Validation of Semantic Interoperability between IoT Platforms

Eftychia Lakka, Nikolaos E. Petroulakis, Emmanouil Michalodimitrakis, Emmanouil Papoutsakis
Foundation for Research and Technology – Hellas (FORTH)
Heraklion, Crete, Greece
elakka@ics.forth.gr, npetro@ics.forth.gr, manmix@ics.forth.gr, paputsak@ics.forth.gr

***Abstract*—The growing trend towards the exchange of data between heterogeneous objects and platforms have increased the level of complexity in the Internet of Things (IoT) area. Beyond those, to enable seamless resource sharing between different IoT systems, efforts by academia and industry have currently focused on achieving semantic interoperability between IoT platforms. That involves the delivery of dynamic interaction between platforms to interconnect the variety of smart objects covering both multiple technologies and intelligent services. With this in mind, the main goal of this work is to propose a mechanism to validate and achieve interoperability between IoT platforms. A motivating example is described and analyzed along with the implementation of the proposed system.**

***Keywords*-Semantic Interoperability, IoT Platform, Smart Objects, Thing Description**

I. Introduction

While the world is faced with economic, energy and societal challenges, the introduction of advanced, real-time and interactive services that are highly relevant to the Internet of Things (IoT) paradigm appears to be the key to address these challenges. To this extend, the capabilities of IoT for sensing, accumulating and transferring data is related to the interconnection of physical and virtual object, devices and actors over the network. That includes the classification of different categories such as (i) people to people, (ii) people to machine/things, (iii) things/machine to things/machine [1]. Taking to account that billions of sensors and actuators already deployed into a number of domain-specific platforms, at the same time this appears to be one of the biggest conceptual and technological challenges of our time [2].

As a result, the widespread interest of both industry and academy is increased for new research and developing IoT solutions [3]. Specifically, many IoT platforms have been proposed in order to manage and maintenance the huge amount of heterogeneous data formats from multiple sources that has already been increasing tremendously. For this reason, the main aim of this research is to investigate the interoperability between the different IoT platforms. In fact, the proposed solutions require to provide interconnection and interoperability between the components of different platforms. This increase in connectivity creates many prominent challenges such as integration issues, thing interaction, security, privacy and monitoring of devices. The first case is mentioned in the fact that the IoT domain requires standards to enable horizontal platforms that are communicable, operable, and programmable across devices, regardless of their make, model, manufacturer or industrial applications [4]. Moreover, for the interaction of *things* to occur, a user or device needs a way to search for *things* and access the data they produce. This includes many fundamental communication issues such as those relating to how *things* are represented, searched and accessed on the Internet [5]. There are also additional issues that should be considered related to the security (authorization, authentication, trust, integrity, validation, etc.) and privacy for both of the users and devices [4].

Hence, the interconnection between different IoT platforms requires the deployment of a bridge capable to read and write inserted data, necessary to communicate with the targeted IoT platform. Depending on the level of required communication, the bridge could be classified in two main categories related to the devices and the services. The *Device Layer Bridge* includes the communication between towards and from the sensors and actuators and the *Service Layer Bridge* that manages communication towards and from services (e.g. for applications or for other IoT platforms) [6].

Motivated by the above, this work aims to provide integration and interoperability of heterogeneous IoT platforms. For that reason, a framework is proposed to validate the compatibility between different IoT platforms. The proposed approach is based on the SEMIoTICS [7] architecture[1]. The remainder of this paper is organized as follows: Section II provides the background and related works regarding the interoperability between IoT platforms. Section III details the methodology and the implementation details of the proposed approach. Section IV gives a brief overview of the testing and the performance evaluation. Finally, Section V features the concluding remarks and pointers to future work.

