

# Towards Cognitive Ports of the Future



**Santiago Cáceres, Francisco Valverde, Carlos E. Palau, Andreu Belsa Pellicer, Christos A. Gizelis, Dimosthenes Krassas, Hanane Becha, Réda Khouani, Andreas Metzger, Nikos Tzagkarakis, Anthousa Karkoglou, Anastasios Nikolakopoulos, Achilleas Marinakis, Vrettos Moulos, Antonios Litke, Amir Shayan Ahmadian, and Jan Jürjens**

**Abstract** In modern societies, the rampant growth of data management technologies—that have access to data sources from a plethora of heterogeneous systems—enables data analysts to leverage their advantages to new areas and critical infrastructures. However, there is no global reference standard for data platform technology. Data platforms scenarios are characterized by a high degree of heterogeneity at all levels (middleware, application service, data/semantics, scalability, and governance), preventing deployment, federation,

---

S. Cáceres · F. Valverde  
Instituto Tecnológico de Informática, Valencia, Spain  
e-mail: [scaceres@iti.es](mailto:scaceres@iti.es); [fvalverde@iti.es](mailto:fvalverde@iti.es)

C. E. Palau · A. B. Pellicer  
Universitat Politècnica de Valencia, Valencia, Spain  
e-mail: [cpalau@dcom.upv.es](mailto:cpalau@dcom.upv.es); [anbelpel@upv.es](mailto:anbelpel@upv.es)

C. A. Gizelis · D. Krassas  
Hellenic Telecommunications Organization S.A., Maroussi, Athens, Greece  
e-mail: [cgkizelis@cosmote.gr](mailto:cgkizelis@cosmote.gr); [dimkrass@ote.gr](mailto:dimkrass@ote.gr)

H. Becha · R. Khouani  
Traxens, Marseille, France  
e-mail: [h.becha@traxens.com](mailto:h.becha@traxens.com); [r.khouani@traxens.com](mailto:r.khouani@traxens.com)

A. Metzger  
Ruhr Institute for Software Technology, University of Duisburg-Essen, Essen, Germany  
e-mail: [andreas.metzger@paluno.uni-due.de](mailto:andreas.metzger@paluno.uni-due.de)

N. Tzagkarakis · A. Karkoglou · A. Nikolakopoulos · A. Marinakis · V. Moulos (✉) · A. Litke  
National Technical University of Athens, Zografou, Athens, Greece  
e-mail: [ntzagkarakis@mail.ntua.gr](mailto:ntzagkarakis@mail.ntua.gr); [akarkoglou@mail.ntua.gr](mailto:akarkoglou@mail.ntua.gr); [tasosnikolakop@mail.ntua.gr](mailto:tasosnikolakop@mail.ntua.gr); [achmarin@mail.ntua.gr](mailto:achmarin@mail.ntua.gr); [vrettos@mail.ntua.gr](mailto:vrettos@mail.ntua.gr); [litke@mail.ntua.gr](mailto:litke@mail.ntua.gr)

A. S. Ahmadian  
Institute for Software Technology, University of Koblenz-Landau, Koblenz, Germany  
e-mail: [ahmadian@uni-koblenz.de](mailto:ahmadian@uni-koblenz.de)

J. Jürjens  
Fraunhofer-Institute for Software and Systems Engineering ISST, Dortmund, Germany  
e-mail: [juerjens@uni-koblenz.de](mailto:juerjens@uni-koblenz.de)

and interoperability of existing solutions. Although many initiatives are dealing with developing data platform architectures in diversified application domains, not many projects have addressed integration in port environments with the possibility of including cognitive services. Unlike other cases, port environment is a complex system that consists of multiple heterogeneous critical infrastructures, which are connected and dependent on each other. The key pillar is to define the design of a secure interoperable system facilitating the exchange of data through standardized data models, based on common semantics, and offering advanced interconnection capabilities leading to cooperation between different IT/IoT/Objects platforms. This contribution deals with scalability, interoperability, and standardization features of data platforms from a business point of view in a smart and cognitive port case study. The main goal is to design an innovative platform, named DataPorts, which will overcome these obstacles and provide an ecosystem where port authorities, external data platforms, transportation, and logistics companies can cooperate and create the basis to offer cognitive services. The chapter relates to knowledge and learning as well as to systems, methodologies, hardware, and tools cross-sectorial technology enablers of the AI, Data and Robotics Strategic Research, Innovation & Deployment Agenda (Milano et al., Strategic research, innovation and deployment agenda - AI, data and robotics partnership. Third release. Big Data Value Association, 2020).

**Keywords** Industry 4.0 · Data for AI · Port authorities

## 1 Introduction

Serverless architecture is based on the ground that the deployment is a transparent process where the developer is not aware of the cluster/server that the stateless functions are deployed. Although the inability to specify where the functions should run seems to weaken the overall architecture caused by a lack of control, that is compensated by the benefit of performance of the application.

The digital transformation of the ports toward the Fourth Industrial Revolution is revealing opportunities related to the add-on services that can be provided. Port platforms are now integrating available data sources capturing the potential needs that arise from the increased demand for more accurate and complete information [20]. The value of that new trend is also boosted by the new wave of startups that implement over-the-top services. However, most port authorities are not technologically ready to host these services and frameworks. The necessary technology infrastructure has anti-diametric different requirements from the one that is deployed. The same drawback applies to the architecture where in most cases must be shifted to serverless-oriented solutions.

For example, port authorities have the chance to enrich their services with smart containers infrastructure. This add-on IoT infrastructure enables the online information exchange capability about the entire journey and the conditions of the cargo directly to the rest of the supply chain without human intervention. This

provides greater visibility to the stakeholders within the transaction as well as to regulatory agencies who need detailed information on the consignments before they arrive at the border. This technology can be combined with other innovations such as blockchain, big data, or data pipelines to provide even more facilitation to the trading community. In all of these cases, though, we see that creating clear, unambiguous message exchange standards will allow to capitalize the full potential of the enhanced data. This shared data will enable the creation of new value-added services that the port authorities can benefit from.

Unfortunately, port authorities are unprepared nowadays to host that massive information exchange system. One reason is investments in IT that should be made, as the port systems will be upgraded to the main exchange point where companies and start-ups will connect to buy/sell data, information, and services. Another reason is the lack of expertise in new design implementations from the port authority staff since the orientation of the authority is anti-diametric different. Although a few of these needed features are addressed by serverless platforms like Openwhisk [3], OpenFaaS [33], OpenLambda [23], and Kubeless [26], some open challenges remain. Scalability, interoperability, architectural design, and standardization are the key pillars for a successful ecosystem. Definitions from a port authority view is presented in Sect. 2, where a complete analysis of these concepts is given in Sects. 3, 4, 5, and 6. Finally, we conclude with the business challenges that future systems will face after the transition to the new era.

