Secure Semantic Interoperability for IoT Applications with Linked Data

George Hatzivasilis, Othonas Soultatos, Eftychia Lakka, Sotiris Ioannidis
Institute of Computer Science
Foundation for Research and Technology – Hellas (FORTH)
Heraklion, Crete, Greece
hatzivas@ics.forth.gr, soultatos@ics.forth.gr, elakka@ics.forth.gr, sotiris@ics.forth.gr

Darko Anicic, Arne Bröring
Siemens AG
Corporate Technology Siemens
Munich, Germeny
darko.anicic@siemens.com, arne.broering@siemens.com

Konstantinos Fysarakis, George Spanoudakis
Sphynx Technology Solutions AG
Zug, Switzerland
fysarakis@sphynx.ch, spanoudakis@sphynx.ch

Mirko Falchetto
STMicroelectronics S.r.l.
Agrate Brianza, Italy
mirko.falchetto@st.com

Lukasz Ciechomski
BlueSoft Sp. z.o.o.
Warsaw, Poland
lciechomski@bluesoft.net.pl

***Abstract*—Interoperability stands for the capacity of a system to interact with the units of another entity. Although it is quite easy to accomplish this within the products of the same brand, it is not facile to provide compatibility for the whole spectrum of the Internet-of-Things (IoT) and the Linked Data (LD) world. Currently, the different applications and devices operate in their own cloud/platform, without supporting sufficient interaction with different vendor-products. As it concerns the meaning of data, which is the main focus of this paper, semantics can settle commonly agreed information models and ontologies for the used terms. However, as there are several ontologies for describing each distinct 'Thing', we need Semantic Mediators (SMs) in order to perform common data mapping across the various utilized formats (i.e. XML or JSON) and ontology alignment (e.g. resolve conflicts). Our goal is to enable end-to-end vertical compatibility and horizontal cooperation at all levels (field/network/backend). Moreover, the implication of security must be taken into consideration as the unsafe adoption of semantic technologies exposes the linking data and the user's privacy, issues that are neglected by the majority of the semantic-web studies. A motivating example of smart sensing is described along with a preliminary implementation on real heterogeneous devices. Two different IoT platforms are integrating in the case study, detailing the main SM features. The proposed setting is secure, scalable, and the overall overhead is sufficient for runtime operation, while providing significant advances over state-of-the-art solutions.**

Keywords—semantics, linked data, data mapping, ontology alignment, interoperability, IoT, JSON-LD, SPARQL-LD

I. Introduction

This paper tackles the semantic interoperability issues that arise in the Internet of Things (IoT) domain [1]. Semantic interoperability is the designed property where various systems can interact with each other and exchange data with unambiguous, shared meaning. This enables knowledge discovery, machine computable reasoning, and federation of different information systems.

Interoperability is materialized by including information regarding the data (metadata) and linking each element to a commonly shared vocabulary (e.g. [2], [3]). Thus, the meaning of the data is exchanged along the data itself in a self-describing information package. The shared vocabulary and the associations to an ontology enable machine interoperation, logic, and inference. Ontology is the explicit specification of a conceptualization and includes a formal representation of the properties and relations between the entities, concepts and data of a specific application domain. In general, technologies from the Semantic Web are adapted in order to capture the inherited properties of an IoT ecosystem [4], [5]. They are mainly XML schemes, such as the RDF, RDFS, and OWL for ontologies, and for services the WSDL. These primitives provide common definitions of data or services, describe things with the underlying properties, and accommodate the semantic annotations, discovery of resources, inference of knowledge, and access control, in an interoperable and machine-readable fashion.

The common format and meaning of semantics in a universally accepted ontology, as suggested above, would be fruitful. Yet, this is not the current status [1]. While various systems could employ standardized or popular ontologies, eventually they extend them and settle own interfaces and semantics (e.g. [4], [5]). Thereby, the direct interaction of such systems is infeasible. A smart watch for example, which is developed in IOS could not interwork with smart bulbs without a relevant proprietary gated application from the same brand. Therefore, islands of IoT functionality are established, leading towards a vertical 'Intranet-of-Things' instead of the actual vision of an 'Internet-of-Things'. To presume upon the full potential of the IoT setting, we require standards for accomplishing the desired horizontal and

