologies employed at the IT layer, with respect to those used at the OT level, is directly connected to the evolution of IT protocols/software adopted also in non-industrial sectors. The unbalanced evolutionary path of information and operational techs is favoring a progressive penetration of IT (historically more dynamic) into the OT world.

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 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.

### 1.1.2 IT/OT convergence 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. Placing a layer of convergence between OT and IT will enable machine data sharing and processing. So far, typically machine data have been confined in information silos, so no chance was given that machines could take advantage of each other's produced data. We make a short, non-exhaustive list of advantages of sharing data across the factory.

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 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, 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. Besides the improved automation, visibility, control, and responsiveness of business software, another main advantage of convergence is the enabling of predictive maintenance, a key concept in the transition to I4.0. This approach improves the technique of preventive maintenance, in which periodically interventions are scheduled in fixed times, even if at that moment the machine may be working well. The latter is a rather static approach, in which we could have unnecessary interventions on perfect working machines, unlike predictive maintenance that analyzes the current condition of the machine, and thanks to the convergence, the surrounding context and the operations of other machines, to predict future problems, malfunctions, wear or simple checks by specialized operators. Predictive maintenance uses the real factory floor intelligence, instead of estimates and best guesses. The OT layer, responsible for collecting data from machines, PLCs, and sensors, can benefit from the cooperation with the IT layer, which provides data aggregation and data analytic tools. This attitude of communication inside production plants avoids or mitigates the unplanned downtimes of the working machines, which we have already said is directly related to the operating expense (OPEX) incurred by the company, besides the extension of the machine's useful life.

Manufacturing machine vendors, thanks to the convergence of IT and OT data, not only increase the profit of the company but also improve their relations with customers.

The embracing of analytic data-driven approach, conventional in IT domains such as big data and machine learning, combined with the analysis of information coming from the working assets help to better understand customer behavior and preferences, reinforcing suppliers, partners, and customers relationship. The convergence of IT and OT layers enables the deep knowledge of the work machines fleet, and in a more complex scenarios, of all fleets of a vendor, and of all their operations. The implementation of convergence can be particularly convenient in cases of vendors who have a large pool of customers to whom they sell the work machines. For example, consider a company that sells the same machine model to several customers having assembly lines in different places in the world. With the convergence of data, the vendor can have the overall sight of the customers' shop floors, and the managerial departments can benefit from the exhaustive view of the data coming from all their machines deployed in different places, and consequently, they can carry out a comparison in terms of current operating parameters with respect to the context in which the machine is. Furthermore, vendors and customers can build their own historic trend on which to frame a variance model for estimating how healthy a production line is, knowing its own application domain.

Nonetheless, tearing down the barrier that has historically kept OT and IT physically apart has strong implications which may not be neglected.

Undoubtedly, *security* is the most relevant issue manufacturing companies need to face. Being OT disconnected from the network, attacks in this area were limited to physical or near proximity. The convergence to IT increases the Industrial Automation & Control Systems (IACS) attack surface. As such, there is a strong need to enable monitoring and secure data flow also in machine operating contexts because potential flaws in IACS systems can lead to severe consequences both for the steal of sensitive company data (through sniffing techniques) and for the concrete hazard to humans working very close to the machines. Many safety aspects have been addressed by the ISA/IEC 62443 standard, developed by the ISA99 committee [2], which aims at developing and sharing standards and best practices for designing, implementing, and managing manufacturing and control systems in a secure way.

*Integration of legacy* systems and protocols employed in the shop floors is a very sought feature in IT/OT convergence and in I4.0 transition in general [2] [6] [7]. Most of the OT asset was not designed to be connected to any network, so it is not able to interact with modern communication protocols largely used in IT environments. For that reason, it is not prone to remote monitoring and control. The integration layer will have to take into account

[2]https : / / www . isa . org / standards-and-publications / isa-standards / isa-standards-committees/isa99

the issue of legacy assets for all SMEs that do not have economic resources to upgrade their fleet.

A further drawback of the convergence is the need to *retrain employees* of the two departments [8] [9] [10]. Convergence inevitably leads to adopting practices and technologies unsuited for those environments. It is therefore compulsory that the different departments build a shared knowledge base and reach a common understanding of the convergent system, which is no longer confined in separate areas of the company. Reskilling and upskilling, as we will see in Section 1.2.1, is an industrial feature very sought by Industry 5.0 practitioners.

### 1.1.3 International standardization initiatives and models

Below we will report are the main standardization initiatives that have identified the IT/OT convergence as a crucial aspect in industrial development. These standards and models demonstrate the preponderance of IT/OT integration the realities that want to address the Industry 4.0 philosophy. Having a modern vision that includes the human component in the industrial resource loop, these standards could pave the way for the upcoming Industry 5.0 transition. Study and deepen the main standardization intitiatives at global level helped us in the formal definition of the requirements of a modern industrial framework. The respect of the specifications dictated by Industry 4.0 will make our platform ready to envelope the changes foreseen by the Industry 5.0 movement.

#### Reference Architectural Model Industrie 4.0 (EU)

The **Reference Architectural Model Industrie 4.0** (**RAMI 4.0**) [3] is a three-dimensional map proposing a structured approach to the development of an Industry 4.0 platform. It represents a service-oriented architecture (SOA) covering all the concepts that are common to most of the stakeholders involved in the I4.0 transition. It provides a common knowledge base of the needs and issues raised by the new industrial revolution. Within the RAMI 4.0 specification, the corporate resource to be monitored and controlled, i.e., the *asset*, takes on a central role.

As shown in Figure 1.1, where the overall architecture is depicted, complex interrelations between the controlled resources are broken down into smaller sections, while keeping the asset's lifecycle status a key and central concept. In this way, the interconnection among components became an easier problem thanks to the identification of its relevant aspects. The three dimensions of the RAMI 4.0 architecture, representing different conceptual levels of the complex I4.0 scenarios, are the following:

[3] https://www.beuth.de/en/technical-rule/din-spec-91345/250940128

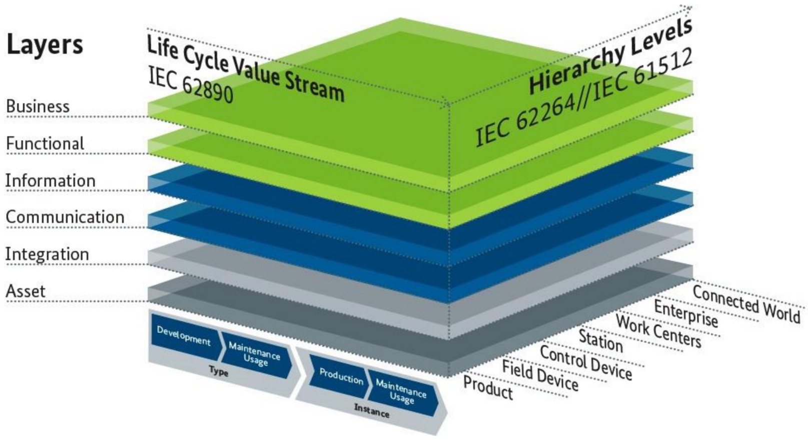



Fig. 1.1 RAMI 4.0 overall architecture

- The **Layers** vertical dimension provides the classic layered approach to the design of the IT architecture that facilitates the development of novel solutions by splitting complex technological issues in smaller and simpler problems. The Layers dimension includes, at the bottom level, the Asset layer (the physical machines), and at upper levels the Integration, Communication and Information levels giving data to the Functional level that offers high-level capabilities to monitor/control underlying physical objects. The top level deals with all processes related to the organization, and it is called Business layer,

- The **Hierarchy levels** horizontal dimension (based on IEC 62264/IEC 61512) provides functional description of components acting in the factory, such as field and control devices, stations, work centers, and the enterprise as a whole. Last element in this axis is the Connected World that underlines the importance of extending the factory environment to the external world. This is the axis that forecasts the use or the presence of old communication protocols in the firm as it emerges from the name given to this axis that includes the IEC 62264 protocol, based on the international standard ANSI/ISA-95. The ISA-95 model is a 5-layers standard that forecasts SCADA systems at layer two that encompasses activities of monitoring and controlling of physical processes,

- The **Life cycle and value stream** key axis (based on IEC 62890) focuses on the whole production cycle of a smart product, and it describes the current lifetime point and location of an object. The model conceptually identifies two phases in I4.0 products lifecycle: the *Type* phase identifies the development of a new product, while the *Instance* phase focuses on the instances of the product itself encompassing servitization, monitoring and e-maintenance processes. The feedback coming back from the Instance phase can impact the Type phase suggesting changes in the project of the product to avoid undesired behaviors detected during the operation. This layer is also a key connection point between enterprises, in the sense that a product in Instance phase in a company could be an input for the Type phase of a product in another company. This aspect helps the enterprises to identify interconnections and dependencies of a product with respect to other companies.

RAMI 4.0 describes the logical organization of assets and combinations, but we want to stress the point that a generic implementation may differ from the rigid specification division, in terms of which layer actually implements the requested service. For example, an MES service logically described in the tier *Work centers*, could be actually implemented in the hierarchical level *Station*.

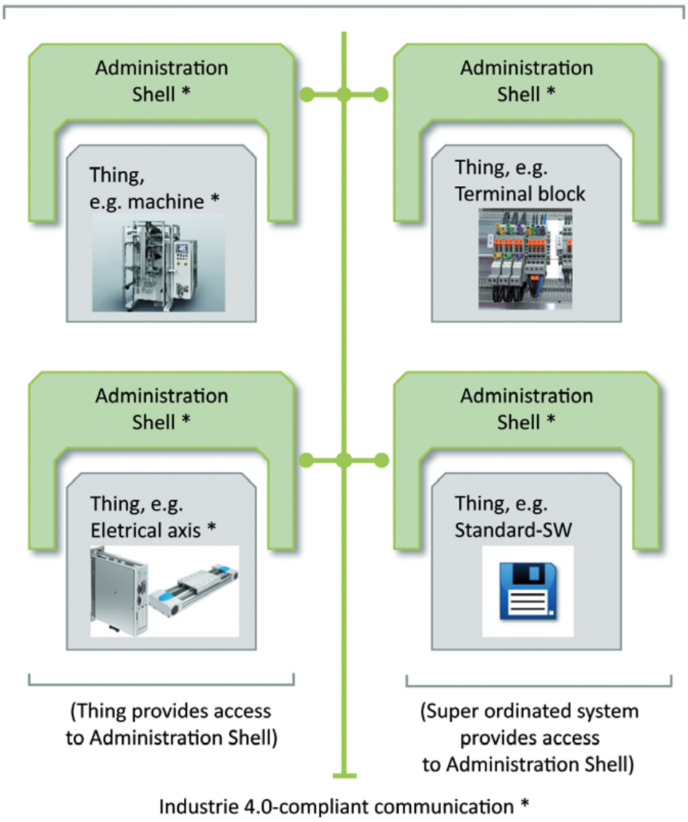



Fig. 1.2 Examples for Industrie 4.0 components

The RAMI 4.0 specification remarks the centrality of the asset in the I4.0 world defining the concept of the **I4.0 component** applicable to all the company's resources: a production system, an individual machine/unit, or a module within a machine. A I4.0 component is an object at least capable of (passive) communication and consists of the asset itself and an *administration shell*. Passive communication is the minimum level of machine interconnection with the corporate network and represents an asset with an interface from which we can read information, such as an RFID or a barcode. Due to the participation of assets to a network with particular pipelines and coordinated tasks, an I4.0 component has the capability to be queried out about its own state at any moment.

The I4.0 components can be described through a common semantic syntax and can be structured in any way their connection allows them to connect with each other. Within an I4.0 system, components have the nesting capability, so a component could be formed by a composition of I4.0 subcomponents. The specification also refers to *Encapsulability* feature of the components which states that the core functionality of a component must be assured also if the external network is disrupted. Figure 1.2 depicts an Industrie 4.0-compliant communication schema between distinct hardware and software components inside a production plant.

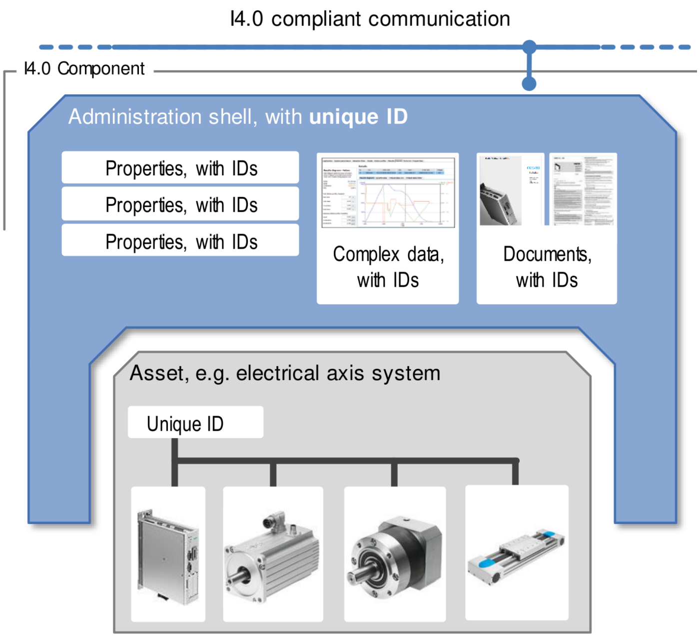



Fig. 1.3 RAMI 4.0 administration shell

The administration shell is what converts an asset into an I4.0 component and it must include all the information to represent the asset, such as a unique ID, and its technical functionality. How you can see from Figure 1.3, both the asset and the shell have their own unique ID, so you can identify them unambiguously. This component is responsible for turning lifecycle data in viewable information for users and administrators. The shell must respect some fundamental security principles, such as confidentiality, integrity, and availability (CIA) of data. If the company uses resources outside its complete control, such as cloud resources, the privacy of sensitive information must be assured. In the case of embedded systems, the administration shell could be part of the asset itself, otherwise, it can be distributed in the IT enterprise system.

The structure of the administration shell has two logical parts: the *Digital Factory (DF) header* and the *DF body*, the first containing the manifest with the asset identifiers and related administration shell, while the second containing a manifest and a component manager, whose functions are available externally via API. The manifest inside the DF body contains partial models written by domain experts with knowledge of data, properties, and functions of the specific domain. The administration shell manifest stores information in the form of readable properties, pointing out whether they belong to the asset in the instance phase or if their value must be identical to that of the asset in the type phase. Properties can reference other properties and can be enumerable and hierarchically organized. An asset can have more than one administration shell for different purposes and an administration shell can represent more than one asset. In the case of multiple shells for a single asset, they must refer to each other, with an administration shell acting as a copy of (a portion of) another shell.

### Industrial Internet Reference Architecture (USA)

The **Industrial Internet Reference Architecture** (**IIRA**) [4] provides a direction in developing smart IIoT architectures, following the developers at every level with the objective to achieve and to cope with the current IT/OT convergence in the industrial production environments. Given the great variety of tools and standards inside industrial production environments, the specification deliberately presents a high-level description of the entities involved, proposing common architecture patterns fitting all industrial sectors. The specification strongly stresses the concept of convergence of the IT and OT layers, underlining that it is a common and required practice for enabling the transition to the new industrial automation, and therefore whose concepts must be covered by the specification itself. The complexity of such industrial ecosystems in which many stakeholders are interested in the complete products' life cycle drove the designers of the standard to distribute the concerns of the different stakeholders into specific categories/layers. Based on ISO/IEC/IEEE 42010:2011 (Systems and Software Engineering - Architecture Description), the standard defines the Industrial Internet Architecture Framework (IIAF), which describes principles and guidelines for architectures based on IIoT. The IIRA document is the result of the application of the IIAF to the IIoT systems, and it identifies and highlights the most important topics and issues coming from the employment of IIoT architectures in different industrial sectors.

[4]https://www.iiconsortium.org/IIRA.htm

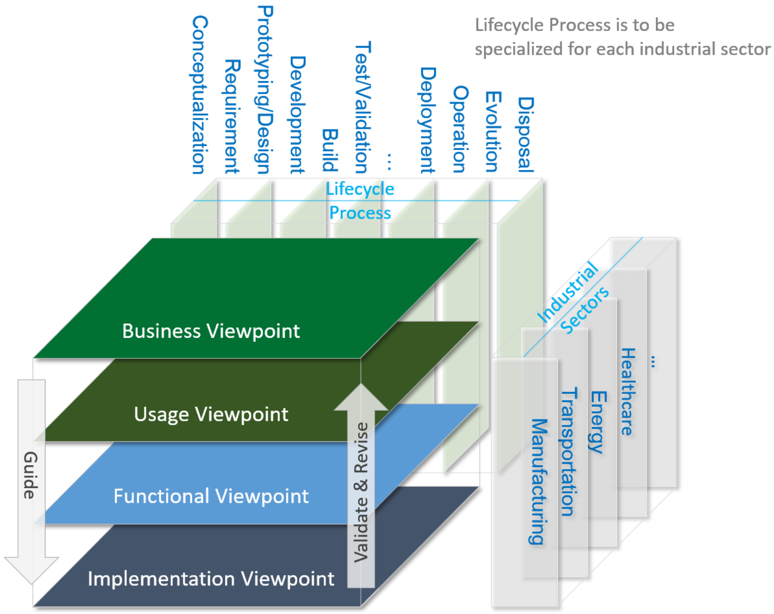



Fig. 1.4 IIRA overall architecture

Figure 1.4 relates the IIRA *Viewpoints* with industrial sectors and product lifecycle stages. We briefly analyze the concepts and the several layers resulting from the division of responsibility that IIRA provides for any corporate production environment. The *viewpoint*, the *concerns*, and the *stakeholders* are the crosscutting concepts that we find in all specification's parts. The *stakeholders* are individuals, teams, or organizations interested in a (or a group of) *concern*, a topic of interest in several domains in industry. A *viewpoint* is a logical plane framing one or more concerns. For each *viewpoint*, there are several notable abstractions, representing real concepts and tools employed in the company.

The **Business Viewpoint** is focused on the values and objectives of business departments and stakeholders, such as the expected return of investment, the costs concerning the maintenance, and the product liability. This viewpoint brings the concepts of *Vision*, the directions and the future state of the company, of *Values*, the perception of the vision by the stakeholders, of *Key Objectives*, the expected business results deriving from the values delivering, and of *Fundamental Capabilities*, the high-level representation, neutral to the real implementation, of the ability of the system in completing the business tasks. The concerns relative to the **Usage Viewpoint** are sequences of activities carried out to achieve the system's key capabilities specified in the business viewpoint.

A key concept of this layer is the **Task**, an operation inside the IIoT system, such as the authentication of a user request, the registration of a new device with the gateway, aggregation of data streams from a set of sensors, etc. The access and the execution of a task are regulated by the *Role* concept, a set of permissions related to an entity that wishes to execute or collect the results of a task. The combination of tasks is an *Activity* which represents a process inside the IIoT system and can have *Triggers* conditions under which the activity is executed.

The richest viewpoint is the **Functional Viewpoint** one, divided into several subdomains: Control, Operations, Information, Application, Business. These domains identify all the functional components of the IIoT system, their characteristics, their inter-connections, and the relationships with the outside world. The *Control domain* is the most sensitive domain of all, including actions such as the sensing and gathering of data coming from sensors, and the control of the system's physical assets through actuators. The *Operations domain* relies on the management of resources representing the functions for provisioning monitoring and optimization of the assets' operation. This level assumes that managerial and design offices have to retrofit the legacy asset still operating in the IIoT system with new computing, storing, and connection competencies. Monitoring, Diagnostics, and Prognostics are the concepts that enable real-time monitoring, detection, and prediction of failures of working machines. In the *Information domain* we can find how gathering, processing, storing and analyzing information to get high-level knowledge from raw data. The deployment of these functions may be on-premise or in the cloud. The *Application domain* contains application logic without executing low-level operations delegated to the Control domain via secure calls. This domain maintains local rules and models in the event of connectivity loss and exposes API and user interfaces for external interactions. Last domain, the *Business domain*, has the business logic proper of the industrial IT world, such as the Customer Relationship Management (CRM), the ERP, the MES, the Product Lifecycle Management (PLM), the Human Resource Management (HRM). This viewpoint has a set of crosscutting functions using data from several domains, such as security and connectivity.

Inside the **Implementation viewpoint** there are components to implement activities and functional requirements reported in the functional viewpoint. The tools proper to this layer must be compliant with all the business viewpoint constraints such as the time-to-market and the costs. In essence, this is a tier that describes the structure of the IIoT deployment, including the level of distribution, the communication protocols, and the interfaces to access the services.

Another notable aspect of this architecture is the foreseeing of different distributed configurations, which already have proven to be remarkable in the past from the point of view of the IIoT business systems deployment. The first conceivable deployment has three levels: Edge, Platform, and Enterprise layers. The *Edge* tier gets data from physical devices (called *edge notes*) through the *Proximity network*. This tier has all the requirements of the OT layer, or low latency, high bandwidth, and enhanced security. The *Access network* forwards the data from this tier to the Platform tier and transfers the control commands in the opposite direction. The *Platform* middle tier processes, analyzes, and eventually aggregates data from the Edge layer. Data flows to the Enterprise layer via the *Service network*, used also to pass commands backward, from the upper layers to the physical level. In the *Enterprise* tier, there are domain-specific applications and big data analytics tools. From this deployment, we can learn the importance of network separation which results in a separation of objectives and better security.

Another deployment encompasses the use of an Edge/Gateway Hub as a bridge between the edge nodes and the WAN. The gateway represents the access endpoint to the edge nodes and isolates the local network and in some cases, it could act as a management entity that aggregates, processes and analyzes data.

The last proposed approach in the IIRA specification is the layered one. This is a deployment suitable for systems with many interactions between the components, and which, thanks to the stratified way, ensures low-latency for the real-time analytics in lower layers and secure P2P data communications among layers. In the higher levels, it is possible to deploy supervisory, control and monitoring tools. Since each layer can have its own data model, adapters could be used to align data models among different layers, or for integrating legacy assets, also using publish-subscribe protocols.

### Intelligent Manufacturing System Architecture (CH)

With the **Made In China** (**MIC**) **2025** initiative [5], inspired by Germany's I4.0, China formalized its 10-year industrial plan for the 10 most important production sectors in the region. The strategy focuses on the implementation of smart manufacturing techniques improving efficiency, quality, and productivity inside the factory's shop floor, in order to compete with emerging low-cost competitors: such as South Korea, Japan, and Germany. Since the actuation of the initiative forecasts an enhanced interconnection and digitization, the information technology and the IoT play a crucial role to connect SMEs production chains with the global production network. The target is the whole production industry, with all their stakeholders, from technical to managerial departments, whereas the demand is the formalization and the adoption of international technical standards. In this context, the committee defined the Intelligent Manufacturing System Architecture (IMSA) specification, whose architecture is shown in Figure 1.5.

The reference model for the transition to the fourth industrial revolution is substantially similar to RAMI 4.0. The standardization of all the company procedures, such as business processes and technical tools for implementation and testing, is a key concept in order to

[5] https://www.isdp.eu/publication/made-china-2025/

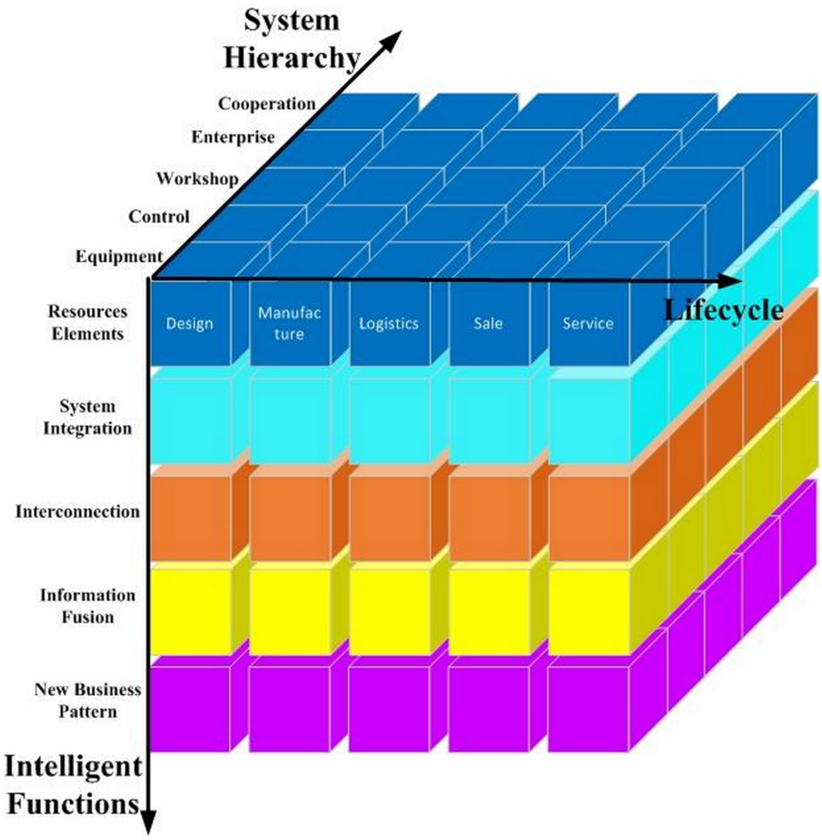



Fig. 1.5 IMSA overall architecture

conform and integrate all manufacturing sectors that intend to adopt a smart manufacturing approach. In the following, we expand the three dimensions of the IMSA architecture:

- The **Lifecycle** axis refers to value creation activities, covering the whole life of a product. The specification identifies these layers for the lifecycle axis: it starts from the R&D phase, with the *design* of the product, continuing with the *manufacturing* process to create the goods, the *logistic* process to set the asset to its destination, the *sale* and the *service* representing the transfer of goods from seller to customer and the post-sale services, such as smart maintenance and recycling.

- The **System Hierarchy** axis identifies the entities involved in manufacturing process: the *equipment* hierarchy covers all the devices responsible for asset management such as sensors, metering tools, radio frequency identification (RFID) devices, and data gathering tools, the *control* hierarchy encompasses all the instruments to effectively realize monitor and control of the physical processes, such as PLC, SCADA protocols, field bus control systems (FCSs). The third hierarchy, the *workshop* ones, is intended as all the components managing the factory, such as the MES software, and it is different from the *enterprise* hierarchy containing the enterprise actual management tools, such as the ERP, the PLM, the supply-chain-management (SCM), and the CRM software. The last *cooperation* layer represents the internal and external interconnection and sharing processes.

- The **intelligent functions** axis relies on the real implementation of smart manufacturing functions, such as self-sensing, self-adaptation, predictive maintenance, accomplished through the exploiting of modern information and communication technologies. In this axis, we can find the *resource* layer referring to resources and tools used during the manufacturing process, the *system integration* layer used to achieve integration of the assets with the intelligent production line, the *interconnection* layer, which trivially describes the connections between devices, such as wireless/wired protocols, the *information fusion* layer describing collaborative information sharing modalities, such as cloud computing and big data, and, finally, the *new business* layer used to drive the intersection of the enterprises value chains in order to reach the goal of conformity of the new unified industrial ecosystem.

The specification flanks a structural diagram to the 3D layer model. The aim of this further document is to frame the great number of standards that are part of the specification, let us briefly see the three main layers:

- The bottom layer is called **Basic Generality** and with five subcategories it identifies the basic concepts associated with the smart manufacturing, the safety standards for safe control and maintenance of smart assets, and those for performing the tests and identifying the crucial indexes to be kept under control within the production environment. The last subcategory, on the other hand, covers standards for analyzing the reliability of the system and satisfying the required requirements.

- The immediately above layer, called **Key Technique**, is the richest one with several subgroups and subcategories. The *Intelligent Equipment* subgroup collects standards relating to the cyber-physical system inside the workshops, while the *Intelligent Factory* subgroup involves those relating to planning, design, integration of legacy components and logistics. The subgroup on top of the previous two, the *Intelligent Service* relies on modern smart manufacturing standards, embracing the customization, and the remote maintenance and operations of the smart goods. Two subgroups are crosscutting to the whole Key Technique category and deal respectively with data management standards (such as big data and distributed industrial software) and with standards relating to the industrial internet, covering various architectural deployments, resource management, and network equipment.

- The highest layer, namely the **Industrial Applications**, covers all the industrial areas of interest, encompassing all the standards dealing with requirements of specific application domains. Examples of application areas covered by the Chinese specification are the biomedical, aerospace, energy, agriculture, and transport sectors.

### Open Platform Communications Unified Architecture

**OPC** is a series of standards specifications and it was originally based on Microsoft's OLE COM (Component Object Model) and DCOM (Distributed COM) technologies. The specification defines a standard set of objects and interfaces facilitating the interoperability among control processes and manufacturing automation applications [6]. The many challenges of I4.0 transition forced the OPC foundation to think about the heterogeneity of components (software and hardware) and about a unifying architecture tackling these diversities, namely the **Unified Architecture** (UA) specification [7], released in 2006 and become the IEC-62541 standard. OPC UA is a platform-independent, service-oriented architecture specification applicable to all industrial domains. It defines a common infrastructure to exchange information and control industrial processes, by specifying the interaction between applications, the communication models and the semantic structure of the information.

OPC UA supports a lot of communication patterns such as direct messages between Client and Server or network messages from Producers to Consumers. The great interoperability of the model targets SCADA, PLC and DCS interfaces, proposing ways to get interoperability among these systems and the higher-levels such as the MES and advanced control functions. The tiered approach isolates the transport network mechanisms from the core design, furthermore, its developers have foreseen three data encodings (XML, UA Binary and JSON) to remain as interoperable as possible with a wide range of protocols and applications, and also providing for different transport protocols (TCP, HTTPS, WebSockets).

In our opinion, the OPC UA specification is the most technically detailed. Below we will list a non-exhaustive set of key parts from the specification which show the great compatibility and completeness of this standard.

Parts 3 to 6 (respectively titled **Address Space Model**, **Services**, **Information Model** and **Mappings**) define how client/server communication (C/S) should be regulated, together with the exchange of messages required by the specification. A client can rely on many servers and vice versa, and also the combination of clients and servers in a single agent is allowed in some scenarios. The C/S interaction also allows the Server-to-Server communication, enabling various features such as the redundancy, the aggregation of data from lower-layer

[6] https://opcfoundation.org/developer-tools/specifications-classic/data-access/
[7] https://opcfoundation.org/about/opc-technologies/opc-ua/

servers, the ability to manage many client's requests with a single dispatcher server which propagates them to the involved server. The Services part describes the interfaces regulating the C/S communication, whereas the Information Model part manages the interaction model between the counterparts.

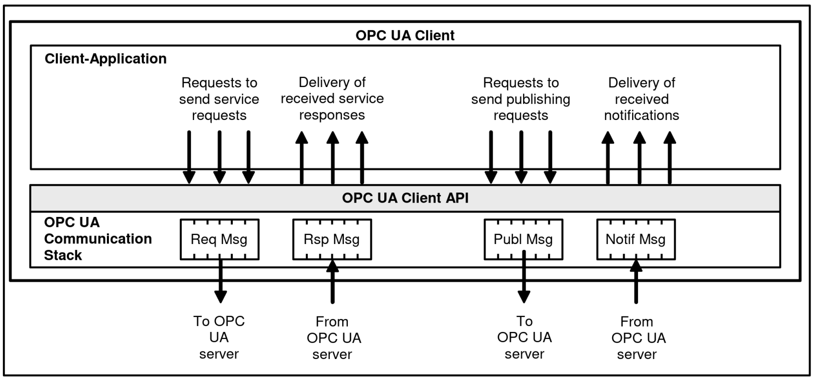



Fig. 1.6 OPC UA Client architecture

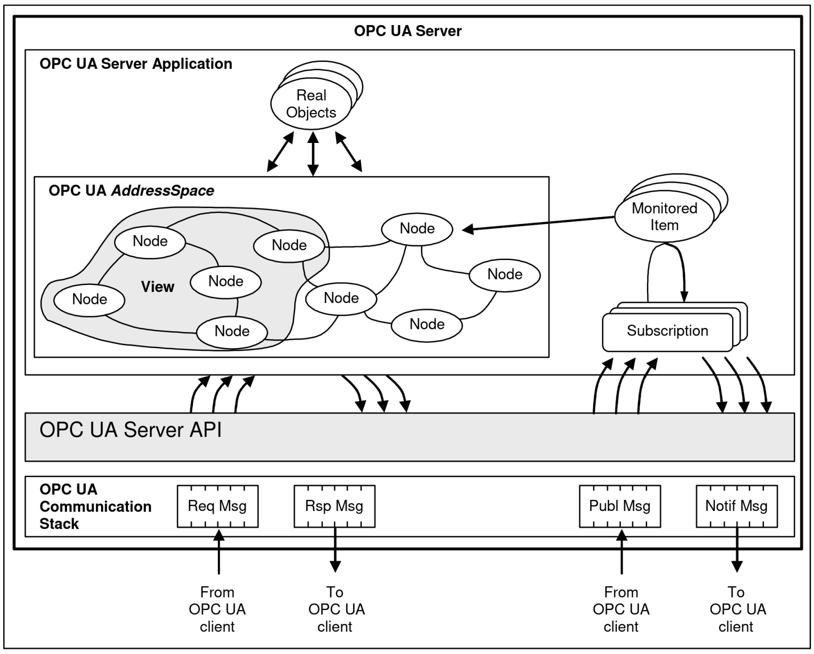



Fig. 1.7 OPC UA Server architecture

As we can see from Figure 1.6, the **Client** has several components inside: the client application is the implementation of the client's functions, the client APIs communicate with the server isolating the business client code from the communication stack, which converts the API calls and the notifications in messages. Figure 1.7 depicts the **Server**'s internal units:

the Server Application is the implementation of all the server's functions, the Real Objects are physical or software objects (such as physical devices or diagnostics counters) accessible by or internal to the server application. API servers have the same purpose as those of the client, i.e., communication interfaces that decouple the implementation of the service code from the underlying communication system, transforming the information exchanged into messages.

The *AddressSpace* describes a set of Nodes accessible to clients calling server's APIs. The concept of *Node* is the representation of a real object and its correlations and references with other objects. The AddressSpace enhances the interoperability of clients and servers and the specification defines a tree of hierarchically organized *NodeClasses* with a common top-level for all servers. The NodeClasses represent the objects in terms of their own variables, events, methods, and relationships with other objects. The views inside the AddressSpace are isolation mechanisms that permit the server to choose the nodes to make visible and accessible to a certain client. The *MonitoredItems* are objects created by client's requests to be notified when data change, alarm or event on one Node, or on their real-world counterparts, occurs, generating a notification propagated to the subscribed clients.

The part 4 of the OPC UA specification (i.e., **Services**) describes many services, organized in *Service Sets* grouped by functionality kinds. For example, the *Discovery* service set comprises the *FindServer*, *GetEndpoints*, and *RegisterServer* services, the *Secure Channel* service set consists of *Open* and *ClosesecureChannel* services. Or the *Session* service set encompasses the *Create*, *Activate*, and *CloseSession* services, whereas the *Browse* service set includes *Browse/BrowseNext* and *Register/UnregisterNodes* services. One of the most common services is the subscription of a client to the server notifications occurring with relevant events such as Alarms, data value changes, and method execution results. Another very common service set is the *Subscription* one, allowing clients to subscribe to specific events so it can get notifications relative to Alarms, data value changes, and method execution results in an asynchronous way.

In addition to the session ones for secure and reliable communication and to the monitoring services, very attention is given to redundancy services. OPC UA enables redundancy at the client, server, and network levels. *Server Redundancy* allows Clients to have multiple sources from which to obtain the same data and it can be transparent or not for the clients. *Client Redundancy* allows identically configured Clients to behave as if they were single Clients, but not all Clients are obtaining data at a given time. *Network Redundancy* allows a Client and Server to have multiple communication paths to obtain the same data.

The part 14 of the OPC UA specifiaction (namely **PubSub**) describes the alternative communication pattern, based on message exchanges. OPC UA does not constrain the developers to bound their software to a specific message-oriented middleware, leaving the choice free, and providing for the use of broker-based and brokerless messaging systems and different transport layer protocols: OPC UA UDP (a simple UDP based protocol), OPC UA (an Ethernet-based protocol), AMQP/MQTT (two of the most used IoT protocols). This interaction model is based on OPC UA Information Model, as the C/S interaction, therefore PubSub and C/S communications can be combined in complex use cases, where often a Server is a publisher and a client is a consumer.