## 2 Challenges for Port-Oriented Cognitive Services

A cognitive service is a software component that uses AI and big data capabilities to solve a business task. Cognitive services are ready-to-use solutions to be integrated into the context of software products for improving the decision-making process related to data. Some cloud providers offer general cognitive services such as image classification or natural text recognition/translation. However, this novel paradigm has not been applied in the ports' domain. Ports share common business tasks in which cognitive services could provide answers. Some examples are the prediction of the Expected Time of Arrival (ETA) of a container or a vessel, the truck turnaround time to deliver a container to a terminal, or the definition of a booking system to reduce environmental impact. The main goal is to build such services to be generic enough to be applied in different ports and use cases. To enable the cognitive services approach envisioned in the context of the project, the DataPorts [13] platform must address the following technical challenges:

1. One issue to address is to enable the data sharing between an undefined number of port stakeholders, such as terminal operators, port authorities, logistic carriers, and so on. The accuracy of cognitive services is directly related to the amount of available data. For building such a data ecosystem, the defined architecture must include scalability as the first design principle. To address this challenge,

we foresee the introduction of the International Data Spaces Association (IDSA) Reference Architecture Model [4], a standard solution with the required building blocks for achieving a seamless integration between organizations.

2. **Scalability for AI training:** Training models for cognitive services is a time-consuming task even with a powerful computing infrastructure for supporting it. As the state of the art evolves, new frameworks and techniques must be tested in order to find the optimal one, which provides better accuracy for the problem at hand. A cognitive service vision implies that not specific know-how from the data science domain is required: the end-user is who defines the training process with little manual intervention. This fact leads to the definition of several training alternatives, which must run simultaneously over a distributed infrastructure. This challenge will be addressed by the DataPorts platform, introducing the most suitable technological approaches from the ML DevOps area.
3. **Heterogeneous data processing:** Ports field is a domain in which several IT infrastructures, information systems like TOS or IoT sensing devices, are potential candidates to become a valuable data source. In this scenario, two main challenges arise: how to deal with the heterogeneity of the data sources (formats, schema, etc.) and with an undefined volume of data. The DataPorts platform will support the heterogeneity of schemas, applying techniques from the semantic interoperability domain and taking into account vocabularies or taxonomies from standardization bodies. Following the good practices for big data processing, such as the use of containerized application and distributed databases, we expect to provide the required tools for enabling a scalable data processing.
4. **A trusted data governance:** Ownership of the data is a key issue in any discussion related to data sharing between different organizations. To enhance the data sharing needed to build cognitive services, the DataPorts platform must first of all provide a trusted framework for defining data sharing rules to specific users, roles, and organizations. This framework must also enforce that the data is used following the specifications the data owner has formally defined. Data management, when this data is outside the boundaries of the organization, is a challenge that requires a set of trusted software components and clear security procedures. We foresee the use of smart contracts, in the context of a blockchain network among organizations, as the technological foundation to address this challenge.

The next sections introduce in more detail how we expect to address these overall challenges in terms of the overall architecture management, scalability, interoperability, and standardization. These challenges are in direct relation with the DaaS and FaaS overall strategic plan as well as with the horizontal concern of data management, data analytics, and data visualization of the BDV Technical Reference Model [43].

### 3 Scalability

In order to define a programming model and architecture where small code snippets are executed in the cloud without any control over the resources on which the code runs, the industry came with the term “Serverless Computing.” [5] It is by no means an indication that no servers exist, simply that the developer should leave most operational issues to the cloud provider, such as resource provisioning, monitoring, maintenance, scalability, and fault tolerance. The platform must guarantee the scalability and elasticity of the functions of the users. In response to load, and in anticipation of potential load, this means proactively provisioning resources. This is a more daunting serverless issue because these forecasts and provisioning decisions must be made with little to no application-level information. For instance, as an indicator of the load, the system can use request queue lengths but is blind to the nature of these requests.

In a few words, serverless computing allows application developers to decompose large applications into small functions, allowing application components to scale individually [29]. The majority of Function-as-a-Service systems use Kubernetes’ built-in Horizontal Pod Autoscaling (HPA) for auto-scaling, which implements compute-resource-dependent auto-scaling of function instances [6]. However, custom scaling mechanisms, such as auto-scaling based on the number of concurrent in-flight requests, can also be implemented. Generally speaking, there exist many implementations regarding scaling and auto-scaling on serverless functions.

Scalability is the ability of a system to handle a growing amount of work by adding resources to itself. This means that scalability stands as a direct solution to any workload issue that might emerge in a system. Therefore, today’s frameworks should take a full advantage of scalability’s benefits by implementing tools that achieve exactly that. How can a system be scalable? Solutions vary. However, the selection field narrows down a lot when it comes to serverless computing systems. That’s because serverless applications are created to have the scalability issue solved in advance. What remains to be answered is how we can further improve scalability in the serverless world.

Nevertheless, a solution to the scalability issue, distinct from the majority of the available ones, is through the prism of a Data-as-a-Service Marketplace [36]. When combined with a fully working Function-as-a-Service (FaaS) platform, this approach can lead to optimum scaling results. The core idea is about creating a serverless platform as part of a Data-as-a-Service (DaaS) marketplace repository framework, which enables dynamic scaling in order to ensure business continuity, such as real-time accommodation of rapidly evolving user numbers, and fault tolerance. That framework contains a multitude of readily accessible APIs to serve the needs of a growing and changing DaaS platform marketplace, while it provides great flexibility in selecting topologies and architectures for the storage pool. In essence, any node, either located in the cloud, at the physical location of the marketplace, or even at the edge of the network, may be used to store data as part of

the repository cluster. That DaaS strategy uses the cloud to deliver data storage, analytics services, and processing orchestration tools in order to offer data in a manner that new-age applications can use.