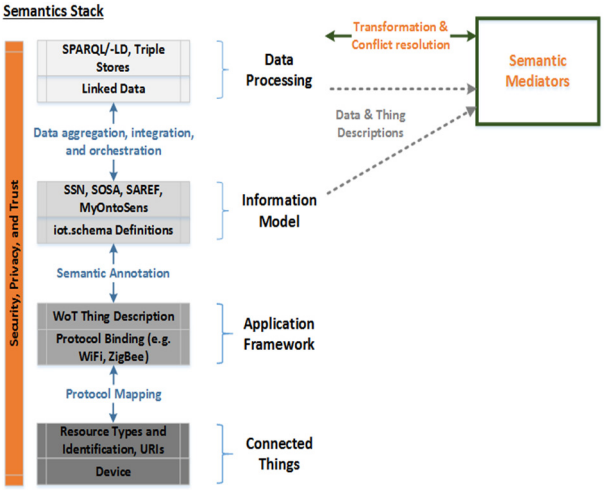


Fig. 1. The semantics stack in IoT.

vertical operation, communication, and programming across platforms/devices, independent of their vendor and/or model.

Nevertheless, the cyber-security concerns must be also taken into account throughout the whole semantic resolution process. The mainstream network defences alone (i.e. TLS) are not adequate in protecting the communication against an emerging type of malicious entity, the *semantic attacker* [6], [7]. While the ordinary attacks could exploit the lack or bad configuration of cryptography, adversaries in the Semantic Web try to manipulate the semantic relations and the RDF rules. The goal is to control the inference operation of the reasoning components that collect, correlate, and process data. The semantic attacks exploit network or Web level vulnerabilities. They do not attack the reasoning system itself but they try to compromise the input data to influence the deduced conclusions. The distributed nature of the IoT and the linking data of the social Web exaggerates the problem, especially in the case of the *follow your nose* algorithm [8] that is performed by many Linked Data (LD) applications.

Thus, the deployment of **Semantic Mediators (SMs)** is recommended in this article in order to correlate the required information and materialize cross-domain interaction with interoperability between systems of different semantics. The SMs transform data in the same format and resolve potential conflicts between the different thing descriptions. Security countermeasures are also deployed, protecting the data both in transit and at rest. The main contributions of the proposed SMs include: i) Cooperation with legacy, XML, and JSON formats, ii) Support of W3C initiatives for IoT and LD (i.e. standardized ontologies, iot.schema.org, JSON-LD), iii) Direct processing of JSON-LD data by the inference and reasoning modules (with SPARQL-LD [9]), iv) Secure transmission of data (TLS at all communications) defending the system against data in transit attacks, v) Validation of the data legitimacy prior their usage (i.e. JWS/JOSE framework [10], [11]), protecting against the data in rest semantic attacks, vi) Distributed functionality across the edge, network, and backend systems, vii) Efficient and scalable operation, viii) Integration of two EU funded IoT initiatives: the SEMIOTICS project and the FIWARE cluster, and viiii) Advancements over state-of-the-art solutions.

The rest article is structured as: Section 2 refers the background theory regarding the semantic interoperability in the IoT domain and related works for semantic mediators/brokers. Section 3 evinces the security perspective. Section 4 describes the proposed SM component. Section 5 details the implementation of a preliminary version along with the performance evaluation. Section 6 discusses the evaluation results and presents a comparative study with alternative candidates, while Section 7 concludes and mentions future extensions.

II. Background Theory & Related Work

A. Semantics

There are several Sematic Web initiatives that try to describe and model specific domain ontologies [12]. The most notable effort for semantics formation in the IoT field are the Semantic Sensor Network (SSN) and Sensor Observation Sampling Actuator (SOSA) ontologies by the W3C community [2]. Combined together, SOSA/SSN model sensors, actuators, samplers as well as their observation, actuation, and sampling activities. The ontologies capture the sensor and actuator capabilities, usage environment, performance, and enabling contextual data discovery. This also constitutes the standardized ontologies for the semantic sensor networks. The cooperation of SSN and SOSA offers different scope and degrees of axiomatization that enable a wide range of application scenarios of Web 3.0 [13].

The general approach regarding the semantic interoperability that is followed by several IoT initiatives, like the EU funded projects OpenIoT [4] and INTER-IoT [5], is the usage of the SSN/SOSA ontologies as the semantic base. The ontologies are then extended with the additional required concepts to model the targeted application scenarios. Such concepts usually include relevant standards and ontologies for specific application areas, like e-health [14], and less often extensions at the sensor level (as the relevant SSN/SOSA information is quite complete). Other similar and popular IoT ontologies include the MyOntoSens [15] and the SAREF [16].