OPC UA developers paid close attention to the safety and security of the actors. Given the use of the OPC UA specification in several enterprise levels, from industrial production plants to manager departments, it is obviously an attractive target for security threats aimed at espionage or sabotage. Part 2 of the specification, called **Security Model**, identifies a list of non-negligible vulnerabilities and threats and provides countermeasures. As for any information system belonging to the IT sphere, OPC UA identifies the same properties to be guaranteed for a secure operation inside production plants: authentication, authorization, confidentiality, integrity, non-repudiation, availability. Among the addressed vulnerabilities, we can count: denial of service, message spoofing or alteration, server profiling, session hijacking, compromising of the user credentials. For each threat faced, the standard offers practical solutions that rely on security tools coming from the IT world or refers to the use of security patterns, leaving developers free to use their own security mechanisms, as long as they are compliant with the security objectives of OPC UA.

### Other standardization initiatives related to Industry 4.0 transition

Other minor standards and initiatives that play a role in the Industry 4.0 transition exist. Not having a central role in the industrialization movement, we report them below in a non-exhaustive list.

- The Japanese **Industrial Value Chain Initiative** (IVI) [8] envisions a society in which manufacturing realities and IT domain are fused rethinking the relationship between industry and people. The Japanese Initiative, which as we will see will be extended by the more modern *Society 5.0* paradigm, is an active forum aiming to direct societal change by nearing creators and users, analyzing the relationship between people through things and information.

- The **MTConnect** [9]standard uses a no proprietary format to provide a semantic and structured vocabulary for manufacturing equipment. Uniforming data allows devel-

[8]https://iv-i.org/
[9]https://www.mtconnect.org/

opers and industry integrators to focus only on business activities, rather than on translations.

- The **ISA 95** [10] and **ISA 104** [11] standards focus on integration, from the point of view of control systems the first and of devices the last.
- The standard *IEC 62264* [12], based on ISA 95, is focused on systems integration, while the *IEC 61850* [13] and *IEC 60870* [14] are focused on issues related to a specific industrial sector, in this case, the communication and control of modern power grids. The *IEC 61131* [15] regulates the programmable logic controllers and *IEC 61499* [16] describes the function blocks in processes and communication for embedded devices, resources, and application.

### 1.1.4 Integration Driving Communities

The first part of this section talks about CPSs and IIoT research communities, very focused on the IT/OT layers integration topic. Although it is not mentioned in the classical definitions of these two systems, the goal of the CPS and IIoT communities is to enable IT/OT convergence by bringing integration capabilities inside production plants. Alongside the main research communities, we also selected a series of key technologies which in our opinion represent the fundamentals for the implementation of the transition.

#### Cyber Physical Systems and Industrial IoT

**CPSs** represent the evolution of the manufacturing sector, where mechanical and IT skills are combined, thus making the convergence of the physical and digital worlds possible. CPSs are systems that integrate sensors monitor different kinds of physical quantities, elements that are part of the low and high levels of the ISO/OSI stack, respectively. The I4.0 movement is embracing the CPSs philosophy to integrate intelligent functionalities and smart monitoring with enterprise processes and shop floor environments. Furthermore, the Cyber-Physical Production System (CPPS) represents a novel research area merging the CPSs capabilities

[10] https://isa-95.com/
[11] https : / / www . isa . org / standards-and-publications / isa-standards / isa-standards-committees/isa104
[12] https://www.iso.org/standard/57308.html
[13] https : / / www . automation . com / en-us / articles / 2003-1 / iec-61850-power-industry-communications-standard
[14] https://webstore.iec.ch/publication/3755
[15] https://plcopen.org/iec-61131-1
[16] https://www.iec61499.de/

with the production lines, employing the interaction interfaces for orchestrating the actors of the production context: human resources, production equipment, and aggregated logical and physical enterprise products.

On the other hand, **IIoT** systems are increasingly employing advanced sensors and actuators, integrated with the software that manages the production environment. The IIoT, known as the industrial employment of IoT devices, is another combination of hardware sensors (extracting valuable information from the surrounding environments) and software capable of data gathering and processing, sometimes overlapping with CPS/CPPS definitions. Specifically for the manufacturing sector, but also more generally for all production realities, IIoT systems promotes the combination of networking and computing resources with control and automation manufacturing equipment.

### IT/OT Convergence key enablers

Besides communities such as IIoT and CPS, there are many IT technologies that with their timing enabled the evolution and implementation of the I4.0 concepts.

The deterministic capabilities of the **Time-Sensitive Networking** (**TSN**) pool of standards will help the companies to accomplish a successful IT/OT convergence inside their production plants. TSN grants deterministic communication over Ethernet, introducing different traffic flows sharing the same physical link. The integration of multiple communication standards can bring IT and OT communication coexist in the same network, integrating many legacy machines and not-deterministic data flows. Time awareness of TSN protocols enables the real-time capabilities that operational assets in production plants need, while keeping the benefits of best-effort communications used for the purpose of typical IT tasks like offload information processing [11]. The filtering policies regulate the priority of the packets at every layer, whereas the network configurations (Central network controller (CNC) and Centralized user configuration (CUC)) allow scheduling for transmissions and receptions, enabling a reliable and fine-grade configurable communication inside convergent environments.

The spread of the **5G** technology will allow companies to rely on radio communications to run processes that require latencies in the order of milliseconds. The 5G networking supports three communication modes potentially suitable for industrial production scenarios: enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), massive Machine Type Communications (mMTC). The network slicing feature introduced by the 5G will enable the coexistence of multiple data flows, with diverse timing requirements, which is considered one key requirement for the convergence of OT and IT systems. The innovative technical features of modern cellular networks will be able to favor the technological development of corporate networks. At the same time, however, it will be necessary to respect the safety specifications dictated by prolonged exposure to the electromagnetic fields (EMF), and to keep in mind the safety measures proposed by international bodies [12] in order to protect the health of workers, one of the main objectives of the I5.0 transformation.

**SDN** and **NFV** allow smart management of converged IT/OT networks, providing agility and end-to-end control over infrastructure deployments. The SDN paradigm will help companies to abandon the IT and OT silos approach in favor of an integrated common networking environment. SDN will supersede VPN-based mechanisms that introduce non-negligible latencies and hinder the management of resources in full autonomy. Network segmentation offered by SDN will allow companies to carry out agile provisioning of enterprise-level services that is requested in modern industrial scenarios. NFV provides Virtualized Network Functions (VNFs) that enhance the reliability and robustness of currently operating services and hide the complexity of managing the underlying network. Both IT and OT departments can deploy network services on existing COTS hardware without dedicated network assets, thus making a step forward to a reliable integration of the two worlds.

IT/OT convergence requires high IT security mainly to defend work machines from external access. We believe modern **cybersecurity** holistic approaches are a key enabler for a secure and safer IT/OT integration. Besides taking care of access control, cybersecurity is useful to keep track of communications among machines and to identify deviations that may occur in automatic communications, which are symptoms of attacks intended to induce unwanted machine behavior that will eventually lead to dangerous situations for humans at work. In IT/OT converged networks, cybersecurity will help companies to keep under control both IT security and employee safety.

The five benefits of **Cloud Computing** meet many industrial production requirements, by providing on-demand services, with great scalability, on a pay-per-use model, sharing the available resources, and making them accessible through broad network access. Cloud computing is a great enabler of modern manufacturing innovation, as it helps to reduce costs and grants access to complex IT services also to SMEs that can not afford an IT team. An agile approach to data storage, management, and analytics favors the integration of data and communication demanded by the IT/OT convergence. The remote cloud is the right place to gather data coming from several partners, platforms, and data sources, and combine them to create a high-value knowledge base exploitable by all I4.0 stakeholders.

**Edge** and **Fog Computing** models foster the placement of computational resources as close as possible to data sources. In production scenarios, they will bring further improvement to cloud-based smart manufacturing processes. Besides the clear advantages Edge-Fog will bring in terms of reduced network latency and a larger bandwidth, it enhances the cybersecurity of the plants since data and computing processes are naturally defended by the enterprise's domain boundary. With computation taking place at the network edge, processes showing very different requirements like those run by IT and OT departments can be served. Furthermore, the edge/fog models overcome the problem of outsourcing sensitive data.

**Digital Twins** established itself as a paradigm to connect OT assets to IT tools in order to monitor and control the behavior and performance of physical assets. Digital Twins in the manufacturing sector is recognized to be a fundamental enabler for convergence for its capability of guessing the behavior of the system in response to certain stimuli. With that a priori knowledge, it is possible to programmatically understand how the system will react, for example, to the connection of the machines to a convergent network, reachable by IT departments and from the outside, or how the IT tools will be able to support a workload increase due to the addition of new machines to the fleet.

One of the most important steps in the IT/OT integration process is to find a consistent and common **semantic** framework for the two departments. An effective implementation of a common vocabulary will make the integration of the data coming from both departments possible, thus allowing full data exploitation from all stakeholders. A standard set of the key terms used in production environments will favor seamless integration of IT and OT assets and operations.

The modern technological advancement of analysis software based on **Artificial intelligence (AI)** techniques makes these tools easier to use, so a lot of companies have employed them in production plants. A wide used feature is the plant analysis in order to enable predictive maintenance, where technicians can predict the failure of the asset before the machine breaks, and fix it accordingly. Machine-learning applied to work machines allows to improve their performance and to learn their behavior in different situations, only by monitoring them during their operation. The support these technologies are giving to IT/OT convergence is surprising and AI-powered applications promise benefits for both OT and IT environments. From the point of view of the company's software improvement, business insights and data-driven decisions can easily create a competitive advantage for any enterprise. ML and AI improve also the OT layer performance. Monitoring assets' health and operations provide a huge amount of data and, to effectively combine them with the information converging from the IT layer, it is paramount to use smart approaches such as reasoning and machine learning.

Figure 1.8 groups under three macro-categories the main enabling technologies that promote and enable the IT/OT levels convergence in modern industrial production environments. The three logic categories groups respectively the Infrastructures enabling the IT/OT convergence, such as cloud and edge computing, the Data Management strategies, i.e., AI/ML, DT, Cybersecurity, Semantic ontologies. The last layer is Communication one comprising advancements in connections, such as cellular networks, deterministic and softwarized networking.

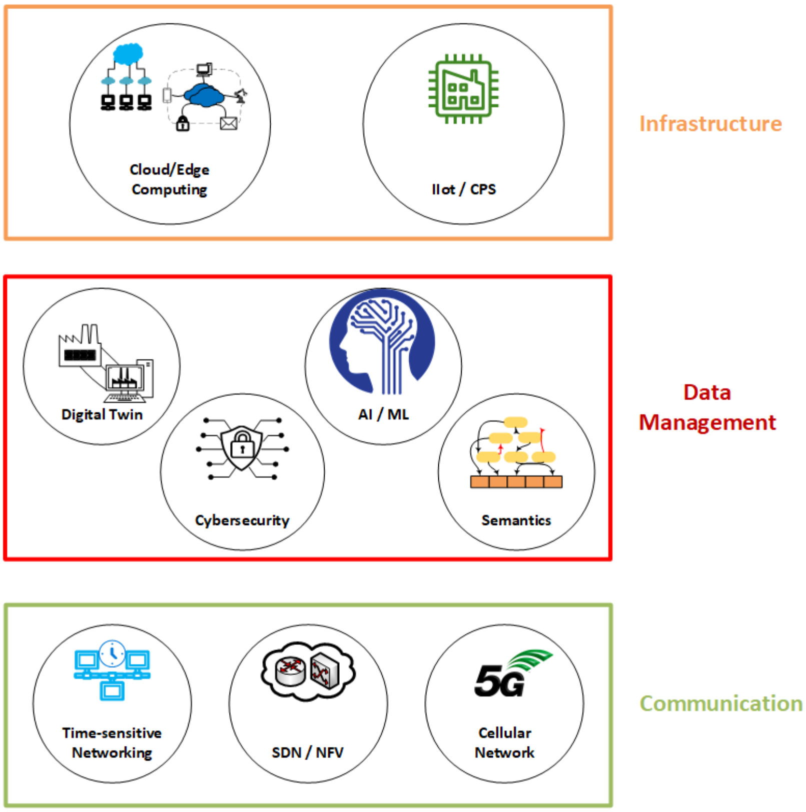



Fig. 1.8 Main IT/OT convergence enablers

Most of the enabling technologies introduced are already well established from an IT point of view, but thanks to the transition to Industry 4.0 they have been largely employed also into the OT domain and are the most used to implement IT/OT convergence. The enabling technologies of the previous sections have mostly been inferred from the literature, but they also are extracted from collaborations with local companies and from the trends that the latter follow.

## 1.2 The emerging human-centric vision

The main idea behind the emerging Industry 5.0 movement is putting the industry ecosystem in service of humankind and the environment. In this section, we will see how the new industrial transition is reshaping society and what are the common ideas behind the European Industry 5.0 and the Japanese Society 5.0 movements.

### 1.2.1 The 5th industrial revolution

The European industry, although in the midst of Industry 4.0, is embracing a new reaction to the conception envisioned by the fourth industrial revolution. In order to keep the industry the central point of the modern economic and societal transitions, the Industry 5.0 paradigm is emerging as a new approach complementary to the I4.0 transition. The experts encouraging this new vision aim to build an industry that has objectives and goals closer to the development of the whole society, besides mere productivity and efficiency improvements.

I5.0 put the well-being of the workers and the societal progress at the heart of its transition. We want to remark that the new I5.0 idea is not a completely new revolution as the previous ones, but it is complementary to I4.0, given the events that in recent years have marked the society and the relationship between industry and workers/environment. The recent health crisis caused by the COVID-19 pandemic that broke out in 2020 and the increasingly frequent disastrous phenomena due to climate change pose new challenges to industrial evolution. Figure 1.9 depicts the three pillars of the I5.0 perspective, complementary to what was stated by I4.0.

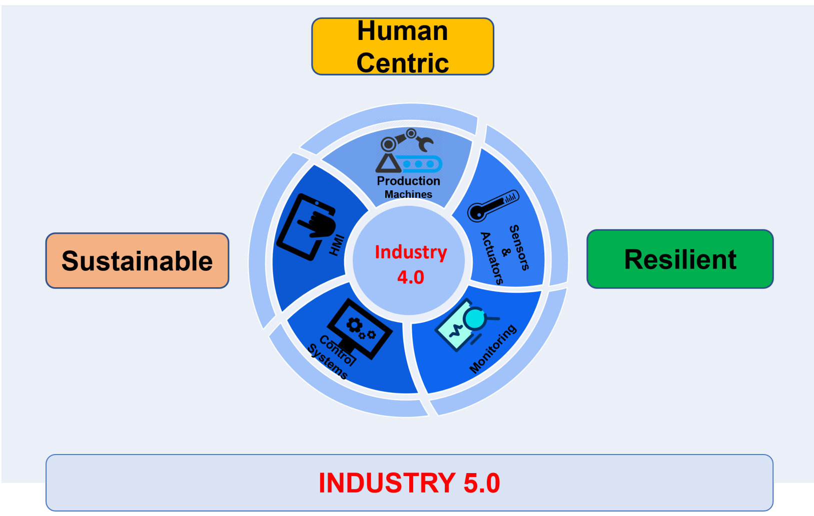



Fig. 1.9 The three pillars of Industry 5.0 transition

With its pervasive role in current society, the industry can play an active role in societal advancement, putting research and innovations at the service of people. The stakeholders involved in the I4.0 transition should pursue the objectives set by I5.0, that is to place humans at the center of evolution, and to create a sustainable and resilient industry.

The focus on society foreseen by the I5.0 needs for a shift of the focus from the technology-driven progress to a human-centric approach taking into account the societal constraints in order to not leave anyone behind. The core of this process is not only the respect of human rights but also the improvement of the working environment and workers' skills. This means that technology used in industrial sectors should be adapted to the needs and heterogeneity of workers instead of having the workers constantly adapt to ever-evolving technologies. Rather than replacing workers with automation tools and Artificial Intelligence, these kinds of new technologies should be a support to the workers' duties. Furthermore, the capabilities of the employees should be always improved by upskilling and reskilling with respect to the advancements of the technology they are using to overcome the skill shortage that plagues and slows down the I4.0 transition. People with reduced mental abilities can exploit the power of robots and AI-based technologies, as well as virtual/augmented reality tools. People with minor motor skills can take advantage of exoskeletons and robots to make certain tasks more physically demanding.

Resilience is the other concept on which I5.0 puts its focus. Workers also have to go through the same evolution from the point of view of resilience that machines and production processes have gone through to undertake the I4.0 transition. Never like these years in which the health crisis has hit our society, modern industry has had to face removal from jobs and unscheduled disruptions due to the ongoing emergency situation. The policies of remote jobs should be improved and become a habit in all workplaces where it is possible to remotely work.

The need to have a resilient industry has also emerged thanks to the **Next Generation EU** (**NGEU**) program, which in Italy has materialized in the **National Recovery and Resilience Plan** (**NRRP**) [17]. The plan developed by the Italian government, originally meant for repairing the economic and social damage caused by the pandemic crisis, is in line with the precepts of I5.0 and it provides measures for digitization and innovation, ecological transition, and social inclusion. With the NRRP Italy aims to reduce territorial, generational, and gender gaps, keeping emissions under control and addressing environmental transition.

From the point of view of sustainability, usually increasing industrial production requires more energy and increases carbon emissions. We have the solutions to keep this problem under control, such as the employment of energy-efficient technologies, smarter production planning, and the use of modern materials. Bio-inspired technologies, raw materials generated from waste, or materials with intrinsic traceability are all solutions capable of reducing

[17]https://italiadomani.gov.it/en/home.html

emissions. Furthermore, digital twins, simulations, and AI-powered analysis can drive the industry to reduce wastes and optimize the processes and the resources they use. From now, modern industries should think about the exploitation of renewable energy sources, energy-autonomous sensors, and low energy data transmission and data analysis technologies.

### 1.2.2 Relation between Industry 5.0 and Society 5.0 movements

The Industry 5.0 concepts overlap with some ideas introduced by the Japanese **Society 5.0** [18] societal-digital transformation plan. The number 5 results from a different and much longer timescale than that of industrial revolutions. The first two "Societies" correspond to the pre-industrial periods (until the end of the 18th century) and are respectively related to the hunting/gathering and the agricultural economies. Society 3.0 is an industrial society and corresponds more or less to the period of the first, the second, and part of the third industrial revolutions. Society 4.0 is characterized by the dominance of "information" and we can say that it evolved from a highly digitized version of the third industrial revolution, up until today.

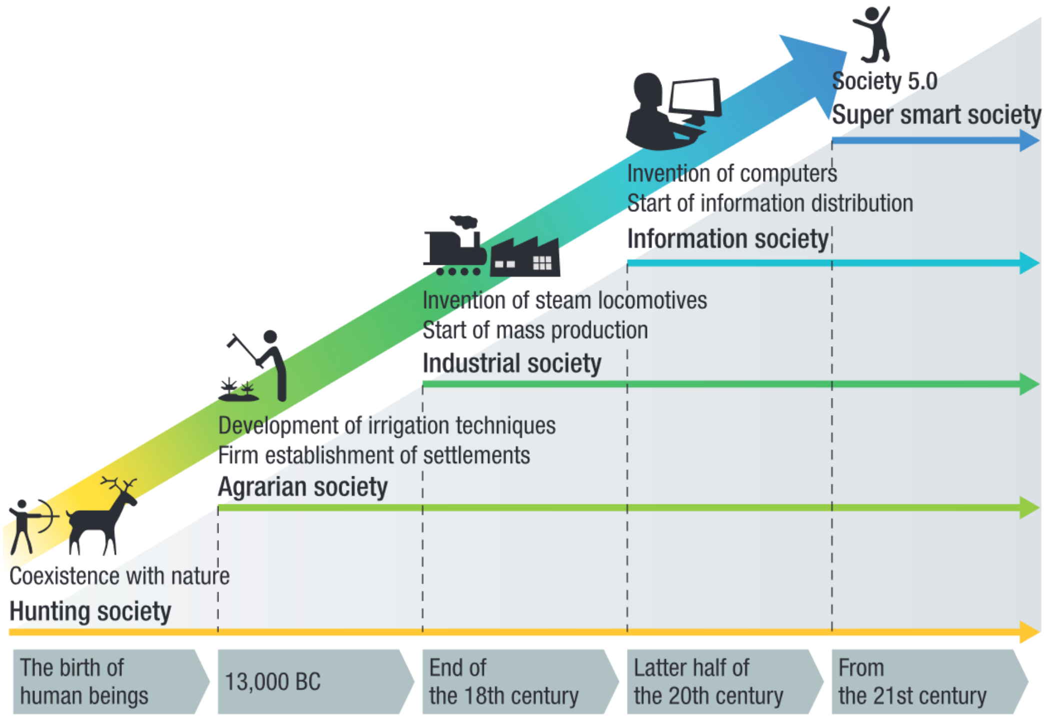



Fig. 1.10 Society historical stages [13]

[18] https://www.japan.go.jp/abenomics/_userdata/abenomics/pdf/society_5.0.pdf

Japan is trying to take the digitalization progresses achieved on the level of individual organizations to a full national transformation strategy, policy, and even philosophy. Society 5.0, as the name suggests, it is not restricted to the manufacturing sector only, but it is an attempt to balance economic development with the resolution of societal and environmental problems. The transition should address large social challenges with the help of advanced IT technologies, such as the Internet of Things, robots, artificial intelligence, and augmented reality, that will be part of the society and not employed only in the industry and only for economic advantages.

There are two main enablers of S5.0 in Japan. One of them is the great amount of collected data from the advanced health care system and from the manufacturing realities. The other main enabler is the *monozukuri* [19], the Japan's excellence in manufacturing which over the years has led to strong progress in the manufacturing sector by creating products that exploit modern technologies such as Big Data and AI that can now be released into the whole society. By taking advantage of these unique factors, Japan expects to overcome big social challenges such as a decrease in the productive-age population, the aging of local communities, energy, and environmental issues, also improving crucial societal sectors such as healthcare, mobility, infrastructures, and fintech.

[19] https://en.wikipedia.org/wiki/Monozukuri

# Chapter 2

# Modern Manufacturing Related Works

We grouped the main contributions from the literature explored during our three-year research into different categories. The first largest category of related works concerns our central SIRDAM platform and platforms/architectures that address our same topic from different points of view, i.e., the IT/OT integration. The last section will report the main state-of-the-art regarding the concepts of Industry 5.0 and Society 5.0, mainly from the perspective of the technical advancements and IT tools employed in these big transformations.

## 2.1 Layers convergence state of the art

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 the factory premise. 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.

This section collects research contributions that address the IT/OT convergence from the data management perspective. Specifically, in Section 2.1.1 we discuss relevant literature works that propose data acquisition and management solutions in IT/OT converging environments. In Section 2.1.3, 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 2.1 to help the reader grasp the advantages offered by our platform compared to those offered by state-of-art proposals.

### 2.1.1 Data acquisition and management

In [14], 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 [15] 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 [16], 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 [17], 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 [18] 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 [19], 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.

[20] proposes a novel mechanism for the autoconfiguration 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.

### 2.1.2 Safety and security

In [21], authors propose a layered approach to tackle safety and security issues in modern IT/OT convergent networks. After summarizing the most used network protocols at IIoT devices level, they map a risk level to every security threat. They discuss a number of good practices for implementing safety in an industrial environment such as access control, enabling remote access only in case of real need, and controlling changes made to devices. The proposed strategy is to use the "defense in depth" approach, a layered military security strategy, according to which no single tier has to hold on its own, but the global defense depends on a set of choices made by a pool of tools acting in different layers. Authors propose to adopt such a vision in regular system activities, such as maintenance and asset management.

Many security companies [22] [23] are aware that convergence raises the need to face new security issues and suggest guidelines for implementing a system that safely integrates IT and OT data. They claim the first rule manufacturing companies should follow is keeping a complete and updated list of their asset. IT/OT converged infrastructures should use a next-generation firewall (NGFW) which, by combining a traditional firewall with network filtering functions, can categorize OT application protocols and passively observe the network and encrypted traffic, thus providing the ground on which to profile and classify physical devices on the network. According to them, network architects have to separate the network in logical realms and define precise rules regulating the information and entities that can access and communicate in each network segment.

It is common opinion that a holistic approach to security is the optimal solution to cope with the complexity added by the IT/OT convergence. In [24], authors focus on the cybersecurity problem in an IT/OT convergent network. They regard OT as a no longer standalone and relatively secure environment. They identify the attacks brought to OT systems during the last years, and summarize the possible consequences. Finally, authors underline how the convergence of two tiers yields a social change that involves the choices made by the two departments, sharing the knowledge and the protection mechanisms, which have hitherto been confined to their respective domains.

Besides the cybersecurity of access, the compliance of new assets with the shop floor policy is also a notable feature in terms of security. In [25], authors formalize an algorithm for checking the degree of security of new proposed product designs. We believe this is an interesting contribution as the same approach can be adopted in I4.0 environments to determine the manufacturing feasibility of a new design with a given set of machine tools.

### 2.1.3 IT/OT integration experiences

In [26], 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 [1] 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 [27], 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 [27], 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 [28], 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 [29], 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

[1] https://www.isa.org/

adopted in critical environments such as corporate production sites in the manufacturing sector.

Authors of [30] 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. [26] and [27] mostly deal with the problem of semantic convergence of shop floors entities. [28] and [30] 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 [29] 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.

## 2.2 Industry 5.0 and Society 5.0 state of the art

In this section, we report the contributions in the literature that deal with implementations and architectural solutions from the world of information technology adapted to the purposes of the Industry 5.0 and Society 5.0 movements. In conclusion, we introduce some works to corroborate our decision regarding the use of crowdsensing technology to emphasize human centrality within the technological transition.

Work [31] talks about the centrality of IT technologies in the industrial and societal evolutionary process. The technologies proposed by the authors derive directly from those promoted during the fourth industrial revolution and aim to implement the incoming industry 5.0. The study identifies a set of technologies that will speed up the I5.0 adoption for many industrial domains, from private companies to the public sector. Advanced distributed computing, the Internet of Everything, ontological approaches, artificial intelligence, green energy policies, and enterprise architecture are the main technologies to rely on in order to make the next industrial transition. In conclusion, the importance of developing and implementing an IT system architecture inside an enterprise planning a transition is the most promising solution for dealing with the issues arising from the transformation.

In work [32], the authors strictly relate the IT/OT convergence movement with the Industry 5.0 transition. The architecture provided by the authors enables fast, reliable, and secure operational OT data exchange towards the IT layer. Their project relies on a middleware layer on the boundary between the OT layer and the above IT, Cloud, or SCADA layer. In the OT layer, the massages are homogenized with respect to original protocols and sent via the OPC UA Pub/Sub protocol. The gateway passes the OPC UA data to the upper layer equipped with Apache Kafka, a general-purpose, fast, and reliable MOM.

Work [33] analyzes existing digital platforms related to the management of enterprises in the upcoming I5.0 era, then proposes a conceptual model of a platform used for advanced adaptive company management. The paper takes into account all the entities in the industrial ecosystem and considers it as a system-of-system populated of smart objects capable of coordinating and resolving conflicts through multi-party negotiations. Authors provide the classification of services that the envisaged digital platform should provide in order to cope with the I5.0 transition in many industrial sectors.

Work [34] provides another attempt to enable the I5.0 transition in the manufacturing sector through an IT architecture that allows the automation of industrial processes. For the authors, a legacy from Industry 4.0, or the CPS, is a key element in the next I5.0 scenario, so populated of many Smart Cyber-Physical Systems (SCPS). The paper proposes a heterogeneous architecture where different types of data can coexist, be integrated, and used in complex business processes. The SCPS proposed is capable of partitioning a given computing grid into multiple voltage-frequency domains and minimizing the energy consumption of each of them by assigning a threshold voltage. The system also takes into account the disturbances that can arise in production environments where a high concentration of machines with very different voltages coexist. The system analyzes, predicts, and corrects disturbances and alterations in communication. The platform responds to the integration needs of the modern manufacturing industry, populated from data exchanged among sensors with different priority and electrical values. It is also capable of managing energy-saving optimally and automatically, in favor of sustainability and cost-saving.

The authors of work [35] propose an innovation management framework labeled Absolute Innovation Management (AIM) to help organizations in facing problems arising from the implementation of IoT and Industry 5.0 concepts inside their production plants. The authors state that the framework links the innovation ecosystem with the corporate strategy, and it helps in progress implementation and economic growth, and making innovation more user/human-centered. Since Industry 5.0 is shifting the paradigm from digital manufacturing to creating a modern digital society, a framework is implementable if it creates values for customers and users, besides the business for the company. The innovation ecosystem presented in this work is not limited to technical aspects, but it spans non-technical areas and spreads the responsibility of innovation all across the organization. Furthermore, the innovation should cross the firm's boundaries, taking care also of inter- and intra-firm networking and integration. Downstream of all these considerations, the authors state that a transversal framework that addresses the development of all company departments is necessary to face the transition to industry 5.0 and project both the company and the workers within this new philosophy of progress social.

Work [36] provides an excellent analysis of the employment of Mobile Crowd Sensing (MCS) in modern scenarios involving smart communities. The authors identify Industry 5.0 as one of the main investigation areas for the application of modern MCS techniques. The work analyzes the methodologies to build a knowledge base of relationships and ties between users and communities of users, to predict more accurately the dynamics of mobility and sociability. Taking into account the I5.0 requirements about human centrality and safety, the MCS paradigm could be helpful to analyze the social dynamics within the workplace and in the application of the safety rules in force. Furthermore, the MCS can facilitate tracking inside buildings, difficult to achieve with only satellite technology, which would enable the prevention of hazardous situations such as workers near dangerous work machines. In conclusion, this work shows that with a good number of active contributors, the MCS could help to achieve many goals that I5.0 aims.

The authors of work [37] analyze the role of Federated Learning (FL) in modern industry and in relation to IIoT devices' communication and monitoring. They envision an industrial scenario in which the FL algorithms coordinate multiple IIoT devices and machines to perform AI training at the network edge while protecting user privacy and confidential business information. In particular, the work focuses on the use case of IIoT mobile crowdsensing. According to the authors, in the next I5.0 scenario, it will be common the employment of MCS in the production environments where humans and machines coexist. Herein, mostly for the human contributions, a problem of privacy arises. The data exchanged must be protected and their processing of information on a remote server must be secure. The authors introduce an FL-based mobile crowdsensing scheme focused on data privacy. This approach provided a secure gradient aggregation algorithm by integrating holomorphic encryption with secret sharing, which prevents the central server from guessing the decryption result before operating aggregation. For the authors, this is a valid example of making use of FL to enable the secure sharing of information among large-scale participation, as in the industrial crowdsensing scenario.

Work [38] is a survey on the enabling technologies of modern smart communities such as those envisaged by Society 5.0. This group also includes modern industrial technologies such as IIoT and its combination with crowdsourcing. The future crowdsourcing campaigns will involve data from humans and environmental sensors. In order for a smart community to benefit from the data coming from heterogeneous systems, we need to build a crowdsensing platform that collects data from different application domains. The authors outline and describe the crowdsourcing approaches in the domains of Vehicular Crowdsourcing (VCS) and Intelligent Transportation Systems (ITS).

## 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.

In the first part of this chapter, 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.

Table 2.1 Literature Review Comparison

| | Legacy Support | Scalable | Fault Tolerant | Cyber Security | Human Centric | Reliable | Sustainable |
|---|---|---|---|---|---|---|---|
| SIRDAM Ecosystem | ✓ | ✓ | ✓ | (✓) | (✓) | (✓) | (✓) |
| [14] | (✓) | ✓ | x | x | x | x | x |
| [15] | x | ✓ | ✓ | x | x | (✓) | x |
| [16] | ✓ | (✓) | x | x | x | x | x |
| [19] | x | x | (✓) | ✓ | x | (✓) | x |
| [20] | ✓ | ✓ | x | x | x | (✓) | x |
| [26] | ✓ | (✓) | x | x | x | x | x |
| [27] | ✓ | x | x | x | x | x | x |
| [29] | (✓) | x | x | x | x | x | x |
| [30] | (✓) | x | x | x | x | x | x |
| [32] | ✓ | x | (✓) | (✓) | x | (✓) | x |
| [34] | (✓) | x | x | (✓) | ✓ | x | x |
| [35] | (✓) | x | (✓) | x | (✓) | (✓) | (✓) |
| [36] | x | x | x | x | ✓ | x | (✓) |
| [37] | x | ✓ | (✓) | x | x | x | (✓) |

The second part of this chapter focused on the works in the literature which take into account the Industry 5.0 movement, and its related concepts. In Table 2.1, we merged all the contributions similar to our work and their main characteristic. The columns represent the KPIs of IT/OT convergence from the point of view of the Industry 4.0 transition and the main pillars of Industry 5.0 philosophy. Each line describes the characteristics of related work, the first of which is our framework (namely *SIRDAM Ecosystem*) presented in the next chapters of this thesis. Depending on the level of support offered by the proposal, aspects may be marked as fully addressed (✓), partially addressed ((✓)), or not addressed (x) respectively.

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 ([19], [27], [28]), while others specifically focus on the implementation of more general-purpose Digital Twins ([14], [15]).

As regards the support of existing communication protocols, the works [15] and [19] do not provide any compliance with the legacy protocols adopted in the production sites. Other works envisage some form of integration, i.e., [14] through the use of Kafka, [29] suggesting connecting TSN and OPC UA, and [30] proposing different configurations of the OPC UA protocol. From this point of view, the ones proposing a full support for legacy protocols are [16], [20], [26], and [27]. Specifically, [16]provides an IoT hub, the architecture introduced in [20] supports the pluggability between OPC UA and other Modbus-like protocols, while [26] and [27] provide, respectively, a layer of interoperability between different protocols and semantic modeling of resources at the OT level.

The platforms introduced in the works [14], [15], and [20] address scalability. [14] and [15] use natively scalable tools such as Kafka and [20] provides for the automatic configuration of OPC UA systems based on the number of controlled PLCs. Some scalability aspects are covered in [16], with the support of the Cloud, and in [26].

Robust fault tolerance is implemented by [15] through the use of intermediate topics in Kafka Stream. However, [19] provides a certain degree of fault tolerance in its ORKAN test architecture, although the implementation is not provided.

[19] 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.

As we can see from Table 2.1, few works address all three pillars of the Industry 5.0 movement, described by the last three columns, namely *Human Centric*, *Reliable*, *Sustainable*. Most of the platforms used for IT/OT convergence have a good reliability from an IT point of view [15, 19, 20, 32, 34], but few have no mechanism to ensure active participation of community and employees in the business development process, except for [34–36]. Only three papers [35–37] examine the corporate sustainability objective that the new platforms should pursue in relation to the optimization of resources' usage and the lowering of the enterprise climate footprint.

The literature analysis reveals a lack of platforms that implement both the needs of Industry 4.0, implementing safely and correctly the IT/OT convergence in production plants, and the objectives of the Industry 5.0 movement. Our platform is the only one that addresses the challenges of IT/OT convergence. Furthermore, thanks to our experiments in different IT domains, the SIRDAM Ecosystem applies its modularity and expandability to the aspects proposed by the modern Industry 5.0 transition.

# Chapter 3

# Enabling the Synergic Integration of Industry and Society

The industrial revolution we are experiencing these years will lead us to the vision advocated by the Industry 5.0 movement and many industrial sectors will become an integral part of society. In this chapter, we introduce the overall design of the platform architecture we conceived to overcome some industrial integration issues and to enable some main features devised by the Industry 5.0 transition. The complexity and vastity of our platform demands the high-level insight that this chapter provides. We will use the following sections to present not only all the components of the *SIRDAM ecosystem* but also the connections with the side topics to the strictly industrial one, namely the cloud computing and crowdsensing expansions. Figure 3.1 is a panoramic view of all macro components of our proposal.

The core part of our platform is the multi-layered architecture to manage data gathering at floor level and provide low latency to the spread of information among upper layers. The use of modern tools from the IT world makes the production environment reliable and fault-tolerant, enabling the safe sharing of information within the working group or with external third parties and partners.

