



# Interoperability of heterogeneous Systems of Systems: from requirements to a reference architecture

Mersedeh Sadeghi<sup>1</sup> · Alessio Carenini<sup>2</sup> · Oscar Corcho<sup>3</sup> · Matteo Rossi<sup>4</sup> · Riccardo Santoro<sup>5</sup> · Andreas Vogelsang<sup>1</sup>

Accepted: 27 October 2023  
© The Author(s) 2023

## Abstract

Interoperability stands as a critical hurdle in developing and overseeing distributed and collaborative systems. Thus, it becomes imperative to gain a deep comprehension of the primary obstacles hindering interoperability and the essential criteria that systems must satisfy to achieve it. In light of this objective, in the initial phase of this research, we conducted a survey questionnaire involving stakeholders and practitioners engaged in distributed and collaborative systems. This effort resulted in the identification of eight essential interoperability requirements, along with their corresponding challenges. Then, the second part of our study encompassed a critical review of the literature to assess the effectiveness of prevailing conceptual approaches and associated technologies in addressing the identified requirements. This analysis led to the identification of a set of components that promise to deliver the desired interoperability by addressing the requirements identified earlier. These elements subsequently form the foundation for the third part of our study, a reference architecture for interoperability-fostering frameworks that is proposed in this paper. The results of our research can significantly impact the software engineering of interoperable systems by introducing their fundamental requirements and the best practices to address them, but also by identifying the key elements of a framework facilitating interoperability in Systems of Systems.

**Keywords** Interoperability · System of Systems · Interoperability requirements · Interoperability challenges · Interoperability architecture · Distributed collaborative systems

## 1 Introduction

The term *System of Systems* (SoS) refers to a collaborative and interactive ecosystem, typically characterized as an environment which is *open at the top* (i.e., there is no pre-defined top-level application, and new applications can be created at any

---

Extended author information available on the last page of the article

time), *open at the bottom* (i.e., system primitives are defined functionally rather than concretely), and *continually evolving* (i.e., functions are stable enough to be useful, but they are understood to be subject to modification) [1, 10, 55]. The concept of SoS can be applied to a broad range of application domains including telecommunication, Internet of Things, cloud computing, enterprise, e-commerce, healthcare and transportation systems [63]. One of the main obstacles to the creation of such large, distributed and collaborative systems is the lack of *interoperability*, which can severely hamper the seamless collaboration and interaction of the SoS's heterogeneous constituent systems [63]. Interoperability enables the interaction, cooperation and sharing among systems; it facilitates the distribution of technology and resources, avoids vendor lock-in, and promotes the establishment of an open, fair market. This makes interoperability key to the enterprise survival, rather than just a technological preference [16].

Nevertheless, despite a large body of research on interoperability, there is a shortage of practical solutions. One reason might be that the current state of the art in the study and analysis of interoperability has been heavily academically oriented, relying on scholarly resources and focusing especially on understanding the theoretical and conceptual aspects of interoperability. To address this shortcoming, in this work, we aim to identify and categorize the core interoperability requirements directly from the practitioners' point of view. Indeed, understanding the core interoperability requirements is the first step toward building interoperable systems. A clear and precise definition of the requirements allows software engineers, on the one hand, to create systems that are interoperable by design and, on the other hand, to measure and test their interoperation capabilities. Though there are in the literature other contributions to the identification of interoperability requirements, they mainly pursued a theoretical analysis of interoperability aspects, but overlooked and in-depth exploration of the tools, technologies and best practices to realize/overcome the concepts/challenges they introduced. As a result, there is a gap between requirement specifications and practical solutions to address them. In this work, we contribute to filling this gap by conducting a critical literature study and analysis of common interoperability approaches and techniques with respect to the extracted requirements. This analysis led to the definition of the key elements that a framework facilitating interoperability in SoSs should have. These elements are the basis for the architecture for our proposed *Interoperability Framework* (IF), which provides the foundations for a new generation of interoperability solutions by offering the following characteristics. **It enables interoperation while preserving the autonomy of individual systems** with respect to the adoption of any desired standards and technologies. Indeed, making systems interoperable should not equate to preventing heterogeneity, not only because such an approach is unfeasible in practice, but also because diversity is the key to innovation and technological advancements. Accordingly, a proper interoperability solution should allow systems to interact even if they employ heterogeneous technologies. To provide a solution that has staying power, the IF is coherent with the distributed nature of SoSs, and **can cope with—and adapt to—the changes and the dynamics of the ecosystem**, in terms of both business and technological aspects. Rather than providing a single medium that makes systems interoperable—such as a software bus or a middleware—**it defines a set of interoperability**

**building blocks that can be created, shared, discovered** and, in general, that can collaboratively evolve. Hence, it allows individual systems to define and build their own interoperability solutions, which are tailored to deal with their specific interoperability obstacles.

Note that, even though for simplicity we refer in the paper to *the* IF, we are not advocating for there to be a single product, called “IF,” which is to be used to achieve interoperability in any SoS. Rather, we present a set of components that should constitute the core of interoperability-fostering frameworks. As such, the IF described in this paper is to be intended as a proposed *reference architecture* for frameworks enabling interoperability in SoSs.

The contributions of this paper are manifold:

- i. the identification of eight fundamental and domain-agnostic requirements for the interoperation of SoSs;
- ii. the incorporation of practitioners’ insight in the requirement extraction, to discover the real challenges they are dealing with and to specify pragmatic requirements for building interoperability frameworks that are effective in practice;
- iii. a thorough literature analysis to determine the best practices for addressing the identified requirements and challenges;
- iv. the introduction of a reference architecture for frameworks that facilitate interoperability among collaborative systems.

Points (i)-(iii) have been tackled in [71], of which this paper is an extension. In addition, this article proposes a reference architecture for interoperability-promoting frameworks and discusses how it addresses the requirements identified in point (i).

The rest of the paper is organized as follows: Sect. 2 reviews the background concepts and discusses related works; Sect. 3 describes the structure of our research, while Sect. 4 and Sect. 5 report and discuss its main findings, which include the extracted requirements and the corresponding literature mapping to identify the best practices; Sect. 6 distills these findings in a proposed reference architecture for an interoperability-promoting framework; finally, Sect. 7 concludes the paper.

## 2 Background and related works

Interoperability is a long-standing challenge, and a lot of research has been carried out in this context. In this section, we provide an overview of the most relevant works related to our research, which consist of studies aimed at better understanding interoperability concepts and challenges mainly through the formalization and categorization of interoperability-related notions. In particular, we focus on the works that, like us, studied the interoperability of distributed systems and SoSs, and provided a generic conceptualization and analysis.

In the Oxford dictionary, the word “interoperable” is defined as “(of computer systems or programs) able to exchange information.” In information systems, however, a unified definition of this concept seems to be difficult to achieve, as discussed

in [37]. Interoperability in software engineering goes beyond its generic definition—i.e., the ability to exchange data (see [39])—and refers to a broader concept: the reciprocal understanding and compatibility among two—or more—systems, which allows them to operate on and use the functions of each other [18, 81].

Indeed, interoperability is not only a technical or business concern, but it has also entered the political sphere, where governmental efforts have pushed toward achieving interoperability in various technology, logistics and social domains. For example, ISA2(Interoperability Solutions for European public Administrations, businesses and citizens) [22, 23] is a European Council project that aims to establish a program focusing on interoperability solutions for European public administrations, businesses and citizens. In addition, various regulations and agreements have been defined regarding interoperability among information systems [25], ranging from interoperable databases [24], to information systems dealing with visas and borders [26], to those regarding police and judicial cooperation [27].

This shows that, when it comes to large distributed systems, interoperation requires cooperation at various levels (physical/infrastructure, hardware, conceptual, application, network, business, regulatory, etc.) among systems, organizations and sectors [17, 84]. Hence, different works have tried to structurally identify various interoperability layers. For instance, the European Interoperability Framework (EIF) [33]—an initiative of the European Commission promoting seamless service interoperability and data flows for European public administrations—defines four levels of interoperability. Legal Interoperability (LI) ensures the cooperation of organizations under different legal frameworks, policies and strategies. Organizational Interoperability (OI) concerns the alignment of business processes. Semantic Interoperability (SI) achieves the semantic and common understanding of data elements, and Technical Interoperability (TI) manages the interoperation at the application level of the infrastructure. Inspired by the EIF—from which it borrows the interoperability layers (Technical, Semantic, Organizational, and Legal)—EOSC IF (The European Open Science Cloud Interoperability Framework, [21]) aims to facilitate interoperability in the research and science domain according to FAIR (Findability, Accessibility, Interoperability, and Reusability) principles for scientific data management [83]. EOSC IF tackles interoperability issues with the help of semantic technologies and a set of loosely coupled services and software; it provides a set of recommendations, best practices, a conceptual reference architecture, and a governance and legal structure to guide and organize the target community.

Similar to the EIF approach and in line with our research, there have been various projects and initiatives working on the formalization of interoperability in enterprise systems, multi-organization systems, Systems of Systems, etc. [16, 28]. For example, [16] conceptualizes enterprise interoperability by introducing four interoperability viewpoints—data, service, process and business—that must be addressed to achieve full interoperability in an enterprise system. Data interoperability refers to operating on various data formats and query languages. Service interoperability involves identifying, composing, and operating various computer-based or non-computer-based applications and services together. Process interoperability is needed to make various processes work together. Finally, business interoperability is necessary to harmonize decision-making processes, rules, company cultures, commercial

approaches, etc., to develop and share business between companies. IDEAS (Interoperability Development for Enterprise Applications and Software) [3, 15] was among the earliest works to suggest a multi-layer solution to envision interoperability. It states interoperability must be achieved on different levels, including business, knowledge, application, data, and communication. It also introduced the semantic dimension concerned with understanding the actual meaning of concepts at each level.

ATHENA (Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications) [5, 70] complements IDEAS and addresses interoperability through a holistic approach that considers the business and technical viewpoints. In short, it defines three meta-levels over IDEAS levels for approaching interoperability in enterprise systems and applications: *Conceptual*, *Applicative* and *Technical*. SOSI (System of Systems Interoperability) [31, 60] and MoBIE (Model-Based Interoperability Engineering) [59] focus on SoSs. SOSI's interoperability model is composed of three layers, programmatic, construction and operational. SOSI maintains that to make two systems fully interoperable at the operational level, interoperability must be achieved first at the programmatic level (e.g., between program offices and contractors), and second at the constructive level (e.g., of the SoS architecture, protocols and standards). MoBIE is a conceptual modeling framework for SoSs, whose main contribution is the development of an ontology describing the interoperability domain and enabling the modeling, use and analysis of all significant interoperability aspects. Following the MoBIE framework, one can model, from the point of view of interoperability requirements, the systems, the involved agents, the nature of interfaces and functional interactions with other systems, and the types of engagements. In another work, the framework for federated interoperability proposed by [52] combines graph theory and Model-Driven Engineering to enable on-the-fly data transformation and integration among heterogeneous relational database systems. In particular, their proposed solution first explores the original relational databases to discover the source and target data models and create their graph representations. It then computes the similarity between the elements (nodes and edges) of the two corresponding graphs to generate a set of transformation rules to map source data to the target data structure.