In the ongoing shift towards the Web 3.0, we move from a *Web of linked documents* into a *Web of linked data* [3], [13]. Except from modelling ontology schemas as mentioned before, this also includes methods for publishing structured data in a manner that it can be interlinking and accessed by semantic queries, like in the LD approach. Just recently, the working group Web of Things (WoT) was initiated by the W3C in an attempt to circumscribe the fragmentation issues in IoT and enable interoperable services and devices, therefore decreasing the overall development costs. The Thing Description (TD) constitutes a considerable aspect of this W3C WoT interplay. TDs describe the interfaces of (physical) Things and their metadata in a machine interpretable manner. They are built upon the W3C's extensive efforts on RDF and JSON- LD, and determine a domain agnostic vocabulary for defining any Thing in terms of its properties, actions, and events. Here, several semantic models can express the semantic meaning of these attributes for each particular Thing. The iot.schema.org is such a meritorious communal effort to establish a semantic schema for the IoT ecosystems. Jointly, W3C WoT and iot.schema.org, instate a layer for semantic interoperability which renders the software capable in interacting with the physical world. This interplay is abstracted in such a manner where the development of applications across various IoT settings and domains is ease and simplified.

Then, data can be transmitted in an RDF format and stored in triple stores (e.g. Sesame and Virtuoso) [17], [18]. Thereafter, tools are used which process semantic queries [19]. The standardized SPARQL is the query language for the Semantic Web [20]. It acts as a semantic database and constitutes the main option for semantic reasoning. Methods are also supported to interrogate multiple triple-stores over HTTP. SPARQL can process data with XML format and exploit the RDF rules in order to answer queries for the stored information. An interesting variant is the SPARQL-LD [9]. This version parses JSON-LD data and can also gather linking information directly from the Internet. The implementation extends a popular SPARQL processor for the Jena Apache server [9] and is quite efficient (an ask query on DPedia requires around 300ms on average).

Fig. 1 illustrates the overall semantics stack of a modern IoT setting. Thereafter, this semantic layers are adopted in the SEMIOTICS project and we utilize the SMs in order to align the semantics of other cooperative platforms, like the FIWARE cluster. Also, the SMs embodies the SPARQL-LD

for the implementation of the semantic reasoning and the direct processing of JSON-LD data across the Internet.

B. Ontology Alignment

Depending on the completeness and expressiveness of the processed ontologies, the aforementioned reasoning tools can infer associations among the different semantic domains. OWL rules are exploited for this purpose, like the *owl:sameAs*, *owl:equivalentClass*, and *owl:subClassOf*. They provide the basic ontology alignment functionality between the underlying linking information. Thus, data are adapted and transformed across signified ontologies.

However, as Halpin et al. state [21], expressing relationships on LD is a much more complex problem than just applying the *owl:sameAs* rule. Currently, there is a plethora of distinct datasets that have been developed independently. The problem raises when someone tries to integrate/correlate these pieces of knowledge together, as many of the applied *owl:sameAs* rules tend to be mutually incompatible. Indicative cases of erroneous usage include definitions for: i) the same thing but in different context, ii) the same thing but in referentially opaque, iii) different representations of the same thing, and iv) very similar things [21]. Moreover, the expressive capabilities of OWL are constrained and are not always sufficient for modeling complex interrelations or resolving semantic conflicts [22], [23]. Doulaverakis et al. also mention [23] that the knowledge base have to be extended with rules in order to capture situations that cannot be defined by OWL alone.

Therefore, semantic brokers are deployed to perform ontology alignment and resolve semantic conflicts [22], [23], retaining the reliability and Quality of Service (QoS) in the IoT setting. The Semantic Information Brokers (SIBs) [22] is a typical option for semantic interoperability in pervasive computing. The semantics are described by mainstream solutions, like RDF/OWL, and distributed SIBs resolve semantics in the local smart space scale. Then, Resolve Servers interlink the knowledge originated by SIBs and permit cross-domain interaction, exploiting the modelling capabilities of OWL. SPARQL is utilized for reasoning.

The Intelligent Information Fusion (IIF) [23] recommends a low level mechanism for consolidating information from heterogeneous sensory equipment. The case study considers a city-wide public surveillance system that has to process information from multiple resources (e.g. visible spectrum or IR cameras, and acoustic sensors) in real-time. The semantics of each domain are modelled by the mainstream XML/RDF technologies and the reasoning is implemented as an SQL-like procedural language in SPARQL. Then, the user defines fusion functions describing which pieces of information are to be retrieved (e.g. number of persons, detection of smoke, etc.) by each platform and how to use them. IIF successfully correlates semantics of the same format for different domains and overcomes some restrictions of the OWL expressiveness. On the other hand, it does not resolve semantic conflicts.