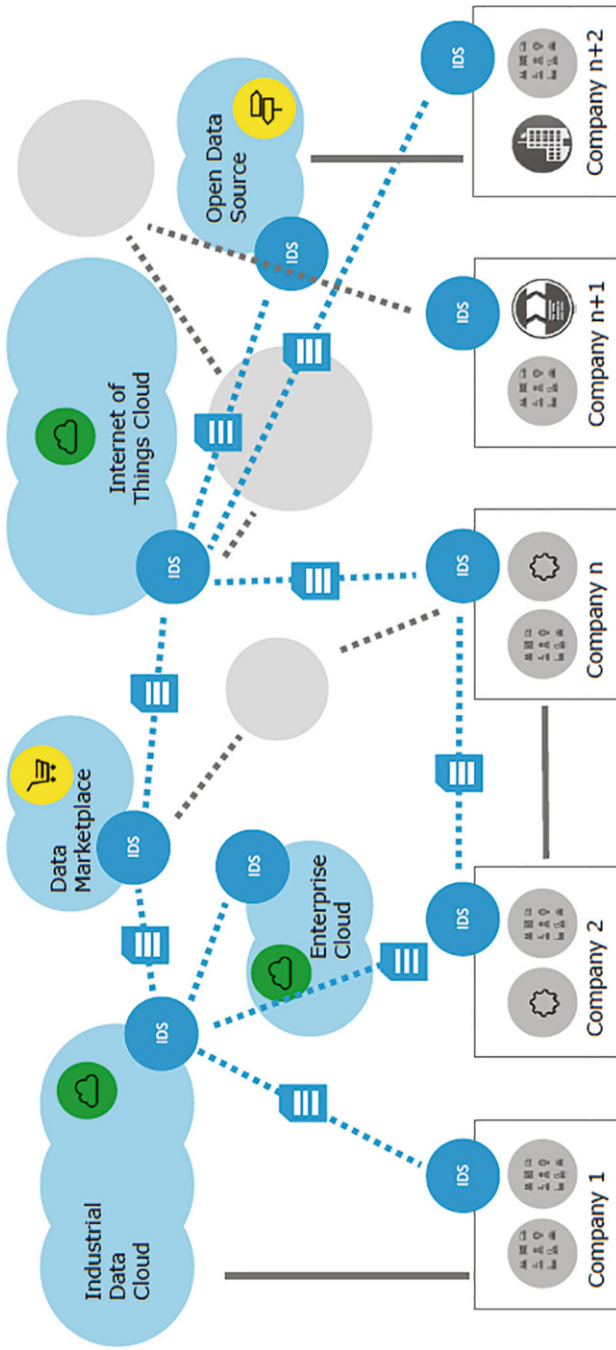
## 4 International Data Spaces Architecture Reference Model

However, the adopted proposal in the context of the DataPorts project [13] is the International Data Spaces Architecture (IDSA) Reference Model [4]. According to the official documentation, the International Data Spaces (IDS) is a virtual data space leveraging existing standards and technologies, as well as governance models well accepted in the data economy, in order to facilitate secure and standardized data exchange/linkage in a trusted business ecosystem (illustration of the architecture is shown later on in Fig. 1). Therefore, it provides a basis for creating smart-service scenarios and facilitating innovative cross-company business processes. At the same time, it guarantees data sovereignty for data owners. Regarding the IDSA's Reference Model, it is highly scalable. In short, the IDSA reference architecture model's high scalability is attributable to the fact that this model is a decentralized architecture ("peer-to-peer" data exchange with redundant replicated connectors and brokers) without a central bottleneck.

The two main and most important components of the IDSA Reference Model are the "Broker" and the "Connector." These components are responsible for the model's decentralized architecture and its ability to be highly scalable, as mentioned before. Additional to these two are the "Data Apps," which are data services encapsulating the functionality of data processing and/or data transformation, packaged as container images for easy installation through the application container management. The Data Apps are distributed through a secure platform, the "IDS App Store." The IDS App Store contains a list of available Data Apps. An App Store therefore facilitates the registration, release, maintenance, and query of the Data App operations, as well as the provisioning of the Data App to a Connector. The presence of an App Store, with different types and categories of applications, means that the IDSA is highly scalable, since connectors, as the main component of the IDSA Reference Model, can be modified or expanded to increase the ability and functionality of the connectors based on different requirements and domains.

To sum up, the key point (and what is worth noting) is that this architecture model is capable of achieving high scalability, in combination with an existing FaaS platform. Regarding the connection of IDSA with the DataPorts project, an architecture design for cognitive ports has to enable data sharing and data governance in a trusted ecosystem including various ports stakeholders. In the context of cognitive ports, to realize trusted data sharing and data governance, one can benefit from two main approaches:

1. **Data is stored off-chain:** Generally in this approach, we may leverage the concept of the International Data Spaces (IDS) reference architecture model



**Fig. 1** International Data Spaces provides an ecosystem where various data sources are connected [4]

(RAM) [4]. In a nutshell, a peer-to-peer data exchange between the data owners and data consumers through the IDS connectors is considered. To ensure the security and privacy aspects, blockchain manages consent of access to data. Smart contracts decide if a particular access to data is allowed concerning the invoker's credentials and the specification of access rules for the particular data [1].

2. **Data is stored on chain (blockchain platform for shared data):** Blockchain platform records transactions related to shared data and processes of all participants in a business network. For instance, coldchain temperature-alarms of a container for verification of its state such as if conditions have been compromised, allows everybody in the network to be aware of the event and act upon it. This approach in fact ensures verifiable and immutable information on shared data through the entire chain to all business network participants serving as a single source of truth and providing transparency and a not-repudiation process.

As depicted in Fig. 1 and mentioned above, the International Data Spaces (IDS) is a virtual environment that leverages existing standards and technologies, as well as governance models well-accepted in the data economy, to facilitate secure and standardized data exchange and data linkage in a trusted ecosystem [4].

As stated before and already analyzed, two main components of the IDS RAM are a broker and a connector. A brief reminder, the broker is an intermediary that stores and manages information about the data sources available in the IDS. It mainly receives and provides metadata. As mentioned before, data sharing and data exchange are the main fundamental aspect of the IDS. An IDS connector is the main technical component for this purpose.

For the IDS connector, as the central component of the architecture, different variants of implementation are available, may be deployed in various scenarios, and can be acquired from different vendors. However, each connector is able to communicate with any other connector (or other technical components) in the ecosystem of the International Data Spaces.

- **Operation:** Stakeholders should be able to deploy connectors in their own IT environment. Additionally, they may run a connector software on mobile or embedded devices. The operator of the connector must always be able to describe the data workflow inside the connector. Moreover, users of a connector must be identifiable and manageable. Every action, data access, data transmission, and event has to be logged. This logging data allows to draw up statistical evaluations on data usage.
- **Data exchange:** A connector must receive data from an enterprise backend system, either through a push mechanism or a pull mechanism. The data can be either provided via an interface or pushed directly to other participants. Hence each connector has to be uniquely identifiable. Other connectors can subscribe to data sources or pull data from these sources. Data can be written into the backend system of other participants.



In addition to what it is described, the IDS RAM benefits from an information model, which is an essential agreement shared by the participants and components of the IDS, facilitating interoperability and compatibility. The main aim of this formal model is to enable (semi-)automated exchange of digital resources in a trusted ecosystem of distributed various parties, while the sovereignty of data owners is preserved.

Data sovereignty is defined as data subject's capability of being in full control of the provided data. To this end, all the organizations attempting to access the IDS ecosystem have to be certified, and so are the core software components (for instance, IDS connector) used for trusted data exchange and data sharing. Such a certification not only ensures security and trust, but the existence of certified components guarantees compliance with technical requirements ensuring interoperability.

## **5 Interoperability**

### ***5.1 Introduction***

Interoperability among disparate computer systems is the ability to consume services and data with one another. Each software solution provides its own infrastructure, devices, APIs, and data formats, leading to compatibility issues and therefore to the need for specifications in terms of interaction with other software systems. Interoperability, as a complex concept, entails multiple aspects that address the effective communication and coordination between components and systems that might consist of a uniform platform at a larger scale. This section focuses on interoperability from the point of view of semantic interoperability and Application Programming Interfaces. On the one hand, semantic interoperability constructs a consolidated ontology model that ensures the unambiguity of data exchanges, since it is guaranteed that the requester and provider have a common understanding of the meaning of services and data. On the other hand, APIs constitute an interoperability tool that documents all the available services that are exposed by a software system, as well as the information about the respective communication protocols. Therefore, APIs are considered a significant step toward the interoperability of a system through the standardization of the components' communication. The present section describes the evolution of the service architectural models based on the ever-changing needs of communication, as well as the most powerful state-of-the-art tool for API standardization.