II. Background

Interoperability among heterogeneous systems involves different categories [8] that are also associated with the IoT domain. The *technical interoperability* is usually related to

[1]SEMIoTICS: Smart End-to-end Massive IoT Interoperability, Connectivity and Security: https://www.semiotics-project.eu/

978-1-7281-6728-2/20/$31.00 © 2020 IEEE

Authorized licensed use limited to: Foundation for Research and Technology (FORTH). Downloaded on September 07,2020 at 11:41:08 UTC from IEEE Xplore. Restrictions apply.

the hardware/software components, systems and platforms that enable machine-to-machine communication to take place. This kind of interoperability is often focused on the communication protocols and the infrastructure needed by those protocols to operate. The *syntactical interoperability* is usually associated with data formats of high-level transfer syntaxes such as HTML, JSON, XML or ASN.1. Furthermore, the *semantic interoperability* can enable a common understanding between *things* and the content of exchanged information [9]. Moreover, the *organizational interoperability* defines the ability of organizations to effectively communicate and transfer meaningful data information even though they may be using a variety of different information systems over widely different infrastructures, possibly across different geographic regions and cultures. Especially for the IoT domain, interoperability can be classified into a variety of different categories including the device interoperability, networking interoperability, syntactic interoperability, semantic interoperability and platform interoperability. The rest of this section is mostly involved with an overview of the platform interoperability as this is the main focus of this work.

A. Cross-Platform Interoperability

The challenge of information system interoperability has been already identified in 1988 [10]. However, the current topic for research focuses on the platform interoperability investigating how IoT applications can access and integrate data from various platforms [11]. Based on the above, the achievement of interoperability between IoT systems requires cross-platform interoperability. This interoperability level provides the interaction between different IoT platforms to one vertical domain such as smart home and smart healthcare [12]. In fact, heterogeneous data from different sources including semantic and non-semantic data (ie. web data, relational data, RDF, XML, CSV, JSON) have an important effect on IoT service provisioning. Since the data is not always in the same type or format, it is required its transformation to a machine-readable format to guarantee interconnection and communication [13].

B. Cross-Domain Interoperability

Although, cross-platform solutions focus on specific activities that are limited to one domain, the cross-domain interoperability enables the federation of different platforms within heterogeneous domains to build horizontal IoT applications. In detail, the provision of cross-domain interoperability among collaborating domains is a new challenge which requires more attention from the research community. Such interoperability requires both data and service extraction in order to acquire ordinary subsets of information and services in collaborating domains, such as healthcare [14]. Consequently, in this case, the IoT platforms should have the opportunity to choose the desired protocols in order to achieve the end-to-end communications and data exchange (from sensors to gateways to cloud-based platforms) based on their requirement and purpose [12]. In a representative approach [15], a wrapper IoT solution was proposed such as *Sensing as a Service* procedure to offer an open source middleware platform for enabling the delivery of IoT services in a dynamic method which follows a utility model, in which the service provider makes computing resources and infrastructure management available to the client as needed.

III. Methodology and Implementation Details

A. Interoperability in the SEMIoTICS Framework

The interoperability between IoT devices and the IoT platforms is based and implemented on the SEMIoTICS architectural framework [16]. The interaction of these IoT platforms is enabled within the architecture throughout the Field Layer, SDN/NFV Orchestration Layer and Application Orchestration Layer as illustrated in Figure 1.

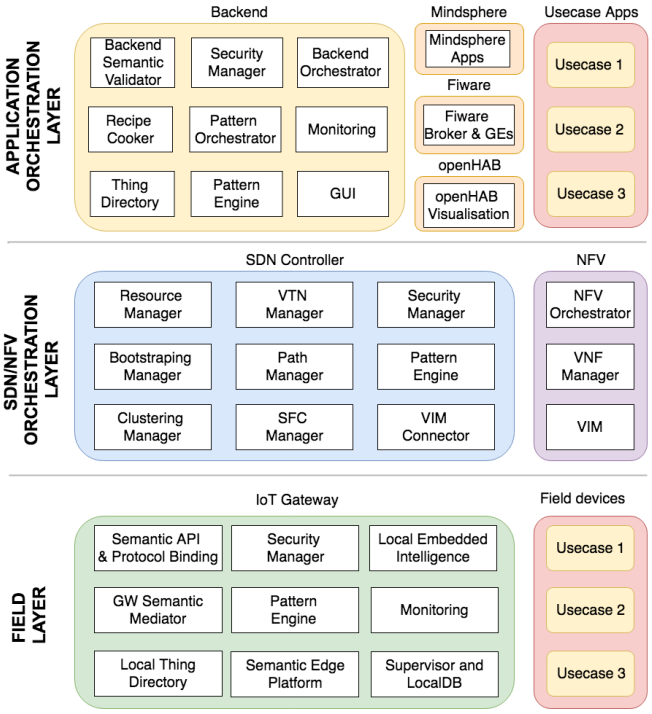


Figure 1: Key components of the SEMIoTICS architecture related to interoperability with external IoT platforms