Similarly, the semantically-enabled Plug & Play approach for the Sensor Web ([24], [25]) facilitates the automatic association of sensors to data hosting Web services. A mediation approach is implemented based on an ontology that extends SSN and a set of SWRL rules. First, the sensor metadata (expressed in standard SensorML) is auto-translated into the ontology. Then, the matchmaking is performed through subsumption reasoning between those advertised sensor metadata and the requirements specified by Web services. Finally, for spatial, temporal and unit matchmaking, SWRL rules are executed and also employed for mediation between convertible mismatches.

III. The Security Inpmlication

The Semantic Web utilizes the widely-known Uniform Resource Identifiers (URIs) as a mean to address and link data and their sources. The consumer (user or service) discovers the required information in the Web, retrieves the data from the related URI, and processes it. However, several security aspects have been neglected throughout the overall operation [26]. W3C standardized the core semantic technologies before the formalization of the Same Origin Policy and the TLS, and thus, the Semantic Web had been designed without taking into account the security implications [26]. Until even today, there are almost no academic works on the Semantic Web security [27], [28]. Furthermore, there is also considerable confusion regarding the underlying security aspects, like the usage of HTTP URIs or the misuse of cryptographic solutions (i.e. TLS, WebID+TLS.2, etc.). These facts also raise significant privacy risks for the personal data that can be exposed and linked across the Internet.

If TLS has not been properly set in the origin, a network attacker acting as a man-in-the-middle can manipulate the transmitted traffic. The malicious entity can gain the control of the exchanged information and perform a series of specialized attacks (e.g. Coercive parsing, SOAPAction spoofing, Metadata spoofing, attack obfuscation, WS-Addressing spoofing, attacks on Web Service Compositions through BPEL state deviation, signature wrapping with namespace injection, etc.) [6], [7]. These exploits could potentially change endpoint URLs, message schemas, cryptographic parameters, or remove security assertions, and even add/delete/change/fake operations. Over and above, the attacks can be done quite easily with open-source tools (e.g. sslstrip or wireshark), while it is impossible for the consumer to discriminate the malicious activity.

Nevertheless, except from securing the information in transit with TLS, we must also protect the data in rest at the backend side. This is an even more neglected research issue for the Semantic Web specialists and practitioners. As Thuraisingham states [28], securing RDF is a much more challenging task than in the ordinary HTML/XML settings as we also need to retain the security of the semantic level. So far the highest majority of researchers and Semantic Web users simply consider the protection mechanisms for access control and secure transmission [27].

Thus, consider the case where the hacker infiltrates a vulnerable server that hosts ontologies or schemas, and replaces them with malicious ones (in a similar fashion as they can change the HTML code of popular sites). The same effect could be accomplished through DNS poisoning, where the attacker makes the traffic to be routed in a compromised server instead to the legitimate one. The result will be successful attacks, as in the aforementioned cases of data in transit assaults ([6], [7]). Note that some versions of these attacks can be performed even if the TLS communication has been set correctly. Then, take as an indicative example, an application that utilizes linking data form the Web in order to determine if a citizen is categorized in a particular class, like

the terrorist group. A semantic attacker who manipulates one of the resources that is parsed by the inference engine (either in transit or at rest) could alter the terrorist definition. The OWL/RDF triples which denote that the crime must be political by virtue of a certain government-approved definition, are erased. Then, any person who exhibits a less important deviating behaviour (e.g. violation of the road traffic code, robbery, etc.) would be erroneously categorized as a terrorist. If more triples are deleted, any citizen could be denoted as a terrorist by this correctly functioning inference procedure, due to the utilization of data with poor quality.

Therefore, as the Semantic Web reasoning is based on collecting and integrating trusted data across the Internet, the whole information retrieval infrastructure must deploy TLS for every involved URI. If triples are originated from Web-level protocols, the protocols must also utilize TLS and retain their security properties. Moreover, in order to protect the inference engine that gathers linked data from the Web for the data in rest attacks, we need to authenticate the legitimacy of the information before proceeding to further processing. For these purposes, the IETF standard JSON Web Signature (JWS) has been proposed [10]. The information that is contained in JSON files is signed and the consumer can verify the source's authenticity along with the integrity of the received data. The full framework, called JSON Object Signing and Encryption (JOSE) [11], can also support encryption for confidentiality.