Modern technology requires more and more specialized work, to allow a human-centric vision it is mandatory that the tools used in the production sites interface with their human collaborators. The human-centric vision will build a win-to-win relationship between industry and society, because of the services provided to customers and of the support to the world of work. A platform acting in such a scenario must drive the workers in the use and understanding not only of the tools they are handling but also of technology internal functioning and its potentials, such as in the case of big data and artificial intelligence. We believe the use of crowdsensing and crowdsourcing techniques facilitates human centrality in

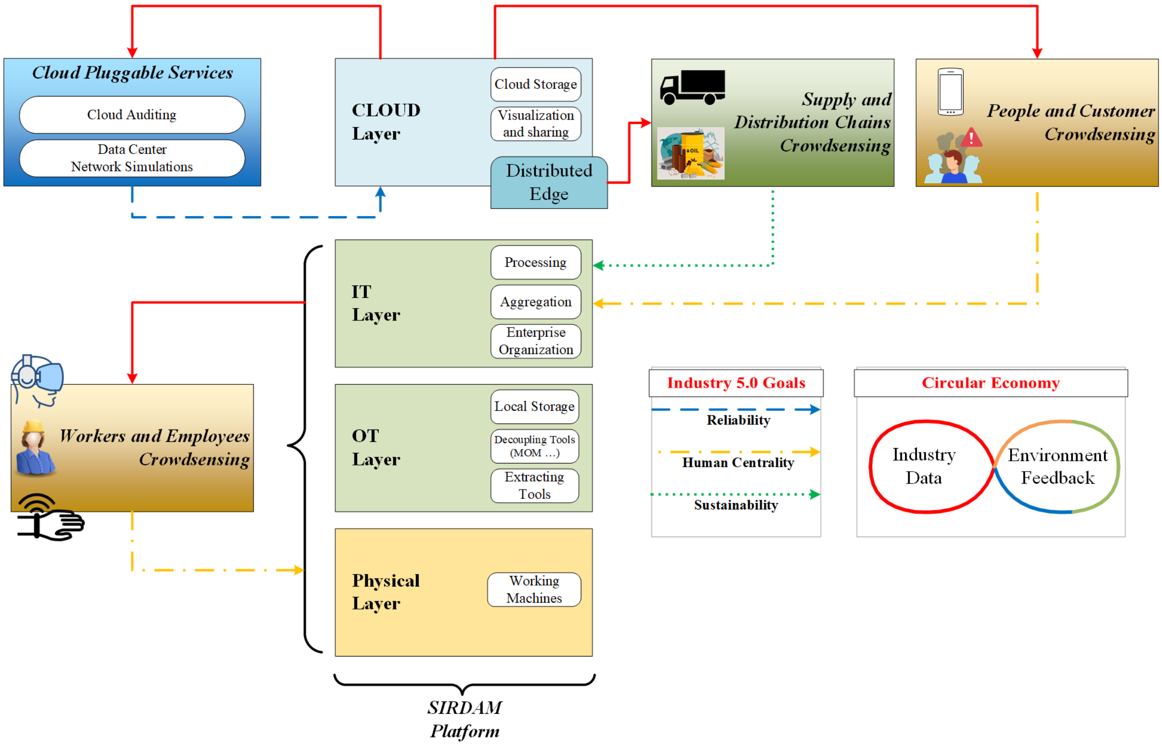



Fig. 3.1 Platform overall architecture

the industrial evolutionary process. This feature is essential for building successful strategies as it sets long-term goals leading to an improvement of society as a whole.

Our studies and experiments on cloud computing address sustainability from the point of view of computational resources' optimization. The cloud enables heavy computationally analysis that cannot be carried out locally, or that requires a centralized point to receive all the information coming from distributed locations. We believe that an I5.0 transition management platform must include techniques to administer its cloud infrastructures, such as simulators to estimate and optimize the use of resources and auditing tools to choose objectively the best solution based on performance requirements.

In the following sections, we will cover in more detail each of the features presented so far.

## 3.1 Dealing with IT/OT convergence

Figure 3.2 depicts a schematic representation of all the layers of the core part of our platform. We have split our approach to the integration problem into different levels to break up the tasks related to each tier and to overcome the issue arising from the layer convergence. The

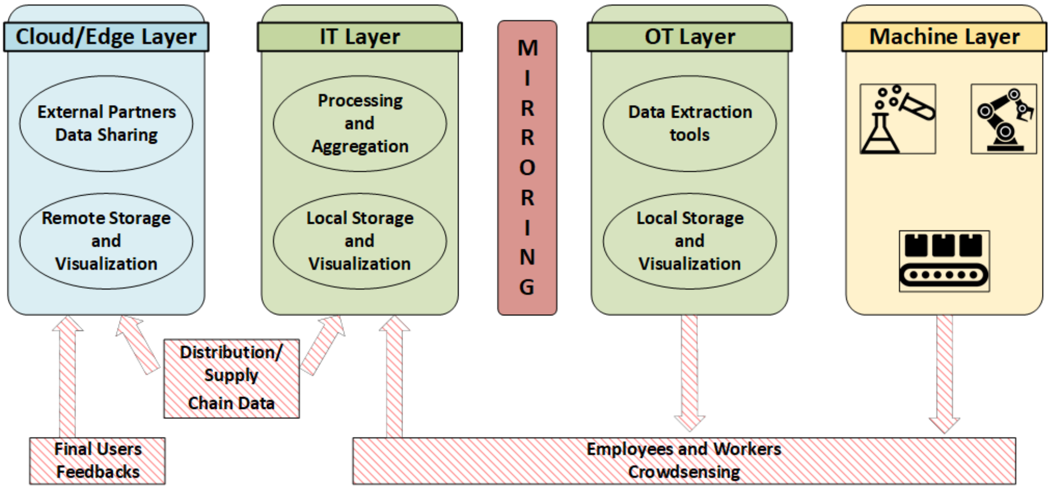



Fig. 3.2 Industry 5.0-enabled data gathering platform

*Machine layer* is the lowest, and it involves all the assets located in the production sites. The immediately above layer is the *OT layer*, having the duty to gather data from work machines in real-time and to send the information to the upper layers. These two layers are the most sensitive in terms of performance and safety/security. The low latency must be guaranteed because the analysis and the decisions that will be taken at the higher levels depend on the reliability of the data collected at this level. Besides the high-performance tools we added to the OT layer, in the Machine-OT layers we foresee security mechanisms that detect malicious intrusions dangerous for humans and machinery. We conducted research about the correctness of the protocols employed at the shop-floor layer, to assure a correct behavior of the machines. We want to ensure secure access to the information stored and exchanged at the shop-floor level, and also the implementation correctness of the low-level protocols, whose wrong behavior could compromise the safety of workers and machines. For the controlled access, we foresee some ACL mechanisms while for the correctness of the implementations we have created a fuzzing tool that allows analyzing the responses of the machines to certain stimuli. We need such an approach because often on the shop floors the companies employ private and not open implementations of the protocols, so it is more difficult to find errors in respect to the open-source implementations.

The *Mirroring layer* contributes to the reliability of the platform by providing customizable support for the replication of information collected at the lowest levels from the work machines. The connection among IT and OT layers is our integration middleware to tackle the convergence issues. View and query tools are common in the upper layers, but we add these facilities also in the OT layer because we think information sharing is a key part of the modernization of the world of work foreseen by the I5.0 transition. There is a need to share data coming from workshops and production sites with the humans at work, even at a low level, in order to drive them in the use and knowledge of new technologies and improve their working place.

The *IT layer* is what enterprises need to manage all the assets, resources, and employees. Here the customers can add the logic to process raw data coming from underlying layers and use the data coming from external stimuli. The IT layer, having a complete view of business processes, can receive feedback and data from the supply and distribution chains to optimize the processes of provisioning and distributing the goods in order to reduce waste and emissions.

The conceptually higher level is the *Cloud layer*. Currently, most SMEs need to connect different distributed sites and convey all the information in a central point, and consequently to make comparisons among different products or plants and selectively share information with the different partners. The Cloud layer, therefore, allows customers to have a high-level view, used by managerial departments to make long-term decisions. Furthermore, the cloud layer allows for greater scalability in terms of long-term information storage and plants management.

Chapter 4 will provide an in-depth view of all the layers of the platform.

## 3.2 Crowdsensing for human-centric vision

Corroborated by works found in literature, we believe that crowdsensing can play a fundamental role in achieving the goals set by Industry 5.0 and Society 5.0. One of the main features of modern Mobile Crowd Sensing (MCS) is the ability to fuse the myriad of information coming from the environment and from sensors in favor of humankind [39]. This aspect makes the MCS technology particularly suitable for use within the manufacturing sector, and in industrial productions in general, to enable the vision of human centrality desired by I5.0.

We want to embody an MCS platform in our integration middleware in order to exploit the workers' data on the production sites and prevent hazardous situations. Through the use of edge computing, our solution could fit all situations that need tracking inside buildings or proximity alerting. Our experiments involved the signaling of crowded areas in order to contain the COVID-19 pandemic [40], but the principles of our work are applicable in each situation needing proximity tracking.

As Figure 3.3 suggests, our tracking model can exploit crowdsensing data passively from wearables, such as smart glasses, smart clothes, smartwatches, and smart bracelets. These environmental and contextual knowledge bases can be interwoven with information coming from data collection campaigns at which users can directly participate providing

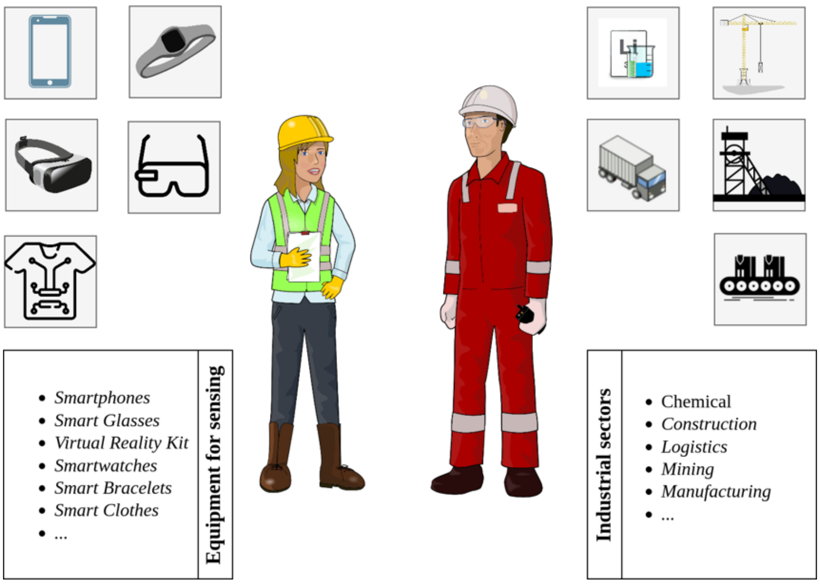



Fig. 3.3 Crowdsensing applications in modern industry

the data gathered by the sensors of their smartphones. We built the crowdsensing extension to our platform relying on the use case explained in Chapter 5. The proposed scenario treats the signaling of dangerous situations to users on the bases of the proximity with other people, even providing suggestions for safer places. Industrial sectors, such as manufacturing, chemical, logistics, mining, need a crowdsensing approach for the prevention of hazardous situations and for the realization of the I5.0 human-centric vision. The integration of MCS technology in the platform used for the transition to I5.0 enables the reskilling and upskilling of the employees, driving them in the understanding of new technologies that emerged from the Industry 4.0 movement and employed in the I5.0 transition.

## 3.3 Cloud Computing for industrial reliability

The Cloud is currently a widespread technology among SMEs in Europe [41]. The benefits deriving from the use of this technology are manifold. The connection to Cloud resources is a cost-saving choice in respect of purchasing and maintaining your own IT infrastructure. It gives scalability to the system to overcome clients fluctuations or company expansions and consequently provide the right amount of resources. With Cloud Computing, enterprises can adopt forms of remote collaborative work and improve their relationship with customers. Cloud and edge computing are some main enablers of the information sharing that the I5.0 needs to become reality. Exploiting the isolation and virtualization of resources enables complex tests, simulations, and analyzes relying on on-demand resources, therefore to the

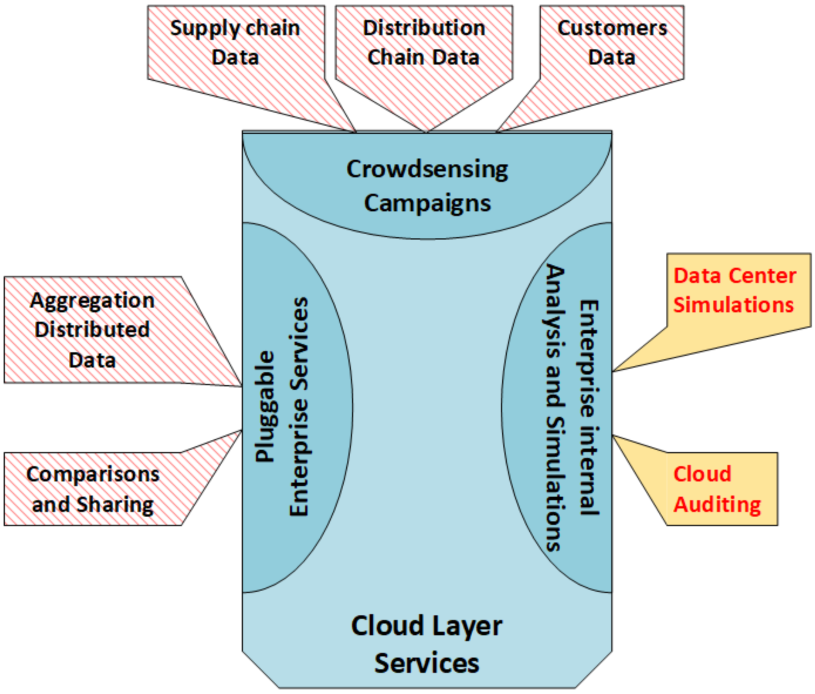



Fig. 3.4 Cloud computing enterprise services

benefit of corporate sustainability. Services provided by a distributed cloud or network edge can help improve supply and supply chains. Furthermore, being centralized with respect to the many distributed customers of our platform, it is the most suitable layer where it brings together all the data. Information can flow into the cloud layer from customers that spontaneously send feedback or from the collections of crowdsensing campaign contributions, integrating the users in the design process of the product itself.

During the design of our architecture, we forecast a pluggable configuration to connect the enterprise IT core to cloud services. As you can see from Figure 3.4, although our architecture has a wide range of applications, to demonstrate its applicability we focused on two use cases, the cloud provider auditing and the in-company simulations to estimate and optimize the use of their own resources in the cloud. The first cloud service helps companies in choosing the right cloud provider based on network performance parameters, i.e., latency, and bandwidth. Our tool audits the cloud providers showing their performance to the customers that can choose the best solution and compare it with other deployments. This tool is useful also from a cloud provider perspective because it allows the discovery of bottlenecks in the network infrastructure, and the setting of the assets' replication degree accordingly.

In case a company needs its own data center, we designed a cloud simulator allowing network architects to find the best deployment fitting with the desired scenario. This tool optimizes not only the usage of the machines but also the emissions due to the use of the technology itself. In addition, accurate simulations of traffic peaks can help in organizing the replication and distribution factor of the deployment and in avoiding unexpected service interruptions.

## 3.4 Cross-domain cybersecurity

We broke down corporate boundaries and built an inclusive architecture that puts the interests of workers and stakeholders at the center of modern industrial transition. In this scenario, information security is of paramount importance as it would not be possible to enable any of the innovations envisaged by Industry 5.0 if the latter endangered the users themselves and their privacy rights.

From the IT point of view, we consider cybersecurity a common feature of each level of our platform. We used common IT security technologies to protect the sharing and expo of data outside the corporate walls. Communications between remote sites and distributed architecture components are encrypted and authenticated. Access to the datasets distributed in the various levels is controlled through authentication mechanisms employing credentials or security keys.

As for performance, we believe that the most fragile level from the point of view of safety is the OT layer, where the work machines' communication protocols reside. The reliability of the company processes and the safety of the workers depend on the correctness of the machines operation and the plant security. For this reason, we focused our efforts and tests in this direction. Our solution provides for the low-level analysis of the M2M and B2M communication, therefore of the protocols for interaction with machines. This protection strategy mitigates the problems deriving from unusual machines behavior improving corporate safety and protecting against attacks from malicious agents that could be within the corporate security perimeter.

Given the attention that the fuzzing technique of the industrial network protocols is increasing [42, 43], we opted to add the support for fuzzing in our platform. Our main goal is to check the correctness of the many implementations of industrial low-level communication protocols specifications. In particular, our use case involves the OPC UA protocol, an industrial standard that is gaining a lot of popularity among enterprises that need for monitoring and controlling work machines [44, 45].

As you can see from Figure 3.5, our cybersecurity extension allows the analysis of the responses of an OPC UA server used in a production site. Our approach verifies if the server under test has bugs, which on closer examination turn out to be real vulnerabilities, which

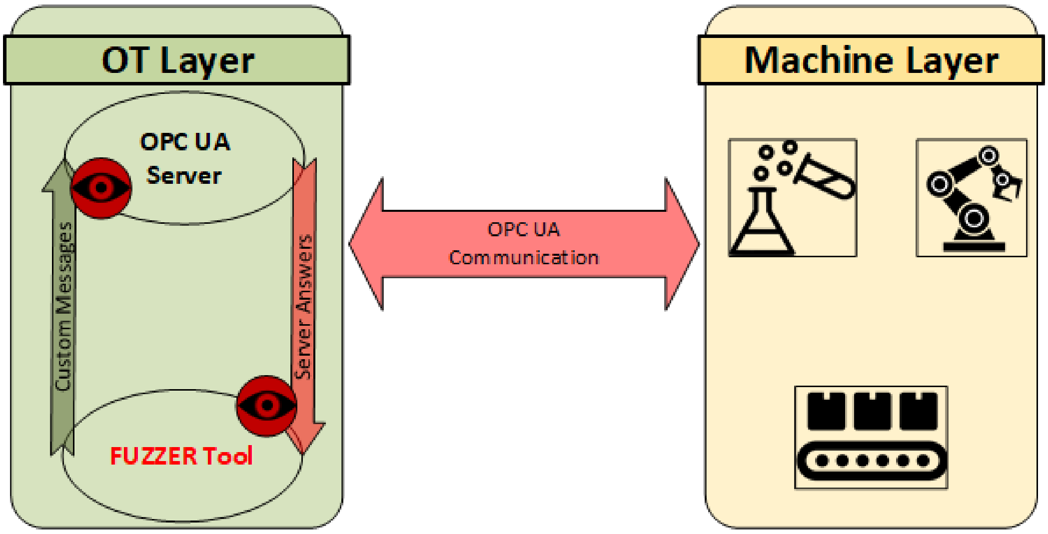



Fig. 3.5 Cybersecurity at lower layers

can bring the system and the running processes in an unexpected state, affecting it with unacceptable delays or even interruption of the service. Following the fuzz testing technique, we automatically send mutated custom and compliant packages from a fake client to our server implementation. Then, we inspect both the answers and the behavior of the server under test. If it returns an exception in responses, we check for implementation correctness, if it is unresponsive maybe it could be crashed or extremely slowed down. For example, after a test session, to see whether a server is responsive we can ping some services and see if a response comes back, otherwise, the server is down, and we need to analyze the previous test session to understand the crash.

We believe that our security plugin applied to the data gathering layer is paramount to prevent accidents on the shop floor. The detected bugs are immediately forwarded to the IT departments for a more careful analysis of the urgency to find a patch. Our tool can be extracted from the integration platform and can work in standalone mode testing any implementation of the protocol under consideration.

# Chapter 4

# A Convergent-Enabled Industrial Platform

Our multi-layer solution promises to overcome the IT and OT layers separation, the main barrier for companies that intend to follow the digital I4.0 transformation, and therefore to pursue the goals of the Industry 5.0 transition. Our SIRDAM platform covers the IT/OT gap and enables convergence, fulfilling many needs that could enable the future I5.0 scenarios. This section provides the motivations that led us to design SIRDAM, the core of our architecture for integration and implementation of the next industrial transition. We developed a prototype to support our findings with many experiments conducted in different scenarios.

## 4.1 Motivating scenario

The present work was born from a collaborative research project carried out 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.

We drafted some guidelines that are the result of the collaboration among researchers and industry experts [2, 46]. We followed this handbook during the design and implementation of the SIRDAM platform [47], aiming to allow the convergence of the factory OT and the IT layers as a way to enable the Industry 4.0 transition.

The platform aims to address some challenges raised by IIRA and OPC-UA in providing data gathering and integration at several levels. 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.

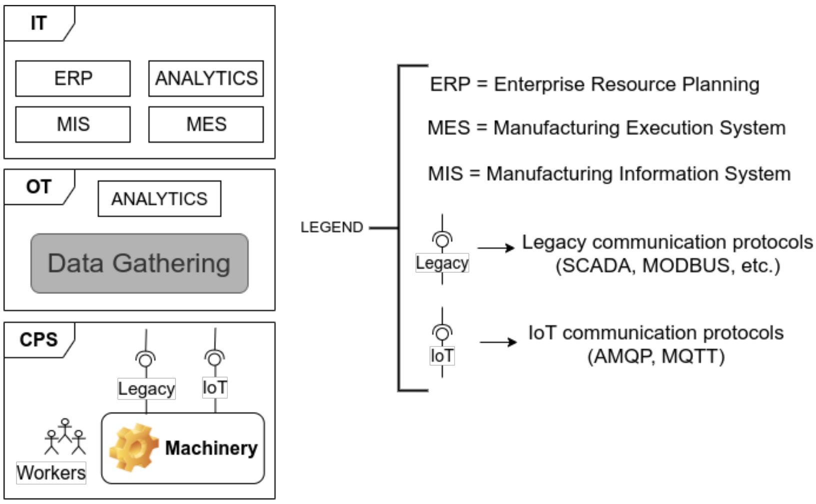



Fig. 4.1 Technological layers in a manufacturing factory

In Figure 4.1, 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 much 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.

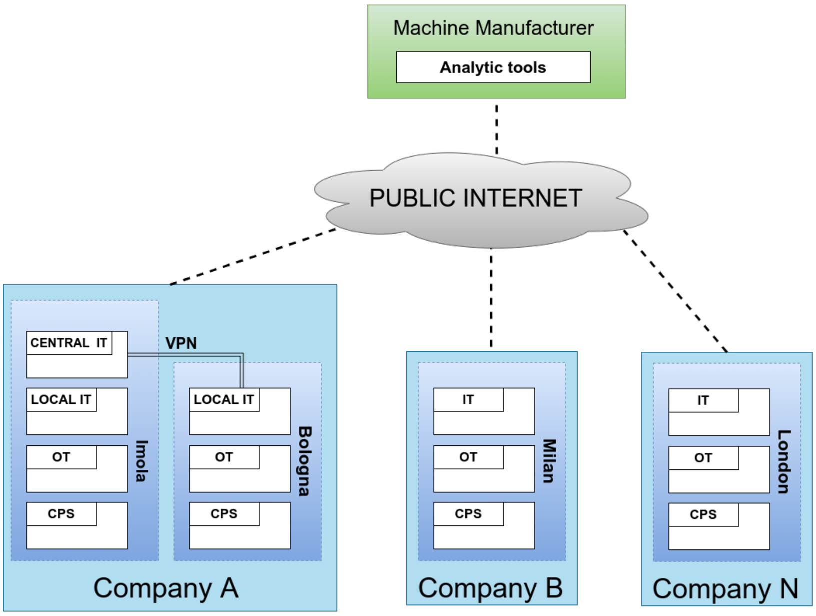



Fig. 4.2 Stakeholders reference scenario

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. Figure 4.2 depicts the entire scenario stakeholders and the technology stacks deployed in each premise.

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, 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).

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 the OT layer and making them safely accessible to stakeholders is a crucial step that poses the basis for the OT/IT convergence in industrial settings. In the following Sections, we will see how we conceived our IT/OT convergent architecture and the lessons learned from its employment in a realistic test bench.

## 4.2 Platform design

This section introduces our **Support Infrastructure for Reliable Data Acquisition and Management in Industry 4.0**(**SIRDAM4.0**). Motivated by the scenario presented in the previous section and driven by the collaborations with important manufacturing enterprises, we developed a prototype of the architectural model shown in Figure 4.3. SIRDAM 4.0 architecture reflects the physical separation of OT and IT layers enforced in most production sites and addresses their integration at data level. We designed the SIRDAM platform to meet the requirements demanded by IT/OT convergence and to provide the main features sought from the various stakeholders of the I4.0 transition.

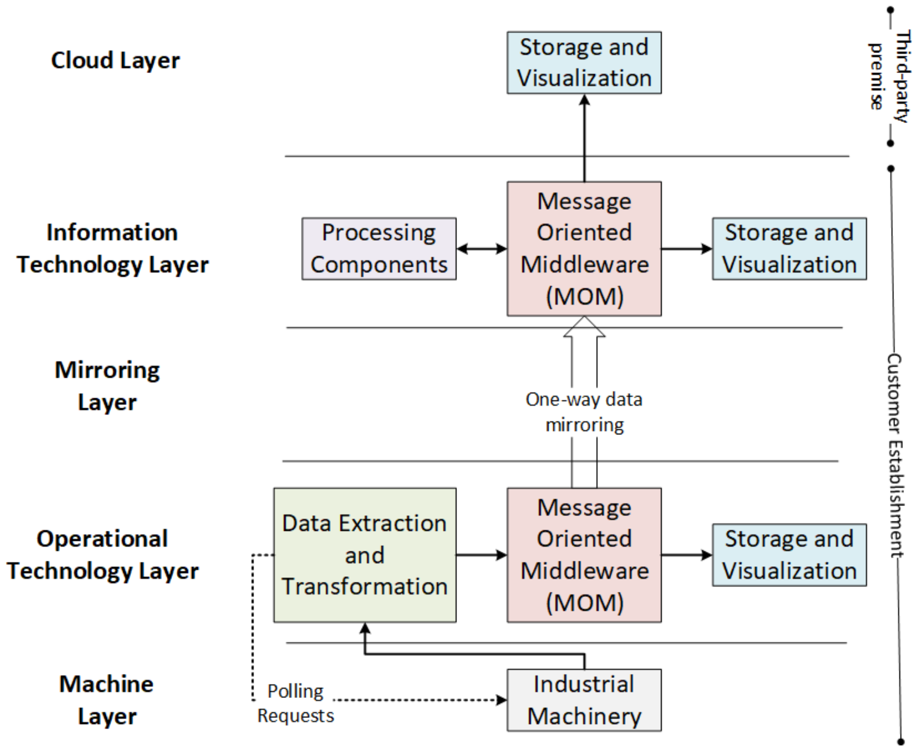



Fig. 4.3 SIRDAM overall architecture schema

The SIRDAM platform yields the following features:

- seamlessly monitoring of production site asset,
- custom data processing, long-term storage, and user-friendly visualization of information and issues,
- mirroring of the operational data to enhance the reliability,
- secure and selective access to information from different layers, for stakeholders and third-party companies.

About the last point, we provided SIRDAM with this feature because both international specifications and guidelines point out the need for data sharing not only among partner stakeholders but also with outside companies (e.g., the machine vendor).

For the SIRDAM platform devising, we were inspired also by the concepts found in the main standardization activities addressing the IT/OT convergence inside the I4.0 movement, such as RAMI 4.0, IIRA, and OPC UA. 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 economic 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 [48–50], 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 in 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 must 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.

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 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.

## 4.3 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 [6].

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).

The main one is Apache Kafka Broker, a message broker instance that can handle a data ingestion rate as high as 420K messages/second [51] 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. We also borrowed other minor tools from the Confluent suite, such as the connector used to decouple the broker from the data sources/sinks which it is connected to.

### 4.3.1 Machine and OT layers

Figure 4.4 depicts a schematic view of machinery populating the *Machine layer* and the software components implementing the functionality of the OT layer.

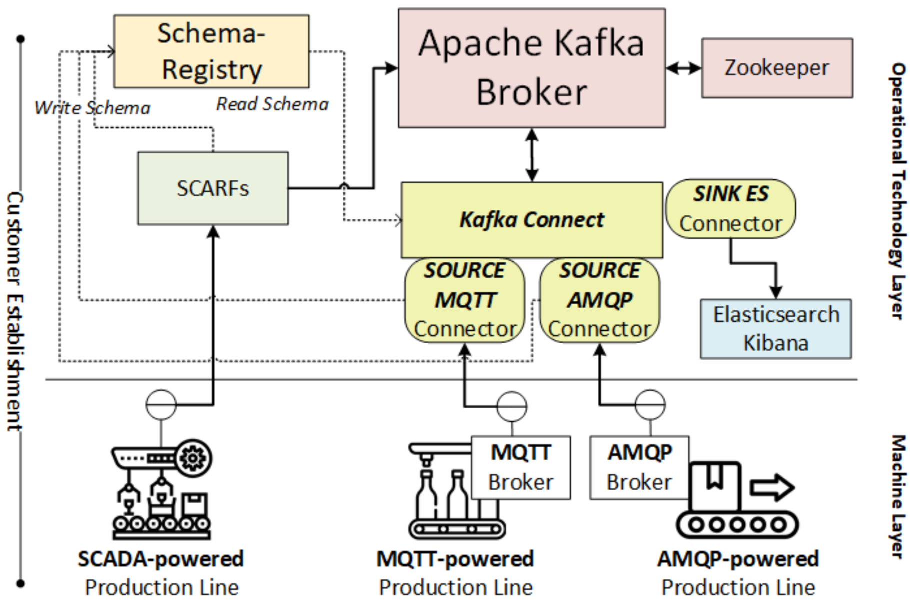



Fig. 4.4 SIRDAM Operation Technology layer

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 are hard-coded in the firmware of manufacturing machines. Although we tested MODBUS TCP [1] (see Section 4.4), other protocols such as Profibus [2], CANOpen [3],

[1] https://modbus.org/
[2] https://www.profibus.com/
[3] https://www.canopensolutions.com/

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. We gave our platform communication and interoperability capabilities decoupling this layer, where data are gathered from the *Machine layer*, where data are produced. The decoupling allows the homogenizing of the data produced at shop floor level in a unique data stream flowing up in the software stack, notwithstanding different machine kinds and their different connection protocols. The communication between the OT and the machine layers is enriched with other components to preserve technologies already employed in the production plants. This avoids tho add complexity directly on the machine firmware, that vendors often tend to not change as the one in use is tested and robust.

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 4.4, 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 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[6], a data serialization framework arranging information in a compact binary format. SchemaRegistry stores the AVRO schemes for serialization

[4] https://www.rtautomation.com/technologies/devicenet/
[5] https://pymodbus.readthedocs.io/en/latest/index.html
[6] https://avro.apache.org/

and deserialization purposes. Eventually, SCARF instances send serialized data to Kafka broker.

- **IoT connectors**. Differently 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.

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.

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

### 4.3.2 Mirror, IT, and Cloud layers

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 customer company. This feature allows the definition of fine-grained configuration policies of the system and in terms of data protection. The Kafka MirrorMaker [7] is a stand-alone component that copies a subset of topics from OT to IT layer (see Figure 4.5). 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.

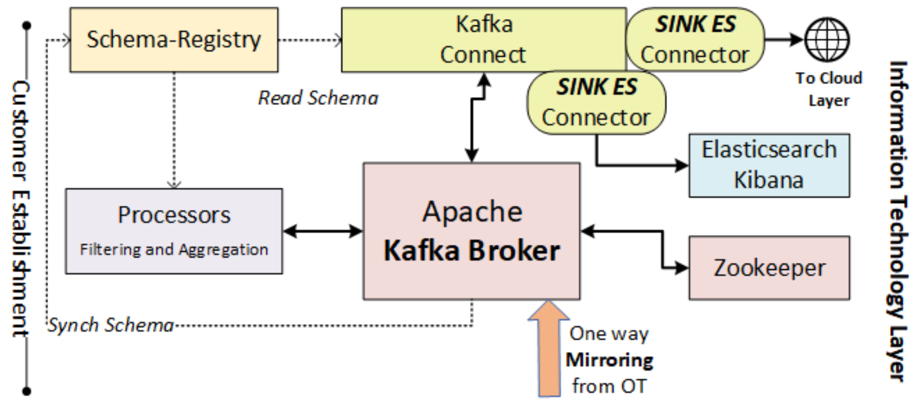



Fig. 4.5 SIRDAM Information Technology layer

The *IT layer* is populated with the same software modules as the OT layer, except for SCARFs, given that the IT layer does not have a direct physical connection with work 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 4.5, we added data processing modules using Kafka Processors, components able to aggregate and

[7] https://docs.confluent.io/3.2.2/multi-dc/mirrormaker.html

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.

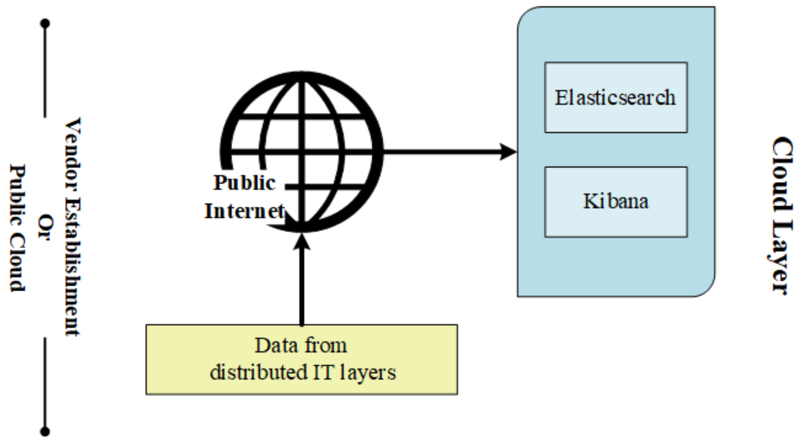



Fig. 4.6 SIRDAM Cloud layer

The main objective of the *Cloud layer* depicted in Figure 4.6 is aggregation and presentation of data coming from production sites. Data coming from all the customer establishments are going to be collected in the cloud environment, be it remote (public or hybrid) or local (private). 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-the-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. Customers here can perform advanced data analysis on the bunch of information coming from all the plants. Typically, this is the layer where ETL tools process data for advanced cross-customer and cross-plants comparisons and examinations. Machine vendors can deploy diagnostic, predictive maintenance, and other after-sales services to completely customize the customer experience. We put the same distribution of Elasticsearch and Kibana also in the Cloud layer because we believe in their ease of use and completeness in querying and viewing the information. We want to recall that any storage and analytical tool can be attached by deploying its connector at the Kafka broker, source of data.

## 4.4 Use Case: the manufacturing sector

In this section, we describe the arrangement of our testbed use case, analogous to the deployment of a typical SME having many production plants. Furthermore, we report the result of a thorough assessment of the platform with respect to the performance indicators previously discussed in Section 4.1: 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 [52], 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 [53], [54].

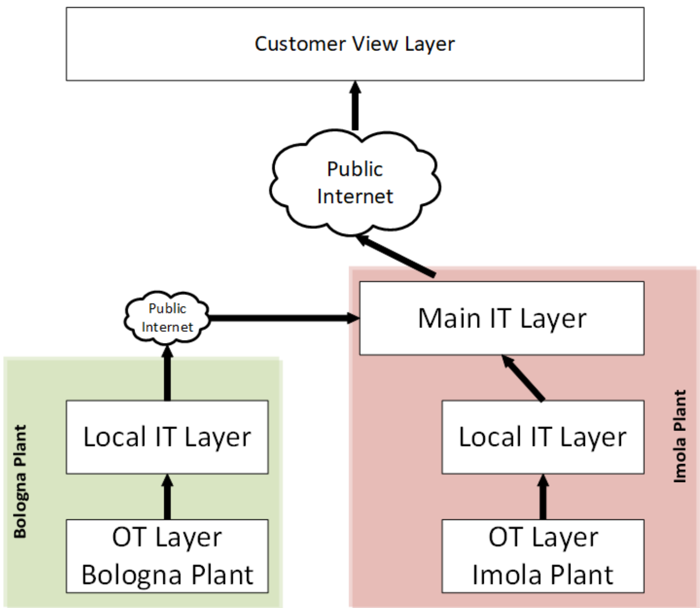



Fig. 4.7 SIRDAM testbed deployment

Figure 4.7 depicts our scenario considering 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 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 4.3 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. Figure 4.8 depicts all VM instances, microservices running within each VM, and the physical location of VMs.