## 5.2 *Semantic Interoperability*

Semantic interoperability provides the ability to computer systems to exchange data with unambiguous, shared meaning [22]. In any system focused on interoperability, it is essential to take into account the production, collection, transmission and processing of large amounts of data. An application consuming those data needs to understand its structure and meaning. The metadata is responsible for representing these aspects in a readable way by a machine. The more expressive is the language used for representing the metadata, the more accurate the description might become. The metadata provides a semantic description of the data and can be utilized for many purposes, such as resource discovery, management, and access control [21].

The concept ontology refers to a structure that provides a vocabulary for a domain of interest, together with the meaning of entities present in that vocabulary. Typically, within an ontology, the entities may be grouped, put into a hierarchy, related to each other, and subdivided according to different notions of similarity. In the last two decades, the development of the Semantic Web resulted in the creation of many ontology-related languages, standards, and tools. Ontologies give the possibility to share a common understanding of the domain, to make its assumptions explicit, and to analyze and reuse the domain knowledge. In order to achieve shared meaning of data, the platforms or systems have to use a common ontology either explicitly, or implicitly, for example, via a semantic mediator [18].

Typically, organizations in transportation and logistics, with a particular focus on port logistics, have their own local standards. Sometimes they have a poor formalization of semantics, or they don't have explicit semantics at all [19]. The development of ontologies for logistics is not a trivial task. Define and use guidelines and best practices are necessary for this domain, especially to bridge the gap between theory and practice. A proper theoretical and methodological support is required for ontology engineering to deliver precise and consistent solutions to the market, as well as to provide solutions to practical issues to be close to the real market needs [12].

DataPorts project is developing a semantic framework for describing ports data together with mappings to standard vocabularies in order to simplify the reuse of data applications for analytics and forecasting. In particular, this semantic framework will codify the domain knowledge of the domain experts, and thus can be reused and exploited by the data experts directly, thereby empowering building cognitive port applications.

The relevance of the use of data platforms and the exploitation of data sharing are boosted by the high volume of different companies and public bodies that need to collaborate among them with different degrees of digital capacities. In this aspect, a semantic interoperability framework with a currently non-existing global ontology will improve such collaboration and data representation. Regarding the target group of users from the logistics domain, DataPorts semantics components make the interpretation of data and metadata more manageable for data users, so that their discovery is straightforward according to common search criteria. On

the other hand, special provisions are taken in the platform to create an active and easy to enter data marketplace for third-party developers and service providers, with minimal integration efforts, clear monetization, and business value creation.

From a technical point of view, the aim of the project is to identify the different data sources to be integrated into the DataPorts platform, including the mechanisms to store and facilitate data management. Ontologies, mechanisms and enablers must also be defined to provide semantic interoperability with the data of these digital port infrastructures. This includes IoT devices, mobile applications, and, legacy databases and systems. Finally, to develop the semantic-based tools and components needed to facilitate the generation of interfaces to interact and manage the information of these data sources through the DataPorts platform. The data platform will guarantee semantic interoperability in order to provide a unified virtualized view of the data for its use by the different data consumers and the data analytic services. Figure 2 shows the architecture designed to achieve this purpose.

The Data Access Component solves the problem of access to different data sources or origins in a secure way. Given the very different types of sources, the platform will have to cope with the variety of data sources. It is necessary to analyze the provided interfaces of each data source. These interfaces should facilitate the way to access, to analyze the format and to understand the way to receive the data.

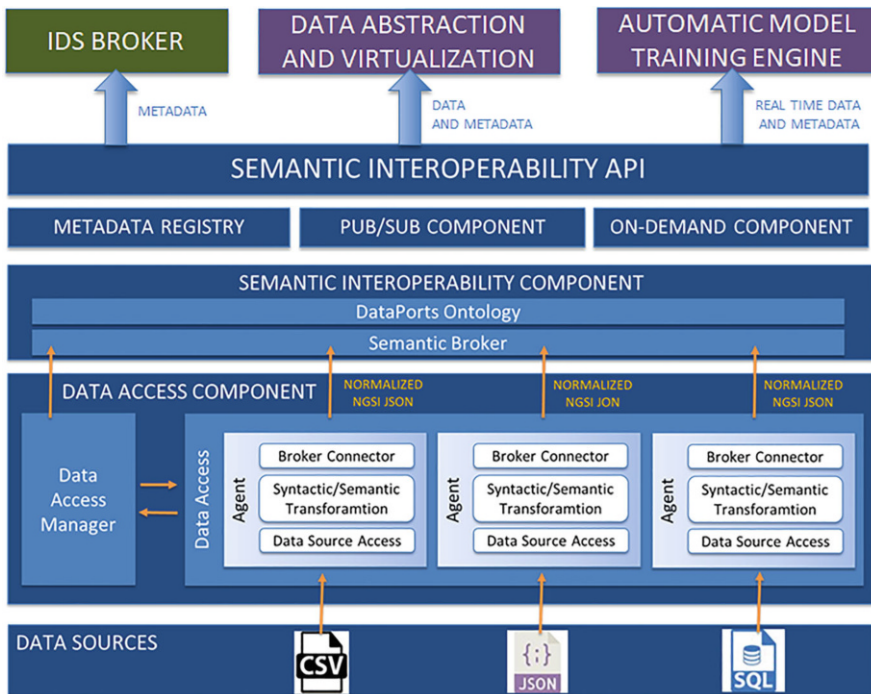


Fig. 2 Data Access and semantic interoperability Layer components of DataPorts Platform

They should recognize the volume, the velocity and the veracity of the information. In addition, to consider the existence of an ontology in the data source. To deal with these heterogeneous data sources, a data access agent is necessary for each data source integrated into DataPorts platform. An agent or adapter is the piece of software in charge of acquiring data from a source under certain conditions. Then the agents transform this data acquired from the sources to data in the DataPorts common data model. Finally, this data is sent to the other platform components. In addition, the Data Access Manager manages the metadata description of the data processed by the agents and the interaction of the agents with the other platform components. To recapitulate, the Data Access Component is the responsible for getting the sharing data and metadata description from the different data sources. It performs the processes to make the data sources understandable and available to the other platform components.

The semantic interoperability Component provides a unified API to access the data from the different data sources connected to the DataPorts platform, providing both real-time and batch data to the data consumers. In collaboration with the Data Access component, it will provide a data shared semantics solution for the interoperability of diverse data sources using existing ontologies. The output data will follow the common DataPorts ontology.