For the proposed SMs, except from deploying TLS and signing data, we mainly utilize JWS to sign the TDs that are processed and validate the trustworthiness of the reported transformation rules. As these rules are pieces of code that will be executed by the legitimate system for accomplishing ontology alignment, they have to be also inspected by the system operator prior their integration to the knowledge base. Otherwise, the SM could be vulnerable to code injection attacks. Nevertheless, this is not considered a significant burden for the operator as it is done only once, when a new or updated TD is parsed. Then, the runtime interoperation of the various components is done automatically.

IV. Semantic Mediators

This section details the operation of the semantic mediators. The SMs utilize the Yet Another Next Generation (YANG) data model in order to map the data in a common format (i.e. JSON). For ontology alignment, they retrieve transformation rules from the related signed JSON-LD files and then perform the rules as regular expressions in Perl.

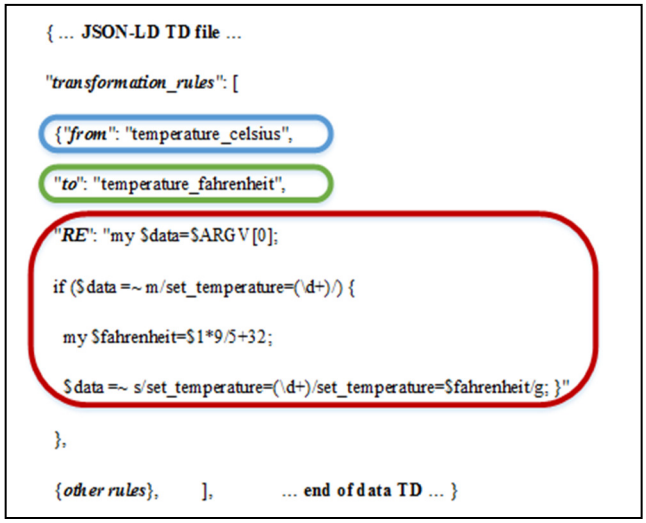
A. Data Mapping

The YANG data language is defined in the RFC 7950. It is a current programming trend and facilitates the deployment of new applications in various platforms. For this purpose, YANG supports the NETCONF and RESTCONF interfaces for the deployment of network and RESTful services, respectively. The service operations are modelled in YANG. Then, the YANG processor parses the model and exports the abstract development project in a denoted programming language (e.g. JAVA, C/C++, etc.). For our motivating example below, we utilize these features in order to deploy the smart functionality that collects, processes, and transmits the sensed information. More specifically, we exploit the RESTCONF and implement RESTful web services that run in the field and backend systems. RESTCONF is defined in the RFC 8040.

Thereafter, we additionally exploit YANG to establish a common data mapping between the involved operations. The interfaces can process messages with semantic information. At the design phase, we have described the structure of these messages in YANG (e.g. get current temperature value from a sensor). Then, at runtime, we can transform XML messages into JSON ones and vice versa, according to the specific format which is supported by each interface. The mapping is accomplished via the IEFT Internet Draft *draft-ietf-netmod-yang-json*, which establishes a one-to-one mapping between JSON and the subset of XML that can be modelled by YANG. The overall functionality is also tailored in order to cooperate with legacy formats, as in the IoT domain there could be several constrained devices, like motes/sensors, that do not process structured data.

B. Ontology Alignment & Semantic Reasoning

After we have achieved the common format, the next step is to resolve semantic conflicts and perform ontology alignment between the interacting domains. Thus, we need **transformation rules** that describe how we can transform data that are processed by one application into a compatible form which is understandable by another machine.



For the proposed SM components, the rules are modelled as specific JSON tags that are included in the related TD/JSON-LD files. Each rule tag contains the identification of the two domains (from-to) and a **Regular Expression (RE)**. The RE is a valid PERL program that models the search pattern (for matching the data to be altered) and the transformation formula itself (how the data will be changed). For example, the next TD sample transforms the temperature value from the Celsius to the Fahrenheit scale. Once parsed to the inference engine, the rule takes as input the JSON-LD file from a FIWARE's *set_temperature* service, searches for the temperature value, and changes it to the other scale. The expressiveness of this RE type is even more advance than just performing a single mathematic formula. REs can perform complex transformations and successfully resolve conflicts that occur by the incorrect OWL correlations [21].

V. Implementation

A. Motivating Example – Smart Sensing

As a motivating example, we consider a smart sensing scenario, where a smart building deploys several sensing equipment in order to support pervasive and ubiquitous functionality. Energy management is such a popular service.