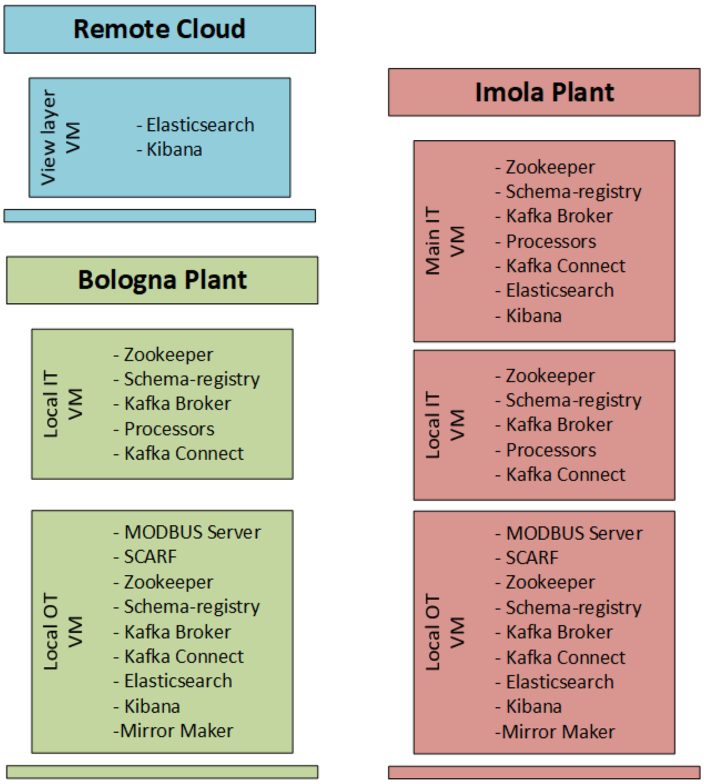



Fig. 4.8 SIRDAM virtual machine deployment

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 4.1, this layer includes IT software serving all company departments, such as those for managing staff or project cycles.

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.

In the following sections, we talk about a series of stress tests carried out on different layers of the platform. 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, 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 4.1.

### 4.4.1 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 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, 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.

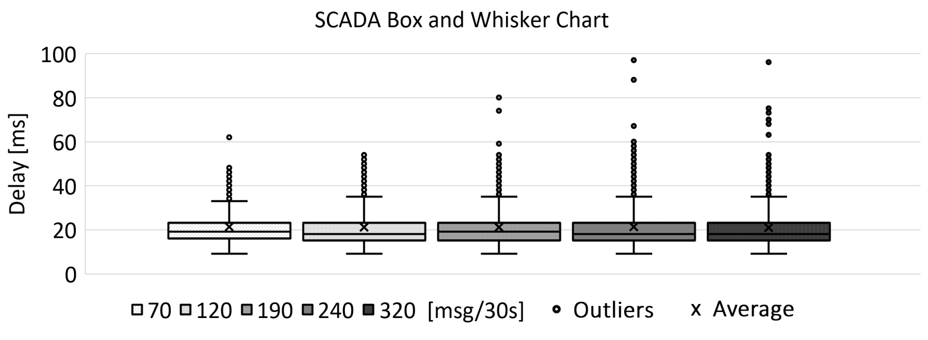



Fig. 4.9 SCADA delay test

Figure 4.9 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.

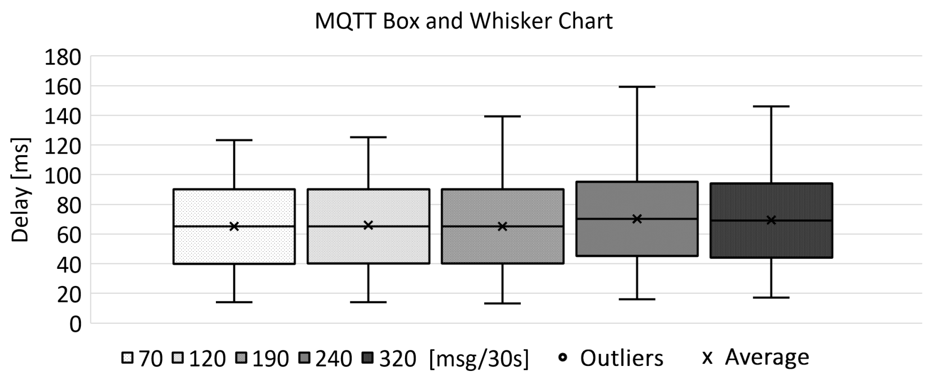



Fig. 4.10 MQTT delay test

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

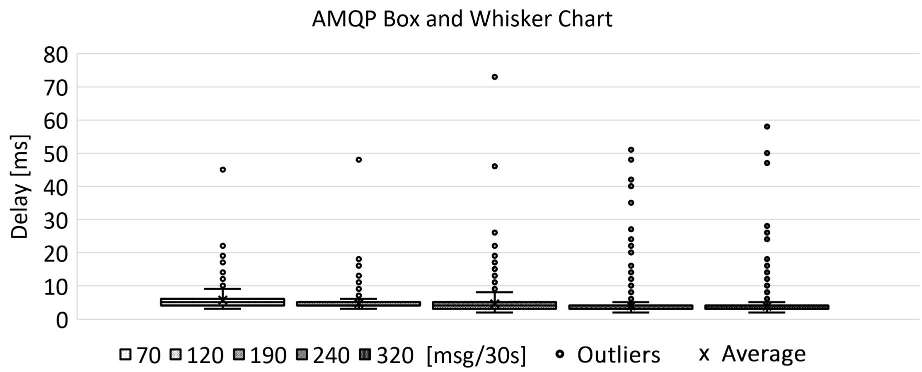



Fig. 4.11 AMQP delay test

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 4.10, in the case of Mosquitto the message delay is around 70 ms, while for RabbitMQ, depicted in Figure 4.11, 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.

### 4.4.2 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.

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.

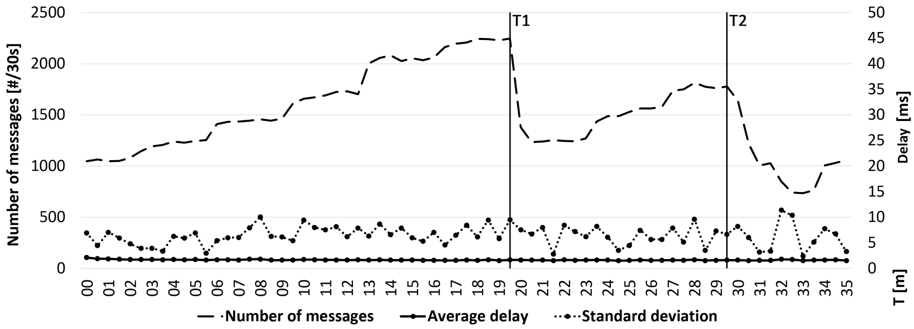



Fig. 4.12 Main IT layer delay test

In Figure 4.12, 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 4.13 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.

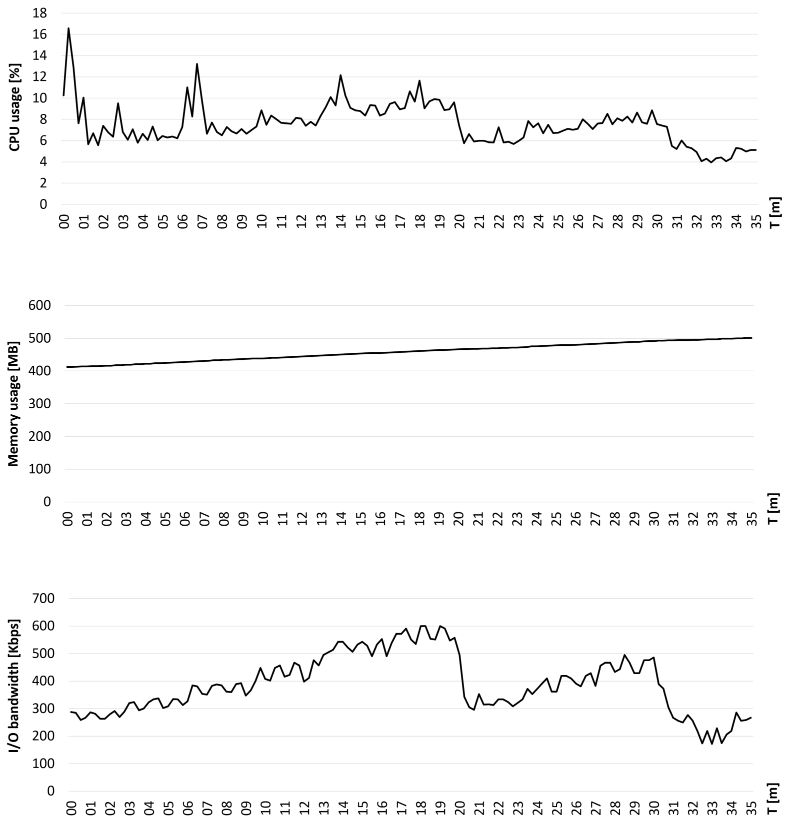



Fig. 4.13 Resources usage of the Kafka broker inside Main IT layer

### 4.4.3 Controlled access-list test

The SIRDAM platform allows data to flow from OT to IT and above layers. Should malicious actors gain access to the message brokers, they could steal precious data or 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, 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.

```
ksql> show topics;
 Kafka Topic
     bologna_capper-1-data-cycle
     bologna_capper-1-data-print
     bologna_capper-1-cooling
     bologna_capper-1-machine-state

     imola_capper-1-data-cycle
     imola_capper-1-data-print
     imola_capper-1-cooling
     imola_capper-1-machine-state
```

Listing 4.1 Imola Main IT broker topics

```
[PRODUCER]
.\kafka-console-producer.sh --broker-list <A.B.C.D>:9092
    --topic bologna_capper-1-data-cycle --producer.config
    sasl-producer-COOLING.properties
>Hello Message
WARN [Producer clientId=console-producer] Error while
    fetching metadata with correlation id 3 : {
    bologna_capper-1-data-cycle=TOPIC_AUTHORIZATION_FAILED
    }
ERROR [Producer clientId=console-producer] Topic
    authorization failed for topics [bologna_capper-1-data
    -cycle]
ERROR Error when sending message to topic bologna_capper
    -1-data-cycle with key: null, value: 4 bytes with
    error...Not authorized to access topics: [
    bologna_capper-1-data-cycle]

[KAFKA AUTHENTICATOR LOGGER]
INFO Principal = User:COOLING is Denied Operation =
    Describe from host=<A.B.C.D> on resource=Topic:LITERAL
    :bologna_capper-1-data-cycle
```

Listing 4.2 Kafka ACL violation

According to the scenario introduced in Figure 4.7, the Imola Central IT layer consumes data coming from the underlying OT layer and data coming from the Bologna OT layer. Listing 4.1 reports an excerpt of topics existing in the Imola Main IT 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 (see Listing 4.2).

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 which they are the intended recipient. Let us conclude by noting that this test demonstrates that the platform meets requirement $R3$ mentioned in Section 4.1.

### 4.4.4 Resilience 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, due to their convergence. We remind that Kafka brokers are implemented as containerized services running inside VMs. Faults can be of many kinds (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. The master replica is the one data producers and consumers rely on. 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. A broker fault implies that all its topics are not available anymore. In case a master replica is unavailable, the Kafka ecosystem automatically elects a new master copy (among the slave replicas) to which all consumers and producers

Table 4.1 Resilience test: brokers per topic

| **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] |

will be redirected. As soon as the crashed broker becomes available again, Kafka restores and synchronizes all the replicas.

With this test, we want to simulate the fault of a broker in a multi-broker Kafka deployment. To accomplish that, 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 4.1 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.

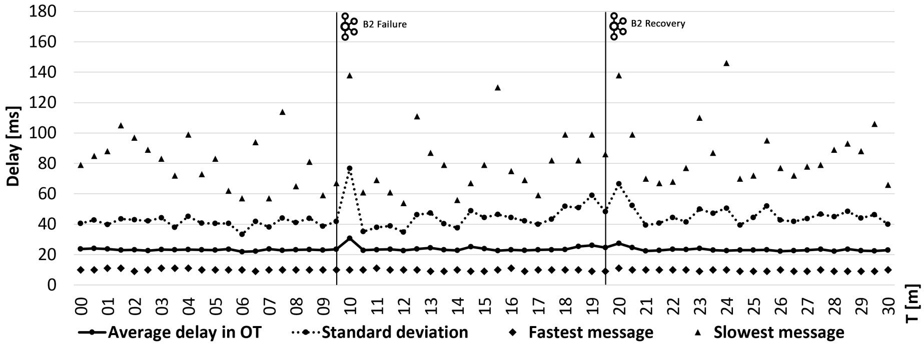



Fig. 4.14 Messages average and standard deviation during resilience tests

In Figure 4.14 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 *R*4 set out in Section 4.1.

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 4.15. 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.

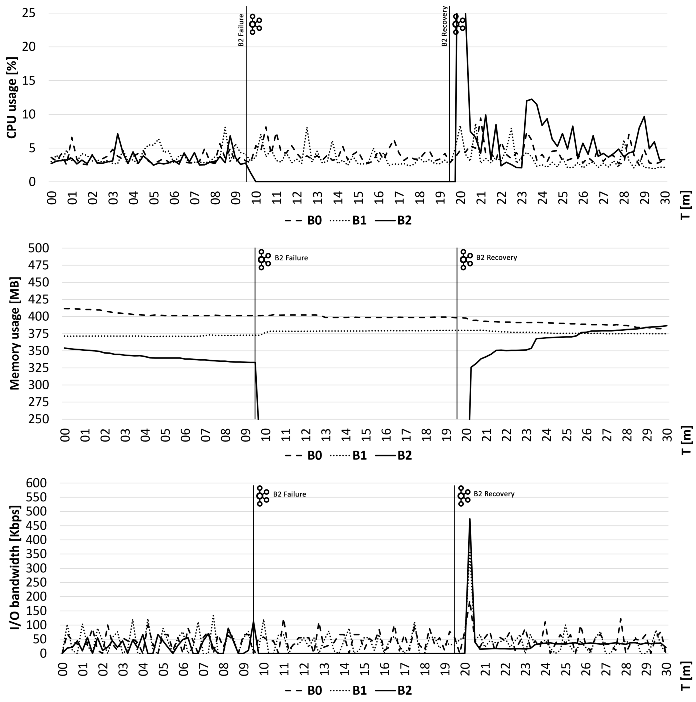



Fig. 4.15 Resource usage of the Docker containers running the three brokers

## 4.5 Lessons learned

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 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.

# Chapter 5

# Mobile Crowdsensing in Aid of Human Centrality

In order to enable the human centrality that the modern I5.0 transition is hoped for, we investigated participatory paths through the employment of **Mobile Crowd-Sensing** techniques. We believe that MCS can enable the massive participation of society in the new industrial transition, allowing to assist workers in the workplace and to include consumers and people in the corporate innovation loop. This Section introduces the design and the implementation of our edge-enabled MCS platform and shows the experimental results about the crowded places tracking scenario [40].

## 5.1 Motivating scenario

Smart cities put digital technology at the service of citizens to improve their quality of life and of city rulers to better govern their service provisioning decisions and to improve sustainability. The concept of a smart factory can benefit from MCS technologies as well, allowing the citizens' involvement in many industrial areas such as manufacturing, urban transport, water supply, waste treatment. Through the use of cloud computing, MCS technology can become pervasive even outside the company walls, reaching all stakeholders focused on the I5.0 transition and contributing not only to the human centrality peculiarity but also to an improvement in corporate sustainability. Actually, MCS can be used to track and improve supply and distribution chains, by way of contributions that people and machinery handling the goods can provide in real-time. MCS could improve the relationship with customers making them participate in the improvement of the product itself, proposing them crowdsensing campaigns that require user feedback, opinions, or active actions.

In any MCS system, the involvement of as many people as possible is crucial for the sensing campaign's success; a greater amount of information leads to more complete inferences, therefore to high-quality data sets. A largely used gimmick to increase and incentivize participation is through reward programs that loyalize and involve the users. In a highly distributed system, participants need to trust the crowdsensing campaigns provided by the platform to assure a good participants number and consequently the campaign's success. In this context, the blockchain federations are a promising technology to augment the trustiness among unknown peers acting in the same environment.

Taking advantage of the pervasive potential provided by **Multi-access Edge Computing** (**MEC**) and the security guarantees of blockchain technology, we developed an edge-enabled platform capable of leveraging the collection of data from mobile users and their analysis in order to identify the crowding degree of urban areas. Furthermore, to improve the effectiveness of users' rewards we propose an edge-enabled distributed ledger architecture to record the reward assignments among untrusted and unknown participants in a generic gamification system. We have used a distributed ledger system for the reward distribution, preventing in this way the stealing or faking of users' accomplishments resulting from an internal or external attack on the server itself.

## 5.2 Platform Design

In this section, we introduce an edge-enabled MCS platform targeted at supporting effective data collection/analysis and rewarding in critical scenarios, such as the recent **COVID-19 pandemic**. We exploited the MCS framework called **ParticipAct** [55] to develop an edge-enabled data collection/analysis module capable of evaluating the crowding degree of an area relying on an edge-based distributed ledger for managing users' rewarding.

ParticipAct is a comprehensive mobile crowdsensing platform developed the University of Bologna that provides us with the proper playground to gather crowd contributions and to test edge-based expansions in the crowdsensing domains of our interest. In ParticipAct the users have a sensing client application installed on their smartphones, and they send collected information to a centralized cloud server, based on targeted sensing campaigns created by researchers and platform administrators. ParticipAct developers followed best practice guidelines in developing the crowdsensing platform. Thanks to the use of *MoST* [56], a high-performance sensing module, the ParticipAct client application has a very low footprint when running on devices, and it requires few actions from the user to collect data, avoiding boring him with requests. At any time, the user can stop the sensing data sharing and can reject tasks and campaigns that are proposed to him. The secure protection of users' data and the mechanisms for administrators and clients authentication guarantee the integrity of the profiles and contributions collected. Any user can freely view its own contributions to keep tabs on everything he is sharing with the community. The database replication assures the availability of data. The server side is built on top of open-source technologies, and its modularity permits easy expansion and reusability for different purposes in several domains, such as smart cities and transportation, people tracking, GPS data gathering, and trajectories drawing. Authorized administrators can create campaigns on the server and can choose the users to whom to propose them, the geo notification and geo activation areas, and a time frame during which the campaign is available. The many actions of a single campaign are called *Tasks* and they must all be completed before a user can declare a campaign concluded, send the collected data to the server, and possibly receive a reward for the quantity and quality of the information provided.

### 5.2.1 Data collection and crowded area identification

The basic cloud-based deployment of ParticipAct lacks some characteristics that are needed to address our requirement of providing support for controlling the spread of a pandemia, such as in the recent COVID-19, by calculating the crowding degree of an area.

This requirement can be achieved with a massive data campaign supported by an effective data collection and user's rewarding management. The basic ParticipAct platform can involve many users around the country, but does not currently have the possibility to efficiently and effectively notify users in a specific geographic area with context-aware and location-aware updated information, being the cloud layer unaware of these data. The edge computing paradigm allows to overcome this limitation. Edge computing extends the cloud resources by offering networking, storing, computing capabilities and services distributed at the edge of the network closer to the final users. Edge nodes allow to reduce the load toward the server and the communication latency because they can perform local computation on data of interest and, most importantly, can promptly provide nearby users with location-aware information. Another limitation of the basic ParticipAct platform is the impossibility to federate different spontaneous systems spawned around the world. In this regard, we would like to achieve the purpose of sharing the user's scoreboard among ParticipAct federated servers deployed in distributed areas (ideally worldwide), maybe for different purposes. In this way, if a contributor should be involved in a crowdsensing campaign created by a server different from her usual one, she can contribute to the campaign and continue to acquire scores on the same profile, common among all federated servers. Thanks to the great customization feature of the platform, in [57] we started to formalize a way to federate different ParticipAct servers spawned around the world. Different servers, with possibly different purposes, can share the user rewards in order to enable the interaction of federated participants even if subscribed to different local servers. Users can benefit from the federation as they find their scores on any of the federated servers, regardless of the crowdsensing campaign they are participating in. Facilitating user's reward sharing among federated servers has a great beneficial impact on data collection. The catchment area participating in each campaign is increased making data collection more complete. In the case of a pandemia this is particularly important because it is possible to take into account the movement of users across different cities. When in a federated city, users can still contribute to the campaign and user's presence can be still considered to precisely evaluate the crowding degree of an area.

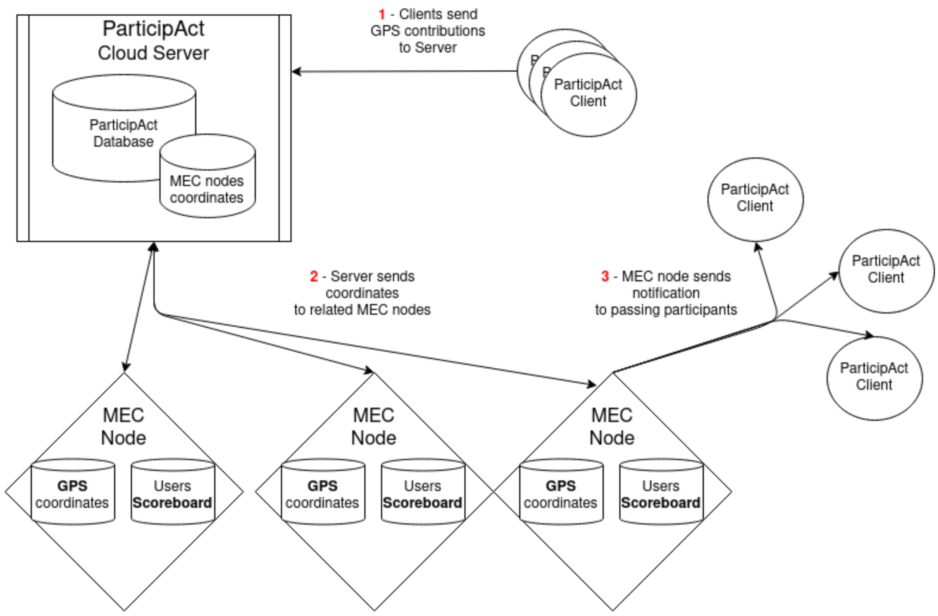



Fig. 5.1 Edge-based ParticipAct Architecture

Figure 5.1 shows the proposed edge-based architectural model of ParticipAct. In particular, each ParticipAct server relies on a pool of base MEC stations, as those defined by ETSI in the ETSI-MEC specification. We can assume that the exact position of the edge stations is always known by the server, and it is stored in a dedicated database containing the GPS coordinates of all associated MEC nodes. The participants in the crowdsensing campaigns have their own specific ParticipAct server to which they send contributes in order to complete the tasks assigned by ParticipAct's administrators and researchers. The ParticipAct platform already has all the tools to enable GPS tracking of users' locations with extreme precision. For the sake of controlling pandemic spreading, the administrators can choose the duration of a campaign. At campaign completion after participants can send all contributions. This policy is used to control the trade-off between monitoring level precision and network overhead.

For example, a short campaign duration augments the precision of the contributions, on the contrary a long duration for a campaign decreases the network load.

For our use case, the contributors send to the server their GPS tracking location data created and stored on their devices while performing a GPS tracking task. In this way, the server keeps the contributions of all users in terms of GPS coordinates and areas visited by contributors during the tracking campaign.

As Section 5.3 will detail, from the ParticipAct's database containing all the contributions, it is possible to obtain information about most frequented areas in urban centers, outlining degrees of density, in terms of number of people, and classifying them based on the indications of the medical authorities. In our deployment, The ParticipAct server is hosted in a private datacenter or in a public cloud. However, during the crowdsensing campaigns, the server is engaged in a high number of tasks, including the receipt of contributions by users. Furthermore, a dataset of location contributions can reach the size of several tens of thousands of entries per day, and the calculation of the density could be a heavy computation task if performed at the server side. For these reasons, we decide to delegate the ParticipAct edge agents to perform the calculation of the high-density zones.

The ParticipAct server, based on an application-dependent policies, sends a subset of coordinates to each associated edge node for the density calculation. We recall that the server knows the positions of the MEC nodes, so it sends to a single edge node only the coordinates that pertain to the coverage area of the node. The policy with which this calculation takes place on MEC nodes depends on the policy according to which the server recursively sends contributions to the edge nodes. For example, if a server sends daily updates, the edge node will calculate the crowding of its related zones on a daily basis. At the end of a generic campaign, including those related to the pandemic control, in addition to the GPS coordinates, the ParticipAct servers will send to the associated edge nodes the prize to be awarded to each user who has completed a campaign, so that all MEC nodes will have a copy of user scores.

The MEC nodes now can notify people who pass within their range with alerts about crowded areas. This information will appear on all devices of people having ParticipAct application, but the data can be also shared with third-party health services to enrich the knowledge base of the smart city.

### 5.2.2 Edge-based blockchain for rewarding management

In our proposal the storage of user's rewards in a federated crowdsensing environment is provided by an integrated edge-based blockchain platform within ParticipAct. The underlying reason for the adoption of a blockchain paradigm is to maintain rewards in a secure and distributed manner ensuring privacy and non-repudiable features. One crucial issue to consider when integrating a blockchain solution within an MCS platform is the architectural model to adopt. It is unrealistic to have a complete instance of the ledger on the end user devices and to rely on them for achieving a consistent ledger state. For saving resources we propose to rely on edge computing to distribute the ledger among multiple close-to-edge deployments. The ledger is distributed and decentralized among MEC nodes and MEC nodes are responsible for achieving a consistent state. We consider the employment of ETSI MEC nodes to improve the scalability of the entire system, in this way in fact the DLT-related functions can be executed exploiting the computing and storage edge resources lighten the cloud servers of in these additional tasks.

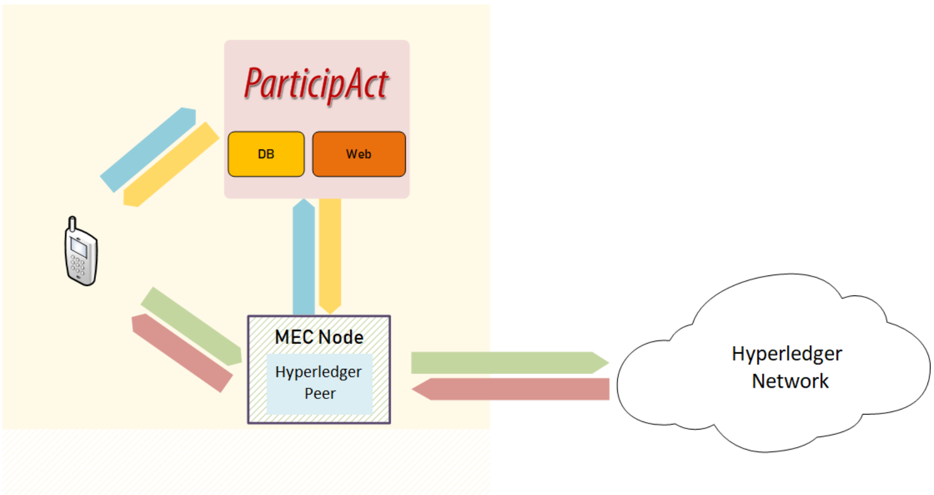



Fig. 5.2 Edge-based blockchain architecture

In the edge-based blockchain architecture, shown in Figure 5.2, the cloud server has a connection with a set of MEC nodes containing a full replica of the distributed ledger, i.e., all immutable concatenated rewards. The participant to the crowd-sensing campaign is still afferent to an individual PartcipAct server to which a group of ETSI MEC nodes have been added. The MEC nodes have numerous services including a complete distributed ledger constituted by a full replica and a wallet service. The clients rely on the blockchain facilities provided by their closest MEC node and through it they can access and interact with the rewarding account records.

The federated infrastructure still remains for all intents and purposes unaltered, with collected data still privately kept by cloud servers independently of each other. After having validated a user's task result, cloud servers calculates and report its relative point allotment to their closest MEC nodes. This information is then added to the blockchain so that it can be then shared by every other MEC node under the control of federated members. More in details, when rewards need to be updated, MEC nodes execute a proper smart contract

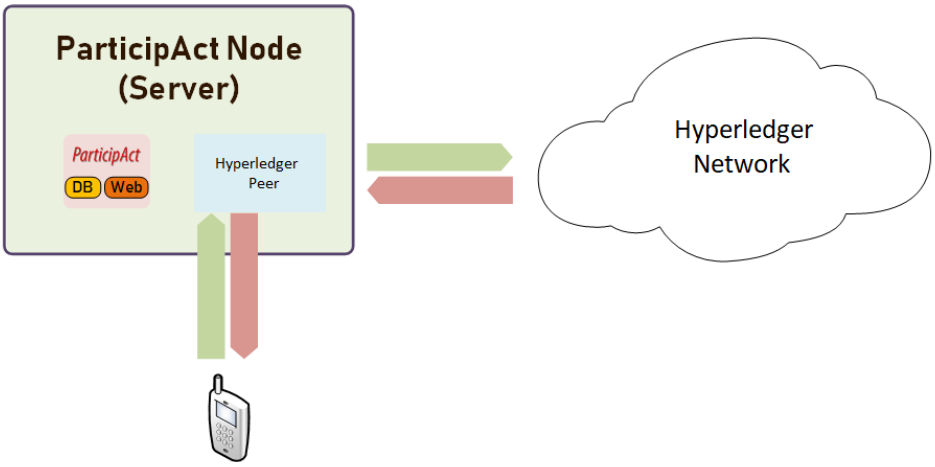



Fig. 5.3 Client-server blockchain architecture

and validate the transaction containing the reward update request. In particular, MEC nodes perform the consensus protocol to add the new block containing the transaction to the blockchain and to maintain a consistent state of the ledger. In [57] we have compared the above described edge-based blockchain architecture with an alternative deployment solution based on the client-server model. In this case, as shown in figure 5.3, the distributed ledger is completely located at the server level, and it keeps the reward data in a distributed manner among federated nodes. The federation is constituted by all the organizations which take part of the MCS campaign such as universities and company which constantly update the ledger in a way completely transparent to the end use. In this configuration the ledger can be placed aside of the ParticipAct database on each server keeping them independent and each reward update is broadcasted to every federated node.

Both the architectural approaches have different benefits and drawbacks. Comparing the client-server and an edge-based blockchain architectures as highlighted in the work [57], we can notice from the security perspective that the client-server architecture is potentially prone to tampering of the ledger since the number of federated institution servers tends to be low in the most common case. A low number of nodes enrolled in the server federation increases the risk of 50% + 1 attacks in which malicious actor can hijack the consensus protocol of the ledger taking over the majority of the nodes. Including the edge infrastructure in our ledger deployment improves the fault tolerance of the MCS platform, in this way in fact the blockchain knowledge base is distributed on many network segments which are more trustworthy since are managed by third party's telecommunication providers.

## 5.3 Platform Implementation

This Section provides a deeper view on the implementation and the algorithms executed in each single layer.

### 5.3.1 Crowded areas calculation

We deployed our ParticipAct servers in the cloud layer. Each server relies on a pool of MEC base stations and has its own users. The clients always know the server's address because they registered with it. We used a private datacenter to run the server application, but being the server a classic web application, its installation can be made also in a public cloud. How we can see in Figure 5.4, the ParticipAct server can create crowdsensing campaigns asking users to provide information from a broad range of sensors, including the GPS. By creating a GPS task, we can obtain information about the places people stay or pass-through.

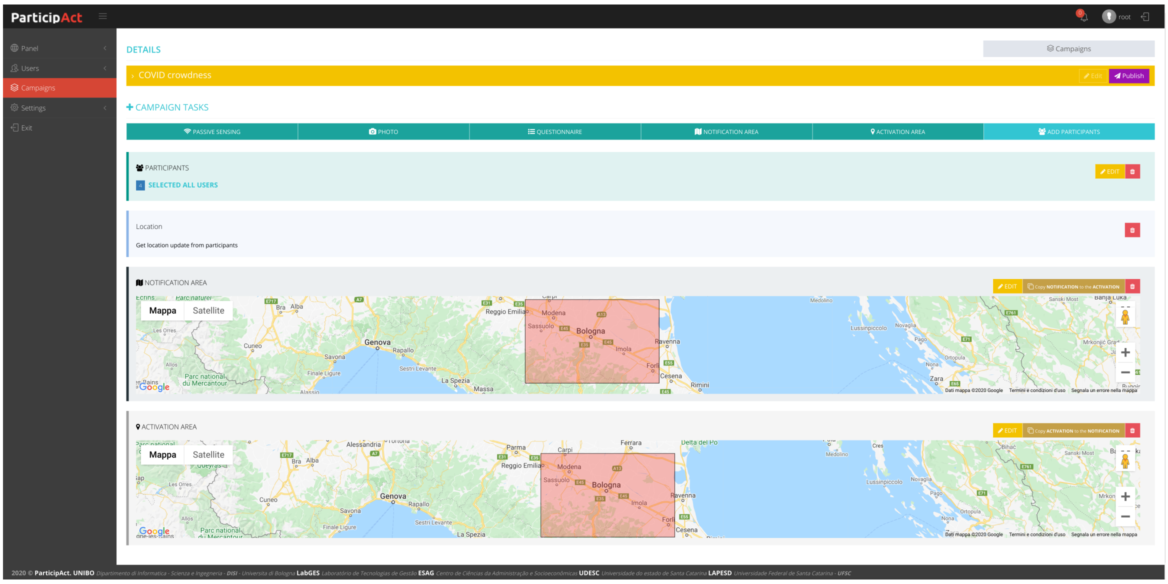



Fig. 5.4 ParticipAct GPS geo notificated and geo activated campaign

The server is responsible for the storage, aggregation, and processing of GPS locations. From the locality data, we can infer the density of the areas visited by users of the platform. For the server, we chose the cloud deployment for the great availability of resources inside a datacenter. This choice gives us the scalability that high demanding operations need. However, the calculation of crowded areas will take place in each edge node associated with a server, for the reasons set out in Section 5.2. For the density area calculation, we started from the *datalocation* table stored inside the DB on which the ParticipAct server backend stores all user contributions. Table 5.1 is an excerpt of the *datalocation* table. We send part of the coordinates to each associated edge node, based on their coverage area

| ***User_Id*** | ***Received Timestamp*** | ***Latitude*** | ***Longitude*** |
|---|---|---|---|
| 24173 | 2017-01-01 00:02:17.384 | 37.4876 | 14.056 |
| 15697 | 2017-01-01 00:12:41.197 | 38.0989 | 13.3997 |
| 48360 | 2017-01-01 00:14:21.547 | 41.2776 | 15.2665 |
| 24173 | 2017-01-01 00:17:17.575 | 37.4922 | 14.0557 |
| 15697 | 2017-01-01 00:27:42.709 | 38.0989 | 13.3996 |
| 48360 | 2017-01-01 00:34:19.583 | 41.2776 | 15.2665 |
| 8659 | 2017-01-01 00:35:27.982 | 44.4974 | 11.3436 |
| 24173 | 2017-01-01 00:37:20.836 | 37.4925 | 14.056 |
| 15697 | 2017-01-01 00:47:41.573 | 38.0989 | 13.3997 |
| 24173 | 2017-01-01 00:52:16.757 | 37.4916 | 14.0528 |

Table 5.1 Excerpt from the *datalocation* table

We used the Geohash coordinates as an identifier of the area for which we want to classify the density of people in transit, we used the Geohash coordinates, a practical geocode system which uses a short string of digits and letters to encode a geographic area. Substantially this system breaks the earth surface into 32 sections, in turn, divided into other subregions identified with a unique string. The width of the area selected by a string depends on the size of the string, the longer the string, the smaller the selected area. This feature allows us to have a dynamic dimension of the areas in which we calculate the level of crowding. For the calculation of the Geohash from the ParticipAct *datalocation* table, we use the PostGIS extension (https://postgis.net/) for the PostgresSQL database. PostGIS enables geographic support to the database, allowing location queries to be run in SQL language.

```
ST_GeoHash(geometry geom, integer maxchars=full_precision_of_point)
```

The previous query realizes the transformation from coordinates to geo hashes, taking a *geometry* point and an integer for the precision. We recall that a shorter geo hash coincides with a larger zone (less precise). If no *maxchars* is provided, the algorithm uses the default maximum precision (20 characters). The first parameter, of geometric type, is created by the function **ST_MakePoint**.

```
ST_MakePoint(float long, float lat)
```

The function alone does not refer to any Spatial References Identifier (**SRID**), a unique unambiguous identifier associated with a specific coordinate system.

```
ST_SetSRID(ST_MakePoint(float long, float lat),integer srid)
```

In our case, we use 4326 as SRID, which corresponds to the World Geodetic System 1984, used by GPS systems. The following query is the result of this algorithm using a geo hash area of 7 digits, or a tile size of 152.9 m x 152.4 m.

```
SELECT ST_GeoHash(ST_SetSRID(ST_MakePoint(long,lat),4326),7)
    INTO public.datalocationgeohash
    FROM public.datalocation;
```

We transformed all the coordinates in the *datalocation* table into geo hashes. We built a new table named *datalocationgeohash* taking the past coordinates gathered via GPS monitoring tasks. The last step to calculate the crowding of geo hash areas is the counting of the number of contributions for each area, our indicator of the population density in that area. The following query performs this operation.

```
SELECT st_geohash, COUNT (*)
    FROM public.datalocationgeohash
    GROUP BY st_geohash
    ORDER BY COUNT(*) DESC
```

### 5.3.2 Users' reward mechanism

For the choice of the most suitable blockchain platform, we have evaluated the main distributed ledger (DL) and their features such as permissioned vs permissionless, tokenized vs tokenless below. Permissioness blockchain means that users need prior approval (credentials like certificates or keys) before take part of the ledger and using it submitting transactions and smart contract whereas a permissionless blockchain lets anyone participate in the system. The second analyzed feature is about tokenized DL which has a mechanism to generate the currency like mining and require fee for transactions. Tokenized DL allow to exchange the currency for fiat currencies and requires a lot of computing power. The tokenless DL does not have any fee or mining mechanism and are prone to spam. The blockchain platforms analyzed for our use case are: Hyperledger Fabric and Ethereum. Fabric is a permissioned tokenless blockchain framework originally contributed by IBM and hosted by the Linux Foundation. Ethereum is a tokenized distributed ledger which provides digital money, data services and distributed applications, it supports permissionless and permissioned deployments and the use of a self-executing code known as smart contracts run by Ethereum Virtual Machine (EVM).