In the context of the industry, particularly with the advent of the fourth industrial revolution, commonly referred to as "Industry 4.0," the pivotal role of SoS interoperability became increasingly evident [11]. Notably, the reference architecture for implementing Industry 4.0, known as the Industrial Internet Reference Architecture<sup>1</sup> (IIRA), consists of four layers (Business, Usage, Functional, and Implementation) that inherently demand vertical interoperability among these layers. Consequently, various initiatives have begun to identify SoS interoperability challenges within this framework. For instance, Gollner et al. [40] recognized the significance of Digital Twins (DTs) in the SoS context. DTs serve as digital representations of physical objects or systems [20] and play a crucial role when different partners in the value chain need to collaborate. This is because all

<sup>1</sup> <https://www.iiconsortium.org/IIRA.html>.

pertinent assets require such digital representations to be seamlessly integrated into a collaborative and interoperable environment. However, unlike our work, it does not primarily focus on an in-depth analysis of interoperability challenges and only proposes a modeling language that enables various partners to collaboratively specify the interactions between different DTs. One limitation of their approach is that this specification process is not fully automated and still relies on human intervention.

In a similar vein, the work in [66] addresses the challenge of heterogeneity in SoS by focusing on digitalization and automation to enhance interoperability within the Industry 4.0 movement. Like our approach, the study proposes architectural design principles and tools to streamline interoperability engineering efforts. However, their focus is primarily on service-oriented systems. This narrows their interoperability solutions to issues specific to Service-Oriented Architecture (SOA), such as service interface and contract heterogeneity. In contrast, our analysis and solutions are more versatile, extending beyond just SOA.

The last group of related works includes other surveys and studies on interoperability challenges and characteristics. Lane and Valerdi [53] characterized influences of interoperability on SoS engineering. In particular, to study the type of engineering work required to make SoSs interoperable, they analyzed 14 interoperability models gathered from the comprehensive survey presented in [37]. In addition, [53], and the similar literature survey presented in [54], provided interoperability definitions, models and techniques. The work in [62] analyzes and classifies available solutions to achieve pragmatic interoperability by systematically over-viewing 13 papers out of more than a thousand. The in-depth analysis presented in [6] shows why interoperability should be considered a crucial characteristic of large collaborative systems. Further, it defines a set of requirements for the subsystems of an SoS to make the latter interoperable. It also presents a relationship between the interoperability of an SoS and that of its composing subsystems to help engineers choose the right subsystems prior to developing the SoS.

Our research and the works mentioned above differ in many ways:

- i. Most of the available contributions on the conceptualization of interoperability and the analysis of its challenges focused solely on academic articles, whereas in our work, we aimed to extract the challenges and requirements by considering the point of view and experience of actual practitioners (both in industry and academia).
- ii. Previous works provided theoretical analyses but did not investigate the tools, technologies, and best practices to realize/address the identified concepts and challenges. In this research, after extracting the relevant requirements, we analyzed the best approaches for addressing them.
- iii. Most of the frameworks mainly focus on the modeling, formalization, and conceptualization of interoperability; however, unlike this work, they do not identify—nor provide—the architectural and technical elements for the implementation and development of components that can enhance the interoperation among systems in a complex ecosystem.

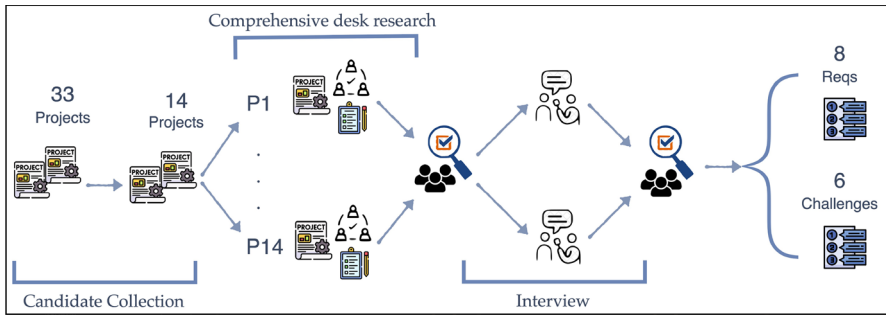


Fig. 1 Procedure followed to carry out the questionnaire-based survey

### 3 Study design

This work has two goals: first, to obtain a deep understanding of the main interoperability challenges, barriers and requirements in large-scale collaborative systems; second, to understand the capabilities of current interoperability methodologies and approaches, along with the tools and technologies that can address the identified requirements. To this end, we structured our research in three phases, each addressing one of the following research questions.

- (RQ1) What are the fundamental requirements and challenges, considering every relevant aspect, that must be addressed to make large collaborative SoS interoperable?
- (RQ2) Which methodology provides the most suitable basis to address the identified requirements and challenges and foster SoS interoperability?
- (RQ3) What components should be part of a framework specifically built to facilitate interoperability among SoSs that addresses the identified requirements using the most suitable methodology?

In the first phase, focusing primarily on the transportation domain as a paradigmatic case of distributed and collaborative eco-system—or *de facto* SoS, we studied the transportation stakeholders, organizations and systems that constitute or are involved in, SoSs. As depicted in Fig. 1, we assembled a collection of 33 projects and initiatives supported by the European Union. These projects were drawn from those funded by the Shift2Rail Joint Undertaking under the umbrella of Innovation Programme 4,<sup>2</sup> and we also included select external projects. These initiatives spanned a wide spectrum, encompassing applications related to transportation for smartphones, as well as academic and industrial projects addressing diverse facets of mobility and transportation systems. The primary selection criteria revolved around two key factors: their relevance

<sup>2</sup> [https://projects.shift2rail.org/s2r\\_ip.aspx?ip=4](https://projects.shift2rail.org/s2r_ip.aspx?ip=4).

to interoperability issues and the presence of substantial documentation and information available in English.

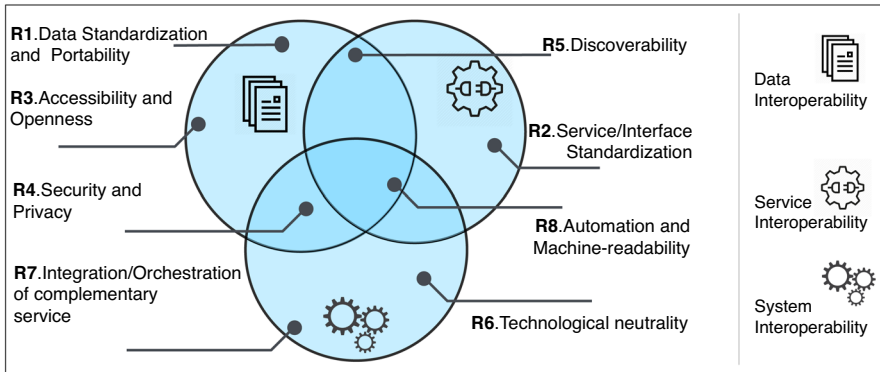
Then, we carried out a screening process to narrow down the list of relevant projects and ultimately identified 16 suitable candidates. After reaching out to these projects to gauge their willingness to participate in our survey and interview processes, we successfully secured the participation of 14 projects. In the next phase, we conducted extensive desk research on the selected projects to gain insights into their existing systems and technologies, as well as their objectives and requirements related to interoperability. A comprehensive list of the projects studied, along with detailed information about them and our research, can be found in [78].

Following this, we developed a questionnaire comprising 16 questions, encompassing both multiple-choice and open-ended inquiries, to delve deeper into the obstacles and challenges faced by practitioners. The complete set of questions is presented in [77]. The questionnaire was distributed to a selected set of managers, engineers and technicians in every project. The answers to the questionnaire were collected in several rounds of communication with the experts for further follow-up questions and clarifications regarding the questions/answers. In some cases, face-to-face interviews were carried out. Section 3 provides an overview of the main findings of this requirement analysis phase. We argue that, while some interoperability challenges may vary according to the specific domain and the desired level and nature of the cooperation, the core set of concerns and the fundamental problems are shared across domains. Hence, while we sometimes articulate the requirements with reference to the transportation domain, we believe they are general and domain-independent, and so is the proposed interoperability model.

After extracting the basic set of requirements, in the second phase of our study, we focused on answering RQ2, and we carried out an in-depth comparative study and critical analysis of the literature. We investigated the state of the art and available techniques and technologies tackling interoperability issues, to understand how effective they are in addressing such requirements and to accordingly identify the best practices for building interoperable SoSs. Section 5 illustrates the results of this phase of the study.

Finally, in the third phase, we tackled RQ3 by defining the reference architecture and core elements of frameworks that aim to facilitate interoperability in SoSs. We first identified the design goals for such frameworks, based on the requirements discussed in Sect. 4 and on the best practices presented in Sect. 5. Then, we defined a set of key elements that interoperability-promoting frameworks should have, where each component addresses at least one (possibly more) of the desired requirements. We also identified several different strategies that could be used to deploy these components in practice. The proposed architecture, with its core elements and the possible deployment strategies, is described in Sect. 6.





**Fig. 2** The interoperability trilogy: Data, Service and System Interoperability, and their corresponding requirements

#### 4 Interoperability challenges and emerging requirements

To frame the identified requirements that must be addressed to overcome the lack of interoperability in any large, distributed and collaborative system, we introduce the *interoperability trilogy* shown in Fig. 2, which has three facets: Data, Service, and System Interoperability. Each facet covers concerns related to a key source of challenges to interoperability: the need to exchange data, use one another's services, and systematically manage the infrastructures for distributing, discovering, sharing and exchanging artifacts.

The first two aspects of the interoperability trilogy, *Data* and *Service Interoperability*, comprise two fundamental interoperability barriers: heterogeneous data, and disparate APIs and services. Most of the contributions addressing this problem have focused on the provision of solutions characterized by a narrow focus, based on the development of specific standards, data models, and technical solutions such as ad hoc plugins; in contrast, little attention has been paid to tackling interoperability issues among such solutions, frameworks and standards themselves. Thus, though solutions focusing merely on data and service aspects can foster some level of interoperability among multiple systems, the extent of the achieved cooperation is still limited and highly fragmented. As a result, instead of individual isolated systems, we obtain isolated *groups* of systems, where the systems of each group are typically under the control of a single organization (e.g., a company) able to impose constraints on the autonomy of its designers or solution providers. In this case, the systems of a given group are highly interoperable with one another, but they have no or very little interoperability with systems of other groups, which are controlled by separate, independent organizations. Hence, for interoperability to be effectively achieved in a large, collaborative environment composed of independent and distributed systems, the presence of standards, unified specifications and interfaces can help, but it is not sufficient. Therefore, we introduce the third facet of the interoperability trilogy, *System Interoperability*. It focuses on the features of a system or infrastructure that facilitate the development, establishment, advertisement, distribution

and collaborative use of interoperability-enabling concepts and technologies—e.g., ontologies, data distribution and sharing, standards, unified service interfaces—that can significantly reduce the loss of autonomy and independence that organizations suffer when they participate in an SoS ecosystem.