Horizontal operation in the field system is mandatory as well as vertical cooperation with the backend. The Customer Energy Manager (CEM) is a logical function for optimizing energy consumption and can be deployed either in the home gateway and/or in the cloud. The interoperability of the underlying IoT devices and the CEM service must be guaranteed regardless their brand or manufacturer. The user should be able to buy and install any smart device while retaining the full functionality of the integrated system.

As an indicative scenario, we consider the case where the user installs temperature sensors in the rooms. Three types of sensory devices are modelled: the first one is bought from a European manufacturer – measures the temperature in the Celsius scale (°C) and transmits data in an XML format; the second one is bought from USA – measures the temperature in the Fahrenheit scale (°F) and transmits JSON messages; and the third sensor is compatible with the FIWARE's semantics – measures the temperature in °C and transmits JSON messages. Then, we model two reasoning processes where the system collects data and takes runtime decisions.

Edge reasoning: The CEM functionality that runs in the local gateway must retain a specified temperature value in the building. At first, the SM component in the gateway maps all gathered data in a common format and aligns all semantics in the SEMIOTICS schema (iot.schema.org). Thus, the temperature information is stored in JSON and in the Celsius scale. If the temperature in a room goes beyond a threshold, the relevant fan is adjusted accordingly.

Backend reasoning: If one device is damaged or malfunctioning, the deductive capabilities of JSON-LD are utilized to search for a technician who can fix it. Thus, the equipment descriptions and the technicians' expertise are collected via Internet. The information is stored in the CEM cloud service (i.e. SPARQL-LD) and the SM's ontology alignment can be performed if it is required.

The SMs are deployed in the gateway and the cloud to ensure common data mappings and ontology alignment. They also include a local repository for maintaining TDs and sensed data. Then, semantic reasoning can be performed, i.e. with SPARQL/SPARQL-LD. Fig. 2 depicts this scenario. From bottom-up, we consider 3 main data flows that implement the abovementioned functionality. The first data flow includes the local communication of the interconnected devices at the edge system. The devices can interact directly (if they are compatible) or indirectly through a gateway. If it is required, the gateway also performs the SM services, applying common data mappings and semantics. In the second setting, the devices or a gateway application interplay with the backend. Here again, the gateway can execute the SM services for semantic interoperability. In cases where the communication between the field and the backend (flow 2) must be encrypted, the SM functionality is performed in the cloud by the end-point that decrypts and processes the data.

B. Performance Evaluation & Comparison

A preliminary version of the proposed setting is implemented. We deploy two different embedded platforms that emulate the smart sensing equipment, consisting of Zolertia Z1 motes and BeagleBone nodes. Two Z1 transmit XML/°C messages with 6LoWPAN, a BeagleBone sends JSON/°F data over Ethernet, while another BeagleBone exchanges JSON/°C information via USB-WiFi. A laptop acts as the local gateway that gathers data from the edge

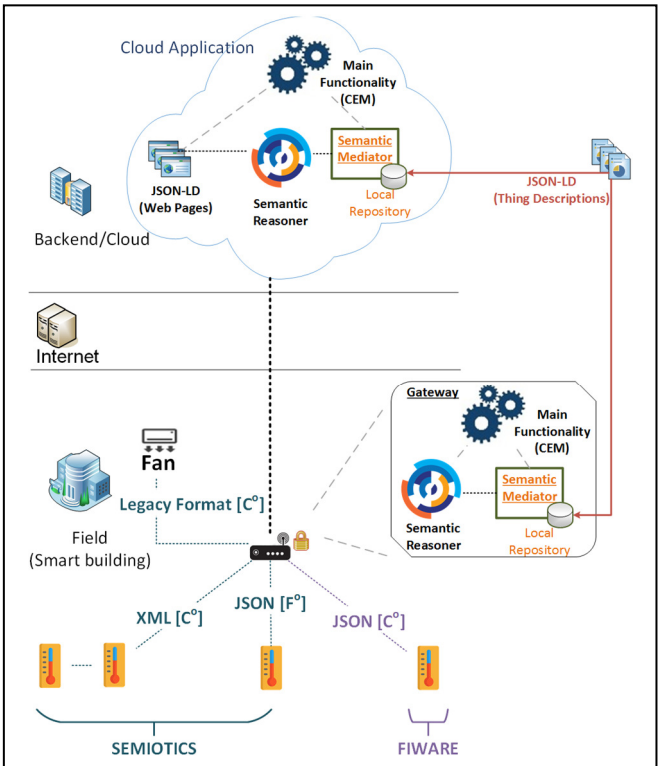


Fig. 2. The smart sensing interoperability scenario.

system. It also runs the SM service and emulates a fan device that exchanges legacy-formatted/°C messages. A similar virtual machine runs the backend SM in the cloud platform Proxmox along with end-user services.