Regarding the metadata, the semantic interoperability Component obtains information about the different data sources from the Data Access Manager and stores it in the metadata registry to make it available for the other subcomponents of this layer. In addition, the metadata is sent to the IDS broker to provide information to the data consumers about the available data sources. Other components of DataPorts platform like Data Abstraction and Virtualization and Automatic Prediction Engine can retrieve this metadata description of the data sources by asking the semantic interoperability API. The output metadata will follow the common DataPorts ontology.

The semantic interoperability Component provides a repository with the DataPorts Data Model and DataPorts Ontology. It offers an ontology definition using OWL [34] and a Data Model description using JSON schema and JSON-LD context documents. The aim is to integrate the following ontologies and data models: Fiware Data Models [14], IDSA Information Model [24], United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) [40] Model, and Smart Applications REFERENCE (SAREF) ontology [38].

Finally, the semantic interoperability Component API will interact with the Security and Privacy component to enable authentication and confidentiality, as well as to enforce the data access policies, in order to ensure proper data protection in the exchanges with the data consumers.

The open-source Fiware platform [16] has been selected to act as a core element of the Data Access and semantic interoperability components of the architecture presented, and it will be adapted, customized, and extended to fit the DataPorts project expectations. Moreover, the Fiware Foundation is involved in the development of IDS implementations and is actively cooperating with the Industrial Data Space Association (IDSA). Together, Fiware and IDSA are working on the first open-source implementation of the IDS Reference Architecture [2].

The aim behind Fiware is that technology should be accessible for everyone, with a focus on interoperability, modularity, and customizability. Data should seamlessly merge with data from other relevant sources. For that reason, Fiware components provide standardized data formats and API to simplify this integration. The platform is open source and can be easily embedded by ecosystem partners in the design of their solutions and reduce vendor lock-in risks. The standardized API means that services can operate on different vendor platforms. Fiware NGSI [17] is the API exposed by the Orion Context Broker and is used for the integration of platform components within a “Powered by Fiware” platform and by applications to update or consume context information. The Fiware NGSI (Next Generation Service Interface) API defines a data model for context information, an interface for exchanging context information, and a context availability interface for queries on how to obtain context information. The agents are the connectors that guarantee the transmission of raw data to Orion Context Broker using their own native protocols.

Some implementation decisions have been made in order to collaborate with the Fiware ecosystem. Firstly, all data and metadata formats are going to be designed to follow Fiware NGSI data models specifications [15]. Secondly, the use of the Orion Context Broker is being adopted as the semantic broker component [17]. Finally, regarding the data access agents, they are not mandatory to be implemented with a closed specific technology, but in order to provide a standardized SDK (Software Development Kit) to develop agents, the aim is to use the pyngsi Python framework [37].

The work done in previous European projects, where Fiware technology is a key element, is taken as a reference in the DataPorts implementation. For example, it is interesting to highlight projects like SynchroniCity [39] and Boost 4.0 [11]. SynchroniCity is aimed to establish a reference architecture for the envisioned IoT-enabled city market place with identified interoperability points and interfaces and data models for different verticals. The baseline of the SynchroniCity [39] data models is the FIWARE Data Models initially created by the FIWARE Community and have been expressed using the ETSI standard NGSI-LD. Regarding Boost 4.0, the aim of the project is to implement the European Data Space with FIWARE technologies.

### ***5.3 Application Programming Interfaces for Serverless Platforms***

In order for the envisioned software components of the DataPorts [13] architecture to be manifested, their interconnection through a standardized API is crucial in order to avoid a monolithic service approach that does not face the challenges presented above and causes a major drawback in the process of building a federated ecosystem. The concept of a service within a computational infrastructure has been fundamental through the evolution of different architectural designs and

implementations. Application Programming Interfaces (APIs), however, constitute a powerful interoperability tool that enables communication within a heterogeneous infrastructure resulting in loosely coupled components. For that reason, their usage has almost dominated the landscape of services, especially within serverless infrastructures.

Considering the WWW as an ecosystem of heterogeneous services that require simplicity, Web APIs, or also known as RESTful services, have been increasingly dominating over the Web services that are based upon WSDL and SOAP. RESTful services conform to the REST architectural principles, which include constraints regarding the client-server communication, the statelessness of the request, and the use of a uniform interface. In addition, these services are characterized by resource-representation decoupling, such that the resource content can manifest in different formats (i.e., JSON, HTML, XML, etc.). Furthermore, the majority of Web APIs reliance on URIs for resource identification and interaction and HTTP protocol for message transmission result in a simple technology stack that provides access to third parties, in order for them to consume and reuse data that originate from diverse services in data-oriented service compositions named mashups [28].

The evolution of the monolithic service-oriented architectures (SOA) has commenced from the management of the complexity of distributed systems in scope of integrating different software applications and has evolved through microservices into serverless architectures. In service-oriented architectures, a service provides functionalities to other services mainly via message passing. With the modularization of these architectures into microservice ecosystems, different services are developed and scaled independently from each other according to their specific requirements and actual request stimuli, leading to the localization of decisions per service regarding programming languages, libraries, frameworks, etc. However, the rise of cloud computing has led to serverless architectures that support the dynamic resource allocation and the corresponding infrastructure management in order to enable auto-scaling based on event stimulus and to minimize operational costs [25]. In that context, Web APIs should be considered the cornerstone of the exploding evolution of service value creation, where enterprise systems are embracing the XaaS (Anything-as-a-Service) paradigm, according to which all business capabilities, products, and processes are considered an interoperable collection of services that can be accessed and leveraged across organizational boundaries [7]. Nevertheless, since the beginning of the APIs prevalence over the traditional Web service technologies, the APIs have evolved in an autonomous way, lacking an established interface definition language [28]. Hence, in terms of serverless infrastructures, the subsequent need for homogeneity in application design and development has risen.

A common denominator in the development of serverless functions is their ability to support different functionalities in a scalable and stateless manner. For instance, there might be a serverless application that is integrated with an already existing ecosystem of functions that support API calls to cloud-based storage. While the former is by definition scalable, the underlying storage system's on-demand scalability is bound to reliability and QoS guarantees. As far as these serverless

implementations are concerned, two major use cases are addressed hereunder. The first one involves the composition of a number of APIs, while filtering and transforming the consumed data. A serverless function that implements this functionality mitigates the danger of network overload between the client and the invoked systems, and offloads the filtering and aggregation logic to the backend. The second serverless application involves API aggregation, not only as a composition mechanism but as a means to reduce API calls in terms of authorization, for example. This composition mechanism simplifies the client-side code that interacts with the aggregated call by disguising multiple API calls into a single one with optional authorization from an external authorization service, e.g., an API gateway [5].