#### 4.1 Key requirements of interoperability

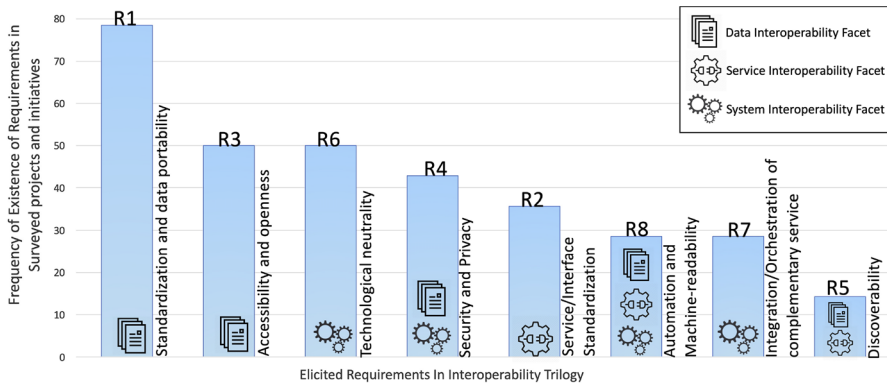
This section outlines the eight identified key interoperability requirements (denoted as **R1–R8** and summarized in Table 2) which are the results of the first phase of our study. Each requirement captures a set of features and prerequisites deemed essential in one or more facets to make SoSs interoperable (e.g., the need for data and services to be discovered in the case of *R5*). In addition, a set of challenges (denoted as **C1–C6** in Table 1) that hinder the satisfaction of such requirements are also outlined in the discussions for each requirement. Finally, as an indicator of the importance of each of these requirements, we defined the relative frequency notion as depicted in Fig. 3. It shows the number of times such a requirement has appeared (implicitly or explicitly) in the answers (the questionnaire or interviews) or in our desk analysis of the surveyed projects.

**Data standardization and portability (R1)** and **Service Standardization (R2)**: Standardization and portability are two key requirements in any collaborative domain where various organizations/systems require data collection from other organizations/systems and in a wide range of categories. Moreover, in a distributed multi-agent enterprise where many products and services can be provided only through a network of cooperating systems, data sharing and the possibility of creating integrated services carry even greater significance. For example, in the transportation domain, the current trends toward a new generation of mobility services providing multi-modal, door-to-door solutions require strong system cooperation, and in particular, the ability to exchange data (to share schedules, fares, etc.) across different sectors (e.g., rail, road, air, urban mobility, vehicle sharing) and countries.

However, in a fragmented environment where individual parties are in charge of data management (generation, utilization, storage, sharing, and distribution of data) data formats and standards tend to diverge (**C1**). This immense data diversity is one of the primary issues obstructing smooth interoperation among different actors. As a consequence, the *Data Standardization and Portability (R1)* requirement of the *Data Interoperability Facet* has the highest rank among all identified requirements (see Fig. 3). More precisely, *R1* refers to the capability of transferring data between different systems in a way that guarantees that data remain functional and processable in the destination system. *R1* is also key for the interoperability of services for a pair of reasons. First and foremost, services must have a consistent interpretation of exchanged data to be able to cooperate. Second, the increasing popularity of servitization and of service- and micro-service-oriented architectures in distributed systems leads, in turn, to increasing heterogeneity in service interfaces and APIs (**C1**). For example, in the transportation domain, the concept of Mobility-as-a-Service—where users can access multiple, heterogeneous, multi-modal transportation services

**Table 1** Interoperability challenges

No.	Challenge	Related Requirements
<b>C1</b>	Huge diversity and heterogeneity in data and service modeling, as well as regulations and policies	<b>R1, R2, R5, R6, R7</b>
<b>C2</b>	Complex organizations have already made their choices of standards and gone through some compromises, efforts and costs to make their internal subsystems interoperable, which might hamper the interoperability with further external systems and services	<b>R2, R6</b>
<b>C3</b>	The tendency of organizations (at the managerial level) to preserve all rights of their data and tools exclusively for their organization	<b>R3</b>
<b>C4</b>	High cost and a generally pessimistic view of security technologies make organizations favor an isolation strategy for maximizing the security and privacy of their internal systems	<b>R4</b>
<b>C5</b>	Modular system development is not yet a common practice, mainly due to the scarcity of practical tools and approaches for modular software engineering	<b>R6, R7</b>
<b>C6</b>	Low level and high cost of digitalization and automation	<b>R8</b>



**Fig. 3** Relative frequency of elicited requirements in the domain of the surveyed projects

**Table 2** Interoperability requirements

No.	Requirement
R1	<i>Data standardization and portability</i> : facilitate data exchange among systems to support collaborative data collection and integrated services
R2	<i>Service standardization</i> : promote common standards for service management to enhance interoperability within diverse organizational contexts
R3	<i>Data accessibility and openness</i> : ensure open, accessible data with a focus on legal and technical openness, discoverability, and reusability
R4	<i>Security and privacy</i> : address data security and privacy concerns in competitive enterprise environments, emphasizing control and secure communication
R5	<i>Discoverability</i> : establish an efficient and reliable discovery process vital for service-oriented computing and data-driven systems, while mitigating data heterogeneity challenges
R6	<i>Technological neutrality</i> : encourage technological neutrality by decoupling services and functions from underlying technologies to foster interoperability in complex systems
R7	<i>Integration/orchestration of complementary services</i> : facilitate the integration and orchestration of complex, complementary services to meet evolving user expectations, particularly in areas like transportation
R8	<i>Automation and machine-readability</i> : promote automation and machine-readability to reduce human intervention

in a seamless manner, typically through a single application<sup>3</sup>—is attracting a lot of interest and is becoming a common practice [47]. As a result, transport actors—from public transport authorities to vehicle sharing companies, infrastructure managers in different sectors, transport operators and retailers—are increasingly required to make their products available through web services and APIs. These services, however, are exclusively tailored to each organization’s internal standards and models,

<sup>3</sup> <https://maas-alliance.eu/homepage/what-is-maas>.

which are often incompatible with the service interfaces and APIs of other actors (C2).

Then, an essential requirement of interoperability is *Service Standardization* (R2), which refers to the adoption of common standards in various aspects of service management, including interfaces, underlying data models, and service descriptions. Indeed, R2 is the most frequent requirement in the *Service Interoperability Facet*.

**Data Accessibility and Openness (R3)** and **Security and Privacy (R4)**: Requirements R3 and R4 concern data ownership, access and sharing. R3 is part of the *Data Interoperability Facet* and includes two complementary requirements. *Data openness* highlights the importance of the publication of data that is legally and technically open and available in the public domain. *Data accessibility*, instead, concerns further properties of open data, including discoverability, assessability, processability, and re-usability.

The other highly influential factor in fostering data sharing is the existence of mechanisms that ensure data *Security and Privacy*. Requirement R4 belongs to both the *Data* and *System Interoperability Facets*. In enterprise environments—e.g., in the transportation domain— organizations compete with one another to achieve a higher market share, companies have many conflicts of interest, and private sector entities have invested considerable assets to create, collect and process data. Then, security and the ability to fully control one's own data become critical aspects. Accordingly, data owners tend to maximize their profits by making their data and assets exclusively accessible to themselves rather than freely available to everyone, including their own competitors (C3). Similarly, they are inclined to avoid compromising their systems by sharing their data through insecure communication channels (C4). Hence, data management solutions such as those promoted by the International Data Spaces (IDS) Association<sup>4</sup> (where data are shared while respecting the sovereignty of the data owner/creator), or initiatives such as [72] (which introduces a security management mechanism for SoSs) are becoming increasingly popular.

**Discoverability (R5)**: As depicted in Fig. 3, *Discoverability* is a requirement in the *Service* and *Data Interoperability Facets*, since the discovery process is a pivotal aspect in service-oriented computing as well as in data-driven systems. Indeed, a simple, dynamic, and reliable discovery process is the foundation of service-oriented architectures, enabling various primary features, including service composition and binding [43]. Also, the Internet, considered as a distributed and data-driven system, shows how the resource discovery process directly impacts the usefulness and applicability of such a system—it is hard to imagine the Internet today without the giant search engines that provide users with search and discovery functions. Likewise, discovery is becoming a major bottleneck for independent organizations working together in collaborative environments and SoS ecosystems, where the relevant services and data are spread across heterogeneous databases and service registries. In other words, if actors need to spend more time and effort discovering the required resources than implementing/creating the desired services/data, then organizations will favor using their own services and data, which exacerbates

<sup>4</sup> <https://internationaldataspaces.org/why/data-spaces>.

interoperability issues. Conversely, the existence of a reliable and flexible discovery mechanism, which allows interested parties to efficiently find the desired data/services, encourages the use of external organizations' data, the integration of their services, and the cooperation and interaction with them. Yet, the discovery mechanism is essentially a structured way of reading and processing data. Hence, the heterogeneity of data structures, meta-models, and interfaces creates a severe challenge toward the implementation of a discovery mechanism that can handle data coming from different sources (C1).

**Technological neutrality (R6):** This requirement belongs to the *System Interoperability Facet* and stems from the inadequacy of monolithic approaches to the software engineering of complex systems (C5). When the hard coupling of internal components and artifacts of complex systems (and constituent subsystems) meets the heterogeneity in the lower layer of the technology stack, it becomes a serious impediment to the adoption of unified standards at the top layers (i.e., the provided services and data). Accordingly, decoupling the services/functions provided by a system from the underlying enabling technologies is necessary to foster interoperability.

R6 is similar to the standardization requirements in the service and data facets (R1 and R2), but it views the heterogeneity problem from a different perspective. Whereas *Data/Service Standardization* pushes involved parties toward the use of a unified standard that prevents divergent data formats/service interfaces, *Technological neutrality* tackles the same issue by encouraging the development of systems, services and functions in a manner that is agnostic with respect to data formats, technologies and specifications. R6 is an interoperability foundation because of two crucial attributes of large-scale SoSs. First, parties might be distributed across countries or even worldwide, and hence, the use of common standards requires unified—or at least compatible—policies, rules and regulations, in addition to technological requirements. Then, the definition of widely accepted standards requires time-consuming and complex bureaucratic and political activities (C1). Second and more importantly, adaptation to specific standards forces organizations to modify their already-developed systems, which is a long and costly process (C2).