We measure the performance of the SM component in the laptop (the cloud version performed similarly). For the initialization process, 100 TDs are parsed by the mediator (for the sake of this testbed, the operators consent is gained automatically, without the manual inspection of the transformation rules). Then, the devices sent totally 100 sensed messages, requiring common data mapping and ontology alignment. Table I details the average evaluation results. As is evidence, the overall overhead of the SM is adequate for real-time applications which will have to process many messages simultaneously.

TABLE I. PERFORMANCE EVALUATION FOR SM

Operation	***CPU (ms)***	***RAM (KB)***
Security		
Sign JSON-LD (RSA-2048)	0.3	7.0
TLS	10.7	11.5
Verify signature (offline)	0.2	0.5
Initialization		
Signature verification of TD	0.2	0.5
Processing of TD and extraction of transformation rules	0.1	0,2
Intermediate data mapping models (Yang)	800	32,658
Total resource consumption	**800.3**	**33.358**
Runtime Processing		
TLS+Signature verification	10.9	12
Data mapping (Yang)	800	32.658

Operation	*CPU (ms)*	*RAM (KB)*
Ontology alignment (Perl execution)	4.7	452
Maximum resource consumption	**815.6**	**33.122**

Table II summarizes the main features that are provided by the proposed SM and the related SIB [22] and IIF [23]. In general, SIB/IIF are suitable for RDF/OWL ecosystems while SM also exploits the capabilities of the modern LD approach. The expressive power of the SM/REs is far more advance than a specific notification schema that must be supported by all entities, as proposed by the related works. Thus, SM resolves a high variety of semantic conflicts, including the four main cases of erroneous OWL usage [21], and offers a general ontology alignment approach. Moreover, security and trust are considered, with the validity of TDs and semantic data being verified before further processing.

TABLE II. COMPARISON

Feature	*SM*	*SIB [22]*	*IIF [23]*
Architecture	Distributed	Distributed	Distributed
Data formats	Legacy, XML, JSON	XML	XML
Data mapping	JSON-LD	RDF/OWL	RDF/OWL
Ontology alignment	Regular Expression	Own physical objects notifications	OWL extensions
Reasoning engine	SPARQL-LD	SPARQL	SPARQL
Semantic Security	TLS and JOSE/JWS	No	No

VI. CONCLUSION

This article presents the landscape for semantic interoperability in the IoT. To do so, the state-of-the-art approaches are reviewed, including technologies for semantics, data mappings, ontologies alignment, semantic reasoning, etc. The main outcome is the proposal of the Semantic Mediator (SM) component which can be deployed across the various IoT layers (field, network, backend) and provide the required common representation and meaning of data. The platform integration of 2 EU funded IoT initiatives (SEMIOTICS and FIWARE) is described in a smart temperature sensing scenario. This includes the appliance of the various interoperability methods from the field to the backend. The overall deployment is scalable and sufficient for real-time operation. In comparison with related settings, the SM retains security and exhibits more advanced data mapping and ontology alignment capabilities. As future extension, the SMs' can be applied as privacy mediators, where the transformation rules anonymize or generalize the exchanged data and enhance the user's privateness.

ACKNOWLEDGMENT

This work has received funding from the European Union Horizon's 2020 research and innovation programme under the grant agreement No. 780315 (SEMIoTICS) and No. 786890 (THREAT-ARREST).

REFERENCES

[1] G. Hatzivasilis, I. Askoxylakis, G. Alexandris, and K. Fysarakism, "The Interoperability of Things," CAMAD, IEEE, Barcelona, Spain, 17-19 September, 2018, pp. 1-7.

[2] A. Haller, et al., "The SOSA/SSN ontology: a joint WeC and OGC standard specifying the semantics of sensors, observations, actuation, and sampling," Semantic Web, IOS Press, vol. 1-0X, 2018, pp. 1-19.

[3] D, Zeng, S. Guo, and Z. Cheng, "The Web of Things: a survey," Journal of Communications, Academy Publisher, vol. 6, no. 6, 2011, pp. 424-438.