The research in this work is focused on the investigation of two different cases in which the semantic interoperability is enabled. The first case includes the interaction between the IoT platforms through the framework. In that way, the use of the exposed interfaces of the said IoT platforms can take advantage of IoT devices whose descriptions are available in repositories outside framework. The second case involves the interaction of the framework with the other

Authorized licensed use limited to: Foundation for Research and Technology (FORTH). Downloaded on September 07,2020 at 11:41:08 UTC from IEEE Xplore. Restrictions apply.

IoT platforms. In a similar way, the said platforms utilize the framework exposed interfaces of selected components in order to employ IoT smart objects and services. In the next subsection, the implementation and the key components of the presented framework architecture to achieve semantic interoperability are described.

To achieve bidirectional communication, the framework can enhance different external IoT platforms including the FIWARE[2], the MindSphere[3] and the openHAB[4]. For the interconnection and the interoperability between FIWARE and the framework, it is required the deployment of a bridge capable to map the semantic concepts between them. Based on FIWARE data model, an adaptation of the FIWARE data model is proposed to be matched with W3C Web of Things including its thing description and iotschema.org [5] (see Table I). In the same way, the data model of MindSphere is being mapped to semantic constructs of iotschema.org as shown in Table I. The model is based on Assets, a digital representations of a machine or an automation system with one or multiple automation units connected to MindSphere. Finally, the openHAB REST and JSON-LD[6] protocols can offer a simple integration path with its protocol suite without the need of any additional parsers. This is achieved by the use of the openHAB Transformation Services that can map the thing descriptions of the framework to the openHAB's Eclipse Smarthome Data Model.

Table I: An overview of mapping FIWARE and MindSphere Model to iotschema.org

FIWARE	**MindSphere**	**Framework**
Entity	Asset	Thing Description
Entity Id/Type	-	Thing Description Id/Type
Attributes	-	Properties
-	AspectType	Capability
-	Aspect	Thing Description's part marked up with a Capability
-	Variable	Thing Description's Event, Property or Action

B. Implementation Analysis

The main goal of the proposed mechanism is to enable the semantic interoperability between the framework and external IoT platforms. More specifically, to enable the interoperability between the respective application programming interfaces (APIs) of the components and the platforms of the framework, a number of different step phases are required, following the corresponding sequence diagram in Figure 2. In addition, to describe the applicability of our proposed interoperability solution, a motivated scenarios is presented. The scenarios includes the interoperability between the framework and the FIWARE for the description and analysis of the development of the proposed approach and steps.

1) Flow Design: The first phase includes the design of the flow interaction between two *things*. The main goal is to validate the semantic interoperability between the identified nodes to ensure the required interconnection. The workflow is designed in the Node-RED framework[7] and is responsible for expressing flows and send them as requirements to the pattern orchestrator using a POST method request. Particularly, this POST parameters includes the flow in JSON format, as body and the header as application/json. Pattern orchestrator is in charge of the automated configuration, coordination, and management of different patterns and their deployment to express requirements of the flows to guarantee interoperability based on architectural patterns [17], [18].

2) Interoperability Requirement: The second phase includes the insertion of the interoperability requirement as a POST from the orchestrator to the pattern engine to enforce the respective pattern rules. Backend pattern engine allows the insertion, modification, execution, and retraction of patterns expressed as through the pattern orchestrator. Patterns are expressed as *Drools*[8] business production rules, running and the associated rule engine by applying and extending the Rete algorithm. The latter is an efficient pattern-matching algorithm known to scale well for large numbers of rules and data sets of facts, thus allowing for an efficient implementation of the pattern-based reasoning process. The interaction between the orchestrator and the pattern engine, can offer not only interoperability but also verification of Security, Privacy, Dependability and along with Interoperability (SPDI) and Quality of Service (QoS) properties as a service to be used from the other IoT platforms. In that way, an external IoT platform may utilize the said service, in order to verify the SPDI/QoS properties to an existing workflow that is comprised of IoT smart objects (IoT service workflow). The IoT service workflow in question is described in a dedicated language that the orchestrator understands and is given as input. It exchanges information with the pattern engine to check whether a specific property holds throughout the whole workflow and corresponds accordingly.