**Integration/orchestration of complementary services (R7):** This requirement, which belongs solely to the *System Interoperability Facet*, stems from the fact that SoS services and products are becoming increasingly complex. For example, in the transportation domain, today's travel services go beyond the simple concepts of booking and ticket generation. Travelers, instead, expect many complementary services (e.g., personalization of travel offers, smart navigation, trip-tracking, re-accommodation, and onboard services such as entertainment) [75] to be available before, during, and after their trips. Furthermore, creating the new forms of travel that are becoming increasingly popular—e.g., door-to-door travel packages—requires an extensive orchestration of multi-modal and cross-border services offered by different operators.

As a result, on one hand, the provision of a complete and full-featured service and product by a single system/organization seems unrealistic and infeasible. On the other hand, fragmented and one-to-one orchestration of multifarious services is a cumbersome and inefficient process (C1,C5). So, the development of mechanisms

and technologies realizing a distributed logical layer that interlinks a vast variety of core and complementary services is a pressing requirement. The existence of such an infrastructure would boost interoperability: Interested parties could collectively discover, compose and orchestrate more complex services/products in a collaborative manner, where atomic services/products are offered by different organizations/systems.

*Automation and Machine-readability (R8)*: Automation is a common concern across all facets of the interoperability trilogy. However, it targets different concepts and approaches at each facet. Concerning services and systems, it mainly refers to the procedures, tools and technologies that remove—possibly partially—the need for human intervention in tasks and operations. This is true also of data-related procedures; however, in the data domain automation also refers to the methodologies and tools that make data machine-readable, and in particular to the structured and semantically enriched data that enable unambiguous interpretation of data without human involvement. As a result, data become assets fed as input or generated as an automated process output.

Automation procedures vary depending on the application domain and on the software design methods [82]. However, we can generally frame automation as a process that breaks a complex procedure down into a chain of multiple intermediate steps that carry a formal—usually abstract—description of the implementation of each task, its requirements—such as libraries or platforms—and expected inputs and outputs. The description is ultimately translated into a machine-readable format that lets machines perform the complete procedure. Hence, an added value of automation is that distributed integration of defined stages becomes possible, which increases the interoperability among involved parties. Moreover, automation makes procedures more structured and portable, which, in turn, greatly improves their interoperability. Accordingly, introducing automated infrastructures and machine-readable data that make inter-organization cooperation smoother is an important requirement to achieve interoperability in complex SoS environments. Nevertheless, the main barriers to reaching full automation in legacy systems are the lack of digitalization and technical infrastructure in creating and managing the automated process (C6).

## 5 Interoperability building blocks

This section reports the results of the second phase of our study as introduced in Sect. 3. It provides an overview of the generic and common practices, techniques and technologies to realize interoperability in SoSs. The main objective is to conduct a critical analysis and comparative evaluation of such approaches to better understand their limitations and opportunities concerning the eight key requirements introduced in Sect. 4.

In the rest of this section, we first introduce the notion of *interoperability stack* in Sect. 5.1, to frame the scope of the interoperability concept that is the target of our research. Then, to better classify the most common interoperability practices and technologies, we structure the discussion based on the ISO categorization of interoperability approaches. More precisely, the well-known ISO 14258 standard

Public Administration	Interoperability Stack	Transportation
Semantic Interoperability	Level III Logical Interoperability	Application, Services and Network
Technical Interoperability		
Legal Interoperability	Level II Legal Interoperability	Legislation
Organizational Interoperability	Level I Physical Interoperability	Infrastructure

Fig. 4 Interoperability stack and its instances in the public administration and transportation domains

[45] identifies three basic approaches to achieving interoperability [16, 17, 65]: *Integrated*, *Unified* and *Federated*. Then, Sects. 5.2–5.4 survey the techniques that are the best practices for each approach and analyze their pros and cons; finally, Sect. 5.5 identifies the approach that best fits the requirements discussed in Sect. 4.1.

## 5.1 Interoperability stack

Based on the characteristics of large, distributed enterprise systems, and with reference to the current state of the art and available interoperability solutions in this domain, we introduce the notion of *interoperability stack*, depicted in Fig. 4, where the realization of interoperability at the higher level of the stack depends on addressing interoperability at the lower levels.

The stack is composed of three levels: physical, legal, and logical. The lower (physical) level refers to everything that is not digital. It includes artifacts such as hardware, infrastructure, and physical materials and objects, but also procedures, strategies such as business models. The middle (legal) level concerns issues related to interoperability and compatibility with regulations and policies, which is an essential element to allow (or hinder) effective cooperation and collaboration among organizations, corporations, sectors and national authorities. Finally, the upper (logical) layer refers to compatibility and interoperation at the conceptual and digital levels, which leads to technological and application interoperability.

The proposed interoperability stack provides a generic and abstract classification that can be instantiated to specific levels in different domains. For example, Fig. 4 shows how the interoperability stack can be instantiated in the public administration and transportation domains. As introduced in Sect. 2, EIF, the reference framework for public administrations, has four layers, which are mapped to our three-layer model as follows. OI covers the integration of business processes, which implies documenting them in an agreed way and with commonly accepted modeling techniques, including the associated information exchanged. In the public administration



domain, the physical artifacts are the documents and business models, and hence, OI could be considered as an instance of the **Physical Interoperability Level**. LI is about ensuring that organizations under different legal frameworks, policies and strategies can work together, which corresponds to the **Legal Interoperability Level**. Finally, SI—which refers to achieving a common understanding of the meaning of data elements and the relationship between them—and TI—which supports the applications and infrastructures linking systems and services—can be mapped to the **Logical Interoperability Level**.

Similarly, seamless and extensive cooperation cannot be achieved in the transportation domain without a multi-layer interoperability model. To make seamless multi-modal cross-border travel a reality, mobility and transportation systems must be interoperable at different levels [13, 34, 35, 44]. At the infrastructure level (which corresponds to the **Physical Interoperability Level**) they must, for example, ensure the physical compatibility of rail tracks, or install suitable telecommunication networks to allow vehicles and infrastructure to communicate. At the legislation level (corresponding to the **Legal Interoperability Level**) they must allow cross-region transportation actors and authorities to smoothly carry out business (in the broad sense of the term) together.

At the application, service, and network level (which corresponds to the **Logical Interoperability Level**) mobility and transportation systems must be able to exchange data, communicate, realize complementary services and—in general—facilitate the cooperation among travel services, application and information providers, MaaS providers, transport authorities, travel agencies, IT and software suppliers.

The focus of our research is the upper level in the interoperability stack. However, as discussed above, it is not possible to fully achieve interoperability at the **Logical Level** before reaching an acceptable degree of interoperability at the **Legal and Physical Levels** (though the digitalization of legal procedures can also help automatically check the compliance with legal interoperability requirements). Hence, the latter should be considered as prerequisites for the former; however, the detailed analysis of lower layers is beyond the scope of this paper. Accordingly, in the rest of this paper, the term “interoperability” refers to the **Logical Level**.

## 5.2 Integrated interoperability

Integrated interoperability is based on the adoption of a single common model—and technology—by interoperating systems; accordingly, it is achieved mainly through standardization. In particular, integrated interoperability is described in [65] as follows:

[Integrated interoperability occurs] where there is a standard format for all constituent systems. Diverse models are interpreted in the standard format, which must be as rich as the constituent system models.

The fundamental characteristic of this approach is that, instead of solving interoperability issues among heterogeneous systems, it tries to prevent incompatibilities at the root, by eliminating any heterogeneity in the first place. The degree of

interoperability achieved by this approach directly depends on the spread of a specific standard. Hence, global interoperability is not achieved in this model unless a model/standard/technology is adopted worldwide.

In principle, if every actor adopted the same standard—and corresponding implementation technology—full harmonization of systems would be achieved. In practice, convincing a large community to adopt a common model/standard/technology can be very difficult—if not outright unfeasible. Hence, the integrated approach seems hardly applicable to address issues related to the *Service* and *System* facets of the interoperability trilogy (see Sect. 4) that are closely related to technologies. That is because access to superior technologies is key in the race among organizations to secure greater market share and popularity. This naturally leads to increased heterogeneity of processes, tools and technologies. Integrated interoperability, however, can be a sound approach for data and service interface interoperability (see requirements **R1** and **R2** of Sect. 4), as witnessed by some well-known examples. A homogeneous representation format can also facilitate the searching and discovery mechanisms up to some extent (**R5**). For example, the XML and WSDL [19] specifications gained worldwide acceptance in the data and service interface domains, respectively, and significantly contributed to the interoperability of web services.

Similarly, in the transportation domain, many initiatives focused on the standardization of data formats [14]. For example, many standards co-exist in the railway systems sub-domain, such as RailML<sup>5</sup> (Railway Markup Language) [61], which covers formats and structures for data exchange among railway applications. TAP TSI<sup>6</sup> (Telematic Applications for Passengers Services), is another widely adopted standard defined by the European Railway Agency, which defines protocols for the exchange of timetables, tariffs, reservations, information about circulating trains, etc. The NeTEx<sup>7</sup> technical standard (which is based on and extends the Transmodel data model<sup>8</sup>) defined by CEN is another widely known specification for the exchange of schedules in public transport. As the examples above show, standards tend to diverge even when considering single modes of transport (e.g., railway) and specific topics (e.g., public transport schedule exchange).

Similar situations are observed in other sectors and domains as well. In general, despite the considerable efforts and interest elicited by the integrated interoperability approach, the successful widespread adoption of common standards has usually been limited to rather small scopes—determined by the scope of control that an organization can manage.

### 5.3 Unified interoperability

Unified interoperability is defined in [65] as follows:

<sup>5</sup> <https://www.railml.org/en/introduction.html>.

<sup>6</sup> [http://taf-jsg.info/?page\\_id=51](http://taf-jsg.info/?page_id=51).

<sup>7</sup> <http://netex-cen.eu>.

<sup>8</sup> <http://www.transmodel-cen.eu>.

[It occurs] where there is a common meta-level structure across constituent models, providing a means for establishing semantic equivalence.

As the definition shows, unified interoperability relies, as in the integrated interoperability approach, on a single common standard/format shared by the different parties. However, unlike in the former case, in unified interoperability this common model is at the meta-level. In other words, it is not used as a standard/format to describe systems, services, or data, but it provides a means to define equivalent representations of different formats through the meta-model: When the meta-model is commonly adopted as a reference, different formats can be translated into one another by first mapping them to the meta-model. The downside of this approach is that while—theoretically—an exact mapping with one hundred percent coverage is achievable, the translation process often involves some semantic and/or information loss. More precisely, the accuracy of the translation—i.e., the amount of semantic and/or information loss that occurs during the translation process—is conditional on the comprehensiveness of the meta-model, completeness of the source and target models with respect to one another, and the precision of the translation process itself. In other words, concept *A* in the source standard would be lost during the translation process, if no equivalent concept exists at the meta-level, or if there is no corresponding concept in the target model, or if the translation mechanism is imperfect. In general, all facets of the interoperability trilogy (data, service and system) can benefit from unified interoperability, though enabling tools are different.