[4] J. Soldatos, et al., "OpenIoT: Open source Internet-of-Things in the Cloud," Interoperability and Open-Source Solutions for the Internet of Things, Springer, LNCS, vol. 9001, 2015, pp. 13-25.

[5] M. Ganzha, et al., "Semantic interoperability in the Internet of Things, as overview from the INTER-IoT perspective," Journal of Network and Computer Applications, Elsevier, vol. 81, issue 1, 2017, pp. 111-124.

[6] V. Patel, R. Mohandas, and A. R. Pais, "Attacks on web services and mitigation schemes," SECRYPT, IEEE, Athens, Greece, July 26-28, 2010, pp. 499-504.

[7] V. R. Mouli and K. P. Jevitha, "Web services attacks and security – a systematic literature review," Procedia Computer Science, Elsevier, vol. 93, issue 1, 2016, pp. 870-877.

[8] H. Halpin, "Social Semantics – The search for meaning on the Web," Database Management & Information Retrieval, Springer, Semantic Web and Beyond Series, vol. 13, 2013, pp. 1-220.

[9] P. Fafalios, T. Yannakis, and Y. Tzitzikas, "Querying the Web of data with SPARQL-LD.," TPDL, Hannover, Germany, 5-9 September, Springer, LNCS, vol. 9819, 2016, pp. 175-187.

[10] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Signature (JWS)," IETF, RFC 7515, May 2015, pp. 1-59.

[11] K. Moriarty and S. Turner, "Javascript Object Signing and Encryption WG (JOSE)," IETF, version 02, 2013.

[12] I. Szilagyi and P. Wira, "Ontologies and Semantic Web for the Internet of Things – A survey," IECON, IEEE, Florence, Italy, 23-26 October 2016, pp. 6949-6954.

[13] C. Scovotti and S. K. Jones, "From Web 2.0 to Web 3.0: implications for advertizing courses," Journal of Advertising Education, vol. 15, issue 1, 2011, pp. 6-15.

[14] J. D. Cameron, A. Ramaprasad, and T. Syn, "An ontology of mHealth," AMCIS, Puerto Rico, 2015, pp. 1-11.

[15] G. Bajaj, et al., "A study of existing ontologies in the IoT-domain," HAL Archives, hal-01556256, 2017, pp. 1-24.

[16] L. Daniele, F. den Hartog, and J. Roes, "Created in close interaction with the industry: the smart appliances reference (SAREF) ontology," FOMI, Springer, LNBIP, vol. 225, 2015, pp. 100-112.

[17] K. Nitta and I. Savnik, "Survey of RDF storage managers," DBKDA, IARIA, Chamonix, France, 20-25 April, 2014, pp. 148-153.

[18] M. Morsey et al., "Usage-centric benchmarking of RDF triple stores," AAAI Conference on Artificial Intelligence, 2012, pp. 2134-2140.

[19] M. T. Ozsu, "A survey of RDF data management systems," Frontiers of Computer Science, Springer, vol. 10, issue 3, 2016, pp. 418-432.

[20] E. Prud'hommeaux and A. Seaborne, "SPARQL query language for RDF," W3C Recommendation, 15 January, 2008.

[21] H. Halpin, et al., "When owl:sameAs isn't the same: an analysis of identity in Linked Data," ISWC, Springer, LNCS, vol. 6496, 2010, pp. 305-320.

[22] J. Kiljander, et al., "Semantic interoperability architecture for pervasive computing and Internet of Things," IEEE Access, IEEE, vol. 2, 2014, pp. 856-873.

[23] C. Doulaverakis, et al., "An approach to intelligent information fusion in sensor saturated urban environments," EISIC, IEEE, Athens, Greece, 12-14 September, 2011, pp. 108-115.

[24] A. Bröring, et al., "Semantic mediation on the sensor web," IGARSS, IEEE, Germany, 2012, pp. 2910-2913.

[25] A. Bröring, et al., "Semantically-enabled sensor plug & play for the sensor web," Sensors, MDPI, vol. 11, issue 8, 2011, pp. 7568-7605.

[26] H. Halpin, "Semantic insecurity: security and the Semantic Web," PrivOn, co-located with ISWC, Vienna Austria, 2017, pp. 1-10.

[27] A. Medic and A. Golubovic, "Making secure semantic web," Universal Journal of Computer Science and Engineering Technology, vol. 1, issue 2, 2010, pp. 99-104.

[28] B. Thuraisingham, Security standards for the semantic web, Computer Standards & Interfaces, Elsevier, vol. 27, issue 3, 2005, pp. 257-268.