For our solution the Hyperledger Fabric platform turned out to be best suited to our needs thanks to its permissioned and tokenless feature which better support the private nature of our rewards network. The Hyperledger Fabric ledger is constituted by two different parts: world state and blockchain. The world state is a database which keeps the current values of the attributes of an object represented by key-value pairs. The world state allows the programs quick access to the blockchain values without having to go through the entire blockchain to calculate it. The second is the blockchain transaction log which keeps the transaction history collected in blocks. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data generally represented as a Merkle tree. In Hyperledger Fabric, each node have a copy of the world state and the blockchain ledger and any update of the ledger is performed by the peers individually executing the consensus algorithm. This algorithm is at the base of the consistency and ensures that every update is executed in a uniform manner on each peer and all the peers have identical copies of the ledger.

Fabric defines three types of peers which are all involved in the consensus protocol with different: i) endorsed peers which receive and validate the transactions and execute smart contracts; ii) ordered peers which create transaction blocks and receive endorsed transaction proposals and insert them in a block together with others in an orderly manner; iii) committer peers which receive and broadcast transactions and blocks. In addition, Hyperledger Fabric offers the chain code (the equivalent of the Ethereum smart contract) implementation in different general-purpose programming languages which are Java, Go, and Node.js. This feature makes easy the development of smart contract because we do not need to learn another language avoiding an additional layer of abstraction at programming level.

The application that manages the users' rewards is based on a Hyperledger Fabric chaincode and is written in Go which is the most performing language as highlighted in the work [58]. It maintains the rewards as a matrix data structure of triple <identifier:integer:timestamp>; the identifier is unique and is used by the ParticipAct application to anonymize the user personal data, the integer represents the number of points per user while the timestamp keeps track of the most recent data upload constituting a simple invalidation mechanism on blockchain. The reward chaincode is accessed in writing only from the server that calculate the increment or decrement of the gamification points based on task user contribution, and it is accessed in reading from any participant user to check the score.

A reward update on the ledger always starts by the server which evaluate the client gamification task and attributes a score to the user sending a transaction proposal to the MEC nodes. The MEC nodes reply with the responses which are received by the server and sent to the orderer node. The orderer first determinates the satisfaction of the endorsement policy and then compare the answers. The endorsement policy defines the number of peers which have to execute the chaincode to validate the transaction. The reading operation on the ledger to get the reward score can be rather executed by all the clients sending a transaction proposal query and immediately getting the result.

## 5.4 Use case: people contributions to face up to the COVID-19 pandemic

This section shows the experimental results grouped in two sets. The first one tests the load of the edge computation on the infrastructure, with respect to the cloud solution. The second part of the experiments regards the performance of the blockchain framework, in which we calculate two different kinds of latency: the query latency and the update latency.

### 5.4.1 Crowding experimental results

We investigate the capabilities of the ParticipAct server to calculate the crowding of geo hash areas, exploiting the ParticipAct dataset created through many crowdsensing real campaigns carried out from 2013 to 2017. We hypothesized that the density calculation takes place not continuously, but on a policy set on the basis of sending data from the server to the edge nodes. A fully functioning crowdsensing system, such as ParticipAct was during past data collection campaigns, could produce tens of thousands of entries containing user contributions. The geo hash calculation on so many entries could be a heavy process.

To obtain the following performance data we averaged ten runs of the SQL query to calculate the density. We simulated the two tiers, the server, and the edge one, using respectively a VM running in a private data center and having 4 vCPU, 16 GB of RAM, and 100 GB of HD, and a laptop with quad-core Intel processor, 8 GB of RAM, and 20 GB of HD. We considered the contributions collected in the date 02/04/2014, so a table with 47384 GPS entries (coordinate points). We executed the algorithm showed in section 5.2 on all the contributions of the selected day.

This calculation could not be negligible, indeed, usually, the server is under pressure during intensive crowdsensing campaigns due to the collection and processing of the data coming from the many sensors into the users' device. In this regard, we thought of transferring the processing of coordinates on the edge stations. Since each server relies on a pool of MEC Radio Access Network (RAN) stations, so the server sends part of the "datalocation" table (latitude, longitude, and contributions) to each edge site, based on their position. For example, if a RAN station covers a certain geographic area, only the positions involved in its range will be sent to that station. This cuts down on the calculation times of crowded areas as each edge station would only have to do with the contributions of users who have passed through this location. For this experiment, we used geo hashes with 7 digits, covering an area of 150 square meters, and we hypothesized a RAN coverage area of about 300 meters.

| ***Deployment*** | ***Time (sec)*** | ***Rows Number*** | ***CPU Usage*** | ***RAM Usage*** |
|---|---|---|---|---|
| Cloud | 5.42 | 47384 | 97% | 0.5% |
| Edge | 0.511 | 3754 | 90% | 0.4% |

Table 5.2 Performance comparison between cloud and edge deployments during geo hashes and density calculation

Table 5.2 shows the overhead due to the calculation of the densities performed respectively on the server and on the edge. The edge had to process only coordinates that belong to its coverage area, which are only 3754 entries. Instead, the server has to process all the contributions of the day. Although we simulated the MEC stations with a less powerful asset, they show a lighter footprint on system resources with respect to the execution on the server, due to the limited number of contributions. The timing for the edge case does not take into account the data splitting between the edge nodes, because it is carried out on the cloud side once a day and with negligible timing compared to the total gain.

| ***Geo Hash Area*** | ***Crowding*** |
|---|---|
| srbj1v8 | 657 |
| srbj1eh | 537 |
| srbj1g9 | 477 |
| srbj1dz | 376 |
| srbj1tr | 329 |

Table 5.3 Geo hash area's density calculation on a single edge node

Table 5.3 is the result of the calculation of the crowding of geo hash areas under the coverage of a hypothetical RAN MEC station located at the engineering faculty of the University of Bologna. The first column represents the geo hash areas under the edge station coverage, whereas the second column reports the number of people passing through that area during the analyzed day.

### 5.4.2 Hyperledger Fabric chaincode experimental results

Since the chaincode performance is crucial for a responsive and secure reward mechanism, we have analyzed and evaluated the transaction latency at varying of the number of participating endorser peers. The endorser peers play a crucial role in the consensus algorithm because are the nodes on which the chaincode is installed, they receive and execute the transaction proposal sent from the clients and reply sending back endorsed result (endorsed transaction proposal).

We defined the transaction latency as the time between when the client sends the request and when the response is received. Analyzing the Fabric operation, we can calculate two different type of latency: the query latency and the update latency which depend on the type of operation executed. For a query in fact the interaction is only between client and a peer, while an update involves also endorser, orderer and committer peer and goes through all the consensus algorithm phases. For these reasons we differentiate between *Query Latency (LpQ)* which is the time interval waited by the client between the sending of the request and the receiving of the response and the *Update Latency (LpT)* which is the time interval waited by the client between the request sending and the receiving a confirmation event to notice the inclusion of the transaction in a block and then to the blockchain.

Both *LpQ* and *LpT* are composed of a sum among several latencies, for the *LpQ* they are: (1) the time taken by the client application, (2) the time taken by a transaction to reach the number of endorser peers declared in the endorsement policy and to get the response, (3) the overall chaincode execution time, (4) the number of concurrent transactions executed, (5) the time spent to read/write the worldstate. For the *LpT* other two latencies are added: (6) the time taken by the orderer to receive transactions and organize them in blocks, (7) the amount of time to receive the new block, validate all the transaction individually and include it to each ledger peer. The latency is influenced by many factors such as, for example, peers and client machine hardware, network latency. Other important factors are the number of endorser peers and endorsement policy which defines the number of endorser peers that have to execute the transaction. Moreover, there is the type of orderer service that can work in one mode or Kafka, in the first one only a single node is involved while Kafka requires coordination between the different orderer nodes. Crucial is also the chaincode implementation details and programming language.

The testing scenario is constituted by VMs connected by a local network and created on a OpenStack-based cloud infrastructure constituted by 4 server hosts. Each VM has 2 CPUs, 2 GB RAM and 20 GB of disk and runs Ubuntu 16.04 LTS and Hyperledger Fabric (version 1.4). All the endorser peers run on different VMs, they have the chaincode installed and use LevelDB as world state. The orderer is implemented in one mode and runs on another distinct VM, while the CA is in a separate container. The client runs on a further VM which belongs to the local network. The tests are designed to measure the query and the update latency at varying of the number of peers. The tests use the Go languages both for the client and for the chaincode at varying the number of nodes of the network. The number of nodes for the tests are 1, 2, 4, 8, 10, 12, 14 and 16 nodes and consider the worst case which is when the transaction proposal is requested and executed by all the network peers to satisfy the endorsement policy.

The evaluation was carried out on the ledger both for the *Update Latency (LpQ)* and for *Query Latency (LpQ)*. For each experiment the client performs 50 transactions in sequence, the interval measured calculates the time from the request sending request to the response receiving. We have repeated each experiment 33 times to reduce error factor and in the graphs are shown the average values; we have not reported the standard deviations which are always below 6% for all tests. We consider the reception of the result for the experiment related to the queries while for the experiments focused on the updates the notification related to the inclusion of a transaction to a block and the appending on the blockchain is considered. To perform a write, we executed a writing of a new reward score for a certain user while for reading we requested the reward points of the same user.

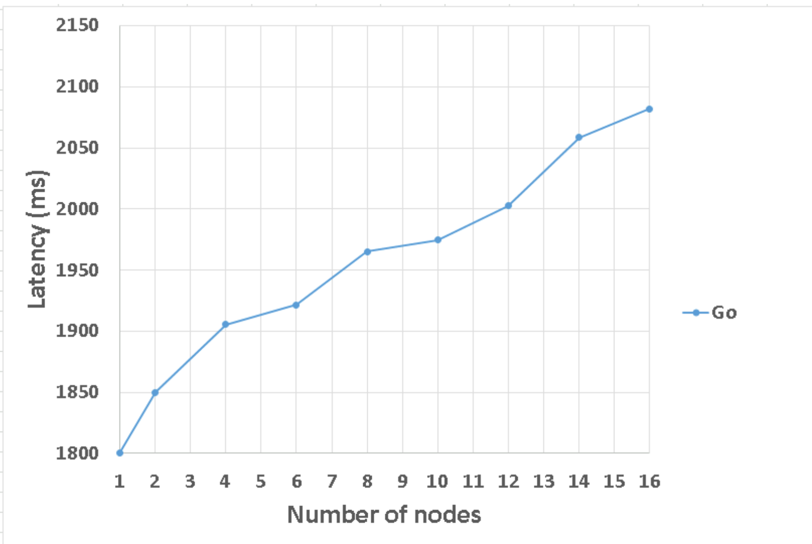



Fig. 5.5 Update transaction latency

Figure 5.5 shows the latency related to a ledger update, the transactions modify the world state and are then included to the blockchain. The update is then propagated to all the peers. We can notice that the graph follows a linear trend, it starts from about 1800ms and grows up to 2100ms. The difference of 300ms is caused by the execution of consensus algorithm and, in particular, the transaction proposal which have to be executed on all the 16 nodes to satisfy the endorsement policy.

Figure 5.6 shows the latency related to the queries which exploits only the first phases of the Hyperledger Fabric consensus algorithm without modify the ledger including the transactions. Differently from above in fact the reply does not have to wait for the replies of all the network nodes interacting only with a single peer. As we can see from the graph, in this case, the trend at the varying of the number of the peers is approximately constant, specifically, the latency starts from about 5ms and grows up to 20ms.

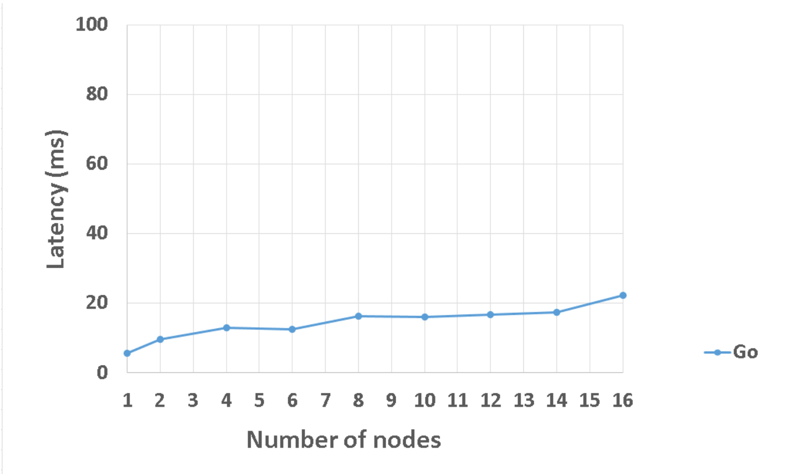



Fig. 5.6 Query transaction latency

Comparing the graphs, we can easily notice that the latency for a query is noticeably lower than for an update of the ledger. In particular, its average time is about 100 times faster than an update transaction execution. As described before, we believe the reason can be researched in the nature of the consensus algorithm and in the way the data is included into the blockchain. In the case of update in fact the algorithm involves several communications and an agreement among peers, in the second case the information is directly generated by a unique peer.

## 5.5 Lessons learned

Mobile crowdsensing is a paradigm that empowers the citizen contributions melting this data with smart city information. Pervasivness of mobile and wearables devices gives good chances to obtain detailed data from the crowd. Modern smart cities need for scalable and interoperable solutions in order to achieve a precise model of the reality and to give citizen and public institutions a good support based on real data. The gamification among the participants to crowdsensing campaigns improves the quality and the quantity of contributions provided by platform users.

We have extended the classic client-server infrastructure of MCS platforms by adding the edge layer and we have proposed a blockchain mechanism to federate different MCS servers in order to increase the catchment area and participation in crowdsensing campaigns. The edge layer supports both a local processing of location data on MEC nodes associated with each server and a full-distributed permissioned blockchain platform to keep the reward user score. Thanks to the edge we have numerous advantages. On the one hand it allows us to free the server from complex calculations that involve a non-negligible overhead. On the other hand, it increases the robustness of the distributed ledger platform replicating it on many nodes. In case of potential server failure the users' scores could be recovered from any replica on an associated edge node.

We tested these new capabilities using our MCS platform, namely ParticipAct, extending it with the middle edge tier and proving its operability. We specialized the platform by adding the capability to find crowded areas based on the presence of users in those areas during the crowdsensing campaigns. This integration can be used to mitigate the spread of pandemic diseases and could be of help, both for the smart city and for citizens, during the current health emergency concerning the spread of COVID-19. The calculation of the most crowded areas is a complex operation that involves many transformations from geographical coordinates to geo hashes values identifying a zone with different precision. In our tests, we put the MEC nodes under stress with the calculation of crowded areas, based on users' positions, and with the updating of the blockchain to estimate the users' score.

We are planning to completely integrate the edge capabilities with our crowdsensing platform model. The next step could be the implementation of our logic inside a real MEC node, following the official ETSI specifications. We will implement the activity recognition to programmatically and dynamically adjust the precision of the crowded radius areas calculated by our algorithm, with the hope that the users contributions could help in improving smart interactive services in the healthcare field and in situations of great emergency such as the one we are experiencing today.

# Chapter 6

# Cloud-Based Solutions for Industrial Reliability

This chapter collects all the tools and experiments aimed at improving corporate reliability. The I5.0 transition focuses on the reliable aspect of modern production realities, both to make the work environment safe and to avoid disruptions due to uncontrollable external factors. We will discuss in more detail, in separate sections, the use cases in which we employed cloud computing, branched in two main projects: a tool for auditing the performance of cloud providers resources and a network simulator for planning the right enterprise data center deployment.

## 6.1 DCNs-2: network simulations for virtualized resources

In the last years, ICT and production companies are undergoing a strong transformation of which Cloud Computing, Software-Defined Networks (SDN), and Network Functions Virtualization (NFV) are some of the most important expressions. The combination of these new technologies is changing not only the network development process but also the associated maintenance costs. The characteristics of the network are improved because, thanks to the SDN and NFV paradigms, it is possible to decouple many network functions from the underlying physical layer.

SDN makes the network control programmable, improving the performance of the above services, the monitoring capabilities, and the efficiency of the system. NFV is a complementary concept using virtualization to virtualize classes of network nodes acting as real equipment, such as load balancers, dispatcher, and signaling systems. Thanks to their capabilities leading to a reduction of costs for deploying a reliable network, SDN and NFV are considered two of the enabling technologies of the novel 5G networks [59].

From the perspective of the companies providing IT services on cloud environments using SDN/NFV concepts, there is a need to have mechanisms for verifying and simulating network behavior [60] [61]. Usually, in a Data Center Network (DCN) many servers provide different concurrent services and often they have to comply with very strict requirements. A big complex network can be arranged in different topologies and architectural models with different features, but almost all of them aim to build a highly scalable and reliable system able to guarantee high performance with a low-cost infrastructure. Scalability and reliability are just two examples of the requirements a DCN has to assure; there are many other elements to consider such as throughput, redundancy, and power consumption.

The growth of the data center size has been followed by the increase of the complexity of interconnecting its internal nodes and, consequently, of network traffic management. An IT network designer should limit the latency of the communication without expensive network equipment and give a right redundancy level with relatively low power consumption. The trade-off is between the compliance with these strict requirements and the adoption of economically sustainable solutions. In any case, to grant agreed quality levels, the DCN should implement a network topology and routing protocols that avoid over subscription of the connections. A good choice of the protocols and the topology during the design of the network leads to an optimal deployment of nodes and so to an efficient implementation of above SDN and NFV networks [62] [63].

To evaluate the performance of a DCN, we have three different possible choices, in general suitable for any complex system. Obviously, we can perform a direct observation of the network parameters during the full operations of the system. In the case of a DCN, this analysis method is often not applicable because of the complexity of the whole system and the cost of deploying a real system, not affordable for Small and Medium-sized Enterprises (SMEs) and researchers. As an alternative, it is possible to arrange an analytic model [64]. This second choice simplifies the complexities of the real system defining an abstract representation through a mathematical model. In many cases, that requires to adopt very simplified assumptions to represent the system, and this could lead to not so accurate results. Finally, a widely used alternative is the adoption of a network simulator [65]. This tool simulates real network components in the simulation playground and triggers events during a test scenario, in order to examine the reaction of the system in response to certain inputs. The simulators help cloud providers and companies with a big network infrastructure to overcome the issue deriving from the difficulty to perform on-field-tests to estimate the network performances.

Indeed, the use of a simulator is the preferred way to test the deployment of a DCN in the research community since researchers typically do not have the possibility to play with expensive real infrastructures, such as those available at big companies and cloud providers. There are many simulators that model common network infrastructures scenarios, analyzing different aspects of their deployment, such as performance and power consumption. However, at the current stage, most simulators have many drawbacks such as the configuration difficulty and the computational overhead. Above all, there is no network simulation software able to offer all the abstractions to describe cloud infrastructures of Infrastructure as a Service (IaaS) providers and their complex network interactions, as we will see in Section II. Additionally, traditional simulators usually do not provide a way to map the virtual resources on physical ones. Moreover, the simulators regarding only SDN and NFV aspects do not cover the issues related to the physical layer, and it becomes impossible to correlate software service failures with the behavior of the underlying network infrastructure [66] [67].

To overcome those limitations, we introduce a flexible and easy to configure simulation platform, suitable to execute performance tests on many network topologies [68]. The platform expands the classical network simulator Ns-2 with data center specific modules and components. With our new simulator, called DCNs-2, it is possible to evaluate the fault-tolerance of a DCN and typical management and operations scenarios, such as the response of the system to certain allocation and migration policies. The simulator can shape all the network equipment, the physical nodes hosting virtual resources, and all the connections that determine the topology of a DCN. Furthermore, we claim the enabling nature of our simulator thanks to the features we added such as migration and provisioning entities which simulate key processes in SDN and NFV architectures. In case of transition from physical network assets to virtual SDN resources, our simulator can estimate the position and the replication level of SDN resources (for instance the SDN controllers) to obtain good scalability and it can check the answer of the network to migration, interaction and interconnections of NFV resources. Our implementation also provides fine-grained statistics to the observer both during the operation time and at the end of the simulation, through classical log trace file.

### 6.1.1 The Ns-2 simulator

We use this section to provide some useful core concepts of the Ns-2 original simulator which can be recalled or extended by our solution, DCNs-2

Ns-2 is a DES tool written in C++ language and using *OTcl* (MIT Object-oriented Tool Command Language) [1] as a scripting language to configure the test scenarios. The simulator

[1]http://www.mathcs.emory.edu/~cheung/Courses/558/Syllabus/A2-Tcl/OTcl.html

is used mainly for analysis and modeling of new network protocols. The diagram in Figure 6.1 highlights the use of the two programming languages, C++ and OTcl, each with its own class hierarchy.

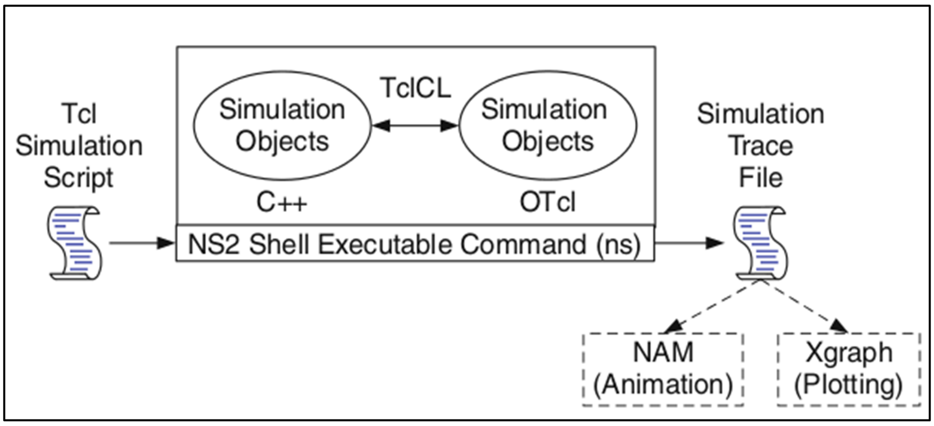



Fig. 6.1 Ns-2 simulator architecture overview

A simulated scenario is a timeline reporting the events that the user wants to trigger at certain time instants. On one hand, in Ns-2 we need a programming language that can efficiently manipulate bytes and implement complex algorithms; on the other hand, we need a programming language faster in iteration-time that in run-time, for example, to change a parameter and re-run a simulated scenario. Using an interpreted language (OTcl) allows you to configure a new simulation without having to re-compile the system primitives. Employing the compiled language (C++), Ns-2 optimizes the execution of the simulator core code. The two hierarchies of classes are closely related to each other with a one-to-one correspondence, and they have a common root class. Through TclCL interface, Ns-2 makes objects and variables available in both programming languages.

Let us overview some components of the original simulator for a better comprehension of our extensions. Figure 6.2 shows the main objects involved in a simulated communication path. The Node object is responsible for the packet multiplexing and can represent the leaf of a communication path. Internally it is constituted by objects in charge of packets switching to other destination nodes (through Links objects type) or representing the endpoint of the communication (object Agent), consuming and destroying the packets. A packet can be delivered to an Agent object through the PortClassifier multiplexer, or to another Node object through the AddrClassifier internal multiplexer.

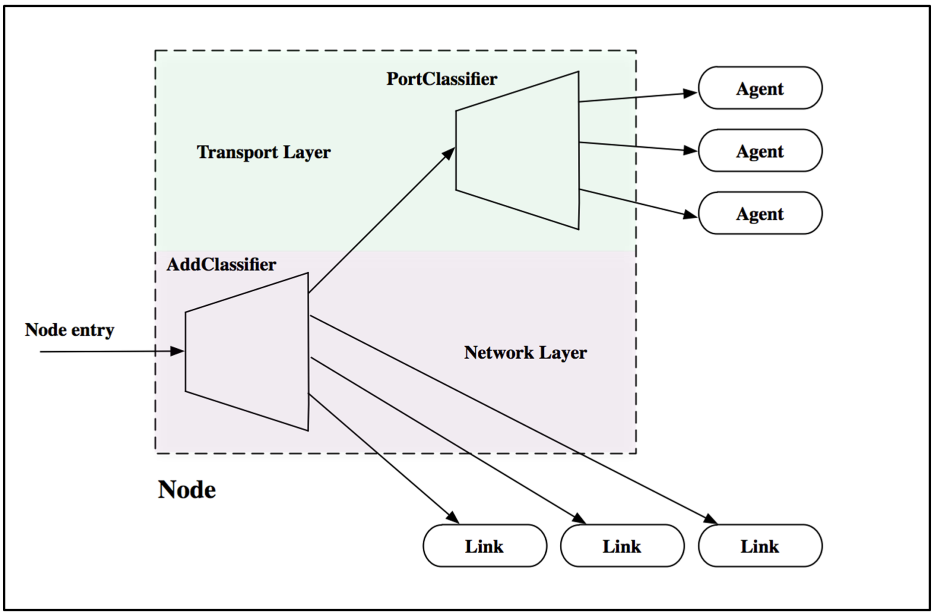



Fig. 6.2 Ns-2: Node object internal schema

The Link objects are responsible for store and forwarding policies. They have some associated queues and if the receiving queue is full, they can simulate a situation of link congestion and consequently the Link object will drop the next packets. Using the Queue-Monitor objects the user can monitor traffic and statistics such as the number of byte/packets received, sent or discarded, and the network performance.

The Application objects allow the creation of traffic generators (for example, a constant bit rate traffic generator) and the simulation of application layer protocols (such as the FTP or HTTP protocols), through Application objects. Regarding the traffic generators, the policies are dictated by internal Timer objects, ruling the rate of the creation of the packets to send.

After that the main parts of a simulation in Ns-2 are introduced, we can show the improvements added to the simulator by our implementation.

### 6.1.2 DCNs-2 platform design

To create DCNs-2 simulator, we extended the Ns-2 original one, focusing the solution on the IaaS simulation [69]. Our platform has the same features of the base simulator, such as the possibility to schedule events during a test scenario and to simulate routing policies. Figure 6.3 reports the overall architecture of our simulator.

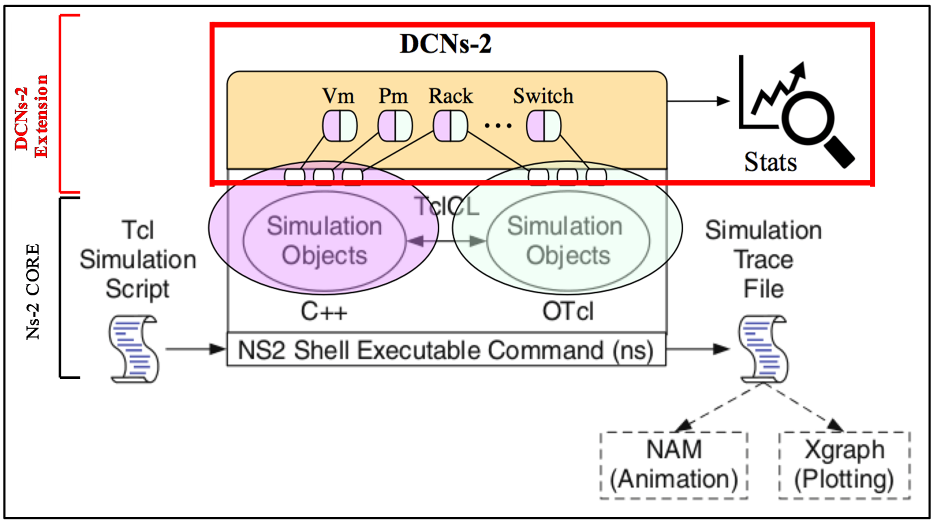



Fig. 6.3 DCNs-2 platform overall architecture

Leaving the double hierarchy of classes, as in the original simulator, we can reconfigure a simulation scenario through directives in OTcl language, without having to recompile the structures written in C ++ language. Figure 6.3 also shows a significant subset of new classes we added to simulate real network assets. The main idea is to add all the entities modeling the components of an IaaS deployment. For instance, Figure 6.3 reports the objects representing a virtual and a physical machine, respectively called Vm and Pm, and the entities simulating the network assets and the racks, called Switch and Rack. A key feature that makes our simulator particularly fitting to represent a virtualized cloud environment is the realization of both physical and virtual abstractions for each main resource. In addition, to manage the complexity of a big simulated network environment, we transferred some module definitions in the compiled domain, reducing the probability to introduce errors during the phase of definition of a new simulation.

Besides the introduction of new entities, we also improved the monitoring capabilities of the original simulator enabling the run-time access to the statistics of the simulation. At any time, the user can access the traffic statistics, regarding the network connections, and, to allow the maximum flexibility, the user can see the complete trace file for the post-execution analysis.

Our simulator is fulfilling also many non-functional requirements. It supports the easy extension of the newly introduced entities that model a DCN. Performance results shown in the following demonstrate that the system presents good scalability, providing the ability to simulate complex test scenarios, modeling big networks with hundreds of nodes and maintaining the execution time proportional to the simulated entities number.

Basically, the DCNs-2 simulator turns out to be a flexible platform, easy to configure, use and expand, capable to simulate and analyze the performance of a big DCN. With DCNs-2, we can study the behavior of a system in response to specific allocation and resource migration policies. Furthermore, we also boosted up the analysis process adding the availability of the statistic reports during the simulation runtime phase. In the next section, we will see the implementation and the usage of some new modules we introduced to simulate the DCNs.

### 6.1.3 Implementation insight

In this section, we analyze the implementation of the new added entities. With these modules, we can represent in a simulated scenario all the main network equipment of a real cloud DCN.

#### Physical and Virtual Resources

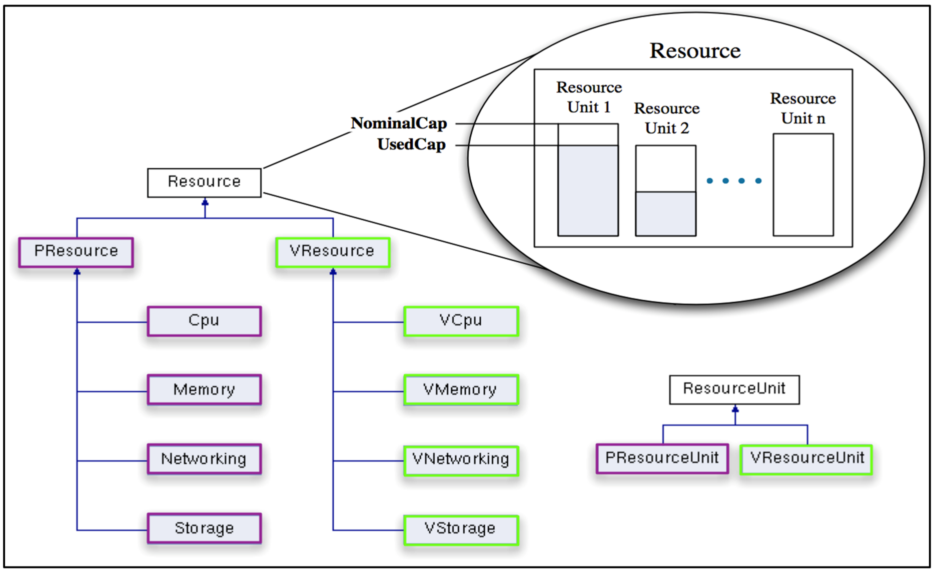



Fig. 6.4 Inheritance diagram of the Resource and ResourceUnit classes

The main brick to build a node in our DCNs-2 simulator is a component modeling a resource. Inside an IaaS deployment there are many virtualized resources, so in our simulator, we chose to introduce two types for a resource, one physical and the virtual counterpart. We created two classes for all the principal components of a real host: Central Processing Unit (CPU), MEMORY, STORAGE, and NETWORKING. Figure 6.4 shows the hierarchy schema of the main resources. In our platform, each of them has two versions that respectively model the physical object and the virtual counterpart, as it is in real DCNs. How we can see in Figure

6.4, each resource is a composition of smaller units with a prefixed nominal capacity. The full capacity of a resource is equal to the sum of nominal capacities of the individual units, which can be arranged according to the user's needs. For example, a 1000 Million Instructions Per Second (MIPS) CPU can be obtained using an object containing 4 units of 250 MIPS each. The only exception is the network resources that can only contain an elementary internal unit expressing the capacity of the network interface in bps. To reduce the risk of error during the definition of a test scenario, we provide objects ready to use with a static capacity, such as CPU_2CORE, MEMORY_16GB and STORAGE_1TB objects. Following our modularity and flexibility objectives, we have left open the possibility to define resources with different characteristics by adding or removing units with arbitrary nominal capacities.

### Computing and memory resources

Following the philosophy of composition adopted for the definition of the resources, we defined a virtual or physical host as a composition of resources. The object modeling a physical host could have a list of virtual machines inside, besides its own basic resources. To complete the modeling of a host server, we added a component that plays the role of hypervisor for the guest virtual machines. The Virtual Machine Monitor object manages the virtual resources of the physical host and the mobility of its virtual machines. It determines for a physical node, the possibility to spawn (or to migrate) a new virtual machine based on the current residual capacity of physical resources.