At the system level, a promising methodology enabling unified interoperability is the Model-Driven Architecture (MDA) [38]. In MDA, the unified meta-models are defined at two levels of abstraction, the Computational Independent Model (CIM) and Platform-Independent Model (PIM), which can be automatically converted to the specific target system through the system-level meta-model (Platform-Specific Model, or PSM). All instances of the CIM, PIM and PSM models derived within the MDA architecture are interoperable with models designed by other organizations following the same meta-models.

At the data and service level, the most widely practiced approach to realize unified interoperability is middleware technology. Middleware-based systems can differ in their types and forms [46], but they all share a common idea. A middleware is generally defined as an intermediate layer between two others (e.g., between hardware and application, or between application and network layer) that provides reusable services and functions to hide the complexity and heterogeneity of one layer from the other. In the context of unified interoperability, by “middleware” we refer to the traditional concept of translation adapter [9], which translates the internal data and interfaces of a system to a specific unified model—agreed in advance—and then exposes them to the external parties. Hence, it shields the idiosyncrasy of heterogeneous organizations and creates a unified layer across different systems, enabling them to cooperate and interoperate.

A prominent example of the realization of unified interoperability in practice is the REST (REpresentational State Transfer, [36]) architectural style that proposes a set of lightweight and loosely coupled principles for web service design and implementation. The REST style has significantly enhanced the interoperability of

applications and services at the network and communication layer in essentially all domains that use web technologies. Due to its advantages, REST quickly replaced existing (so-called WS-\*) approaches and related technologies after its introduction [69].

Nowadays, REST is the *de facto* reference architectural style for web services and, given its spread, it could even be considered a successful example of the integrated interoperability model.

Unified interoperability provides greater flexibility compared with the integrated approach and it does not require existing systems to totally revamp their internal models. Hence, in comparison with integrated interoperability, it is more suitable for addressing **R6**. Nevertheless, it still imposes a hard constraint regarding the compatibility of each internal model used in the domain with a single unified model. Hence, as in the integrated approach, the achieved level of interoperability depends on the extent of the acceptance of a unified (meta-)model throughout the domain.

#### 5.4 Federated interoperability

Federated interoperability essentially generalizes unified interoperability, since it also relies on intermediate meta-models to draw correspondences between heterogeneous formats (in the broad sense of the term); however, in this case, there is no single reference meta-model or standard, but multiple ones, to be selected (possibly dynamically) depending on the involved systems. As a consequence, a federated approach to interoperability eliminates the need for an agreement among the actors of a community/domain on a fixed meta-model and makes the decision regarding which meta-model to use a dynamic one.

More precisely, [65] defines federated interoperability as follows:

[It occurs] where models must be dynamically accommodated rather than having a predetermined meta-model. This assumes that concept mapping is done at an ontology level, i.e., semantic level.

As its definition suggests, federated interoperability calls for a *semantic* approach when drawing correspondences between concepts of different systems. More precisely, *semantic interoperability* refers to the ability to exchange and incorporate data between multiple systems based on the meaning of the exchanged information.

In practice, a *syntactic* approach is often used instead of a semantic one. In this case, one-to-one mappings based on the structure of data representations are created between different federated systems. However, this is only a pale approximation of a truly federated approach as, while it eschews a single common model (as in integrated interoperability) and also a single common meta-model (as in unified interoperability), it requires the development of  $n^2$  mappings in a federation of  $n$  systems.

The full potential of federated interoperability, instead, is realized if the approach is applied at the semantic level.

A semantic approach to federated interoperability is based on the capability of providing, often on-the-fly and in an algorithmic and machine-processable manner, meaning to a set of concepts through a set of propositions. It is achieved through one

or more ontologies that establish the exact and agreed meaning of the concepts, and through a set of interpreters distributed across heterogeneous systems, which allow the creation of a common understanding of the meaning of each concept. These interpreters transform the exchanged information from each of the non-interoperable formats of the federated systems to the shared model. This is a scalable approach, as it requires  $n$  interpreters for a federation of  $n$  systems.

The main enabling technologies for semantic federated interoperability are those of the semantic web—i.e., identifiers, ontologies, RDF (Resource Description Framework) schema, Web Ontology Language, etc. Ontologies, in particular, play a pivotal role in this approach. In computer science, an ontology is referred to as a “specification of the conceptualization” [41] that provides a formal description of a set of concepts and the relations among them in a particular domain. As described in [32], an ontology serves both representational and computational purposes, where the former refers to the enumeration of factual domain knowledge and the latter to the process of deriving inferences from represented facts. Indeed, the ability to infer knowledge makes ontologies extensible: through inference, new information can be derived and added to the domain knowledge [32], which is a set of assertions about a set of concepts and is represented using RDF. In RDF, each concept is associated with a unique identifier that refers to an ontology, vocabulary and data domain model. Hence, any machine can seamlessly retrieve the exact meaning of a concept and so process data without any previous knowledge of it or human input. It could then be considered the best approach to address **R8**.

The best vision for the semantic federated interoperability is arguably captured by the *web of linked data* [8]. Tim Berners-Lee defines the linked data principle<sup>9</sup> as a set of best practices to meaningfully interlink structured data from heterogeneous sources. Through the adoption of this principle and the use of standardized web technologies, a data-level interconnection of distributed data silos can be achieved and create a global data space. This global knowledge graph, firstly, promotes the use of search engines and technologies like SPARQL, which provide an expressive querying capability over linked data that can effectively fulfill **R5**. Secondly, it makes linked data applications seamlessly interoperate with multiple federated data sources over diverse domains [7]. Finally, on top of that, Open Linked Data can effectively address **R3**.

While semantic web and linked data approaches have been mainly designed and used for data interoperability, there also have been significant research activities, initiatives and standardization efforts aiming to apply a semantic approach in the context of web services [51, 56, 57, 67], from semantic-based service description and discovery to semantic-based service mashup, composition and orchestration [30, 50, 68, 76]. Accordingly, within the SoSs and enterprise domains, the semantic federated interoperability approach can be used to enhance both the Data and Service Interoperability viewpoints. In particular, addressing standardization—i.e., **R1** and **R2**—in dynamic, complex, and large-scale SoSs seems more feasible in the federated model compared to the other ones.

<sup>9</sup> <https://www.w3.org/DesignIssues/LinkedData.html>.

The discussion above highlights that the federated style is the most promising approach to address interoperability issues in SoSs. Firstly, it allows organizations to cooperate using their own preferred technologies and data models, which fits very well with the most prominent features of SoS ecosystems: diverse technologies and heterogeneous data. Secondly, it does not require pre-existing agreements among actors as a prerequisite for establishing any interactions. Hence, it suits well the dynamic nature of SoSs and the complex business relationships that exist among actors, since it gives more freedom to organizations to choose and change their partners and cooperating systems. Hence, unlike the integrated and unified approaches, the federated style, by definition, can satisfy **R6** and **R7**. Finally, it is worth noting that, while the federated interoperability approach does not directly contribute to fulfilling **R4**, it still facilitates addressing this requirement.

Indeed, the key tenets of this approach are that systems operate in autonomy and are only loosely coupled, typically increasing their privacy.

## 5.5 Discussion

The literature review presented above shows that many approaches seek interoperability through integration (Integrated and Unified Interoperability), which mandates using a common set of technologies and mature standards at various levels. Indeed, if the entire ecosystem operates using a single standard and the same technologies, interoperability among composing systems is achieved naturally. Unfortunately, this approach has been successfully used only in few, exceptional cases, since wide-scale adoption of a standard/technology by a large community is a complex process.

From a technological point of view, the most common approaches are middleware-based solutions. Middlewares act as brokers, which map different systems' internal data/interfaces to a previously agreed data model or set of interfaces [4, 9]. Though these mechanisms make systems interoperable, this solution suffers from several drawbacks. First, the adoption of a single middleware to make  $N$  systems interoperable requires the development of  $N$  different translators/plugins, which considerably increases the design and implementation efforts, complexity and costs. Moreover, middlewares are often implemented through a centralized architecture which seriously hampers scalability, increases performance concerns, and suffers from the single-point-of-failure problem. In addition, the management of SoSs composed of a large number of administratively and geographically distributed organizations requires a distributed and collaborative approach, which is against the very nature of any centralized solution. Finally, middlewares are not robust with respect to technology upgrades, since they require the substitution of all the underlying plugins. In a dynamic and technology-oriented domain characterized by continuous technological advancements, this shortcoming poses a serious problem.

As summarized in Table 3 and discussed in detail in Sect. 5.4, Federated Interoperability is the most suitable approach to overcome the above-mentioned issues and address the key interoperability requirements.

Nevertheless, semantic federated interoperability can be achieved only if suitable infrastructures and technologies exist [73]. So in this direction, to obtain a greater

**Table 3** Coverage by different interoperability approaches of the identified requirements

Interoperability requirements	Interoperability approach		
	Integrated	Unified	Semantic federated
R1	✓	✓	✓
R2	✓	✓	✓
R3	×	×	✓
R4	×	×	+
R5	+	+	✓
R6	×	+	✓
R7	×	+	✓
R8	×	×	✓

Symbol × means the corresponding approach does not address the selected requirement

+ means it can partially address the requirement and ✓ means it has the potential to fully address the requirement

degree of interoperability, the SoS and interoperability community, researchers and practitioners should foster federated interoperability by filling the technological gap. It calls for more contributions to developing and adapting semantic-aware systems, loosely coupled services, self-contained tools, and modular and composable software architectures, which together provide the plug-and-play technologies enabling on-the-fly federated interoperability.

The next section proposes a reference architecture for interoperability-enabling frameworks, which is a first step in the selected direction.

## 6 A conceptual architecture for interoperability frameworks

The discussion of Sect. 5 highlights that a proper federated interoperability solution for large collaborative SoS must meet the following features.

(I) It should enable interoperation while preserving the full autonomy of individual systems with respect to the adoption of any desired standards and technologies. Indeed, making systems interoperable should not equate to preventing heterogeneity, not only because such an approach is unfeasible in practice, but also because diversity is the key to innovation and technological advancements. Accordingly, a proper interoperability solution should allow systems to interact even if they employ heterogeneous technologies. (II) To provide a durable solution, it must be coherent with the distributed nature of SoSs, and it must be able to cope with—and adapt to—the changes and the dynamics of the ecosystem, in terms of both business and technological aspects. (III) Instead of providing a single medium that makes systems interoperable (e.g., software bus or middleware), it must define a set of interoperability

building blocks that can be created, shared and discovered and that can collaboratively evolve.

These desired features led us to the formulation of the following core design goals (DG) for the IF.<sup>10</sup>

- (DG1) **Enabling a Federated Approach to Interoperability:** The IF shall establish an infrastructure that facilitates a federated approach to interoperability, effectively addressing the distinct requirements within each facet of the interoperability trilogy, as described in Sect. 4.
- (DG2) **Semantic Approach to Federated Interoperability:** One of IF's primary goals is to overcome the fragmentation of SoS ecosystems by fostering semantic interoperability. Hence, it shall focus on provisioning and leveraging technologies to enable a semantic-based approach.
- (DG3) **Flexibility, Extensibility, and Reusability:** The IF shall establish a flexible, extensible, and reusable infrastructure capable of gradual adaptation and evolution in response to emerging technologies and new requirements.