3) Interoperability Verification: The third phase includes the triggering of the semantic interoperability verification pattern inside the pattern engine. The pattern is expressed in a machine-processable Drool rule format of the said semantic interoperability for any inserted flow (Figure 3). The *when* part identify the requested placeholders placed in sequence, required to satisfy the semantic interoperability

[2]FIWARE: www.fiware.org
[3]https://siemens.mindsphere.io
[4]www.openhad.org
[5]https://github.com/iot-schema-collab - http://iotschema.org/
[6]https://json-ld.org/
[7]https://nodered.org/
[8]https://www.drools.org

Authorized licensed use limited to: Foundation for Research and Technology (FORTH). Downloaded on September 07,2020 at 11:41:08 UTC from IEEE Xplore. Restrictions apply.

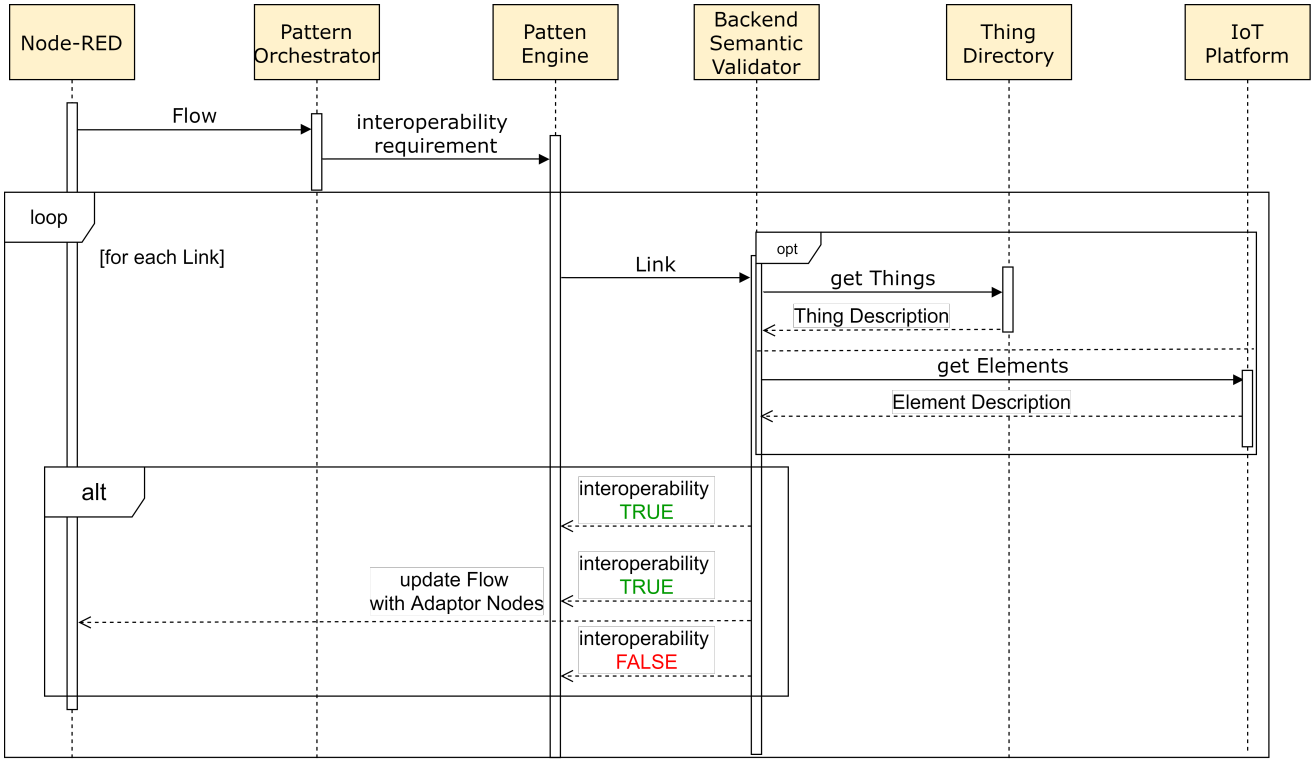


Figure 2: Sequence diagram of SEMIoTICS interoperability with external IoT platforms

```
rule "Semantic Interoperability Verification"
when
 Placeholder($pA:=placeholderId)
 Property($pA:=subject, category=="semantic",
  $prValueIn1:=input_value,
  $prValueOut1:=output_value,satisfied==true)
 Placeholder($pB:=placeholderId)
 Property($pB:=subject,category=="semantic",
  $prValueIn2:=input_value,
  $prValueOut2:=output_value,satisfied==true)
 Sequence($sId:=placeholderId,
  $pA:=placeholderA,$pB:=placeholderB)
 $PR:Property($sId:=subject,
  category=="semantic",
  $prValueOut1==$prValueIn2,satisfied==false)
then
 modify($PR){satisfied=true,
  input_value=$prValueIn1,
  output_value=$prValueOut2};
end
```

Figure 3: Semantic Interoperability Verification Pattern Rule

property. If the conditions are met, the rule in *then* can guarantee that the requested property is satisfied. Hence, the Backend Semantic Validator (BSV), which resolves semantic interoperability issues, receives a request from the Pattern Engine to check the semantic interoperability between said two *things* (link) in JSON-LD/JSON format. The JSON-LD/JSON Parser is implemented as part of the semantic validator, to analyze the received input and extract the meaningful information from these set of data. The communication between the said components is achieved using the POST method.