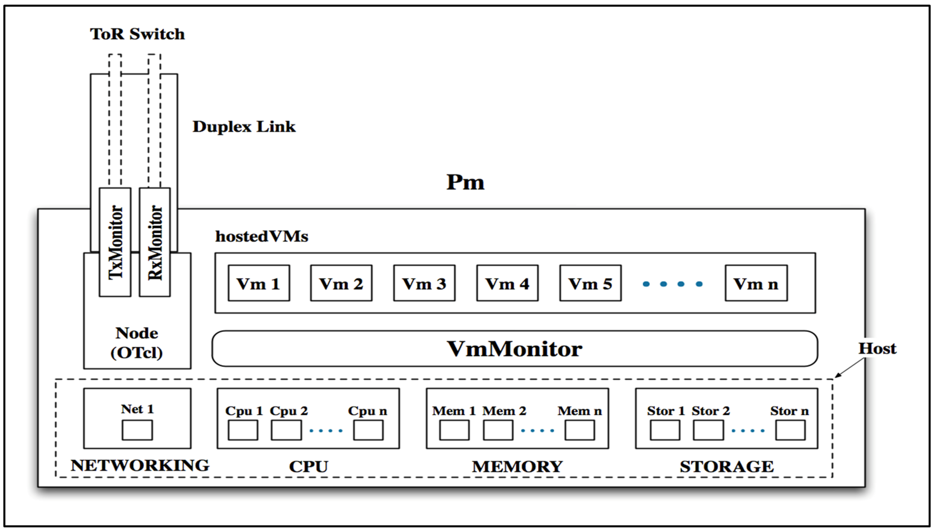



Fig. 6.5 Schema of a physical simulated host in DCNs-2

Figure 6.5 shows a schematic representation of the architecture of a physical host in DCNs-2. In addition to the components listed above, we can also notice another basic unit that represents the communication channel. The Node object records all the Agents used by the virtual machines for packet switching, simulating the network communication. Remember that Agents represent endpoints where network-layer packets are generated or consumed. For example, in Figure 6.5, the node object is connected to a Top-of-Rack (ToR) switch endpoint. In fact, the simulator forecasts the modeling of a classic Rack host containing many physical hosts and a switch for the connectivity, a very common configuration in the DCNs.

Taking the IaaS applications as an example, we created a set of virtual machines with predefined resource capacity, leaving the opportunity to add virtual machines with an arbitrary amount of internal resources. Each virtual machine has one receiving Agent and several transmission Agents, respectively to consume and generate traffic in a simulation. The Agent objects handle the statistics of the traffic.

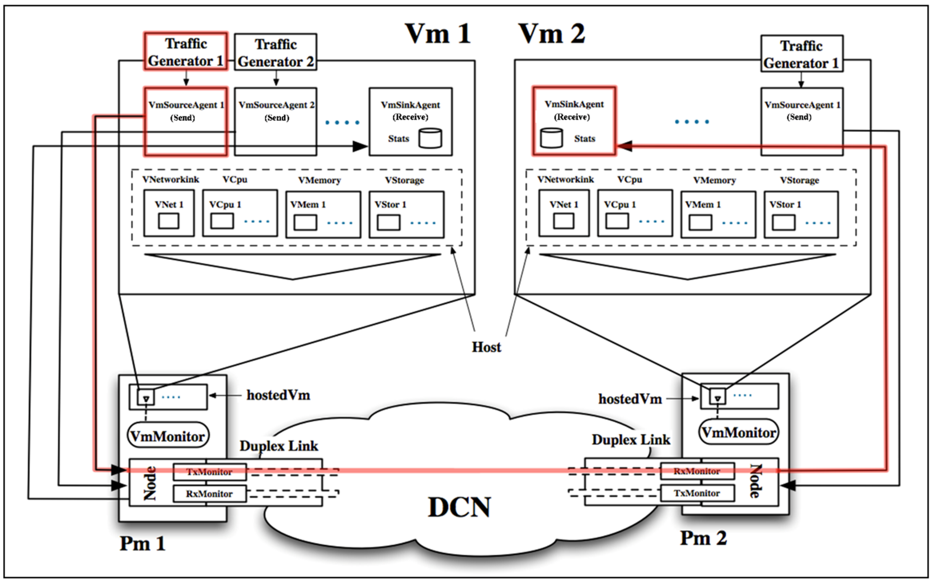



Fig. 6.6 VM communicating in a DCNs-2 simulation

Figure 6.6 shows the communication between two virtual machines, involving the Node objects belonging to the hosts where the communicating virtual machines are allocated. Unlike the physical hosts having only two possible states, ON and OFF, determining if a host can handle a request of allocation/migration of virtual resources, the virtual machines have many possible states:

- **RUNNING**: the active virtual machine occupies both primary and secondary memory of the physical host and it can communicate with other virtual machines,
- **SUSPENDED**: the suspended virtual machine uses only the secondary memory of the hosting node and it cannot communicate with other virtual machines,

- **INACTIVE**: the state used when the virtual machine is turned off or just created and not yet assigned to a physical host. The virtual machine uses the secondary memory of the host and it cannot communicate with other virtual machines.

### Networking Resources

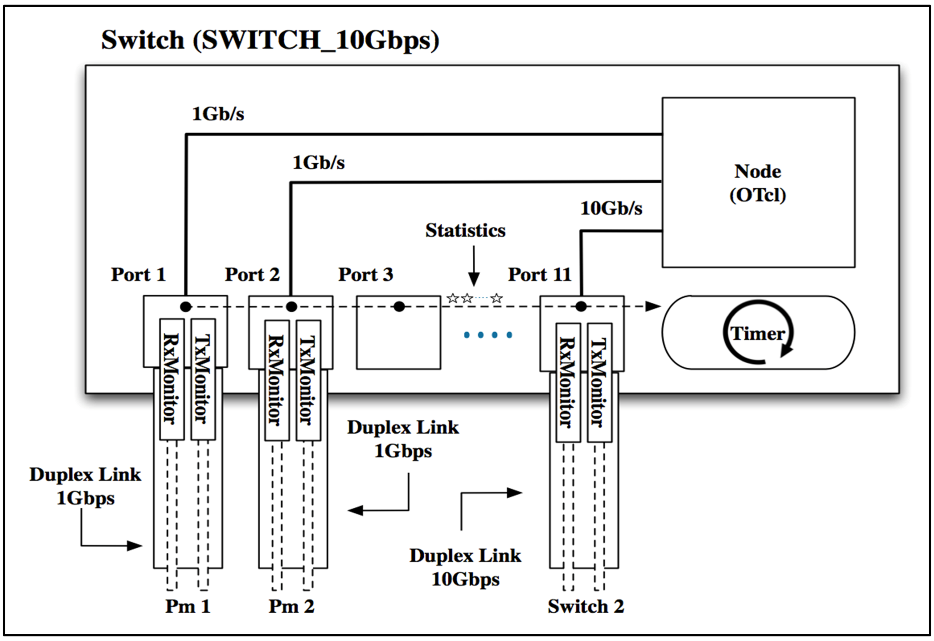



Fig. 6.7 DCNs-2 Switch schema

The Switch objects are responsible for the simulated routing policies and they permit the communication between hosts. Our simulator does not have virtual network resources representation, such as virtual switches and routers. This lack does not affect the communication between physical or virtual resources and therefore we decided to add these peculiarities later. In Figure 6.7, there is a schematic representation of a Switch object, having many ports each with its own bit rate. The user can create Switch objects with an arbitrary bit rate. This component simulating network asset can process the statistics at regular time intervals and the user can change the sampling rate to tune the analysis granularity. The schema in Figure 6.7 shows as many Node objects as the number of connected ports of the switch. The statistics of each port are individually analyzable in terms of channel load and arrived, rejected or sent packets. We implemented also a specialization of the Switch representing a real top-of-rack switch that can be added to a Rack object to simulate the communication equipment of the access level of a DCN.

### Management of Virtual Resources over physical ones

In previous sections, we showed the implementation of the classic life-cycle states of the virtual resources in a traditional cloud IaaS environment. Exploiting these abstractions, our simulator can reproduce the complex operations occurring within the cloud infrastructure, such as the migration and the allocation of the virtualized resources.

We implemented a super-object having the view of all the entities in a simulated scenario. This object, called **DcManager**, store a constantly updated list of associations between physical and virtual machines. Given its global sight, the manager object receives all the migration, allocation and removal requests related to virtual resources. One of the tasks of the manager object is to check the availability of resources following a migration or allocation request. For the allocation, if the check about resources has been successful, the virtual machine is assigned to the physical host and the Virtual Machine Monitor decreases the free capacity of the host. The last step in the allocation process is the connection of the communication Agents to the server's Node object and the embedding of the new virtual machine in the list of the virtual machines hosted by the physical machine.

To remove a virtual machine, the Virtual Machine Monitor first updates the state of the virtual machine placing it in the INACTIVE state, and later it deletes the virtual machine from the list of allocated machines, updating residual capacity of the physical host. The final step here is to detach of the Agents of the virtual machine from the server's Node object. In our simulator, we implemented different types of migration. Regarding the cold migration, the simulator acts as follows: the Virtual Machine Monitor updates the status of the virtual machine to INACTIVE and removes it from the source server before the allocation of the virtual machine on the new host, always in the INACTIVE state. The restore of the virtual machine activity occurs when it is placed in the RUNNING state. This type of migration does not involve the exchange of network messages between the two servers, so not causes communication overhead.

For live migration, we can use Post-copy or Pre-copy strategy. During a Post-copy live migration, the virtual resource is suspended on the physical machine on which it is currently allocated, and all active connections are closed. In a real system, at this point, it should take place a copy of the virtual machine memory pages. This step is simulated by using two migration Agents communicating through a constant bit rate connection established between the physical hosts involved in the migration. To simplify the implementation of the process, we chose a constant to represent the current state of a virtual machine. We modeled this constant as 5% of the total virtual memory of the virtual machine to be transferred. At the end of the process, the machine is resumed to the RUNNING state and the connections are reactivated. During a Pre-copy live migration, the communications of the virtual machine stay active and its status remains RUNNING during the whole process. The communication is interrupted only during a negligible interval of time that starts from the detaching of the Agents on the sender Node till the restore on the target Node. The current state of a transferred virtual machine using the pre-copy process is set as 10% of virtual memory, a greater value than the previous case to simulate the iterative pre-copy step happening in a real scenario. For the sake of implementation simplicity, we neglected the copy process of the dirty pages, which always happens during the pre-copy live migrations of virtual resources [70] and it consists of copying to the destination server the memory pages written during the migration.

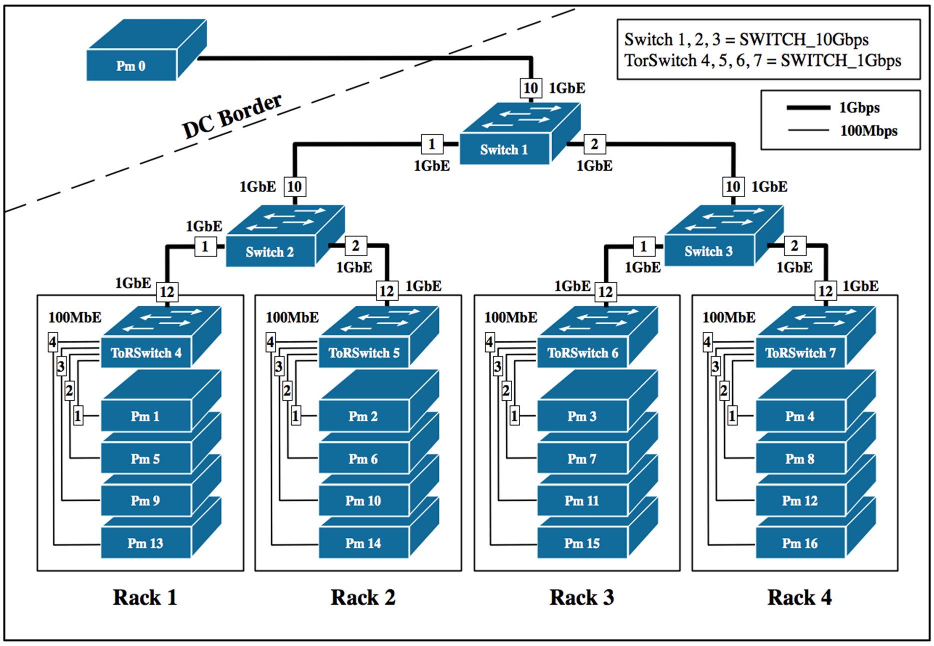



Fig. 6.8 Schema of the simulated scenario

### 6.1.4 Experimental results

This section assesses the effectiveness and efficiency of our network simulation platform. After defining the test scenario, we will analyze the performance and the overhead of the running simulator. In addition to the functional tests, such as the simulation of link faults, for the analysis of the performance of our network simulator, we have chosen the most important parameters for large SDN/NFV networks [71] [72] [73], i.e. migration/provisioning times, and throughput of the communication channels. For the simulations, we arranged a test scenario having the topology shown in Figure 6.8, and the spawning schema of Table 6.1:

- 16 physical machines evenly distributed inside four racks,

| Rack 1 | | Rack 2 | | Rack 3 | | Rack 4 | |
|---|---|---|---|---|---|---|---|
| Pm 1 | Vm 1 | Pm 2 | Vm 2 | Pm 3 | Vm 3 | Pm 4 | Vm 4 |
| | Vm 17 | | Vm 18 | | Vm 19 | | Vm 20 |
| | Vm 33 | | Vm 34 | | Vm 35 | | Vm 36 |
| | Vm 49 | | Vm 50 | | Vm 51 | | Vm 52 |
| Pm 5 | Vm 5 | Pm 6 | Vm 6 | Pm 7 | Vm 7 | Pm 8 | Vm 8 |
| | Vm 21 | | Vm 22 | | Vm 23 | | Vm 24 |
| | Vm 37 | | Vm 38 | | Vm 39 | | Vm 40 |
| | Vm 53 | | Vm 54 | | Vm 55 | | Vm 56 |
| Pm 9 | Vm 9 | Pm 10 | Vm 10 | Pm 11 | Vm 11 | Pm 12 | Vm 12 |
| | Vm 25 | | Vm 26 | | Vm 27 | | Vm 28 |
| | Vm 41 | | Vm 42 | | Vm 43 | | Vm 44 |
| | Vm 57 | | Vm 58 | | Vm 59 | | Vm 60 |
| Pm 13 | Vm 13 | Pm 14 | Vm 14 | Pm 15 | Vm 15 | Pm 16 | Vm 16 |
| | Vm 29 | | Vm 30 | | Vm 31 | | Vm 32 |
| | Vm 45 | | Vm 46 | | Vm 47 | | Vm 48 |
| | Vm 61 | | Vm 62 | | Vm 63 | | Vm 64 |

Table 6.1 Test scenario deployment

- an external host (Pm0 object) to simulate hypothetical communication with the outside networks,
- all the servers have two 4-core CPU of 1000 MIPS, a primary memory of 8 GB and a secondary memory of 1 TB, a network interface of 1GbE bit rate. The external host Pm0 has a single CPU unit of 1000 MIPS instead of two,
- 7 network assets: 3 Switch objects and 4 ToR Switch objects. The Switch of the aggregation and core level have a capacity of 10 Gbps, instead, the Switch of the access level has a capacity of 1 Gbps,
- 64 virtual machines evenly distributed inside the 16 physical hosts and one virtual machine inside the Pm0 external machine.

We forecasted an execution time of 3600 seconds (one hour) for all the simulations. For the following evaluation tests, we suppose physical machines within the data center are sharing the same file system, so the migration process involves only the memory and not also the virtual machine image base file.

**Test 1: Rating of the network connections**

In this early test, we try out the transmission of data between two virtual machines. We activated the virtual machines (changing the state from INACTIVE to RUNNING) at time t=1 (after one second of simulation). For obtaining a transmission time of one hour, we modified the simulation duration from 3600 to 3601 seconds. We defined a constant bit rate (CBR) connection of 1 Mbps with packet dimension of 512 bytes between two virtual machines allocated in different physical machines, Pm1 and Pm5, inside the same rack. We analyzed the traffic on the ports connecting the two physical hosts of the top-of-rack switch. At the end of this simulation, there is an amount of 450 MB exchanged between port 1 and port 2 of the ToRSwitch 4, the connection ports of Pm1 and Pm 5. Using the same configuration, we tested also an incremental number of connections. We tried with four connections:

- a CBR connection of 1 Mbps between Vm1 and Vm5, the same of the previous case (producing 450 MB of traffic);
- a CBR connection of 500 Kbps between Vm1 and Vm0 (producing 225 MB of traffic);
- a CBR connection of 1.5 Mbps between Vm4 and Vm0 (producing 675 MB of traffic);
- a constant bitrate connection of 5 Mbps between Vm17 and Vm33 (producing 2.25 GB of traffic).

```
==========================================================================================
Pm 1: State = ON, Rack_Id = 1, NIC = 1.00Gbps, RxBytes = 0.00kB, TxBytes = 675.00MB
      Cpu = 2000MIPS, usedCpu = 0MIPS, availCpu = 2000MIPS
      Memory = 8.00GB, usedMemory = 0.00kB, availMemory = 8.00GB
      Storage = 1.00TB, usedStorage = 40.00GB, availStorage = 960.00GB
======================================== Resources =======================================
[...]
===================================== Virtual Machines ===================================
Vm 1: Type = NANO, State = INACTIVE, Host_Pm = 1, vNIC = 10.00Mbps, RxBytes = 0.00kB, TxBytes = 675.00MB
      VCpu = 250MIPS(100%), usedVCpu = 0MIPS, availVCpu = 250MIPS
      VMemory = 1.00GB(100%), usedVMemory = 0.00kB, availVMemory = 1.00GB
      VStorage = 10.00GB(100%), usedVStorage = 10.00GB, availVStorage = 0.00kB
------------------------------------------------------------------------------------------
Vm 17: Type = NANO, State = INACTIVE, Host_Pm = 1, vNIC = 10.00Mbps, RxBytes = 0.00kB, TxBytes = 2.25GB
      VCpu = 250MIPS(100%), usedVCpu = 0MIPS, availVCpu = 250MIPS
      VMemory = 1.00GB(100%), usedVMemory = 0.00kB, availVMemory = 1.00GB
      VStorage = 10.00GB(100%), usedVStorage = 10.00GB, availVStorage = 0.00kB
------------------------------------------------------------------------------------------
Vm 33: Type = NANO, State = INACTIVE, Host_Pm = 1, vNIC = 10.00Mbps, RxBytes = 2.25GB, TxBytes = 0.00kB
      VCpu = 250MIPS(100%), usedVCpu = 0MIPS, availVCpu = 250MIPS
      VMemory = 1.00GB(100%), usedVMemory = 0.00kB, availVMemory = 1.00GB
      VStorage = 10.00GB(100%), usedVStorage = 10.00GB, availVStorage = 0.00kB
------------------------------------------------------------------------------------------
Vm 49: Type = NANO, State = INACTIVE, Host_Pm = 1, vNIC = 10.00Mbps, RxBytes = 0.00kB, TxBytes = 0.00kB
      VCpu = 250MIPS(100%), usedVCpu = 0MIPS, availVCpu = 250MIPS
      VMemory = 1.00GB(100%), usedVMemory = 0.00kB, availVMemory = 1.00GB
      VStorage = 10.00GB(100%), usedVStorage = 10.00GB, availVStorage = 0.00kB
------------------------------------------------------------------------------------------
==========================================================================================
```

Fig. 6.9 Pm1 output after the execution of the scenario with 4 connection

Figure 6.9 shows a snippet got from the output of the traffic analysis extracted from physical machine Pm1. In the same figure, there is also the traffic generated by the communication between the virtual machines, but not that of fourth connection due to the resiliency on the same host of the communicating virtual machines, Vm17 and Vm33.

### Test 2: Migration of virtualized resources

A crucial aspect of the modern cloud DCNs is the migration of the virtual resources. In the following simulation, we used a single CBR connection of 1 Mbps between Vm1 and Vm5. At time t = 1801, we scheduled migration of the Vm1 virtual machine from the physical host Pm1 to the destination host Pm13, using the pre-copy migration strategy.

The test reveals a transferring time of 80 seconds, during which the virtual machine was always connected with Vm5 (the amount of data transferred is 225 MB before the migration and 235 MB after). This time-lapse is which needed to transfer the 10% of the memory of the virtual machine Vm1, from source physical host Pm1 to the destination one Pm13, using the CBR connection of 10 Mbps of the tor switch connecting the hosts. We can notice an additional transfer of 4 MB due to the TCP connection protocol and to the acknowledgment mechanisms used by Ns-2 to manage this protocol. A further proof that the connection between Vm1 and Vm5 remains in an active state during the migration is the amount of data exchanged between the virtual machines (450 MB), the same detected in the scenario without the migration. Analyzing the output resulting from the migration of the same Vm1, using a post-copy strategy, we notice that the amount of the exchanged data is the half of the previous case, so we need exactly half the time to complete the migration. However, during the 40 seconds of migration time, the virtual machine is suspended. In fact, the amount of the exchanged data during the communication between Vm1 and Vm5 is now 445 MB, not 450 anymore.

### Test 3: Network link fault

The simulator is suitable also for the simulation of network link failures. For this test, we scheduled a fault at time t = 1801 involving the link between the core Switch1 and the switch at first aggregation level, the Switch2. The rest of the scenario was arranged like the second part of test 1, with 4 active connections. In this case, the communication affected by the fault is the connection between Vm1 and Vm0. At time t = 1801, Switch1 stops receiving packets despite Switch2 continues to receive the packets generated by Vm1 and destined to Vm0 (via the ToRSwitch4).

### Test 4: Comparing two network topologies

This test confirms the suitability of our tool for the analysis of the performance between different network topologies. We created a scenario closer to a real data center than the previous, with 128 servers and 1024 virtual machines, and we chose to evaluate the performance of two network topologies, the tree, and the multipath Clos, which diagram is shown in Figure 6.10.

We chose these two topologies as they are widely used as intra-datacenter network topology [74] [75]. The virtual and physical machines are of the same type of the previous simulations. In the multipath Clos topology, we used a Link State routing protocol for packet switching and Equal-Cost Multi-Path (ECMP) routing strategy to exploit the redundancy of the path provided by this topology. We simulated two types of traffic using 600 connections; with the distribution of 512 connections equitably among the virtual machines inside the data center, we reproduced the east-west traffic, instead, with the remaining 88 connections between Vm0 and other virtualized resources of the data center, we reproduced the north-south traffic.

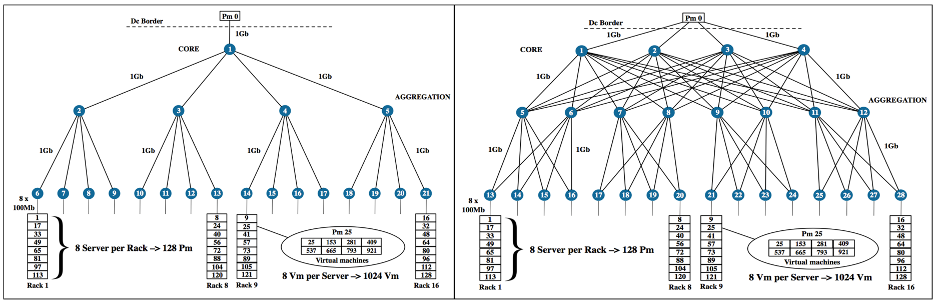



Fig. 6.10 Tree (a) and Clos (b) network topologies

We want to analyze the results from different points of view. Taking into account the network **scalability** investigation, we noticed better scalability of the Clos network topology. Looking at the throughput of the devices of the aggregation level, we can see a higher load of the same channels in the tree topology.

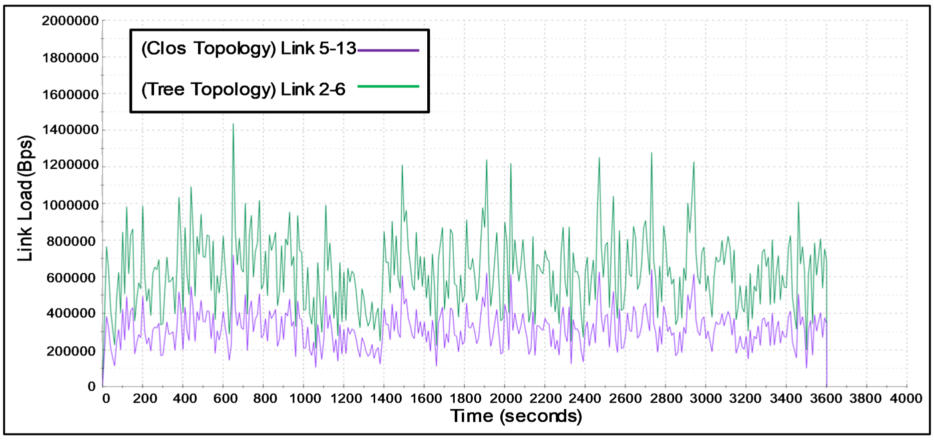



Fig. 6.11 Throughput comparison of links 2-6 and 5-13, respectively of the Tree and Clos topology

In Figure 6.11, we show the throughput of two receiving ports: the first is the port 1 of the first switch of the aggregation level (Switch5) in the Clos topology (link 5-13 in the figure), the second is the same port of the first switch in the aggregation level (Switch2) in the Tree topology (link 2-6 in the figure). We chose these ports because they are the destination of the traffic coming from the 8 servers (more specifically from 64 virtual machines) inside the Rack1. The load on the link 2-6 of the tree topology is twice the load on the link 5-13 of the Clos topology. This result is due mostly to the model of the network: the Clos topology has two uplinks in the aggregation level for each device in the access level and inside the network, there is the ECMP protocol using a multipath routing strategy. The gap of the load on the links is more visible analyzing the links connecting the levels core and aggregation. For example, Figure 6.12 shows the difference of the load for the link 1-2 of the Tree topology and the link 1-5 of the Clos topology.

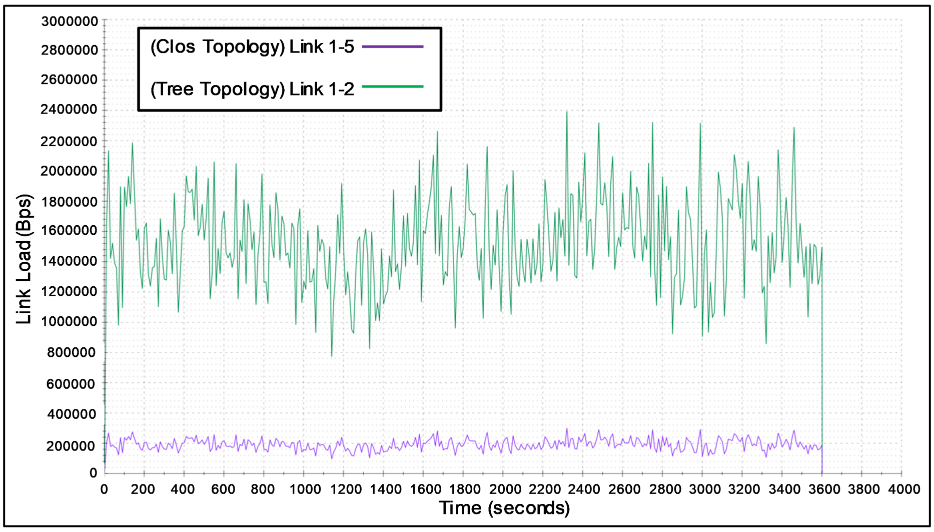



Fig. 6.12 Throughput comparison of links 1-2 and 1-5, respectively of the Tree and Clos topology

We tested also the migration capability of our platform simulating the migration of all the virtual machines from Pm1 to the eight servers of Rack5.

In this scenario, we test the migration of all the virtual machines from Pm1 to the eight servers of Rack5. Starting the migration at time t = 1801, we left only a second of time between two consecutive migrations, so the effects of the simulation are concentrated in a limited time-lapse. Figure 6.13 shows the percentage of the load of the link 1-2 (for the Tree topology) and of the link 1-5 (for the Clos topology). We chose these links because they are directly interested in the migration process. The Clos topology scales better than the tree one because the multipath routing strategy is more efficient and reduces the overhead generated by the migration processes. The load percentage of the core level links never exceeds the 1.3% of their transmission capacity.

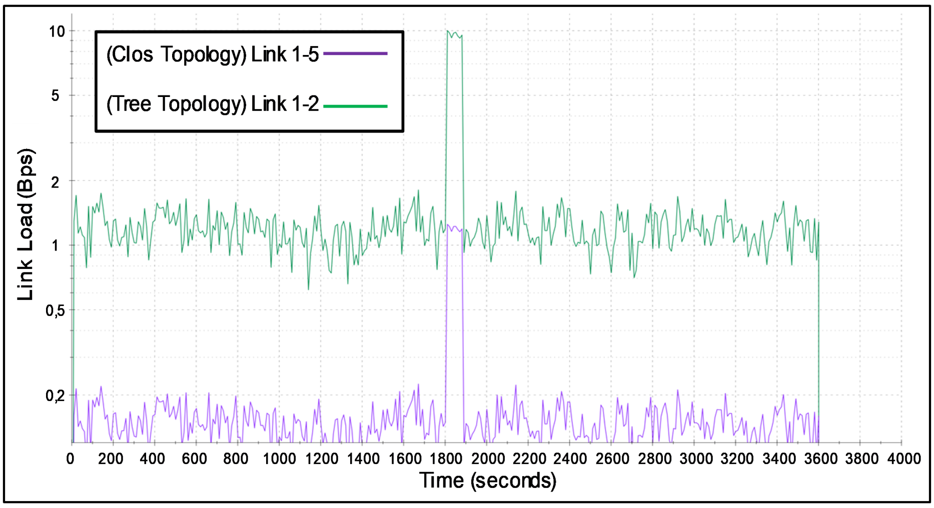



Fig. 6.13 Load percentage comparison between two links of Clos and Tree topology

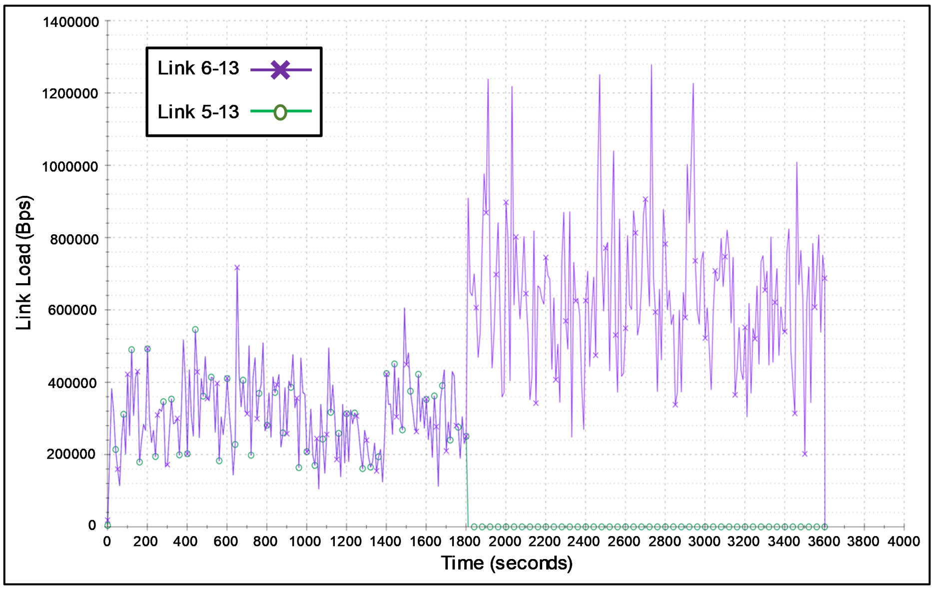



Fig. 6.14 Traffic redistribution after the link 5-13 fault

For both the topologies we tested the fault of the link connecting the first switch of the aggregation level with the first switch of the access level. It is superfluous to demonstrate the low capacity of the tree topology to react to the link faults, whereas in Figure 6.14 we show the redistribution of the traffic performed by the Clos architecture, from the failed 5-13 link to the working 6-13 link. The Clos topology can deal with the full fail of each switch in the aggregation level (single failure hypothesis) or up to three contemporary links fails inside the core level, without compromising the communications among the virtual machines.

### DCNs-2 vs GreenCloud: a qualitative comparison

A simulator modeling all the network equipment and nodes existing in a real DCN is useless if it cannot produce results in a time proportional to the complexity of the whole system. We measured both the execution time and the amount of memory needed to solve a complex scenario. We compared the resource consumption and the execution time with another ns-2-derived simulator, GreenCloud. We tried to reproduce as faithfully as possible the same topology scenario in both simulators, even if, in addition to the connections created for our tests, GreenCloud has to perform all the necessary computations and to install all the modules needed to estimate the energy consumption of a data center.

To execute this test, we used a virtual machine with Ubuntu 14.04 operating system, inside a host system equipped with an Intel Core i5 2.6 GHz CPU and 8 GB of RAM. The virtual machine used one core and4 GBB of the hosting system. We tried 8 variants of the previous Clos scenario, doubling the connections for each scenario, and the results are the average of 5 executions of the same scenario. The simulations lasted for 3600 seconds and we used the same traffic generators for all the network connections.

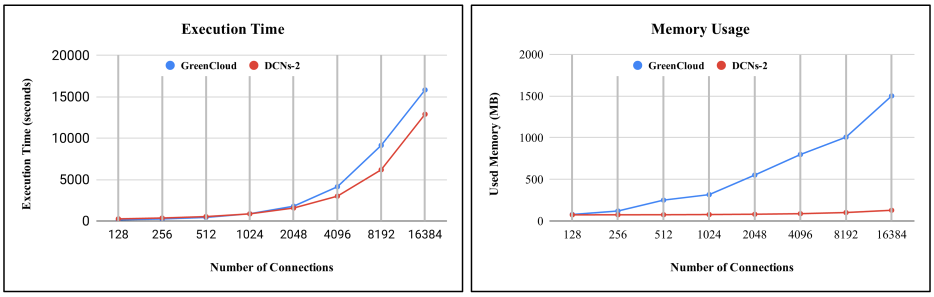



Fig. 6.15 DCNs-2 execution time and memory consumption

Figure 6.15 shows the execution time as a function of connections between virtual machines. As you can see in Figure 6.15, the time grows proportionally to the connections, proving our DCNs-2 simulator is a scalable system. Analyzing the system from the point of view of memory usage, we can notice that the amount of memory used is almost constant. The increase in network connections does not particularly affect the use of memory, exploited mostly by network devices. As stated before, the values of GreenCloud are so bad because the tests are soiled by the logic of the energy calculation, so it is less suitable than our simulator to represent large DCNs, having been born with a very different purpose. The disadvantage shared by all network simulators [76] arises when we have very complex topologies with a huge number of nodes and network devices. In the most complex scenario, GreenCloud requires 22.79% more execution time and more than ten times the memory of DCNs-2.

Currently, these drawbacks are reduced by the high availability of primary memory within contemporary servers.

### 6.1.5 Lessons learned

As shown in the paper, the proposed DCNs-2 simulator fits the needs of performance analysis and evaluation of virtualized resources deployed atop large DCN networks by overcoming the typical limitations of most widespread existing techniques in the field. Moreover, DCNs-2 allows simulating complete cloud environments, so that network analysts can model and consider all the main elements included in real DCN deployments, as well as the virtualized ones at the IaaS level. In fact, DCNs-2 has demonstrated to present a flexible and easily configurable simulation environment, suitable for modeling an arbitrary network infrastructure, thanks to the support of multiple simulated entities such as physical resources, virtual resources, and network assets. The extensive performance results reported in this paper demonstrate the feasibility of the proposed solution: DCNs-2 exhibits a memory consumption that grows only linearly with the size of the targeted scenario (primarily in terms of number of entities and network connections), while its processing time is always below three times the simulated time duration even for the most articulated and complex deployment environments. The encouraging results already obtained are stimulating our further research activities in the field along two primary lines. On the one hand, we are working to integrate our simulator with other softwarization tools, such as Mininet, to realistically mimic a complete testbed with physical deployment elements and virtualized network equipment powered by SDN and NFV. The new features and entities, such as virtual resources and allocation and migration policies, can drive the researchers in modeling complex test scenarios and in the analysis of the network with respect to the replication level and the deployment of SDN and NFV equipment. On the other hand, we are extensively testing, evaluating, and assessing the degree of realism of DCNs-2 with additional cloud data center management scenarios, in particular by considering virtualized resources of multiple data centers working together in a federated way.

## 6.2 Audit4Cloud: a platform for auditing cloud networking performance

Cloud computing infrastructures consist of worldwide fully interconnected data centers offering their computational resources on a pay-per-use basis to ease and fasten the development of sophisticated services and IT management systems on a global scale. After almost a decade, this is a consolidated trend; moreover, several cloud users are considering complex multi-site and multi-cloud deployments (through hybrid and federated cloud solutions) as a way to obtain increased reliability and/or to save costs by leveraging the potential of dynamic pricing schemes currently offered by all main cloud players.