Despite the commonalities among serverless platforms in terms of pricing, deployment, and programming models, the most significant difference between them is the cloud ecosystem [5]. Differences in cloud platforms lead to discrepancies in developing tools and frameworks that are available to developers for creating services native to each platform. The ever-evolving serverless APIs in combination with the corresponding frameworks and libraries represent a significant obstacle for software lifecycle management, service discovery, and brokering. The plethora of incompatible APIs in terms of serverless technology has created the need for multicloud API standardization, interoperability, and portability in order to achieve seamlessness. In this direction, informal standardization has been formed after community efforts toward addressing the lack of a common programming model that enables platform-agnostic development and interoperability of functions [42].

Following the problem identification depicted above, the solution to the lack of a standardized and programming language-agnostic interface description language is fulfilled by the OpenAPI Specification (OAS). The OpenAPI initiative was founded in November 2015 by the collaboration of SmartBear, 3Scale, Apigee, Capital One, Google, IBM, Intuit, Microsoft, PayPal, and Restlet. This initiative was formed as an open-source project under the Linux Foundation and was designed to enable both humans and computers to explore and understand the functionalities of a RESTful service without requiring access to source code, additional documentation, or inspection of network traffic. OAS enables the understanding and interaction with the remote service with a minimal amount of implementation logic according to a vendor neutral description format. The OpenAPI Specification was based on the rebranded Swagger 2.0 specification, donated by SmartBear Software in 2015 [32].

The most important advantages of the OpenAPI Specification are twofold. On the one hand, the business benefit that comes a long is the recognition of this standardization as a useful means for a lot of developers to develop open-source repositories of tools that leverage this enablement. Furthermore, OAS is supported by a group of industry leaders that contribute with their strong awareness and mindshare, while indicating stability across a diverse code base. On the other hand, OAS is registered as a powerful technical tool that is most importantly language-agnostic and provides understanding of an API without the involvement of server implementation. Its documentation is regularly updated by a broad community

that provides additional example implementations, code snippets, and responses to inquiries [32].

According to the above, the significance of the standardization that OAS offers rationalizes its adoption as the proposed solution interface description language for the proposed port platform. The pluralism of different data sources that need to be integrated within this platform, in combination with the existence of different frameworks and technologies of the existing APIs that are already utilized by ports, introduces the need for a uniform description of the exposed interfaces. For instance, the IoT infrastructure that includes APIs that are exposed by smart containers enables the aggregation of information that implements the life cycle assessment (LCA) applied for port logistics operations and needs to be integrated seamlessly with the different components of the platform. Moreover, crucial role in the message exchange between the different infrastructures within the ports ecosystem, play the APIs that facilitate the communication between components. Therefore, the standardization of their interface is of utter importance for the scalability and the interoperability of the platform. The DataPorts architecture can be constructed based on the OpenAPI specification, enabling the development of add-on services and the creation of added value of the available data. Furthermore, the implementation of the platform within a serverless architecture framework underlines the significance of the OpenAPI specification as a powerful standard for the interface description of all services within and exposed by the DataPorts platform.

## 6 Standardization

IDSA [4] aims at open, federated data ecosystems and marketplaces ensuring data sovereignty for the creator of the data by establishing virtual space for the standardized, secure exchange and trade of data. Standards, be they national, regional, or global, are the fruit of collective efforts and guarantee interoperability. They can be revised to meet industry needs and remain relevant over time. Standards organizations, where participants from different segments of the industry gather, are among the few places where competitors work side-by-side. Standards organizations offer a safe place to do so from an antitrust perspective. Standards development participants are industry experts, tech companies, and customers representing all fields of the industry.

The adoption of global multimodal data exchange standards guarantees interoperability. In fact, smart container standardization effort [9, 10] is one of many standardization initiatives [27] supporting global trade. Standards enable stakeholders in the logistics chain to reap the maximum benefits from smart container solutions while enabling them to share data and associated costs. Standards-based data exchange usage increases the ability to collaborate, which in turn increases efficiency. Additionally, such standards reduce development and deployment costs and cut time to market for Internet of Things (IoT) solution providers.



Data exchange standards developed in an open process offer a useful aid to all parties interested in the technical applications and implementation of smart container solutions. Additionally, if solution providers find there are new data elements required to accommodate changing business requirements, it is possible to create a backward-compatible revision of the standard to accommodate their needs.

With the ramp-up of new and emerging technologies, these standards are more necessary than ever. Standards reduce the risk of developing proprietary technologies with significant deployment limitations and the lack of interoperability among systems and devices. Standards enable the parties to avoid costly and time-consuming integration and limit the risk of vendor lock-in. In this context, IDS provides a generic framework that can be leveraged by domain-specific instantiation such as UN/CEFACT smart container standard that offers transport execution and the condition under which the cargo was transported.

The United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) Smart Container Business Requirements Specification (BRS) ensures that the various ecosystem actors share a common understanding of smart container benefits by presenting various use cases. It also details the smart container data elements [41]. Defining the data elements that smart containers can generate accelerates integration and the use of smart container data on different platforms for the enhancement of operations. In addition, utilizing standard smart container data enables open communications channels between supply chain actors.

Standards data models and standard APIs would help stakeholders to make the necessary transformation to achieve supply chain excellence [8]. Indeed, APIs are key to ensuring simplification and acceleration of the integration of digital services from various sources.

The focus of the UN/CEFACT Smart Container project is to define the data elements via varied use cases applicable to smart container usage. Currently, the data model is being developed, which will provide the basis for the smart container standard messaging and Application Programming Interfaces (APIs). The Smart Container API catalog will be the source code-based interface specification enabling software components (services) to communicate with each other. It is crucial to first determine and align the required data elements and their semantics.

Smart containers will revolutionize the capture and timely reporting of data throughout the supply chains. Such containers are an essential building block to meet the emerging requirements for end-to-end supply chains. As leading carriers adopt smart container solutions, they gain valuable data that can be shared with shippers and other supply chain stakeholders.

However, generating and collecting data is not enough to make smart container solutions or supply chains “smart.” Stakeholders already manage huge amounts of data and struggle with multiple technologies that take time away from their core businesses. A smart container solution must deliver data that matters, in a standard format for easy integration into different systems. It must enable unambiguous data interpretation and empower all involved stakeholders with actionable information. Clear semantic standards are essential for effective smart container data exchange ensuring that all stakeholders understand the same information in the same way.

Then and only then, can smart containers truly become part of digital data streams [35].

The UN/CEFACT Smart Container project aims to create multimodal communications standards that can facilitate a state-of-the-art solution in providing and exposing services. Any intermodal ecosystem stakeholder may then orchestrate and enrich these services to meet their business process needs. The availability and exposition of these services can boost the digital transformation of the transportation and logistics industry, fuelling innovation in new applications and services. Physical supply chains that move goods need a parallel digital supply chain that moves data describing the goods and their progress through the supply chain. The smart container data flows ensure that the physical flow is well synchronized with the required documents flow. Data are the raw material of Maritime Informatics. Without data streams emanating from operations, there can be no data analytics. As we digitalize, we improve operational productivity and lay the foundation, through Maritime Informatics, for another round of strategic and operational productivity based on big data analytics and machine learning.