Consequently, the IF design must be oriented toward offering an extensive suite of interchangeable, composable, and loosely coupled interoperability tools and services, complemented by a set of deployment principles. This approach shall enable systems within SoSs to seamlessly communicate, utilize external data and services, and collaboratively develop new tools and applications without necessitating any alterations to their internal interfaces, standards, or systems. An integral aspect of the IF must be an emphasis on the efficient semantic linking of data, thereby reducing the need for data and service replication. This efficiency could be achieved through a mechanism that automates the generation of semantic descriptions and meta-data for submitted data and assets. Rather than acting as another intermediary or middleware for bidirectional translations between diverse organizations, the IF shall serve as an advanced distributed registry for data and services, accompanied by a suite of semantic interoperability tools. By generating uniform semantic-based descriptions, the IF conceals data and service idiosyncrasies, rendering them discoverable across federated IF nodes. However, the scope of the IF shall transcend that of a mere distributed registry. It aspires to realize an open shared distributed processing environment by providing loosely coupled and standard-agnostic interoperability tools. Users can collectively select and assemble these tools based on their specific requirements, creating a tailored and special-purpose middleware that is ready for deployment.

The suggested high-level architecture of the IF, depicted in Fig. 5, offers a comprehensive view of its components based on the interoperability trilogy model (see Fig. 2). At its core, the architecture features the *AM* component, surrounded by a group of sub-components that collectively address *System Interoperability* within the IF. Additionally, two logical abstractions, namely *Data Abstraction* and *Service*

<sup>10</sup> Recall that, as mentioned in Sect. 1, for ease of writing we use the term “the IF” to refer to our proposed reference architecture for interoperability-fostering frameworks.



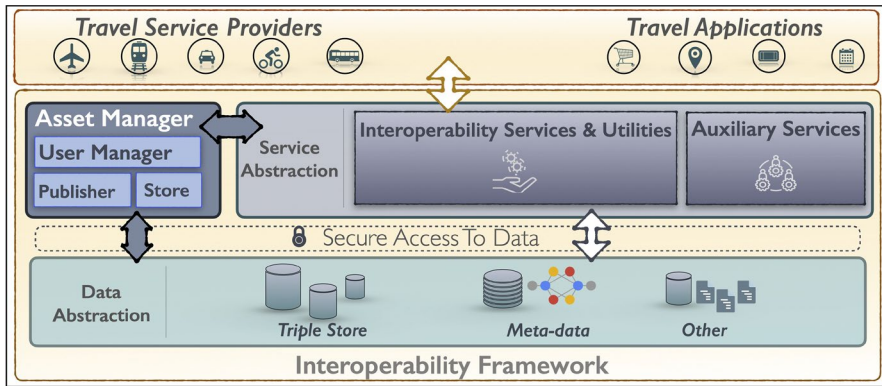


Fig. 5 Conceptual IF architecture

*Abstraction*, encompass the IF components primarily involved in handling *Data* and *Service Interoperability* aspects.

## 6.1 Data abstraction

Data Abstraction plays a crucial role in realizing a web of data and ensuring semantic data interoperability within the IF. The term “data” can encompass various categories and types. For example, in the transportation domain, it includes ontologies, supply chain and logistic data, code lists, ticketing and payment data, historical mobility data, traffic data, fares’ data, and more. In the absence of a unified standard, these data are currently modeled and represented using a heterogeneous set of specifications, vocabularies, and data models. To overcome this challenge, *Data Abstraction* contributes to addressing **DG2** by automatically converting and enriching heterogeneous data into a common shared semantic graph. In line with **DG1**, the primary objective of the *Data Abstraction* is to address requirements **R1**, **R3**, **R5**, and **R8**. It mainly consists of back-end databases and triple stores, complemented by front-end interfaces and mechanisms managed by the AM. These mechanisms handle various operations, including linked-data and meta-data creation and management (related to **R3** and **R5**), as well as the collection, storage, and retrieval of data, ontologies, vocabularies, and meta-data provided by different parties (related to **R1**, **R3**, and **R5**), or created by internal components of the IF (related to **R8**).

## 6.2 Service abstraction

In accordance with **DG2**, the IF prioritizes the utilization of semantic interoperability. As explained earlier, two systems achieve semantic interoperability when the exchanged data can be clearly understood and consistently interpreted in both systems. To meet the objectives outlined in **DG3**, we emphasize the importance of providing flexible and reusable approaches to fulfill interoperability requirements. In

alignment with these goals, the *Service Abstraction* is designed using a modular and service-oriented approach, consisting of a set of self-contained, reusable, and composable services and utilities. The *Service Abstraction* addresses **DG1** by fulfilling requirements **R2**, **R5**, and **R8**.

In this context, we introduce three core types of interoperability services as foundational elements within the IF: converters, mapping mechanisms and ontology management services. More specifically, we can distinguish two groups of semantic interoperability requirements. The first set of requirements is associated with the ontology engineering process involved in creating an ontology in the first place, which is the goal of *Ontology Editors*. These requirements, similar to the system interoperability requirements (see **R6**, **R7**, and **R8**), focus on facilitating ontology engineering through automation or by enabling a collaborative approach to ontology development and publication following FAIR principles [83]—for example, by fostering ontology integration. The second set of requirements pertains to activities after the ontology’s development. They revolve around promoting effective ontology usage, a primary goal of *Converters and Mapping Tools*. Additionally, these requirements encompass the need for accessibility, distribution, and discoverability of ontologies, which are common to all asset types (**R1**, **R3**, and **R5**).

*Converter* Utilizing conversion as a prevalent strategy to tackle interoperability issues involves transforming heterogeneous standards into a formal model comprehensible by all participating systems. Consequently, within the IF, converters emerge as fundamental components. In particular, the IF promotes mechanisms such as those described in [12, 74], which facilitate semantic interoperability by automating the transformation of diverse standards into a semantic reference model serving as an intermediate point of reference. This approach’s value lies in lifting and lowering data to and from the ontological level, allowing for meaningful data conversion at the conceptual level, not merely the syntactic level. Each mapping between the semantic model and other standards directly benefits the conversion process to any already-aligned ontology. This eliminates the need for multiple point-to-point conversions, which can become unwieldy as the number of data formats/standards and their respective data models increases. With semantic interoperability technologies, only the mappings to the ontological level need to be maintained and aligned over time, simplifying the complexity associated with managing multiple conversions.

*Mapping Tool* As mentioned above, addressing the significant diversity in data representations within as SoS often involves adopting effective strategies for conversion mechanisms. Typically, these conversion mechanisms rely on mappings, whether explicitly or implicitly defined. These mappings essentially establish a correspondence between concepts or terms in one standard to their counterparts in another standard. The process of mapping plays a pivotal role in the conversion procedure. Mapping tools, then, are key components to achieve the IF’s goal of enhancing interoperability. An illustrative example of this is the *Mapping Tool* [42, 48, 49, 80], designed to facilitate the creation of one-to-one mappings between analogous concepts in two distinct specifications. To ensure semantic interoperability, this mapping process is executed through a two-phase approach, encompassing both semantic and syntactic matchmaking stages. Specifically, the tool employs the *Word2Vec* algorithm [58] to create word embeddings. It leverages a *Word2Vec*-trained

model to identify linguistic similarities among terms and concepts within the given standards. Subsequently, the second phase employs various heuristics to determine the structural similarity among terms within each standard. The resulting set of selected mappings can then be harnessed by *IF Converters* to streamline the data conversion process.

*Ontology Management* The increasing number of available online ontologies has led to a shift in ontology development, emphasizing the reuse of existing ontologies and modules [79]. This approach considers ontology development as the construction of a network of ontologies, where various resources may be managed by different individuals or organizations. With this collaborative vision of ontology engineering, it becomes vital to provide robust methodological support for the development of ontology networks.

To address this need, the IF can integrate tools such as *OnTology* [2], making it accessible as a utility discoverable and deployable through the AM. *OnTology* serves as a tool for automated and cooperative ontology engineering. Specifically, it focuses on three key aspects: (i) Automating coarse-grained support activities involved in ontology development, including documentation, versioning, evaluation, and publication of ontologies that are maintained and versioned in a git-based environment; (ii) implementing a workflow for continuous integration of support activities whenever new changes are made to an ontology; and (iii) integrating this workflow with a collaborative environment, enabling seamless collaboration among developers.

### 6.3 Assets and asset manager

To align with the objectives of **DG1** and **DG3**, we introduce the *Asset* and *Asset Manager* components as integral parts of the IF. Serving as the gateway to the IF for both consumers and contributors, the AM plays a pivotal role. It provides users with the necessary administrative procedures to operate within a distributed, collaborative, and cross-organizational framework, addressing requirements **R6**, **R7**, and **R8**. Additionally, the AM oversees the essential operations for the comprehensive governance and accessibility of the IF, thus fulfilling the objectives of **DG1**. Furthermore, the AM takes on the responsibility of addressing select security concerns (related to **R4**) that are inherent in a multilateral open ecosystem. This includes managing authorization and access rights, and ensuring the ability to carry out specific processes or actions on assets while maintaining the necessary security protocols. By encompassing these functions, the AM contributes to the overall effectiveness of the IF and enhances its capacity to facilitate the achievement of its core design goals.

Broadly speaking, its function encompasses that of an official asset catalog, overseeing assets through distinct publication procedures. In pursuit of **DG1** and the advancement of the federated interoperability approach, AM offers basic capabilities to publish, share, discover, maintain, and oversee the artifacts utilized and contributed by users, as well as those forming integral components of the IF. Essentially, the AM establishes a unified repository of dispersed digital assets, simplifying the

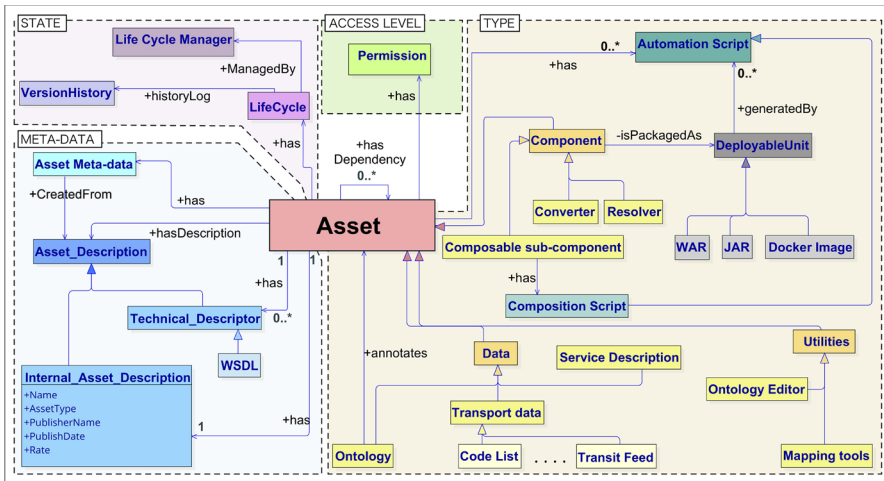


Fig. 6 Asset meta-model

exploration and accessibility of all resources and tools essential for achieving interoperability. This encompasses a wide range of elements, such as ontologies, converters, data schemas, and service descriptors. By fulfilling these roles, the AM contributes significantly to the IF's objectives and the realization of effective interoperability practices.