From the perspective of practitioners and researchers working in this area, one key challenging and still open issue is how to realistically model intra-/inter-datacenter communications so to help cloud users to make unbiased informed decisions. Indeed, cloud communications are subject to several (uncontrollable) factors including intrinsic high variability and periodic re-adjustment, such as variations of the data path/number of hops as time passes by [77]. In addition, the dynamic pricing schemes adopted by most cloud providers could lead users to change the deployment of their resources in favor of more convenient tariffs [78].

That in its turn requires new tools able to calculate metrics (e.g., oriented to cloud networking, such as latency, bandwidth, delays), to store collected measurements, and to massage and present data through adequate data analytics functions. Some commercial services to benchmark and compare cloud providers are emerging in the market, such as Cloud Spectator and Cloud Harmony; however, these solutions still present several weaknesses. They are typically proprietary and not free, thus scarcely adoptable at least by the research community. From a functional perspective, they usually do not allow mining and drilling the history of collected measurements: typically they limit themselves to performing tests just in time, by offering to their user only monitoring information about a specific time instant.

To overcome these issues, this paper proposes a framework, called Audit4Cloud [79], that presents several novel elements of originality. First, it is free, based on widely available tools, and available to the research community as an open source project. Second, it includes big data facilities able to keep track of previous tests/measurements, thus helping users make better informed cloud selection decisions based on past quality and performance history. Third, we hope this project could become an unbiased open-source tool that enables anyone (with the role of third-party auditor) to analyze the performance of public cloud providers. Quantifying the drifts of the functional parameters of the cloud resources enables the drafting of delay patterns concerning bandwidth and latency between the machines of a specific cloud provider, which is of fundamental interest both for researchers and for companies that need to estimate the preferable plan for their own needs. Fourth, Audit4Cloud is original in the related literature because it has been already thoroughly tested with four main public cloud players, and this paper reports some interesting results collected so far (our entire dataset of measurements is available for the community for further study as a further contribution of our work).

### 6.2.1 Distributed architecture

The section concisely reports the primary design choices of our Audit4Cloud platform, which has proven to be an adequately complete tool for performing tests of network performance on any primary commercial cloud provider.

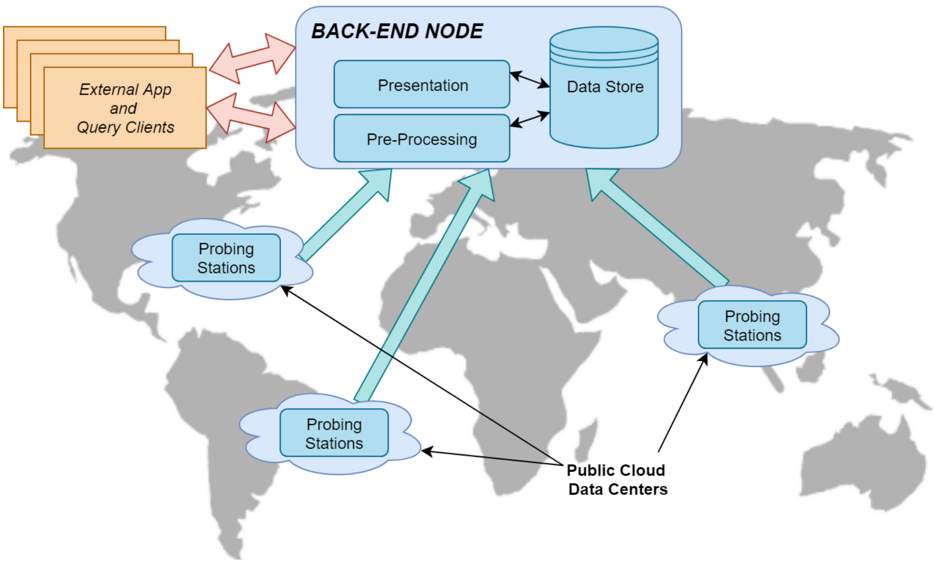



Fig. 6.16 Audit4Cloud overall architecture

The main goal of Audit4Cloud is to collect the statistics of network performance of any cloud provider. To this end, the core element of its distributed architecture is located inside the internal cloud network, from where it operates the performance tests. As you can see in figure 6.16, the virtual machines spawned in the data centers of interest periodically send report files containing results collected from the execution of a given test in a chosen data center. The latency measurement data are processed, stored, and made available for different purposes, e.g., visualization and exploitation as input for latency modelling.

To guarantee a high level of reliability, the data sent by the probing stations are stored in a distributed NoSQL database. The choice of a non-relational database is a key point to ensure the maximum level of decoupling from data types and vendors.

Moreover, the figure 6.16 emphasizes that we have taken great care of the data presentation unit during the development of Audit4Cloud. As a result, the tool turns out to be responsive even in the face of complex performance visualization queries involving many performance indicators. The Audit4Cloud front-end and back-end mask a series of forethoughts adopted to speed up the performance and minimize the waiting time introduced by data extractions from the remote database.

By delving into finer details, for collecting network statistics, we use virtual machines distributed within precise zones of the data centers we want to track. In our test applications, we take care of the concept of the Region that most commercial cloud providers implement in their own infrastructure. By region, we usually mean a geographical position where resources can be spawned. In addition to performance analysis, note that we have intensely worked on cost analysis of the resources used to perform the tests over most widespread public cloud providers. Often, within the same region, it is possible to employ a type of private addressing that usually has a cost much lower than the public addressing used to connect resources across different regions.

In addition, on virtual machines, we run automated applications for network performance tests, which employ well-known tools pre-installed in most operating systems. In particular, we developed two families of tests, one giving results about latency and the other about available bandwidth. Both cases save the related monitoring data in different files discriminated by the date, the time, and whether the associated test has run on resources inside the same region or not. The operations of probing and data sending are scheduled and automatically executed with no need of manual intervention.

#### Storage & Data Presentation

We chose the MongoDB database to persist the data coming from the probing stations. A database classified as NoSQL gives flexibility to the structure of the data. A data persisted in MongoDB is a document with its own schema, stored in JSON format. In non-relational databases there are not predefined schemes, this promotes the addition of new data types without changes to other components of the architecture. MongoDB uses the Aggregation concept to cover the lack of operators giving results in relation to fields of the data stored, such as the JOIN operator for the relational databases. Through a pipeline of operations made available by this mechanism, it is possible to formulate complex queries. For example, through the Group function, we can group multiple documents based on one or more fields and perform operations on them, while with the Sort operation we can express sorting criteria in the return of the results. Furthermore, MongoDB provides the ability to define Indexes on collections of documents. Small portions of the collection are saved in data structures that can be used by the pipeline operations to speed up the search.

The adoption of MongoDB increases the scalability of our platform because we exploit MongoDB clusters of nodes, thus improving performance and increasing the available storage space. The Sharding process actually distributes the database on multiple nodes of a Sharded Cluster. Within a MongoDB cluster, we exploit three core components:

- Config Server: it acts as a controller, managing the cluster structure and data distribution among the nodes. In a production environment, it can be replicated to prevent becoming a single point of failure.
- Shard Server: it is the real data repository. To guarantee reliability in case of faults, the server can be replicated.
- Mongos Router: is a single entry point for the clients that execute the operations on the cluster as if it were a single database.

The Replica Sets in MongoDB are groups of instances maintaining the same data set. The replication provides redundancy and high availability to the data. All these stratagems made MongoDB the solution we prefer because it guarantees the best performance in terms of speed and latency, also gives flexibility to the whole infrastructure.

Our data collection provides a statistical basis on which to elaborate complex models describing inter-datacenter communication delays. The replication of the persistence modules ensures us access to data in their entirety, furthermore, we provide for the end users an intuitive and fast graphically interface to run complex queries. To this purpose, we developed a front-end application through the React framework and arranged several types of charts to show different types of results. React is an open-source JavaScript library particularly suitable for the realization of single page applications. With this tool, it is possible to develop quickly the web pages used for the presentation of data and native applications for the mobile world Android and iOS.

In addition to the development of a clear graphic interface, other back-end software components contribute to an easy visualization of the requested data. The back-end layer runs on Node.js, a JavaScript runtime environment with an asynchronous event-driven engine. This means the application makes a request of some data and then can perform next tasks, without blocking itself on the request, before receiving the reply. In Audit4Cloud platform, Node.js performs the calls to the MongoDB database and it handles the clients' requests. Through the use of the Node Package Manager (NPM), we found many ready-to-use modules to take care of common programming needs.

To complete the MERN stack (MongoDB, Express, React, Node.js) we used Express.js, a web framework for Node.js which implements the routing for requests coming from customers, determining how an application responds to a client call to a particular endpoint, which is a URI (or a path). Even being a minimalist framework, it provides the Middleware mechanism to process requests in a complex way. The Middlewares in Express.js are the external software components that, taking as input Request and Response of a web call, perform operations on them, returning the result to the requesting application

### 6.2.2 Audit4Cloud implementation

In this section, we report some primary implementation insights about our original Audit4Cloud platform. We start with the presentation of the public cloud environments and continue with the explanation on how we get and store the statistical data. In conclusion, we introduce the implementation of the presentation layer.

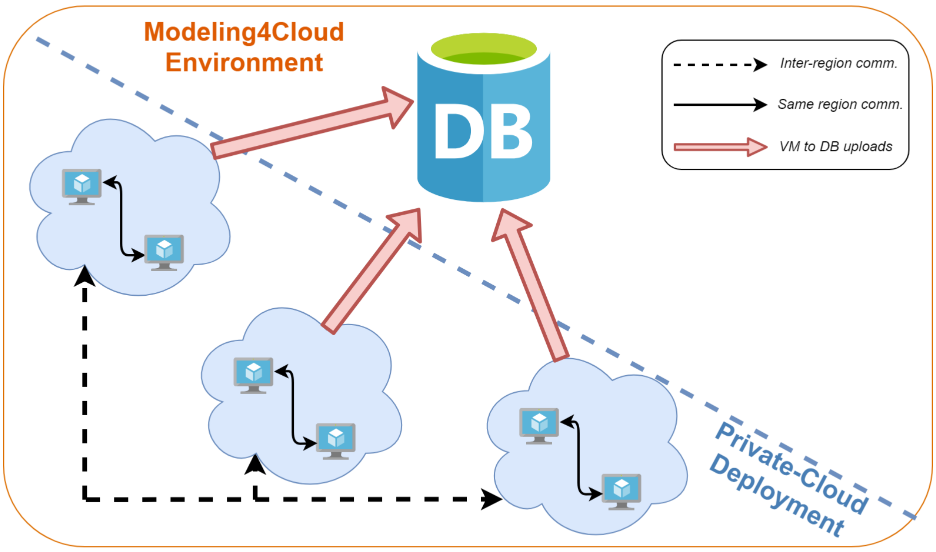



Fig. 6.17 Probing stations deployment

As you can see in Figure 6.17 reporting our virtual machines' deployment to test all regions of a Cloud provider we need a machine for each geographic area to carry out tests between it and machines in other areas and an additional machine for each geographical area to perform a performance test between two instances inside the same area. The virtual machines gathering the data run the Ubuntu 16.04 LTS operating system. We chose a Linux distribution for the built-in mature shell support, that helps us in scheduling the data collection and the result transmission to the database. Our scripts use open-source protocols and tools available for most operating systems in order to execute and automate information gathering and sending to the data store. The Audit4Cloud platform performs two types of network tests.

**TCP Ping**: usually the *ping* Unix command estimates the time of ICMP packets to reach a network host and go back (also called Round Trip Time, RTT). By our choice, we prefer to use a tool that would allow monitoring packets to travel on TCP protocol. The choice regards the Cloud providers, that often block the ICMP traffic in their networks to avoid DoS (Denial of Service) attacks on virtual machines. Most suitable tool for our purpose is the **hping** application [2]. It isn't only able to send ICMP echo requests, but it also supports TCP and UDP protocols and traceroute method to find the path of the packets. More, in its statistics, we can find the parameter *Time To Live*, useful in building a delay model of the network. The TTL parameter is not shown in the output of other similar-purpose applications such as qperf, psping and paping. With the hping tool, we are able to take both the RTT and the TTL statistics of the TCP packets.

**TCP Bandwidth**: this test needs two open doors on two virtual machines because it creates a TCP upon them channel and puts the machines in communication by transmitting data and measuring the statistics. Using **iperf3** [3] we can get both the bandwidth and the number of lost packets, and subsequently transmitted again, during a test communication. For this test, you need to install two parts of the application, a client and a server, on the two virtual machines. The tool allows us to control configuration parameters, such as duration, execution sub-intervals, and the number of parallel connections. With the iperf3 tool, we get statistics about the bandwidth and the number of packets lost on a channel. Regarding the test on the bandwidth, we would like to point out that the duration is an important parameter to choose with attention during definition of a test. During the communication, it could occur the *TCP slow-start* effect, the result of a network congestion control algorithm. This consists in starting a connection by first transmitting a few packets and progressively increasing them according to the network response. Thus, a short duration test could be distorted by the *non-real* slowness of the connection during the early stages of the transmission.

After a test execution, the system saves a new entry in a CSV file (Comma Separated Value). The results are saved in different files based on the type of test and the date of execution. Both tests save a subset of common information: the Cloud provider name, the source and destination virtual machine geographic zone identifier, the source and destination IP address of the virtual machines, and the timestamp. The latency tests add this information to the common ones: the incremental number of the ping sequence, the Time To Live (TTL) parameter, and the Round Trip Time (RTT) in milliseconds. Instead, the bandwidth tests

[2]http://www.hping.org/
[3]https://iperf.fr/

add: the bandwidth in MB/s, the duration of the test in seconds, the number of parallel connections, and the megabytes of data exchanged.

At the end of each day, the virtual machines send these report files to the database and delete them on local machines, to limit the disk space used on each probing station.

As anticipated, the installation and execution of the tests are completely automated. A series of bash scripts, connecting to the probing station via SSH protocol, install the necessary software to carry out the tests and trigger their execution. For each Cloud provider, we provide a configuration file for tuning the parameters of the test scenario.

The common information for each virtual machine is:

- The private key to access the machine (.pem file),
- The geographical area on which the machine is hosted
- The public IP address or the private one, if you intend to run a test inside the same region.

In a configuration file, for each provider we can specify the following information that affects the execution:

- For Ping test: the starting port in case of consecutive hping tests, the time interval between them, the bi-directionality;
- For Bandwidth test: the port, the time interval between tests, the duration, the number of parallel connections, the bi-directionality.

At the end of each day (at midnight UTC time zone) each machine sends the results of all the tests carried out during the day to the database. In our MongoDB there are two collections containing the data related to respectively to the TCP ping (collection **pings**) or to the bandwidth (collection **bandwidth**).

To store the data in a smart way and to make the database responsive also in case of heavy computation due to complex queries, we made some modifications to the default behavior of MongoDB. Initially, we implemented the database as a single instance without any type of replication or sharding (see Section 6.2.1). The queries made to the early versions of the system took a very long time to answer, already after a relatively short analysis timespan. For example, to store the result of a test executing a ping per second, a single machine would produce 86400 documents a day. Below, we will see different improvements whose integration allows us to have an acceptable latency during the execution of queries of any complexity.

First, we created a MongoDB **Cluster** and distributed the data on two separate **Shards**. The application can now scale horizontally, in fact having the database on multiple nodes increases the computing power and consequently decreases the query response time. We also use the replication, through the **Replica Set**, a concept proper of the MongoDB world. The replication of the Shards gives fault resistance and reliability to the whole system. Furthermore, we divide the Shard on a Cloud provider basis, so the platform can answer to parallel queries regarding different providers.

In MongoDB, a complex query may require the analysis and the aggregation of several documents, copied in main memory. If the requested data covers a long time-lapse, it may happen that the size of the documents exceeds the available space in memory, forcing the system to use paging mechanisms that further slow down the sending of a response. To limit this drawback, we added **Indexes** to the collections, with the opportunity to exploit them during the definition of aggregation pipelines that return the result of complex queries.

Last shrewdness to speed up the performance of the platform is the **Precomputing** of the latency tests. Clustering and indexing yield to good results concerning queries about bandwidth statistics, but complex queries on ping statistics on a long time-lapse still have an unacceptable response time. For all queries on latency data, the system calculates the averages of the RTT values, for a given station. We add a new collection, **pingdayavgs**, containing the daily latency averages of a station towards another. The platform counts these average values before storing data in the database and exposing them to the final users. In this way, we introduce overhead before storing data, not impacting the users' experience; furthermore, the parallelization of Shard per Cloud provider speeds up the process. Now the answer time is acceptable also for complex queries on latency data. Each document of the new collection contains the Cloud provider name, the name of the sending and the receiving machine's geographical area, the date, the average of the performed tests, and the number of the performed tests.

Last but not least, we protect the database with a double authentication level. The **modules' Authentication** takes place among the distributed software components of the MongoDB cluster, using a private key mechanism, instead we provide the **clients' Authentication** with private credentials to use the database and to read/write data on the back-end.

The last layer of the Audit4Cloud architecture is the back-end and it has three main purposes. The first one is to receive data from virtual machines, exposing special endpoint for this aim. We create the logic behind the endpoints to which the machines deliver the report files using the Multer [4] plug-in of Node.js., which makes assertions about the data received and, based on the result, sends the document to the correct collection. The back-end

[4]https://github.com/expressjs/multer

also performs the data conversion. For each file received, a process performs the extraction and the transformation for writing the data into the database. In the case of a file reporting latency values, this module aggregates the results. For the operations on the database, we use Mongoose [5], a module of Node.js for object modeling, acting like the ORM (Object Relational Mapping) support for the other programming languages. The last duty of this component is to provide REST API useful for data presentation. Some features available for viewing by web browser applications are:

- Average of each provider for each region, considering if the test executed to different regions or not (custom test),
- Average of all tests between zones of the same provider (testing a provider),
- Average of all tests from one area to all the others (testing an area).

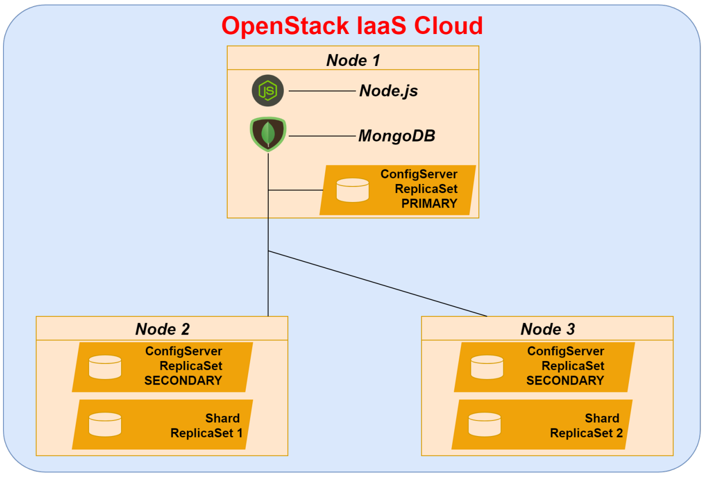


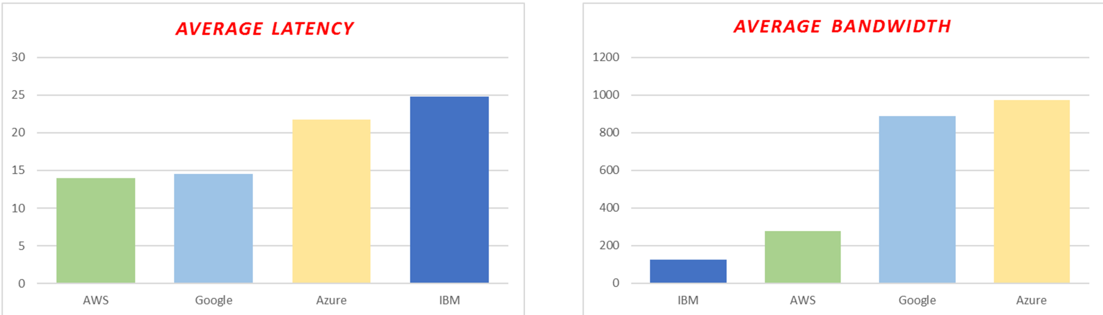

Fig. 6.18 Backend deployment

Most of the application business logic of the whole platform is inside the back-end. We deploy the back-end layer on three virtual machines spawned on the OpenStack IaaS, made available by the University of Bologna. The machines running the replicated instance of MongoDB and an instance of Node.js have the following characteristics: 2 vCPUs, 4 GB of RAM and 100 GB of HD useful for storing MongoDB data (see Figure 6.18).

[5]https://mongoosejs.com/

### 6.2.3 Results and lessons learned

The results section first presents an analysis of the costs and performance of Audit4Cloud; then, to show its feasibility in practical cases of commercial interest, we extensively report about our Audit4Cloud tests when working with the Amazon cloud provider.



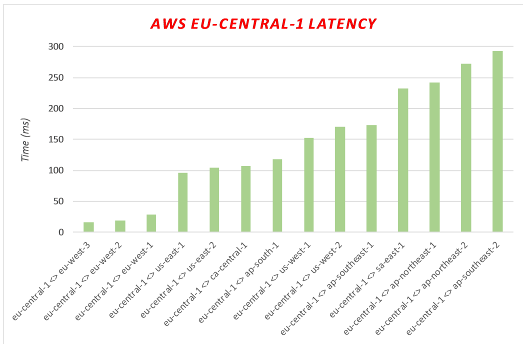

(a) Average latency expressed in ms (b) Average bandwidth expressed in ms

Fig. 6.19 Public cloud providers performance comparison

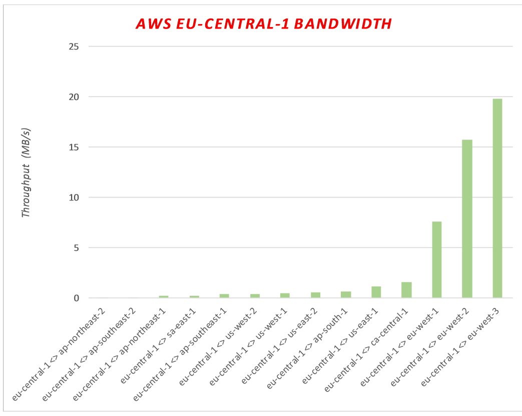
We worked to minimize the intrusion that data gathering tools have on virtual machines. In fact, our experiment was successfully completed by using the following minimum features for virtual machines on each provider: 1 vCPU, 1 GB of RAM, 10 GB of HD.

Considering the high number of regions for each provider, it is clear that a high economic budget would be needed to perform tests on any possible cloud provider. For practical motivations, therefore, we carried out extensive tests only on the Amazon AWS provider (to demonstrate the usability and scalability of the platform); while more superficial comparison tests have been run over all the 4 providers: Google Cloud, Amazon AWS, Microsoft Azure, and IBM Cloud. Fig. 6.19a, we show the average latency for each cloud provider. Fig. 6.19b reports about average bandwidth.

Our tests were performed by paying the commercial fee for using the involved cloud resources: 5.42 \$/month for AWS, 7 \$/month for Google Cloud, 13.5 for Azure and 27 \$/month for IBM Cloud. Also based on this cost difference, we chose the Amazon provider for a thorough test of network performance. The complete monitoring of the Amazon provider generates about 1 GB of data every 5 days. Obviously, the network performance between zones is influenced by their geographical distance. We performed our tests during the month of September 2018, on all the regions of the Amazon cloud provider. We forecast that the number of regions is constantly growing, but we tested all the 15 regions available at the experiment time.




(a) Latency analysis of AWS region eu-central-1 vs all other AWS regions




(b) Bandwidth analysis of AWS region eu-central-1 vs all other AWS regions

Fig. 6.20 AWS regions performace comparison

Considering an average value for every possible combination between two different regions, we would have a graph of 105 values, that for the sake of clarity we do not report entirely here; for the sake of readability and conciseness, we report here the data regarding the network performance from a region to all others. Fig. 6.20a shows the RTT in milliseconds between eu-central-1 region and the others, while in Fig. 6.20b we can see the available bandwidth in MB/s between the same regions used in the previous experiment. The previous figures denote an aggravation of network performance between regions that are not directly connected or geographically distant.

Our Audit4Cloud framework fits the needs of an open tool for the auditing of public cloud provider network parameters. It overcomes the non-negligible limitations of existing tools and services, such as the limitations in terms of stored/available history and the proprietary code base (making it hard to exploit them for research purposes).

The encouraging results already obtained are stimulating our further research activities in the field. On the one hand, we are using collected results to feed our realistic network modeller and simulator [77]. On the other hand, we are working to add also a processing auditing probe to assess also computing capabilities.

# Chapter 7

# Industrial Control Systems Cybersecurity: a black-box approach

This chapter introduces our collaboration with the Department of Electrical and Computer Engineering of the University of Patras. The security in the exchange of information in the low levels of integration architectures is a fundamental requirement to guarantee the correctness of production plants operation. For this reason, we implemented a security mechanism for testing the operational correctness of the OPC UA Server implementations, a very common communication protocol for monitoring and controlling remote machines in production environments. Our findings increase the reliability of the software in the production plant and the safety of humans at work. The approach chosen for our experiments is fuzz testing, a technique based on random, invalid, or unexpected data input packets in order to monitor the behavior of the counterpart. Usually, this technique is used for protocols for which the tester knows the packet input structure.

## 7.1 Sulley: the blind fuzzer

The fuzz testing procedures can be classified in different ways, but their common goal is the detection of bugs in the software of a System-Under-Test (SUT) that can lead to memory errors, crashes, unwanted timeout triggering, and operation delays. Taking into consideration the classification relying on the protocol (under-test) awareness, we have three main categories of fuzzer, as depicted in Figure 7.1: black-box, grey-box, and white-box. The last set of fuzzers includes cases in which the tester has complete knowledge of the SUT, such as he knows the inner details of the server implementation, the documentation, the sources, the built program, and even the developers. As opposite, black-box approaches have no knowledge of the SUT, so they are blind about its internal implementation. In the middle, the grey-box fuzzers know something about the protocol under test, mostly relying on the binary code from which infer source code and operation information. We chose to implement a black-box fuzzer because the only thing these fuzzers can rely on is the server answers (the response packets) and the server behavior; usually, the only knowledge available when dealing with corporate protocols. The software employed on work machines is mostly proprietary implementations, of which the internal functioning, the source code, datasets of examples are not provided. Using a black-box fuzzing approach, the SMEs and the enterprises that want to employ new machine communication protocols could be more sure of implementation correctness. A tester using a black-box fuzzer knows the formal definition of the protocol and the only assumption is that the server implementation follows that protocol.


Fig. 7.1 Fuzzer classification based on protocol awareness

We used the modern Sulley fuzzer [1] to test the correctness of OPC UA server implementations. The goal of the framework is to simplify not only data representation but data transmission and target monitoring as well. Sulley is capable of monitoring the network and the health of the target and reverting it to a good state if some running tests crashed the server. Sulley detects, tracks, and categorizes the incurred faults, it can determine what unique sequence of test cases triggers faults. The fuzzer can fuzz in parallel, significantly increasing analysis speed. The operation of this framework essentially consists of three steps. The first one is the *data representation*, the tester has to break down the protocol into individual requests and represent them as blocks in Sulley. The second step is the linking of requests to form a *session*. In this step, the tester should attach the monitors (such as network and process monitors) to the fuzzer and commence fuzzing. The last step involves the *postmortem analysis*, i.e., the review of the data generated from the session execution. In case of crashes, it is recommended to replay the cases involved in the crash and examine results, server responses, and status.

[1]https://github.com/OpenRCE/sulley

## 7.2 The six fuzz phases

Whatever the approach used for fuzzing, the six basic steps shown in Figure 7.2 are covered by all fuzzing tools. In the next sections, we retrace the development of our fuzzing tool focusing on the common steps that this process requires for analyzing a network protocol.

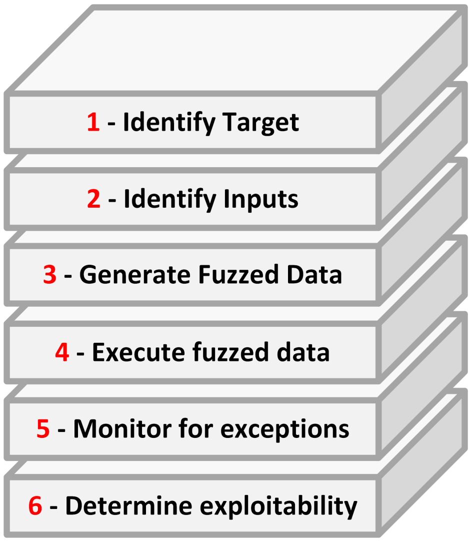



Fig. 7.2 The six phases of the fuzz testing [80]

### 7.2.1 Target identification

To select the right fuzzing approach you must have a target to test, so the first choice was to find what implementations to put under examination. Due to the absence of assumptions on the target system, we chose to test two famous open-source implementations of the OPC UA server, the Open62541 [2] (maintained by many companies including the Fraunhofer IOSB research center[3]) written in C/C++, and the Python FreeOpcUa [4] written in Python. We have chosen these implementations for their completeness in the realization of all the functions that the OPC UA protocol provides and for the simplicity of use and installation.

We reiterate that, since our tool has no preliminary assumptions, it is executable on any implementation of the OPC UA protocol. For example, it can be used against any of the open-source implementations shown in Figure 7.3. We want to point out that it is more frequent to find errors in proprietary implementations as the catchment area is lower and consequently the reports of bugs and vulnerabilities, partially covered in open-source implementations.

[2] https://github.com/open62541/open62541
[3] https://www.iosb.fraunhofer.de/
[4] https://github.com/FreeOpcUa/opcua-asyncio

Even if we expect to find fewer vulnerabilities than in the proprietary versions, we preferred to start with open-source protocols, in which all the features are implemented, to build our fuzzing tool in a complete way and to be able to freely analyze the code in case we encounter errors and bugs.

### 7.2.2 Input identification

Starting to think about what messages to test, we organized the set of all possible inputs that in *allowed* and *not allowed* subsets, as in Figure 7.4. We are not interested in not allowed inputs because this is a mature protocol and the server will discard packets with "not allowed" values inside. For the same reason, we did not test the cases in which the user is not authenticated. We muted the allowed values of valid packets using the heuristics of Sulley fuzzer. Thus, we tested the edge values for each packet field and interesting combinations of values chosen by Sulley (based on the data type of the field), furthermore, we manually added some values of interest for some fields. With Sulley, it is possible also to test all the possible values for each field, but we must consider this case wisely because, for example, a field containing a 4 bytes Integer value will have 4 billion possible mutations.

| Name | Language | License | Client/Server | Link |
|---|---|---|---|---|
| open62541 | C | MPL-2.0 | Client and Server | http://open62541.org/ |
| UA.NET Standard | C# | GPL, RPC for OPC Foundation Members | Client and Server | https://github.com/OPCFoundation /UA-.NETStandard |
| node-opcua | JavaScript | MIT | Client and Server | http://node-opcua.github.io/ |
| FreeOpcUa | C++ | LGPL | Client and Server | http://freeopcua.github.io/ |
| Python FreeOpcUa | Python | LGPL | Client and Server | https://github.com/FreeOpcUa/opcua-asyncio |
| OpenOpcUa | C++ | Cecill-C, source code access costs a one-time fee for non commercial use and a paid support and maintenance for commercial use. | Client and Server | http://www.openopcua.org/ |
| OpenScada UA Interface | C++ | GPL | Server | http://oscada.org/websvn/... |
| ASNeG | C++ | Apache | Server and Client | Git Repository |
| Eclipse Milo | Java | Eclipse Public License | Stack / Client / Server | https://github.com/eclipse/milo |
| opcua4j | Java | Creative Commons 3.0 BY-SA, depends on redistributable jar-files from the OPC Foundation | Server | https://code.google.com/p/opcua4j/ |
| uaf | C++/Python | GNU Lesser General Public License | Client (wrapper over proprietary sdk) | https://github.com/uaf/uaf |

Fig. 7.3 OPC UA open source implementations

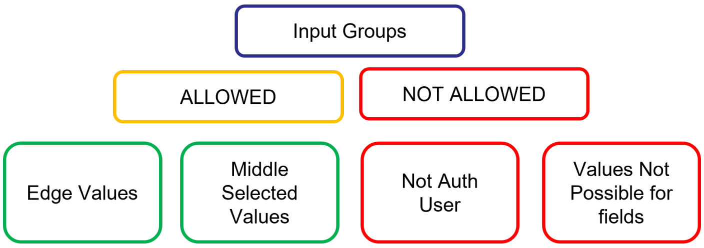



Fig. 7.4 Input classification in fuzzing tests

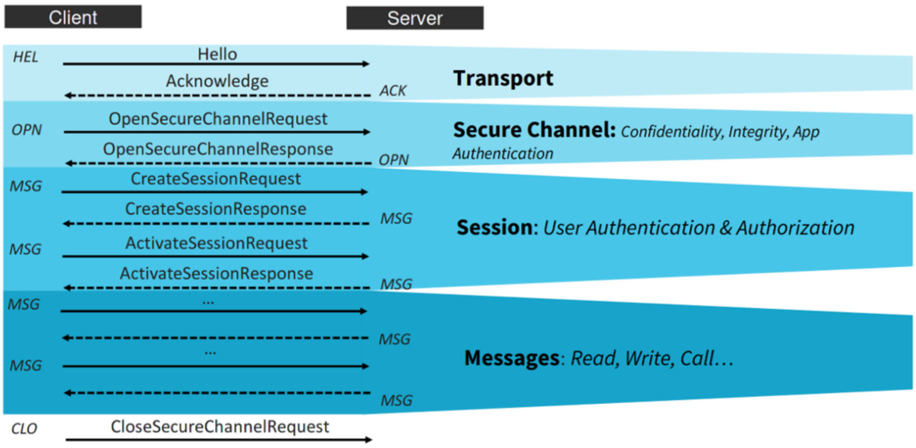



Fig. 7.5 OPC UA establishing secure channel connection

We choose to split the test cases into two macro groups. The *implementation* tests regard the service sets of the target. The second group of tests will evaluate the *information model*, trying to read/write variables and execute methods of the specific custom information model. In order to read/write variables and execute methods, we must first establish a secure connection. When our tool is testing the custom information model, we do not test the packets used to establish a connection to avoid a proliferation of useless tests. In particular, when the fuzzing tool is testing the reading/writing of custom variables and/or custom method execution, we do not test the messages shown in Figure 7.5, used to set up a secure connection.

Keep in mind that this is a crucial step in protocol fuzz testing because a too large input base will lead to test cases unnecessarily long while failing to locate potential sources of input values can severely limit testing reliability.

### 7.2.3 Mutations generation

We generated the input messages respecting the official specification of the OPC UA protocol and using the Sulley primitives. We exploited the Wireshark [5] packet capturing tool to make comparisons between our implementation and the fields of the packets exchanged between client and server.

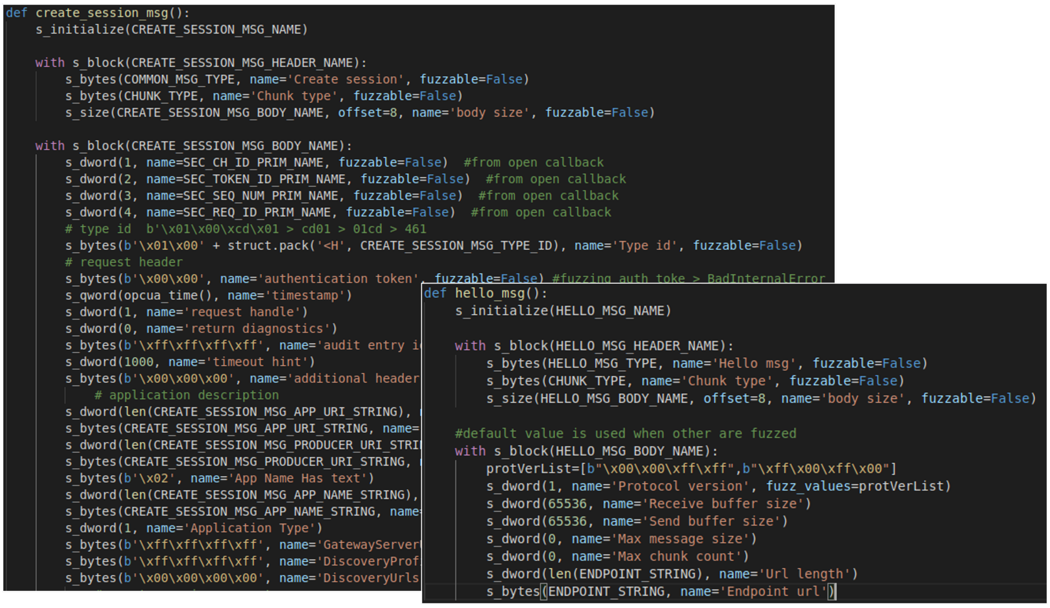



Fig. 7.6 Implementation of OPC UA packets using Sulley primitives

Figure 7.6 shows the implementation of some packet fields using the Sulley primitives. For each field in the packets, we gave a default value that will be used when that field is not fuzzed. To analyze the challenge/response parts of the protocol, we used callbacks called between two messages. Basically, they are functions that read the server response and pass needed information from the first message to the second.