4) Semantic Validation: The fourth phase includes the semantic validator that begins the procedure to tackle the semantic interoperability issues between these two *things* from the said flow. In order to give this answer, the semantic description for any *thing* is required (ie. for FIWARE Sensor and SEMIoTICS Thermostat). For that reason, it sends two requests: (i) SPARQL query to thing directory (based on the Thingweb Directory[9]) in order to receive the Thing Description of a sensor such as thermostat and (ii) GET request to the Orion Context Broker FIWARE platform to receive the context data Description of FIWARE Sensor. The response consists of JSON format with the FIWARE Sensor attributes (type, metadata elements).

5) Interoperability Adaptation: The fifth phase involves the received by the validator information in order to decide regarding the interoperability between the *things* and harmonize the semantic model capabilities with the registration of extra Adaptor Nodes in the node-red if required. Particularly,

[9]https://github.com/thingweb/thingweb-directory

Authorized licensed use limited to: Foundation for Research and Technology (FORTH). Downloaded on September 07,2020 at 11:41:08 UTC from IEEE Xplore. Restrictions apply.

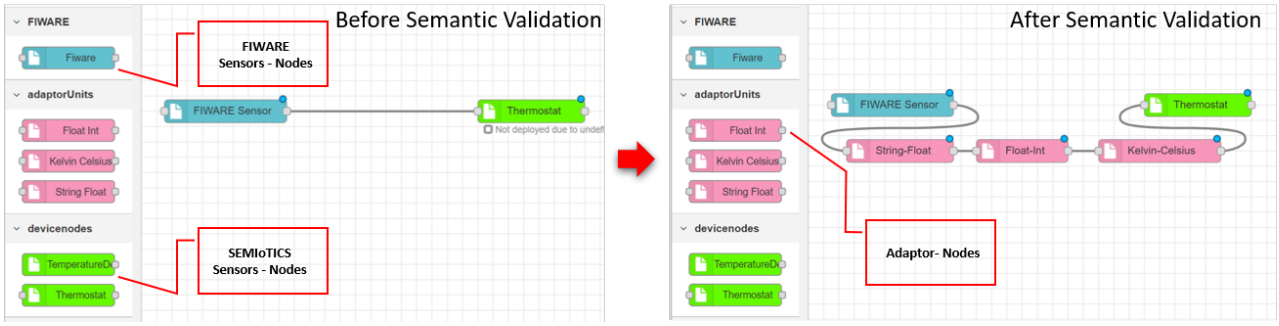


Figure 4: Interoperability between SEMIoTICS and FIWARE

there are three possible results. Firstly, the link source and destination are interoperable, so the validator updates the pattern engine with the TRUE response. Secondly, the link source and destination are not interoperable and the validator can add adaptor nodes in order to guarantee the interoperability. In this case, the semantic validator not only sends the TRUE response to pattern engine, but it also updates the workflow in the node-red using the corresponding adaptor nodes. Lastly, when the link source and destination are not interoperable and the validator does not have the required information to develop the adaptor nodes, the validator sends FALSE response to pattern engine.