Within the IF, the notion of “Asset” (see Fig. 6) plays a pivotal role in the pursuit of **DG3**. An asset is defined as any discernible artifact provided with a comprehensive description, rendering it easily discoverable by users and other interconnected assets. An asset has a *Type*, a *Lifecycle*, some *Meta-Data* that describes it and some *Permissions* that determine who can access it. Every asset has a description, which is the main element in the materialization of the *Data Abstraction* in the IF. In particular, an asset has a *Generic Description* (which can be built upon the DCAT-AP [64] and ADMS [29] specifications, for example), which is generated during asset registration. In addition, depending on its *Type*, an asset might have *Technical Descriptions* containing detailed information on how to use it.

The AM should automatically transform these descriptions into RDF format (shown as *Asset Meta-data* in Fig. 6) and store them in the available *RDF Repositories*, which could be then used by the distributed SPARQL engines to enable the federated and semantic discoverability of assets.

There are three categories of assets: **Data**, **Utilities** and **Components**. An asset of type **Data**, besides the asset description and meta-data, is itself part of the materialization of the *Data Abstraction* and it may include any kind of data in the relevant domain. For example, in the transportation domain data could include transportation ontologies, supply chain and logistic data, code lists, ticketing and payment data, historical mobility data, traffic data, etc. **Utilities** (e.g., *Ontology Editor*, *Mapping Tool*) and **Components** (e.g., *Converter*) are the realization of the

*Service Abstraction.* As discussed above, **Utility** and **Component** assets are tools and services that might be provided and used by external actors as well as the IF itself. Furthermore, as represented in Fig. 6, a **Component** asset could be packaged as different deployable units. This enables multiple deployment and engagement choices for IF's clients and increases the usability of the IF. Also, to fulfill requirements **R6** and **R8**, we argue that the IF should also offer the possibility to automate the whole lifecycle of a **Component**, from its creation to the deployment stage.

Finally, the diverse user landscape that is an integral part of SoSs prompted us to design the deployment of the IF so as to cover a wide range of use cases. Direct Access is a standard mechanism for facilitating loosely coupled and service-oriented interoperability. In this case, the publisher provides—for each of its components—a description with an endpoint, which the AM transforms into discoverable metadata. The IF acts as a repository, facilitating discovery, and then redirects the client to the provider system for service interaction. The Runtime Executable Environment deployment strategy, on the other hand, promotes a Platform as a Service (PaaS) approach. In this case, the AM takes a proactive role, managing deployable artifacts on a cloud platform. Authorization is granted by the service provider or an external infrastructure provider. After discovery, the AM orchestrates execution and provides the client with an access endpoint. Lastly, for local or integrated use, the IF supports Direct Downloading of packaged component implementations. Various artifacts, such as container images or WAR files, cater to different preferences. Clients choose the most suitable deployment option depending on their system architecture and requirements.

## 7 Conclusions

This paper focused on the practical challenges, requirements and approaches for addressing interoperability problems in distributed and collaborative systems. We collected practitioners' views and experiences on dealing with interoperability in actual projects by conducting a questionnaire-based survey. Starting from this survey, we identified in (Sect. 3) three research questions, which have been addressed as follows. To answer RQ1, we have identified (Sect. 4) eight fundamental and domain-agnostic requirements that every system must address to ensure full interoperability. Furthermore, six challenges that hinder addressing such requirements have been specified. Question RQ2 was addressed by performing a critical analysis of the literature and current state of the art in interoperability approaches, through which we identified the best practices to fulfill the identified requirements. The analysis concluded (Sect. 5.5) that a semantic federated approach is best suited to address the identified requirements.

To the best of our knowledge, none of the available interoperability solutions (see Sect. 2) fully address the requirements identified in Sect. 4. For this reason, to address RQ3, we proposed (Sect. 6) a reference architecture—the IF—for interoperability-enabling frameworks for SoSs; for each component of the architecture we pointed out the requirements that it addresses, which provides the rationale for including the component in the architecture. The identified components cover,

altogether, all the identified requirements. The architecture exploits the best practices discussed in Sect. 5 and, to the best of our knowledge, is the first one to specifically address the interoperability problem through a semantic federated approach.

The creation and deployment of stable implementations of the IF components—some prototypes are available, as mentioned in Sect. 6.2—will be an important step to facilitate, in practice, interoperability in SoSs. Notice that, since different components address different requirements, a specific SoS might necessitate only a subset of the elements identified in Sect. 6, depending on its features and interoperability needs. Hence, an SoS might deploy only a limited number of IF components. This can decrease the effort of realizing the IF in practice, thus increasing its impact.

**Acknowledgements** This work was supported by Shift2Rail and the EU Horizon 2020 research and innovation program under Grant agreement No: 826172 (SPRINT). The authors would like to thank all partners of the SPRINT project for their contributions.

**Author Contributions** MS, AC, OC, MR, and RS collectively participated in the conceptualization, design, and execution of the entire research project. In addition, MS and AV were particularly involved in the requirement analysis, design, and analysis of the survey study section. All authors played roles in the discussions, analysis, and refinement of results and concepts. MS initiated the initial draft of the manuscript, which underwent a collaborative review by all authors. Subsequently, MS and MR consolidated the feedback and comments to prepare the final version of the manuscript.

**Funding** Open Access funding enabled and organized by Projekt DEAL. This work was supported by Shift2Rail and the EU Horizon 2020 research and innovation program under Grant agreement No: 826172 (SPRINT).

**Availability of data and materials** The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest** The authors have no financial or proprietary interests in any material discussed in this article.

**Ethical approval** Not applicable.

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

## References

1. Abbott R (2006) Open at the top; open at the bottom; and continually (but slowly) evolving. In: 2006 IEEE/SMC International Conference on System of Systems Engineering, IEEE, pp 6–pp
2. Alobaid A, Garijo D, Poveda-Villalón M et al (2019) Automating ontology engineering support activities with OnToology. *J Web Semant* 57:100472

3. Archimède B, Vallespir B (2017) Enterprise interoperability: INTEROP-PGSO vision. John Wiley & Sons, New Jersey
4. Baresi L, Sadeghi M, Valla M (2018) Tdex: A description model for heterogeneous smart devices and gui generation. In: 2018 IEEE International Conference on Internet of Things (iThings), IEEE, pp 97–104
5. Berre AJ, Elvesæter B, Figay N et al (2007) The athena interoperability framework. Enterprise interoperability II: new challenges and approaches. Springer, Berlin, pp 569–580
6. Billaud S et al (2015) Interoperability as a key concept for the control and evolution of the system of systems (sos). International IFIP Working Conference on Enterprise Interoperability. Springer, Berlin, pp 53–63
7. Bizer C (2009) The emerging web of linked data. IEEE Intell Syst 24(5):87–92
8. Bizer C, Heath T, Berners-Lee T (2011) Linked data: The story so far. Semantic services, interoperability and web applications: emerging concepts. IGI Global, Pennsylvania, pp 205–227
9. Blair GS, Paolucci M, Grace P et al (2011) Interoperability in complex distributed systems. International school on formal methods for the design of computer. Springer, Communication and Software Systems, pp 1–26
10. Boardman J, Sauser B (2006) System of systems-the meaning of of. In: 2006 IEEE/SMC International Conference on System of Systems Engineering, IEEE, pp 6–pp
11. Burns T, Cosgrove J, Doyle F (2019) A review of interoperability standards for industry 4.0. Proc Manuf 38:646–653
12. Carenini A, Ugo D, Stefanos G et al (2018) ST4RT-semantic transformations for rail transportation. Transp Res Arena TRA 2018:1–10
13. CEN (n.d.) Cen european reference data model for public transport information (transmodel). <http://www.transmodel-cen.eu>, Accessed 11 Oct 2021
14. Chaves-Fraga D, et al (2023) Editorial of transport data on the web. Semantic Web
15. Chen D (2006) Enterprise interoperability framework. In: EMOI-INTEROP, pp 1–5
16. Chen D, Daclin N (2006) Framework for enterprise interoperability. In: Interoperability for Enterprise Software and Applications: Proceedings of the Workshops and the Doctorial Symposium of the Second IFAC/IFIP I-ESA International Conference: EI2N, WSI, IS-TSPQ 2006, Wiley Online Library, pp 77–88
17. Chen D, Vernadat FB (2002) Enterprise interoperability: a standardisation view. International Conference on Enterprise Integration and Modeling Technology. Springer, Berlin, pp 273–282
18. Chen D, Doumeingts G, Vernadat F (2008) Architectures for enterprise integration and interoperability: past, present and future. Comput Ind 59(7):647–659
19. Chinnici R, Moreau JJ, Ryman A, et al (2007) WSDL 0.2 specification. <https://www.w3.org/TR/2007/REC-wsdl20-20070626/>
20. Ciepły J, Disselkamp JP, Dyck F et al (2023) A systematic literature review on the digital twin of the factory in the context of the product and factory lifecycle. Proc CIRP 119:834–840
21. Corcho O, Eriksson M, Kurowski K, et al (2021) EOSC interoperability framework: Report from the EOSC executive board working groups FAIR and architecture
22. Council of the European Union (2015) Digital services interoperability programme-isa2. <https://www.consilium.europa.eu/en/press/press-releases/2015/06/12/digital-services-interoperability-programme-isa2>, Accessed 11 Oct 2021
23. Council of the European Union (2015) Interoperability for EU public administrations, businesses and citizens (ISA2). <https://data.consilium.europa.eu/doc/document/ST-9366-2015-INIT/en/pdf>, Accessed 11 Oct 2021
24. Council of the European Union (2018) Infographic - How interoperable databases will boost Europe's security. <https://www.consilium.europa.eu/en/infographics/interoperability/>, Accessed 11 Oct 2021
25. Council of the European Union (2019) Interoperability between EU information systems. <https://www.consilium.europa.eu/en/press/press-releases/2019/02/05/interoperability-between-eu-information-systems-council-presidency-and-european-parliament-reach-provisional-agreement/>, Accessed 11 Oct 2021
26. Council of the European Union (2019) Interoperability between EU information systems in the field of borders and visa. <https://data.consilium.europa.eu/doc/document/PE-30-2019-INIT/en/pdf>
27. Council of the European Union (2019) Interoperability between EU information systems in the field of police and judicial cooperation, asylum and migration. <https://data.consilium.europa.eu/doc/document/PE-31-2019-INIT/en/pdf>, Accessed 11 Oct 2021