### 7.2.4 Running the fuzzer

After identifying the input vectors, fuzz data must be generated. At this point, we had to make a decision about mutating packet fields, and we had three choices: taking predetermined values, mutating existing data, or generating data dynamically depending on the target. Being OPC UA very customizable protocol, we did not rely on an existing data dataset which could be very different from case to case, but we muted the values of the packet fields in an arbitrary way, using the Sulley heuristics and selecting the most interesting values for every single

[5] https://www.wireshark.org/

field. The Sulley engine applies the automation in this process building a chain of messages for the session under test. For example, our chain to establish a secure connection includes the Hello, Open, Create Session, and Activate Session messages. For every message in the chain, a packet will be sent, from client to server, with one of its fields changed according to what is specified in the heuristics or in the set of other interesting values added manually.

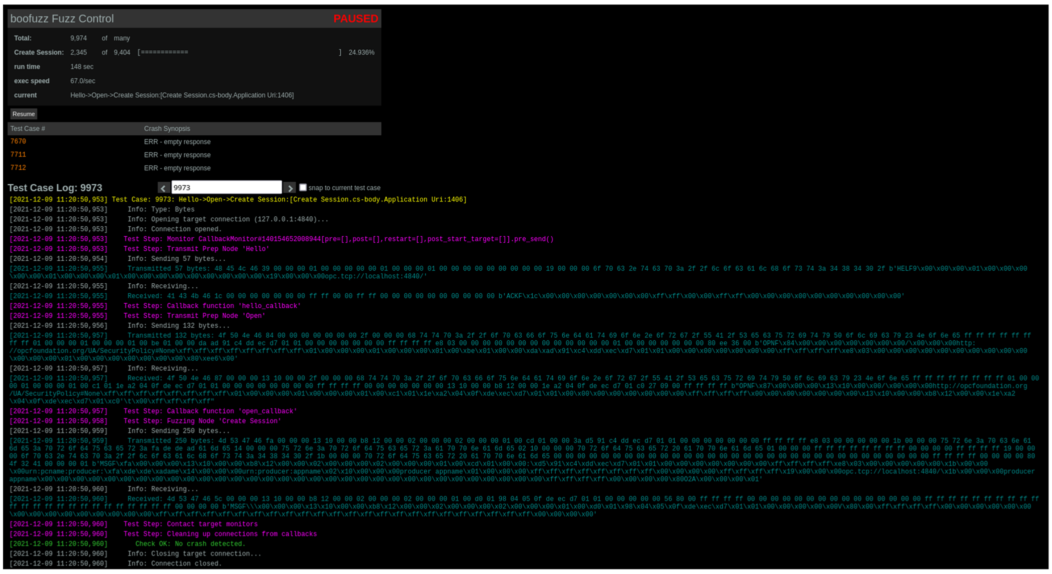


Fig. 7.7 Web dashboard of a running Sulley test

The Sulley session class has a built-in minimal Web server that is hard-coded to bind to port 26000. Once the *fuzz()* method of the session class is called and the fuzzing session starts, the tester can see the progress of the fuzzer including intermediary results. The previous screenshot in Figure 7.7 shows the web dashboard reporting the log and the number of all the test cases. As you can see, Sulley gives us the possibility to stop/resume the tests and to repeat a specific test case.

Figure 7.8 depicts the main steps in the fuzzing on the information model. To test the writing of a custom variable, we first send messages without mutations to accomplish the session creation and security key exchange. In this case, these messages are not mutated because we want to test the custom model, not the usual protocol messages (tested with the other test set). After establishing a secure connection and session, we start to fuzz the writing of the variable under test. As you can see in Figure 7.8, we use the first step to browse the information model (we want to recall that we did not make any assumption on the server Address Space and Information Model), in the second step we write the custom variables with fuzzing values and execute custom methods with mutated parameters.

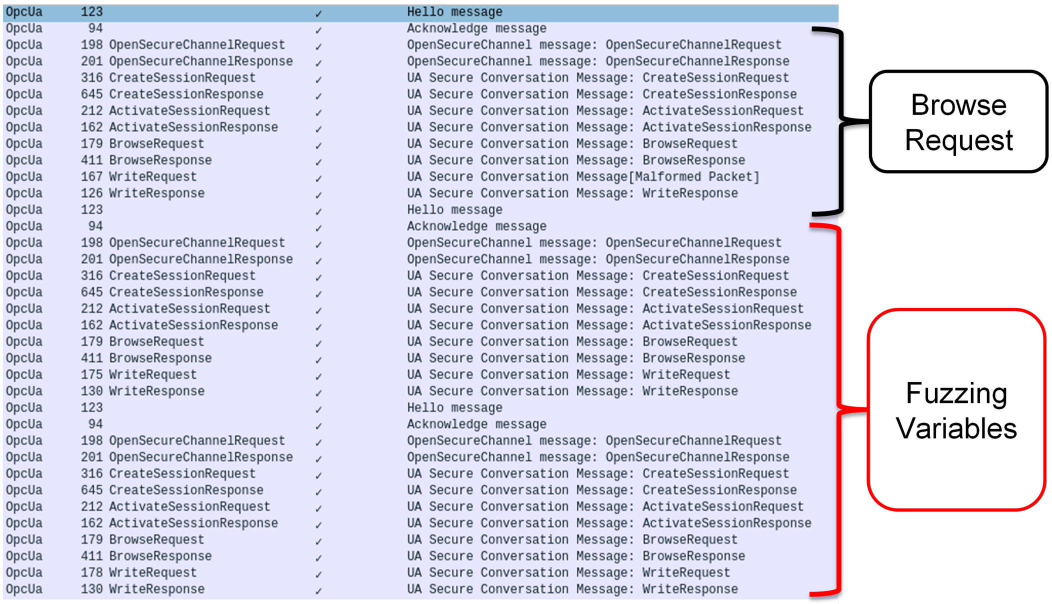
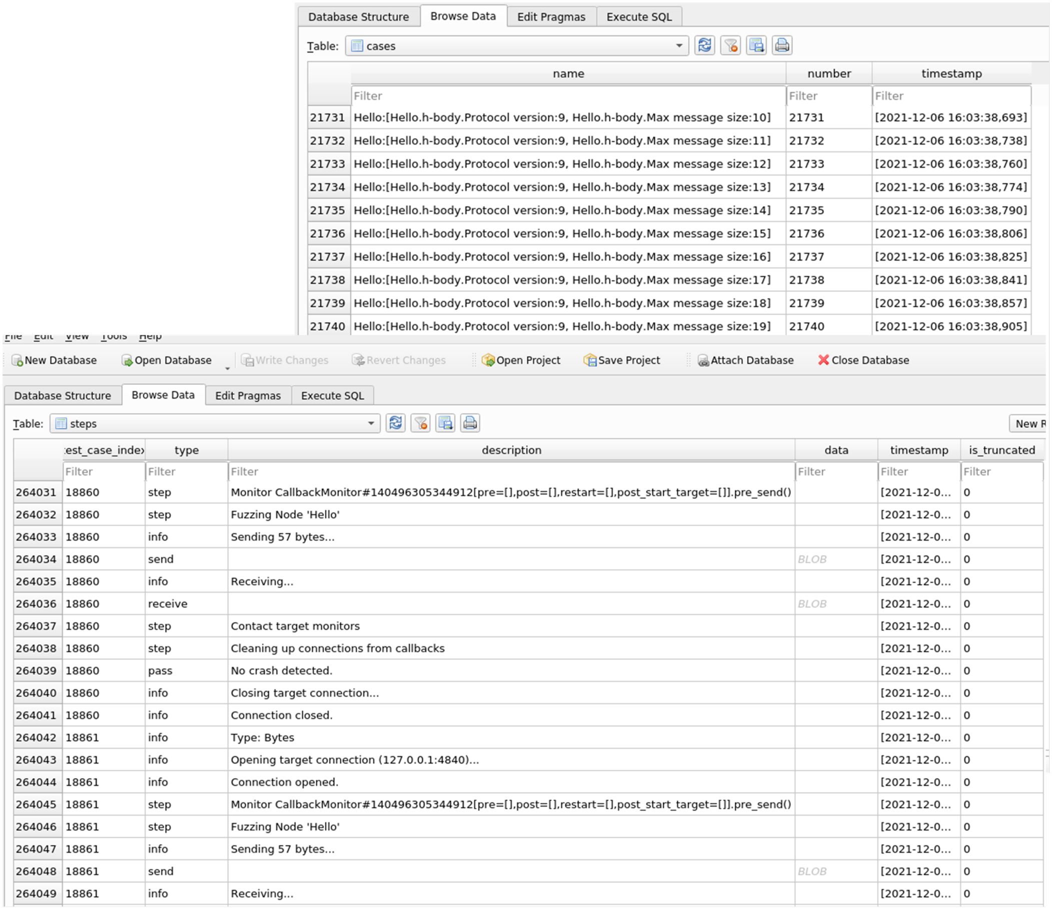



Fig. 7.8 OPC UA custom information fuzz testing

### 7.2.5 Monitor for exceptions and exploitability

The Sulley fuzzer provides us with a DB that collects all the messages sent and all the test cases tested for a run. Figure 7.9 shows two different tables inside the same DB regarding a fuzz testing chain.




| | name | number | timestamp |
|---|---|---|---|
| 21731 | Hello:[Hello.h-body.Protocol version:9, Hello.h-body.Max message size:10] | 21731 | [2021-12-06 16:03:38,693] |
| 21732 | Hello:[Hello.h-body.Protocol version:9, Hello.h-body.Max message size:11] | 21732 | [2021-12-06 16:03:38,738] |
| 21733 | Hello:[Hello.h-body.Protocol version:9, Hello.h-body.Max message size:12] | 21733 | [2021-12-06 16:03:38,760] |
| 21734 | Hello:[Hello.h-body.Protocol version:9, Hello.h-body.Max message size:13] | 21734 | [2021-12-06 16:03:38,774] |
| 21735 | Hello:[Hello.h-body.Protocol version:9, Hello.h-body.Max message size:14] | 21735 | [2021-12-06 16:03:38,790] |
| 21736 | Hello:[Hello.h-body.Protocol version:9, Hello.h-body.Max message size:15] | 21736 | [2021-12-06 16:03:38,806] |
| 21737 | Hello:[Hello.h-body.Protocol version:9, Hello.h-body.Max message size:16] | 21737 | [2021-12-06 16:03:38,825] |
| 21738 | Hello:[Hello.h-body.Protocol version:9, Hello.h-body.Max message size:17] | 21738 | [2021-12-06 16:03:38,841] |
| 21739 | Hello:[Hello.h-body.Protocol version:9, Hello.h-body.Max message size:18] | 21739 | [2021-12-06 16:03:38,857] |
| 21740 | Hello:[Hello.h-body.Protocol version:9, Hello.h-body.Max message size:19] | 21740 | [2021-12-06 16:03:38,905] |

| | est_case_inde) | type | description | data | timestamp | is_truncated |
|---|---|---|---|---|---|---|
| 264031 | 18860 | step | Monitor CallbackMonitor#140496305344912[pre=[],post=[],restart=[],post_start_target=[]].pre_send() | | [2021-12-0... | 0 |
| 264032 | 18860 | step | Fuzzing Node 'Hello' | | [2021-12-0... | 0 |
| 264033 | 18860 | info | Sending 57 bytes... | | [2021-12-0... | 0 |
| 264034 | 18860 | send | | BLOB | [2021-12-0... | 0 |
| 264035 | 18860 | info | Receiving... | | [2021-12-0... | 0 |
| 264036 | 18860 | receive | | BLOB | [2021-12-0... | 0 |
| 264037 | 18860 | step | Contact target monitors | | [2021-12-0... | 0 |
| 264038 | 18860 | step | Cleaning up connections from callbacks | | [2021-12-0... | 0 |
| 264039 | 18860 | pass | No crash detected. | | [2021-12-0... | 0 |
| 264040 | 18860 | info | Closing target connection... | | [2021-12-0... | 0 |
| 264041 | 18860 | info | Connection closed. | | [2021-12-0... | 0 |
| 264042 | 18861 | info | Type: Bytes | | [2021-12-0... | 0 |
| 264043 | 18861 | info | Opening target connection (127.0.0.1:4840)... | | [2021-12-0... | 0 |
| 264044 | 18861 | info | Connection opened. | | [2021-12-0... | 0 |
| 264045 | 18861 | step | Monitor CallbackMonitor#140496305344912[pre=[],post=[],restart=[],post_start_target=[]].pre_send() | | [2021-12-0... | 0 |
| 264046 | 18861 | step | Fuzzing Node 'Hello' | | [2021-12-0... | 0 |
| 264047 | 18861 | info | Sending 57 bytes... | | [2021-12-0... | 0 |
| 264048 | 18861 | send | | BLOB | [2021-12-0... | 0 |
| 264049 | 18861 | info | Receiving... | | [2021-12-0... | 0 |

Fig. 7.9 Sulley fuzzer output database

The **Cases** table contains a test case for each row, and we have the complete chain under test (with the mutated field underlined), the number of the test, and the timestamp in the columns. The **Steps** table includes the state of the fuzzer for each row, i.e., the description of the step the data sent/received, the timestamp, and if the SUT has crashed or no error has been found.

## 7.3 Lessons learned

With our first fuzzer milestone, we provide a flexible, reusable, and homogenous way to fuzz the many implementations of OPC UA protocol. Although in the world of fuzz testing there is no single winner in choosing the most suitable instrument, we chose Sulley for its characteristic of being a black-box fuzzer, capable of testing challenge/response protocols. We believe that fuzzing is a must approach for protocols used in industrial environment lower layers in order to ensure better reliability and protect machines and humans. The cybersecurity is essential in reliability because it helps in assuring that a malicious agent does not take control of machines or assets due to bugs and code vulnerabilities that can be identified in advance. Our tool is a pioneering tool in identifying operational errors when using the OPC UA protocol, helping to ensure the correctness of both open source and proprietary protocol implementations. With next steps, we are planning to cover all the default services of the OPC UA protocol, integrating a way to fuzz the pub/sub features. To automate the error recognition and analysis steps, we will add a gray-box fuzzing tool downstream in those cases where we have more knowledge of the server response only, in order to better analyze the code responsible for bugs and vulnerabilities found by the black-box fuzzer.

# Conclusion

The successful integration of the IT and OT layers in industrial environments is enabling the implementation of the I4.0 transition. New platforms are emerging to tackle convergence and research communities, such as CPS and IIoT, together with attempts at national standardization, such as RAMI 4.0 and IIRA, are favoring an overall vision of the convergence and panorama of the fourth industrial revolution. At the same time, the new progressive Industry 5.0 movement is arising, bringing the need to put humanity, society, and the environment back at the center of technological progress.

Although there are still many features to be integrated, this thesis has contributed to the creation of a skeleton on which to ground future integration implementations used in companies that want to face technological progress and approach the scenario proposed by Industry 5.0. We have touched various domains close to the industrial world and demonstrated the feasibility of the transition, by using safe and performing technologies already employed in critical sectors. We grouped the many milestones reached by our project under the big domains listed below.

- **Industrial data convergence**. We started the engagement in industrial challenges from the manufacturing sector where the IT/OT convergence question is more preponderant. Companies facing the transitions to Industry 4.0 and Industry 5.0 are demanding for a general approach tackling all the challenges arising from the convergence. We conceived the SIRDAM middleware to support large-scale OT data gathering and information spreading in a fast, secure, and scalable way. Our geographically distributed test-bed demonstrated the reliability of the platform in multiple scenarios, also comprising SMEs with distributed production sites and offices. We covered and tested the scenario in which legacy (SCADA-based) and modern machines are employed in the same shop floor. The results show that our integration approach can fulfill the demands of the IT/OT convergence, granting the performance parameters and security it needs. SIRDAM is the core of our solution designed for the integration and achievement of the Industry 4.0 transition and the enabling of the Industry 5.0 scenarios, in an affordable way even for SMEs.

- **Mobile crowdsensing and human centrality**. The employment of mobile crowdsensing in the industrial revolution is a little-explored domain in literature. Keeping in mind the new events envisaged by Industry 5.0, we investigated the scenarios that this technology can enable in the industrial field. Since the MCS is strongly focused on the humans and their contributions provided to improve the services disbursed in the smart cities, we tested a critical use case that could become the prelude to the employment of this technology within production environments. Mobile phones and wearables devices are very common objects nowadays, so there is a good probability to obtain detailed information from people and from surrounding. Gamification techniques among the participants to crowdsensing campaigns improve the quality/number of contributions provided by platform users. Our MCS platform currently solves the problem of alerting about crowded places, which can be shopping, squares but also company data centers and workshops. Having heavy and contextualized computations to do, we extended the classic client-server infrastructure of MCS platforms with an extra edge layer. At this middle layer we demanded the management of blockchain mechanisms and the heavy calculations. The blockchain system used is capable of federating different MCS servers, increasing the number of participants in crowdsensing campaigns. Edge computing, a technology already employed in a lot of production realities, increases the reliability of the system and free the server from complex calculations that could involve a non-negligible overhead

- **Industrial support Cloud services**. The computational needs of a company can vary a lot over time, so Cloud and Edge computing have become one of the most used tools to cover the scalability requirement of modern production companies. Since our integration platform is strongly focused on this kind of distributed computational model, we hypothesized two services that could support the management of the corporate IT infrastructure. On one hand, we developed a cloud provider auditing tool, to compare the cloud performances. The Audit4Cloud tool allows the monitoring of latency and bandwidth parameters of virtual resources provided in public clouds. With this tool, the cloud customers, such as companies using public cloud services, can choose the best provider based on the performance they need. On the other side, the providers can identify bottlenecks between distributed regions and improve their deployment. The second cloud service we designed is a data center network simulator that we used to optimize and arrange an enterprise IT infrastructure deployment. Companies with big DCN need a performant tool to evaluate the virtualized resources' deployment. The DCNs-2 solution completely emulates complete cloud environments, allowing the IT departments to choose the best deployment to minimize energy waste and consequently emissions, making business processes more sustainable. The simulator is flexible, easily configurable, and complete due to its support to all kind of DCN entities emulation, such as network asset and physical/virtualized resources.

- **Cybersecurity at lower layers**. The convergence of data necessary goes through cybersecurity which ensures its integrity and confidentiality, characteristics on which many industrial companies base their reliability policy. We designed and implemented an analysis tool for the fuzz testing of the OPC UA protocol implementations, for identifying and storing bugs and crashes in the system under test. Our fuzzing tool tests any server implementation consistent with the definition of the OPC UA protocol, mutating the input vectors in a smart way. The fuzzer stores status, operational, and protocol errors and send them to IT departments and security expert for further analysis on their exploitability by malicious agents to attack the integrity of the production site.

## Future Work

Within the ambitious project of creating an industry at the service of humanity, society, and the environment, our project is a small contribution, however encouraging for experts in the industrial sector, given the results achieved. This section addresses the future research directions of our work.

The key role of IT/OT convergence within manufacturing environments and production sites demands for software platforms capable to tackle the resulting challenges. The next step to validate the effectiveness of our platform in convergent environments is its employing in production environments with a medium-high number of working machines. We are ready to try the sending of operational commands from the managerial departments to the production sites, enabling more direct control of machines by the business departments. The high modularity of our platform will allow in the future the addition of components and services directly connected to the enabling of the Industry 5.0 philosophy. For each pillar of the fifth industrial revolution, we are planning to enrich our platform with targeted services, aiming to make our middleware a central Industry 5.0 player.

The main I5.0 idea dwells into the integration between industrial and social revolutions. Observing the European and global panorama, we think that in the next super smart society many industrial sectors will be an integral part in the lives of citizens. Mobile crowdsensing is the most suitable technology to encourage the inclusion of people in the collections of useful data to improve their own lifestyle. The embodying of data gathered from targeted crowdsensing campaigns among workers or citizens, will improve the employees' status, their confidence with new technologies, their surroundings, and the citizens' involvement in industrial progress. From the point of view of the enterprise processes organization, connecting production and distribution chains in the crowdsensing gathering will optimize the provisioning and delivering processes, identifying bottlenecks and improving the customers' experience. Furthermore, the MCS is paramount to enable indoor localization, useful for guaranteeing the safety of workers in production sites. Adding this valuable information to the company knowledge base lets managerial departments develop novel long-term successful strategies. Obviously, the massive adoption of crowdsensing campaigns in the evolutionary process opens privacy challenges that will need to be resolved to allow everyone to provide their contribution in a safe and confidential manner.

From the point of view of reliable communication, we will integrate the new low-level timed network protocols, such as TSN, in the production sites in order to have more fine-grain control over the execution timing of the work machines. The TSN protocol will be able to enable the automation of processes needing strict timing requirements to be executed, assisting convergence even in environments potentially of high danger for workers, such as smart grids, foundries, and the manufacturing sector in general. Cybersecurity also contributes to corporate reliability, and our example is only one small use case of all possible employments of information technology to assure data confidentiality, integrity, and availability in the complex industrial domain. We are grouping the security mechanisms we used in our middleware to build a unique holistic platform that can achieve global cybersecurity protecting enterprise distributed and centralized deployment and the horizontal and vertical exchange of information and commands.

The last point on which we can focus our future efforts to enable the I5.0 vision is ecological sustainability. Cloud services would help us to enable the tracking of emissions produced by the company and optimize consumption through in-depth analysis of the resources used and their time of use. We will merge the simulation of the data center network with the digital twins platforms and with modern SDN/NFV simulators, in order to reach a complete forecast analysis of the industrial environment. We are planning to cover the multi-cloud approach simulation, in which many cloud providers and hybrid cloud deployments are used. We could use the auditing tool presented in this thesis to build a detailed model of the performance trends of the many public cloud providers' regions. With this kind of analysis, our platform can be an incentive in the integration of renewable energy sources, modern materials and composite fibers, energy-autonomous sensors, and low energy data transmission and data analysis in order to reduce emissions and keep costs under control.

In the end, the customers of our integration platform will be able to choose among a wide range of services available in the cloud or on private infrastructure (or yet in a hybrid way) to improve the reliability and the sustainability of their companies.

# List of Abbreviations

| | |
|---|---|
| ADMS | Advanced Distribution Management Solutions |
| CNC | Central Network Controller |
| CPS | Cyber-Physical Systems |
| CUC | Centralized User Configuration |
| DCS | Distributed Control System |
| ERP | Enterprise Resource Planning |
| ETL | Extract, Transform, Load |
| HMI | Human-Machine Interface |
| I4.0 | Industry 4.0 |
| I5.0 | Industry 5.0 |
| ICS | Industrial Control Systems |
| IEC | International Electrotechnical Commission |
| IIoT | Industrial Internet of Things |
| ISA | International Society of Automation |
| IT | Information Technology |
| KPI | Key Performance Indicators |
| MCS | Mobile Crowd Sensing |
| MEC | Multi-access Edge Computing |

MES Manufacturing Execution Systems

MIS Management Information Systems

MMC Machine Manufacturer Company

MMI Machine-Machine Interface

MOM Manufacturing Operations Management

NFV Network Functions Virtualization

OT Operational Technology

PLC Programmable Logic Controller

PLM Product Lifecycle Management

RTU Remote Terminal Unit

S5.0 Society 5.0

SCADA Supervisory Control And Data Acquisition

SME Small and Medium Enterprises

SDN Software-Defined Networks

SUT System-Under-Test

# References


[1] “Italy manufacturing pmimarch 2022 data - 2012-2021 historical - april forecast.” [Online]. Available: https://tradingeconomics.com/italy/manufacturing-pmi

[2] 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.

[3] “Home - participact,” Nov 2018. [Online]. Available: http://www.participact.com.br/

[4] M. Foehr, J. Vollmar, A. Calà, P. Leitão, S. Karnouskos, and A. W. Colombo, “Engineering of next generation cyber-physical automation system architectures,” *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*, p. 185–206, 2017.

[5] C. Li, S. Mantravadi, and C. Møller, “Aau open source mes architecture for smart factories – exploiting isa 95,” in *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)*, vol. 1, 2020, pp. 369–373.

[6] 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.

[7] 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.

[8] L. Mladkova, “Industry 4.0: Human-technology interaction: Experience learned from the aviation industry,” pp. 571–578,XXIII, 09 2018, name - Federal Aviation Administration–FAA; Copyright - Copyright Academic Conferences International Limited Sep 2018; Last updated - 2020-11-19. [Online]. Available: https://search.proquest.com/conference-papers-proceedings/industry-4-0-human-technology-interaction/docview/2116815603/se-2?accountid=9652

[9] K. Yonemura, J. Sato, R. Komura, and M. Matsuoka, “Practical security education on combination of ot and ict using gamfication method,” in *2018 IEEE Global Engineering Education Conference (EDUCON)*, 2018, pp. 746–750.

[10] K. Karampidis, S. Panagiotakis, M. Vasilakis, E. K. Markakis, and G. Papadourakis, "Industrial cybersecurity 4.0: Preparing the operational technicians for industry 4.0," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2019, pp. 1–6.

[11] C. Khona, "Key attributes of an intelligent iiot edge platform," *Xilinx, San Jose, CA, USA, White Paper, Sep*, 2017.

[12] "Health impact of 5g: Think tank: European parliament." [Online]. Available: https://www.europarl.europa.eu/thinktank/en/document/EPRS_STU(2021)690012

[13] M. Fukuyama, "Society 5.0: Aiming for a new human-centered society," *Japan Spotlight*, vol. 1, pp. 47–50, 2018.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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

[19] 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.

[20] 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

[21] R. Paes, D. C. Mazur, B. K. Venne, and J. Ostrzenski, "A guide to securing industrial control networks: Integrating it and ot systems," *IEEE Industry Applications Magazine*, vol. 26, no. 2, pp. 47–53, 2020.

[22] D. R. Harp and B. Gregory-Brown, "It/ot convergence bridging the divide," *NexDefense*, p. 23, 2015.

[23] D. Fortinet GUIDE, “Securing industrial control systems with fortinet,” *D GUIDE"*, 2019, [Accessed: Jan 2022].

[24] G. Murray, M. N. Johnstone, and C. Valli, “The convergence of it and ot in critical infrastructure,” *Australian Information Security Management Conference*, 2017, [Accessed: Jan 2022]. [Online]. Available: http://ro.ecu.edu.au/ism/217/

[25] A. Giehl and N. Wiedermann, “Security verification of third party design files in manufacturing,” in *Proceedings of the 2018 10th International Conference on Computer and Automation Engineering*, ser. ICCAE 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 166–173. [Online]. Available: https://doi.org/10.1145/3192975.3192984

[26] 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.

[27] 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

[28] S. Hagner, “Optimizing transmission asset health with it/ot integration,” in *2016 Saudi Arabia Smart Grid (SASG)*, 2016, pp. 1–6.

[29] 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.

[30] 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.

[31] V. V. Martynov, D. N. Shavaleeva, and A. A. Zaytseva, “Information technology as the basis for transformation into a digital society and industry 5.0,” in *2019 International Conference "Quality Management, Transport and Information Security, Information Technologies" (IT QM IS)*, 2019, pp. 539–543.

[32] L. Patera, A. Garbugli, A. Bujari, D. Scotece, and A. Corradi, “A layered middleware for ot/it convergence to empower industry 5.0 applications,” *Sensors*, vol. 22, no. 1, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/1/190

[33] V. Gorodetsky, V. Larukchin, and P. Skobelev, “Conceptual model of digital platform for enterprises of industry 5.0,” in *Intelligent Distributed Computing XIII*. Springer International Publishing, Oct. 2019, pp. 35–40. [Online]. Available: https://doi.org/10.1007/978-3-030-32258-8_4

[34] P. Thakur and V. Kumar Sehgal, “Emerging architecture for heterogeneous smart cyber-physical systems for industry 5.0,” *Computers & Industrial Engineering*, vol. 162, p. 107750, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835221006549

[35] F. Aslam, W. Aimin, M. Li, and K. Ur Rehman, “Innovation in the era of iot and industry 5.0: Absolute innovation management (aim) framework,” *Information*, vol. 11, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/124

[36] M. Girolami, D. Belli, S. Chessa, and L. Foschini, “How mobility and sociality reshape the context: A decade of experience in mobile crowdsensing,” *Sensors*, vol. 21, no. 19, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/19/6397

[37] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, “Federated learning for industrial internet of things in future industries,” *IEEE Wireless Communications*, pp. 1–8, 2021.

[38] A. Iqbal and S. Olariu, “A survey of enabling technologies for smart communities,” *Smart Cities*, vol. 4, no. 1, pp. 54–77, 2021. [Online]. Available: https://www.mdpi.com/2624-6511/4/1/4

[39] Z. Yu, H. Ma, B. Guo, and Z. Yang, “Crowdsensing 2.0,” *Communications of the ACM*, vol. 64, no. 11, pp. 76–80, 2021.

[40] L. Foschini, G. Martuscelli, R. Montanari, and M. Solimando, “Edge-enabled mobile crowdsensing to support effective rewarding for data collection in pandemic events,” *Journal of Grid Computing*, vol. 19, no. 3, Jul. 2021. [Online]. Available: https://doi.org/10.1007/s10723-021-09569-9

[41] Eurostat, “Cloud computing - statistics on the use by enterprises,” *Europe Statistics*, 2021. [Online]. Available: https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=isoc_cicce_use&lang=en

[42] D. Serpanos and K. Katsigiannis, “Fuzzing: Cyberphysical system testing for security and dependability,” *Computer*, vol. 54, no. 9, pp. 86–89, 2021.

[43] K. Katsigiannis and D. Serpanos, “Mtf -storm: a high performance fuzzer for modbus/tcp,” in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2018, pp. 926–931.

[44] S. Hoppe, “There is no industrie 4.0 without opc ua,” *OPC Found*, 2017.

[45] 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, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/5/510

[46] F. Bosi, A. Corradi, G. Di Modica, L. Foschini, R. Montanari, L. Patera, and M. Solimando, “Enabling smart manufacturing by empowering data integration with industrial iot support,” in *2020 International Conference on Technology and Entrepreneurship (ICTE)*, 2020, pp. 1–8.

[47] A. Corradi, G. Di Modica, L. Foschini, L. Patera, and M. Solimando, “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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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

[53] 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.

[54] Cloudera, “Architectural patterns for near real-time data processing with apache hadoop,” Sep 2019, [Accessed: Jan 2022]. [Online]. Available: https://blog.cloudera.com/architectural-patterns-for-near-real-time-data-processing-with-apache-hadoop/

[55] G. Cardone, A. Corradi, L. Foschini, and R. Ianniello, “Participact: A large-scale crowdsensing platform,” *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 1, pp. 21–32, 2016.

[56] G. Cardone, A. Cirri, A. Corradi, L. Foschini, and R. Montanari, “Activity recognition for smart city scenarios: Google play services vs. most facilities,” in *2014 IEEE Symposium on Computers and Communications (ISCC)*, 2014, pp. 1–6.

[57] P. Bellavista, M. Cilloni, G. Di Modica, R. Montanari, P. Carlo Maiorano Picone, and M. Solimando, “An edge-based distributed ledger architecture for supporting decentralized incentives in mobile crowdsensing,” in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, 2020, pp. 781–787.

[58] L. Foschini, A. Gavagna, G. Martuscelli, and R. Montanari, “Hyperledger fabric blockchain: Chaincode performance analysis,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[59] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, “Nfv and sdn—key technology enablers for 5g networks,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2468–2478, Nov 2017.

[60] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano, A. Kanagala, J. Provost, J. Simmons, E. Tanda, J. Wanderer, U. Hölzle, S. Stuart, and A. Vahdat, "Jupiter rising: A decade of clos topologies and centralized control in google's datacenter network," in *Sigcomm '15*, 2015.

[61] A. Andreyev, "Introducing data center fabric, the next-generation facebook data center network," Jun 2018, [Accessed: Jan 2022]. [Online]. Available: https://code.fb.com/production-engineering/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/

[62] A. Blenk, A. Basta, J. Zerwas, and W. Kellerer, "Pairing sdn with network virtualization: The network hypervisor placement problem," in *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, Nov 2015, pp. 198–204.

[63] P. L. Ventre, C. Pisa, S. Salsano, G. Siracusano, F. Schmidt, P. Lungaroni, and N. Blefari-Melazzi, "Performance evaluation and tuning of virtual infrastructure managers for (micro) virtual network functions," in *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2016, pp. 141–147.

[64] A. M. Law and W. D. Kelton, *Simulation modeling and analysis*, 3rd ed. McGraw-Hill, 1999.

[65] J. Banks, J. S. C. II, B. L. Nelson, , and D. M. Nicol, *Discrete-Event System Simulation*, 4th ed. Prentice Hall, 2004.

[66] S. Wang, "Comparison of sdn openflow network simulator and emulators: Estinet vs. mininet," in *2014 IEEE Symposium on Computers and Communications (ISCC)*, June 2014, pp. 1–6.

[67] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *2015 11th International Conference on Network and Service Management (CNSM)*, Nov 2015, pp. 384–389.

[68] P. Bellavista, A. Corradi, L. Foschini, S. Luciano, and M. Solimando, "A simulation framework for virtualized resources in cloud data center networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1808–1819, 2019.

[69] P. Bellavista, L. Foschini, S. Luciano, and M. Solimando, "Dcns-2: A cloud network simulator extension for ns-2," in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWIM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 47–51. [Online]. Available: https://doi.org/10.1145/3242102.3242136

[70] K. Rybina, W. Dargie, S. Umashankar, and A. Schill, "Modelling the live migration time of virtual machines," in *On the Move to Meaningful Internet Systems: OTM 2015 Conferences*, C. Debruyne, H. Panetto, R. Meersman, T. Dillon, G. Weichhart, Y. An, and C. A. Ardagna, Eds. Cham: Springer International Publishing, 2015, pp. 575–593.

[71] T. Kim, T. Koo, and E. Paik, “Sdn and nfv benchmarking for performance and reliability,” in *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2015, pp. 600–603.

[72] “Telco cloud nfv metrics and performance management.” [Online]. Available: https://sdn.ieee.org/newsletter/may-2017/telco-cloud-nfv-metrics-and-performance-management

[73] “Nfv isg poc proposal vnf router performance with ... - etsi.” [Online]. Available: https://nfvwiki.etsi.org/images/NFVPER%2814%29000024a4_NFV_ISG_PoC_proposal_VNF_Router_Performance_with_DDoS_func.pdf

[74] D. Medhi and K. Ramasamy, *Network routing: algorithms, protocols, and architectures*. Morgan kaufmann, 2017.

[75] “Network topology.” [Online]. Available: https://docs.nvidia.com/networking-ethernet-software/cumulus-linux-40/Layer-3/Network-Topology/

[76] D. M. Nicol, “Scalability of network simulators revisited,” in *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, vol. 28, 2003.

[77] W. Cerroni, L. Foschini, G. Y. Grabarnik, L. Shwartz, and M. Tortonesi, “Estimating delay times between cloud datacenters: A pragmatic modeling approach,” *IEEE Communications Letters*, vol. 22, no. 3, pp. 526–529, March 2018.

[78] M. Tortonesi and L. Foschini, “Business-driven service placement for highly dynamic and distributed cloud systems,” *IEEE Transactions on Cloud Computing*, vol. 6, no. 4, pp. 977–990, Oct 2018.

[79] P. Bellavista, A. Corradi, L. Foschini, and M. Solimando, “The audit4cloud platform for auditing the networking performance of public clouds,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[80] M. Sutton, *Fuzzing : brute force vulnerabilty discovery*. Upper Saddle River, N.J: Addison-Wesley, 2007.