## 7 Business Outcomes and Challenges

The fast-growing complexity at seaports makes data management essential, hence the optimal goal is to achieve greater efficiency. The use of large volumes of data (big data) is indisputably a major aid to this goal [31]. AI-based services available in a smart seaport is a new revenue source for many stakeholders. When such data and services are offered through a standard mechanism as is a data-driven platform, this offering acts as a leverage to improve and increase various port operations, especially those that are associated with traffic, vessel, and cargo movement, and is of high importance for third parties. Imagine the case where cargo transfer data are accessible by the shipping lines, and at the same time, all seaport's operations can be available by Port Authority's associates. Passenger mobility patterns may be available not only to the Port Authority but also to the city's decision and policy makers. Commercial or cultural associations may also be interested to access such services, especially from seaports with high passenger activity. Such services, will be used to transform the seaports into smart and cognitive, and eventually will increase the ports' stakeholders and activity boundaries. Hence, through data-driven services, the demand will also be increased. Toward this direction, the research community and the shipping-related SMEs or even the startup community may benefit from analyzing large volumes of data offered by data providers and propose additional offerings. Moreover, Analytics as a Service using data collected from shipping and freight companies, warehouses, customs brokers, and other port operations may be a key for data monetization. The opportunity of data monetization may unlock any considerations regarding data sharing, that are related to the risk of losing competitive advantages.

From a business perspective, in order for data and service sharing to be effective and useful for as many beneficiaries as possible, certain Quality of Service (QoS) characteristics should be followed. Especially in big data, among the key characteristics are considered the Volume, Velocity, Variety, Veracity, and Value. Moreover, offering data and services should match certain needs and be easily accessible for the users. Therefore, in terms of time, data should be up to date, real, and able to be authenticated. Additionally, guarantying a QoS may be difficult for heterogeneous data, especially when the competitiveness is increasing according to the demand for new data and services. Hence, a monitoring mechanism is needed to ensure the above-mentioned characteristics as well as the validity of the transferred data. A main business-related concern with data QoS is considered the regulatory compliance that today is vague, the customer satisfaction which is the goal, the validity and the accuracy of the data to allow decision making, the relevance the data should meet and their completeness for not have missing values and the consistency of data format as expected by the users.

From a technical perspective, all types of applications of the fourth paradigm of science deal with large amounts of data stored in various storage devices or systems. Distributed storage systems are often chosen for storing data of this type, as depicted in Fig. 1, framing requirements for IDS. Some of the requirements posed to those storage systems may concern Quality of Service (QoS) aspects formally expressed in a Service Level Agreement as was the traditional approach in the past. The role of QoS is to provide the necessary technical specifications that specify the system quality of features such as performance, availability, scalability, and serviceability. Within the IDS ecosystem, system qualities are closely interrelated. Requirements for one system quality might affect the requirements and design for other system qualities. For example, within connected IDS managed and framed by various companies may have different and higher levels of security policies that might affect performance, which in turn might affect availability. Adding additional servers to address availability issues affect serviceability (maintenance costs). Understanding how system qualities are interrelated and the trade-offs that must be made is the key to designing a system that successfully satisfies both business requirements and business constraints. Having these QoS attributes in mind, it's evident that the QoS management in distributed and heterogeneous environment is a challenging task given the possible storage device heterogeneity, the dynamically changing data access patterns, the client's concurrency, and storage resource sharing. The problem becomes even more complicated when distributed computing environments with virtualized and shared resources like Clouds and Blockchains are considered. Furthermore, various heterogeneous devices or objects should be integrated for transparent and seamless communication under the umbrella of Internet of Things (IoT). This would facilitate the open-access of data for the growth of various digital services. Building a general framework or selecting an approach for handling QoS becomes a complex task due to the heterogeneity in devices, technologies, platforms, and services operating in the same system. Additionally, Data's Analytics and Governance should follow an all-encompassing approach to consumer privacy and data security as opposed by Compliance Regulations that will become a benchmark

for how personal data are treated in the future. In the area of cognitive ports, regulations introduce major restrictions and complexities for QoS in a technical perspective, especially those parts that address how contact data are handled—and how data quality approach can be used that involves both tools and processes as part of compliance efforts. In such heterogeneous environment, technical aspects of QoS relate to technical translation and treatment of compliance as concerning entities of rights to: “access,” “be informed,” “data portability,” “be forgotten,” “object,” “restrict processing,” “be notified,” “rectification,” and so on in addition to the aforementioned technical specifications.

The majority of technical challenges discussed above, overcame by adopting the serverless architecture approach, as described in section “Scalability”. The use of APIs and the semantic interoperability in “5.2” provide vignettes for the followed approach. Therefore, utilizing serverless as well as microservices paradigms, cognitive ports constitute a PaaS and DaaS environment, in which the majority of traditional QoS aspects are dealt dynamically by inheriting system’s adaptation to current needs.

Additionally, the aspects of Compliance Regulations are approached by the creation of workflows into the Blockchain and Broker infrastructure of the Cognitive Port. For example, upon the request for every provider to comply with existing regulations concerning data and identity attributes, any stakeholder that provides data is responsible for the integrity and compliance of their provided data. Furthermore, risks that might arise from analytical aspects of shared data (such as combining data with new or existing data sources within or external to Cognitive Ports environment, etc) are secured by workflows for approving data processing requests. Therefore, there are controls for either prohibit non-regulated actions, or inform consumers that upon using them, any actions needed (i.e. consents) are their responsibility. Therefore, Cognitive Ports are an ecosystem for dynamically sharing data in IDS communities.

**Acknowledgments** The research leading to these results has received funding from the European Commission under the H2020 Programme’s project DataPorts (grant agreement No. 871493).

## References

1. Ahmadian, A. S., Jürjens, J., & Strüber, D. (2018). Extending model-based privacy analysis for the industrial data space by exploiting privacy level agreements. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing - SAC '18*. ACM Press.
2. Alonso, Á., Pozo, A., Cantera, J., de la Vega, F., & Hierro, J. (2018). Industrial data space architecture implementation using FIWARE. *Sensors*, 18(7), 2226.
3. Apache OpenWhisk. <https://openwhisk.apache.org/>. Online. Accessed 05 January 2021.
4. Auer, S., Jürjens, J., Otto, B., Brost, G., Lange, C., Quix, C., Cirullies, J., Lohmann, S., Eitel, A., Mader, C., Schulz, D., Ernst, T., Menz, N., Schütte, J., Haas, C., Nagel, L., Spiekermann, M., Huber, M., Pettenpohl, H., Wenzel, S., Jung, C., & Pullmann, J. (2019, April). Reference architecture model for the industrial data space. <https://www.internationaldataspaces.org/wp-content/uploads/2019/03/IDS-Reference-Architecture-Model-3.0.pdf>. Online. Accessed 30 October 2020.

5. Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., & Rabbah, R. (2017). Aleksander Slominski, and Philippe Suter. Serverless computing: Current trends and open problems. In *Research Advances in Cloud Computing* (pp. 1–20). Springer.
6. Balla, D., Maliosz, M., & Simon, C. (2020). Open source FaaS performance aspects. In *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE.
7. Basole, R. C. (2018). On the evolution of service ecosystems: A study of the emerging API economy. In *Handbook of Service Science, Volume II* (pp. 479–495). Springer International Publishing.
8. Becha, H. (2019). How standard APIs open the door to powerful digital services. <https://hananebecha.home.blog/2019/11/28/the-un-cefact-smart-container-project/>, November 2019. Online. Accessed 30 October 2020.
9. Becha, H. (2020). Standardization supporting global trade. *Port Technology International*, 2019. Edition 91, Shipping 2020: A Vision for Tomorrow. <https://www.porttechnology.org/editions/shipping-2020-a-vision-for-tomorrow/>
10. Becha, H. (2020). The UN/CEFACT smart container project. *The Report: The Magazine of the International Institute of Marine Surveying*, March 2020. Issue 91. <https://www.iims.org.uk/wp-content/uploads/2020/02/The-Report-March-2020.pdf>
11. Boost 4.0 – Big Data for Factories. <https://boost40.eu/>. Online. Accessed 05 January 2021.
12. Daniele, L., & Pires, L. F. (2013). An ontological approach to logistics. *Enterprise Interoperability, Research and Applications in the Service-Oriented Ecosystem, IWEI, 13*, 199–213.
13. DataPorts H2020 EU Project. <http://dataports-project.eu>. Online. Accessed 30 October 2020.
14. FireWire Data Model. <https://www.fiware.org/developers/data-models/>. Online. Accessed 05 January 2021.
15. Fiware Data Models. <https://fiware-datamodels.readthedocs.io/en/latest/index.html>. Online. Accessed 30 October 2020.
16. Fiware Open Source Platform. <https://www.fiware.org/>. Online. Accessed 30 October 2020.
17. Fiware Orion Context Broker. <https://fiware-orion.readthedocs.io/en/master/>. Online. Accessed 30 October 2020.
18. Fortino, G., Savaglio, C., Palau, C. E., de Puga, J. S., Ganzha, M., Paprzycki, M., Montesinos, M., Liotta, A., & Llop, M. (2017). Towards multi-layer interoperability of heterogeneous IoT platforms: The INTER-IoT approach. In *Internet of Things* (pp. 199–232). Springer International Publishing.
19. Ganzha, M., Paprzycki, M., Pawlowski, W., Szmeja, P., & Wasielewska, K. (2016, April). Semantic technologies for the IoT – an inter-IoT perspective. In *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE.
20. Gizelis, C.-A., Mavroeidakos, T., Marinakis, A., Litke, A., & Moulos, V. (2020). Towards a smart port: The role of the telecom industry. In *Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops* (pp. 128–139). Springer International Publishing.
21. Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43(5–6), 907–928.
22. Heiler, S. (1995). Semantic interoperability. *ACM Computing Surveys*, 27(2), 271–273.
23. Hendrickson, S., Sturdevant, S., Harter, T., Venkataramani, V., Arpaci-Dusseau, A. C., & Arpaci-Dusseau, R. H. (2016, June) Serverless computation with openlambda. In *8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16)*, Denver, CO. USENIX Association.
24. IDSA Information Model. <https://github.com/International-Data-Spaces-Association/InformationModel>. Online. Accessed 05 January 2021.
25. Kratzke, N. (2018). A brief history of cloud application architectures. *Applied Sciences*, 8(8), 1368.
26. Kubeless. <https://kubeless.io/>. Online. Accessed 05 January 2021.

27. Lind, M., Simha, A., & Becha, H. (2020). Creating value for the transport buyer with digital data streams. *The Maritime Executive*. <https://maritime-executive.com/editorials/creating-value-for-the-transport-buyer-with-digital-data-streams>
28. Maleshkova, M., Pedrinaci, C., & Domingue, J. (2010, December). Investigating web APIs on the world wide web. In *2010 Eighth IEEE European Conference on Web Services*. IEEE.
29. McGrath, G., & Brenner, P. R. (2017, June). Serverless computing: Design, implementation, and performance. In *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE.
30. Milano, M., Curry, E., García Robles, A., Hahn, T., Irgens, M., Lafrenz, R., Liepert, B., O'Sullivan, B., Zillner, S., Bisset, D., & Smeulders, A. (2020, September). Strategic research, innovation and deployment agenda - AI, data and robotics partnership. Third release. *Big Data Value Association*.
31. Moulos, V., Chatzikyriakos, G., Kassouras, V., Doulamis, A., Doulamis, N., Leventakis, G., Florakis, T., Varvarigou, T., Mitsokapas, E., Kioumourtzis, G., Klirodetis, P., Psychas, A., Marinakis, A., Sfetsos, T., Koniaris, A., Liapis, D., & Gatzoura, A. (2018). A robust information life cycle management framework for securing and governing critical infrastructure systems. *Inventions*, 3(4), 71.
32. OpenAPI Specification. <https://www.openapis.org/faq>. Online. Accessed 30 October 2020.
33. OpenFaaS. <https://docs.openfaas.com/>. Online. Accessed 05 January 2021.
34. OWL. <https://www.w3.org/TR/owl-guide/>. Online. Accessed 05 January 2021.
35. Pigni, F., Piccoli, G., & Watson, R. (2016). Digital data streams: Creating value from the real-time flow of big data. *California Management Review*, 58(3), 5–25.
36. Psomakelis, E., Nikolakopoulos, A., Marinakis, A., Psychas, A., Moulos, V., Varvarigou, T., & Christou, A. (2020). A scalable and semantic data as a service marketplace for enhancing cloud-based applications. *Future Internet*, 12(5), 77.
37. Pyngsi Python Framework. <https://github.com/pixel-ports/pyngsi>. Online. Accessed 30 October 2020.
38. SAREF Ontology. <https://saref.etsi.org/>. Online. Accessed 05 January 2021.
39. SynchroniCity. <https://synchronicity-iot.eu/>. Online. Accessed 05 January 2021.
40. UN/CEFACT. <https://umm-dev.org/about-umm/>. Online. Accessed 05 January 2021.
41. UN/CEFACT. (2019). The UN/CEFACT Smart Container Business Specifications (BRS). [https://www.unece.org/fileadmin/DAM/cefact/brs/BRS-SmartContainer\\_v1.0.pdf](https://www.unece.org/fileadmin/DAM/cefact/brs/BRS-SmartContainer_v1.0.pdf). Online. Accessed 30 October 2020.
42. van Eyk, E., Toader, L., Talluri, S., Versluis, L., Uta, A., & Iosup, A. (2018). Serverless is more: From PaaS to present cloud computing. *IEEE Internet Computing*, 22(5), 8–17 (2018).
43. Zillner S., Curry E., Metzger A., Auer S., & Seidl R. (2017). European big data value strategic research & innovation agenda. Big Data Value Association.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), 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 chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter'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.