28. Cuenca L, Boza A, Ortiz A, et al (2015) Conceptual interoperability barriers framework (cibf). In: Proceedings of the 17th International Conference on Enterprise Information Systems-Volume 2, SCITEPRESS-Science and Technology Publications, Lda, pp 521–531
29. Dekkers M (2013) Asset description metadata schema (adms). W3C Working Group
30. Di Martino B (2009) Semantic web services discovery based on structural ontology matching. *Int J Web Grid Serv* 5(1):46–65
31. DiMario MJ (2006) System of systems interoperability types and characteristics in joint command and control. In: 2006 IEEE/SMC International Conference on System of Systems Engineering, IEEE, pp 6–pp
32. Ding L, Koları P, Ding Z et al (2007) Using ontologies in the semantic web: a survey. *Ontologies*. Springer, New York City, pp 79–113
33. European Commission (2017) New european interoperability framework. [https://ec.europa.eu/isa2/eif\\_en](https://ec.europa.eu/isa2/eif_en), Accessed 11 Oct 2021
34. European Commission (2021) Transport research and innovation monitoring and information system. <https://trimis.ec.europa.eu>, Accessed 11 Oct 2021
35. European Council (2008) Directive 2008/57/ec on the interoperability of the rail system within the community. [https://ec.europa.eu/transport/modes/rail/interoperability/interoperability\\_en](https://ec.europa.eu/transport/modes/rail/interoperability/interoperability_en), Accessed 11 Oct 2021
36. Fielding RT, Taylor RN (2000) Architectural styles and the design of network-based software architectures, vol 7. University of California, Irvine
37. Ford TC, Colombi JM, Graham SR et al (2007) A survey on interoperability measurement. *Gateways* 2:3
38. Frankel DS (2003) Model driven architecture applying MDA. John Wiley & Sons, New Jersey
39. Geraci A, Katki F, McMonegal L et al (1991) IEEE standard computer dictionary: compilation of IEEE standard computer glossaries. IEEE Press, New York
40. Göllner D, Rasor R, Anacker H et al (2022) Collaborative modeling of interoperable digital twins in a sos context. *Proc CIRP* 107:1089–1094
41. Gruber TR (1993) A translation approach to portable ontology specifications. *Knowl Acquis* 5(2):199–221
42. Hosseini M, Kalwar S, Rossi MG, et al (2019) Automated mapping for semantic-based conversion of transportation data formats. In: CEUR Workshop Proceedings, pp 1–6
43. Huhns MN, Singh MP (2005) Service-oriented computing: key concepts and principles. *IEEE Internet Comput* 9(1):75–81
44. Information Technology for Public Transport (2021) ITxPT specifications. <https://itxpt.org/technology/itxpt-specifications/>, Accessed 11 Oct 2021
45. International Organization for Standardization (1998) ISO 14258: Industrial automation systems - Concepts and rules for enterprise models. <https://www.iso.org/obp/ui/#iso:std:iso:14258:ed-1:v1:en>, Accessed 11 Oct 2021
46. Issarny V, Caporuscio M, Georgantas N (2007) A perspective on the future of middleware-based software engineering. In: Future of software engineering (FOSE'07), IEEE, pp 244–258
47. Jittrapirom P, Caiati V, Feneri AM et al (2017) Mobility as a service: a critical review of definitions, assessments of schemes, and key challenges. *Urban Plan* 2(2):13–25
48. Kalwar S, Sadeghi M, Sabet AJ, et al (2021) Smart: Towards automated mapping between data specifications. In: The 33rd International Conference on Software Engineering and Knowledge Engineering, SEKE, pp 1–10
49. Kalwar S, Rossi M, Sadeghi M (2023) Automated creation of mappings between data specifications through linguistic and structural techniques. *IEEE Access* 11:30324–30339
50. Klusch M, Kapahnke P, Schulte S et al (2016) Semantic web service search: a brief survey. *KI-Künstliche Intelligenz* 30(2):139–147
51. Kopecký J, Vitvar T, Bournez C et al (2007) Sawsdl: semantic annotations for wsdl and xml schema. *IEEE Internet Comput* 11(6):60–67
52. Labreche M, Montarnal A, Truptil S et al (2020) Towards a framework for federated interoperability to implement an automated model transformation. Working Conference on Virtual Enterprises. Springer, Berlin, pp 143–152
53. Lane JA, Valerdi R (2011) System interoperability influence on system of systems engineering effort. In: Proceedings of the Conference on Systems Engineering Research
54. Leal G, et al (2019) Interoperability assessment: a systematic literature review. *Computers in Industry* 106



55. Maier MW (1998) Architecting principles for systems-of-systems. *Syst Eng J Int Council Syst Eng* 1(4):267–284
56. Martin D, Burstein M, Hobbs J, et al (2004) Owl-s: Semantic markup for web services. W3C member submission 22(4)
57. McIlraith SA, Son TC, Zeng H (2001) Semantic web services. *IEEE Intell Syst* 16(2):46–53
58. Mikolov T, Chen K, Corrado G, et al (2013) Efficient estimation of word representations in vector space. arXiv preprint [arXiv:1301.3781](https://arxiv.org/abs/1301.3781)
59. Mordecai Y, Orhof O, Dori D (2016) Model-based interoperability engineering in systems-of-systems and civil aviation. *IEEE Trans Syst Man Cybern Syst* 48(4):637–648
60. Morris E, Levine L, Meyers C et al (2004) System of systems interoperability (sosi). Carnegie Mellon Univ, Software Engineering Institute
61. Nash A, Huerlimann D, Schütte J, et al (2004) RailML—a standard data interface for railroad applications. *WIT Trans Built Environ* 74
62. Neiva FW et al (2016) Towards pragmatic interoperability to support collaboration: a systematic review and mapping of the literature. *Inf Softw Technol* 72:137–150
63. Nielsen CB, Larsen PG, Fitzgerald J et al (2015) Systems of systems engineering: basic concepts, model-based techniques, and research directions. *ACM Comput Surv (CSUR)* 48(2):1–41
64. Nuffelen BV (2020) DCAT application profile for data portals in europe. <https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe/release/201-0>
65. Panetto H, Cecil J (2013) Information systems for enterprise integration, interoperability and networking: theory and applications. *Enterp Inf Syst* 7(1):1–6
66. Paniagua C (2020) Autonomous runtime system of systems interoperability. PhD thesis, Luleå University of Technology
67. Paolucci M, Kawamura T, Payne TR et al (2002) Semantic matching of web services capabilities. *International Semantic Web Conference*. Springer, Berlin, pp 333–347
68. Rao J, Su X (2004) A survey of automated web service composition methods. *International workshop on semantic web services and web process composition*. Springer, Berlin, pp 43–54
69. Richardson L, Ruby S (2008) RESTful web services. O'Reilly, California
70. Ruggaber R (2006) Athena-advanced technologies for interoperability of heterogeneous enterprise networks and their applications. *Interoper Enterp Softw Appl* 1:459–60
71. Sadeghi M, Carenini A, Corcho O, et al (2023) Interoperability of heterogeneous systems of systems: Review of challenges, emerging requirements and options. In: *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, SAC '23*, pp 741–750, <https://doi.org/10.1145/3555776.3577692>
72. Sadeghi M et al (2022) A semantic-based access control approach for systems of systems. *ACM SIGAPP Appl Comput Rev* 21(4):5–19
73. Sadeghi M, et al (2022) Sprint: Semantics for performant and scalable interoperability of multimodal transport. arXiv preprint [arXiv:2203.14568](https://arxiv.org/abs/2203.14568)
74. Scrocca M, Comerio M, Carenini A et al (2020) Turning transport data to comply with EU standards while enabling a multimodal transport knowledge graph. *International Semantic Web Conference*. Springer, Berlin, pp 411–429
75. Shapps G, Williams K (2021) Great british railways: The williams-shapps plan for rail. Tech. rep., Department for Transport, available from: <https://www.gov.uk/government/publications/great-british-railways-williams-shapps-plan-for-rail>
76. Sirin E, Hendler J, Parsia B (2003) Semi-automatic composition of web services using semantic descriptions. In: *1st Workshop on Web Services: Modeling, Architecture and Infrastructure*, pp 17–24
77. SPRINT Consortium (2020) D2.1 – initial analysis of requirements of S2R IP4 projects and other initiatives. available from <http://sprint-transport.eu/Page.aspx?CAT=DELIVERABLES&IdPage=1e2645be-e780-4d99-8117-bae57b67b453>, Accessed 13 Oct 2023
78. SPRINT Consortium (2020) D2.2—requirements for an IF architectural design (C-REL). available from <http://sprint-transport.eu/Page.aspx?CAT=DELIVERABLES&IdPage=1e2645be-e780-4d99-8117-bae57b67b453>, Accessed 14 Dec 2021
79. Suárez-Figueroa MC, Gómez-Pérez A, Fernández-López M (2012) The neon methodology for ontology engineering. *Ontology engineering in a networked world*. Springer, Berlin, pp 9–34
80. Vago NOP, Sacaj M, Sadeghi M, et al (2021) On the visualization of semantic-based mappings. In: *Third International Workshop On Semantics And The Web For Transport*
81. Vernadat F (1996) Enterprise modeling and integration. Boom Koninklijke Uitgevers, Netherlands

82. Vyatkin V (2013) Software engineering in industrial automation: state-of-the-art review. *IEEE Trans Ind Inf* 9(3):1234–1249
83. Wilkinson MD, Dumontier M, Aalbersberg IJ et al (2016) The FAIR guiding principles for scientific data management and stewardship. *Sci Data* 3(1):1–9
84. Yahia E, Aubry A, Panetto H (2012) Formal measures for semantic interoperability assessment in cooperative enterprise information systems. *Comput Ind* 63(5):443–457

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

## Authors and Affiliations

**Mersedeh Sadeghi<sup>1</sup> · Alessio Carenini<sup>2</sup> · Oscar Corcho<sup>3</sup> · Matteo Rossi<sup>4</sup> · Riccardo Santoro<sup>5</sup> · Andreas Vogelsang<sup>1</sup>**

✉ Mersedeh Sadeghi  
sadeghi@cs.uni-koeln.de

Alessio Carenini  
alessio.carenini@cefriel.com

Oscar Corcho  
ocorcho@fi.upm.es

Matteo Rossi  
matteo.rossi@polimi.it

Riccardo Santoro  
r.santoro@fsitaliane.it

Andreas Vogelsang  
vogelsang@cs.uni-koeln.de

<sup>1</sup> Institute of Computer Science, University of Cologne, Cologne, Germany

<sup>2</sup> CEFRIEL, Milan, Italy

<sup>3</sup> ETSI Informáticos, Universidad Politécnica de Madrid, Madrid, Spain

<sup>4</sup> Dipartimento di Meccanica, Politecnico di Milano, Milan, Italy

<sup>5</sup> Ferrovie dello Stato, Rome, Italy