Apart from the developed capability of the framework to use the exposed interfaces of the said IoT external platforms, in the corresponding way, the framework is capable to interconnect with external IoT platforms based on the exposed interfaces of related components to employ IoT smart objects and services. Namely, one of the selected components that has exposed interfaces, is the thing directory which is the repository of knowledge containing the necessary semantic description of *things*. Its interface can be used to retrieve the stored semantic description of *things*. The Thing Description of the corresponding *things* that is returned, complies with the iotschema.org and can be used for consumption from other IoT platforms.

IV. Performance Evaluation

To evaluate the performance and functionalities of the described interoperability framework, the Proxmox Virtual environment[10] is used to create Virtual Machines (VMs) for the different involved components and the external IoT platform (Orion Context Broker FIWARE[11]). All the involved components are instantiated as different VMs running Ubuntu 16.04 operating system inside Proxmox. These components are required to enable the interaction of exchanged data using the APIs between framework and the other IoT platforms as described in the previous section. The overall deployment is scalable and sufficient for real-time operation.

[10]https://www.proxmox.com/
[11]https://fiware-orion.readthedocs.io/

Based on the described methodology, a typical testing workflow includes the validation of semantic interoperability between a FIWARE Sensor and a SEMIoTICS sensor. In the described example, there is only one link/wire identifying the connection between FIWARE Sensor and SEMIoTICS Thermostat. The design of a flow is inserted in the node-red environment including a FIWARE sensor and a sensor thermostat, representing the flow interaction between two *things*. The overall functionality is to check whether the semantic interoperability and the aforementioned communication is ensured. Then, the flow is sent to the pattern orchestrator in order to transform it to a requirement for the pattern engine.

The semantic interoperability is examined regarding the link/wire between sensor and thermostat in the flow as trigger by the Semantic Validator. The validator takes the semantic metadata of two *things* using the Thing Directory and Orion Context Broker FIWARE. Based on this information, the validator decides regarding the interoperability between the *things*. To harmonize the semantic model capabilities, extra adaptor nodes are registered in the flow and the corresponding response of the registration is sent back to pattern engine. Therefore, in the described example, additional adaptor nodes are inserted by the validator between the source and destination, to guarantee the interoperability updated also the initial flow.

The results of the actual evaluation of the workflow status before and after the semantic validation can be seen in Figure 4. Also, it is worthwhile noting that the above adaptor nodes are produced at runtime with the corresponding functionality. In a large-scale scenario, the semantic interoperability is examined for any link/wire between the components of the flow (more than one FIWARE sensors/actuators can belong to the flow components) and a combination of different adaptor nodes can be used parallel to ensure the interoperability of the system. Thus, the evaluation of the proof of concept implementation satisfied the initial requirement for semantic interoperability between *things*, revealing also the capabilities for its enhancement in more complex workflows.

Authorized licensed use limited to: Foundation for Research and Technology (FORTH). Downloaded on September 07,2020 at 11:41:08 UTC from IEEE Xplore. Restrictions apply.

V. Conclusion and Future Work

This paper has presented the development of mechanisms to achieve interoperability and connectivity between different IoT platforms. The proposed approach provides the sharing of smart objects (*things*), which are in different formats, between incompatible IoT platforms. The strong point of our work lies in the fact that overcomes the semantic gap between two (or more) IoT platforms; as a result, *things* from different platforms can interact and exchange data despite their use of different standards and to their high inner heterogeneity. The representative motivated scenario is briefly described involving the interaction of SEMIoTICS framework and FIWARE platform using temperature sensing sensors, which are described by different semantic schemes. As for the future work, our proposal will be extended in order to offer the adaptation between different semantic schemes; in other words, the next step is the bridge for the interoperability between the framework and additional targeted external IoT platforms.

Acknowledgment

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No. 780315 (SEMIoTICS).

References

[1] K. K. Patel, S. M. Patel *et al.*, "Internet of things-iot: definition, characteristics, architecture, enabling technologies, application & future challenges," *International journal of engineering science and computing*, vol. 6, no. 5, 2016.

[2] G. Hatzivasilis, I. Askoxylakis, G. Alexandris, D. Anicic, A. Bröring, V. Kulkarni, K. Fysarakis, and G. Spanoudakis, "The interoperability of things: Interoperable solutions as an enabler for iot and web 3.0," in *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Sep. 2018.

[3] G. Fortino, C. Savaglio, C. E. Palau, J. S. de Puga, M. Ganzha, M. Paprzycki, M. Montesinos, A. Liotta, and M. Llop, "Towards Multi-layer Interoperability of Heterogeneous IoT Platforms: The INTER-IoT Approach." Springer, Cham, 2018, pp. 199–232.

[4] M. Elkhodr, S. Shahrestani, and H. Cheung, "The internet of things: new interoperability, management and security challenges," *International Journal of Network Security Its Applications*, vol. 8, no. 2, 2016.

[5] M. Blackstock and R. Lea, "Toward interoperability in a web of things," in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, ser. UbiComp '13 Adjunct. New York, NY, USA: Association for Computing Machinery, 2013, p. 1565–1574. [Online]. Available: https://doi.org/10.1145/2494091.2497591

[6] "D3.10 ACTIVAGE bridges for European Platforms." ACTIVAGE European Project, 2018.

[7] N. E. Petroulakis, K. Fysarakis, G. Spanoudakis, V. Kulkarni, and S. Ioannidis, "Smart end-to-end massive iot interoperability, connectivity and security (semiotics)," *ERCIM News*, vol. 2019, no. 119, 2019.

[8] A. W. Hans van der Veer, "Achieving Technical Interoperability - the ETSI Approach." ETSI White Paper No. 3, 3rd edition, 2008.

[9] E. Lakka, N. E. Petroulakis, G. Hatzivasilis, O. Soultatos, M. Michalodimitrakis, U. Rak, K. Waledzik, D. Anicic, and V. Kulkarni, "End-to-end semantic interoperability mechanisms for iot," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, 2019, pp. 1–6.

[10] F. Eliassen and J. Veijalainen, "A functional approach to information system interoperability," R. Speth, Ed., Vienna, Austria,, 1988, 1, pp. 1121–1135.

[11] M. A. Mahda Noura and M. Gaedke, "Interoperability in internet of things: Taxonomies and open challenges." Mobile Networks and Applications, 2019.

[12] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in internet of things infrastructure: Classification, challenges, and future work." IOTAAS, 2018.

[13] "D3.1 Initial data and capabilities models for cross-platform interoperability." BRAIN-IoT European Project, 2018.

[14] K. Sartipi and A. Dehmoobad, "Cross-domain information and service interoperability," 2008, p. 25–32.

[15] M. S. John Soldatos, Nikos Kefalakis and M. Hauswirth, "Design principles for utility-driven services and cloud-based computing modelling for the internet of things," 2014.

[16] N. E. Petroulakis, E. Lakka, E. Sakic, V. Kulkarni, K. Fysarakis, I. Somarakis, J. Serra, L. Sanabria-Russo, D. Pau, M. Falchetto *et al.*, "Semiotics architectural framework: End-to-end security, connectivity and interoperability for industrial iot," in *2019 Global IoT Summit (GIoTS)*. IEEE, 2019, pp. 1–6.

[17] K. Fysarakis, G. Spanoudakis, N. Petroulakis, O. Soultatos, A. Bröring, and T. Marktscheffel, "Architectural patterns for secure iot orchestrations," in *2019 Global IoT Summit (GIoTS)*. IEEE, 2019, pp. 1–6.

[18] K. Fysarakis, M. Papoutsakis, N. E. Petroulakis, and G. Spanoudakis, "Towards iot orchestrations with security, privacy, dependability and interoperability guarantees," in *2019 IEEE Global Communications Conference, GLOBECOM 2019, Waikoloa, HI, USA, December 9-13, 2019*, 2019, pp. 1–6. [Online]. Available: https://doi.org/10.1109/GLOBECOM38437.2019.9013275

Authorized licensed use limited to: Foundation for Research and Technology (FORTH). Downloaded on September 07,2020 at 11:41:08 UTC from IEEE Xplore. Restrictions apply.