Developing and Integrating a Semantic Interoperability Testing Tool in F-Interop Platform

Soumya Kanti Datta, Christian Bonnet
EURECOM
Sophia Antipolis, France
{dattas, bonnet}@eurecom.fr

Hamza Baqa, Mengxuan Zhao, Franck Le-Gall
Easy Global Market
Sophia Antipolis, France
{hamza.baqa, mengxuan.zhao, franck.le-gall}@eglobalmark.com

***Abstract*—The full potential of the Internet of Things (IoT) is challenged by heterogeneous IoT data sources, data formats, and fragmented IoT ecosystems. Semantic interoperability is identified as a key to address these challenges. But majority of the current IoT ecosystems lack any tool to verify if two IoT platforms are semantically interoperable. This paper proposes a semantic interoperability testing tool called SemTest. It performs conformance and interoperability tests to ensure whether two IoT systems under test (SUT) are semantically interoperable. The architecture of each testing methodology is presented along with technical discussion on the tool development. A major contribution of the paper is to integrate the SemTest tool into the F-Interop platform which aims at online conformance, interoperability, and performance tests for IoT. Feedbacks from the semantic web technology experts from IETF and W3C communities highlight that the proposed tool is novel, timely, and will have a strong impact across the IoT ecosystems.**

***Keywords*-F-Interop Platform; Internet of Things; Semantic Interoperability; Testing.**

I. Introduction

The Internet of Things (IoT) provides useful services in home automation, health care, transportation and other domains. Despite several successful pilots from the EU H2020 projects like BIG-IoT[1], Autopilot[2] showcasing the benefits of the IoT, its adoption has not yet reached the full potential. This is due to fragmentation at the IoT platforms, data exchange models, security, and privacy challenges. Given this context, semantic interoperability has been identified as a key to harmonize the IoT ecosystems.

Semantic Interoperability refers to the ability of two or more IoT systems to automatically interpret the meaning of high-level information communicated to them and arrive at equivalent meanings [1]. Utilization of semantic web technologies allow (i) the IoT devices to exchange machine-readable data, (ii) uniform annotations, (iii) easy service discovery, and IoT data processing. They in turn pave way for settling the heterogeneity of IoT data sources, data models, data formats, and generation of equivalent high-level information from IoT data processing [2]. These are foundations that lead to semantic interoperability in the IoT ecosystems.

Few academic research initiatives have considered developing a tool to examine semantic interoperability between two IoT systems [3]. The primary focus of this work is to develop a semantic interoperability testing tool (called SemTest) and integrate it onto the F-Interop platform[3]. It offers online conformance, interoperability, and performance tests for the IoT. SemTest extends the current capabilities of the platform by offering two specific types of semantic testing to the platform users - conformance testing and interoperability testing. In addition to that, the proposed tool can benefit the W3C Web of Things (WoT)[4] Working Group as well. WoT aims at solving the IoT interoperability puzzle and uses semantic web technologies in its Thing Description [4]. But the group currently does not provide any guidelines on testing WoT implementations. Therefore, SemTest significantly advances the current state of both the F-Interop and W3C WoT.

To develop the testing tool, we have identified the requirements of semantic tests [3]:

1) **Conformance test** - It inspects if a piece of semantic data is conforming to a reference ontology using three checks:
   a) **Lexical check** for validating the textual serialization (i.e. RDF/XML) of the semantic data.
   b) **Syntactic check** for finding the errors in the semantic data such as untyped resources and literals, ill-formed URIs, problematic prefix and namespaces, unknown classes and properties.
   c) **Semantic check** for finding the logic inconsistency in semantic data after a successful syntactic check, such as cardinality in consistency, problematic relationship or inheritance.
2) **Interoperability test** - It examines if two IoT SUTs understand correctly the meaning of exchanged semantic data using three checks:
   a) **Communication level check** to validate the correct reception of messages.
   b) **Lexical/format level check** to validate the format serialization/de-serialization of the exchanged messages between two systems.
   c) **Data processing level check** to determine whether the two systems understand the data in the same way.

[1]http://big-iot.eu/
[2]http://autopilot-project.eu/
[3]https://www.f-interop.eu
[4]https://www.w3.org/WoT/

One conformance test scenario and two interoperability test scenarios are developed to cover the above aspects using the SemTest tool integrated onto the F-Interop platform.

The remainder of the paper is structured as follows. Section II describes an overview of the F-Interop platform. In Section III, technical details of the development and integration of SemTest onto the F-Interop platform are presented. Section IV covers the execution of testing scenarios and validation of SemTest integration in F-Interop platform. Section V outlines the feedback received from IETF and W3C community experts on the tool requirements and testing scenarios. Section VI concludes the paper.

II. Overview of F-Interop Platform

This section gives an overview of F-Interop Platform [5], its functional architecture, and remote interoperability tests. The platform provides an open framework for online interoperability and performance tests for the IoT ecosystems [6] for IoT. In addition to that, its scope includes conformance testing, scalability testing, QoS, and QoE testing. The high-level architecture is portrayed in Fig. 1.

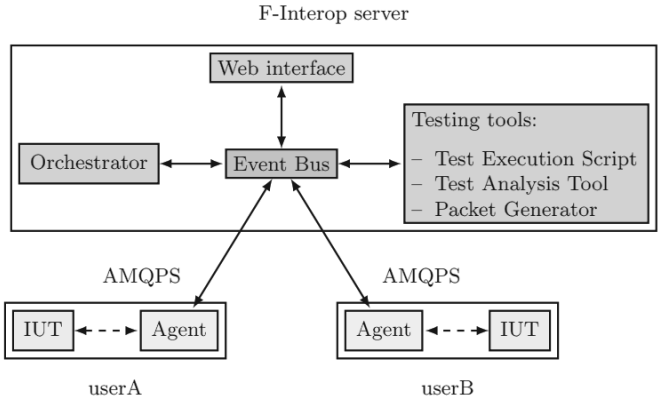

Fig. 1. F-Interop architecture [7].

The above architecture components (e.g. orchestrator, testing tools) exchange messages amongst themselves through a central event bus. It is responsible for control messages, raw data packets, and software log communication. A unique aspect of this architecture is modularity and scalability. The component agent is provided by the platform and allows an user to securely connect an Implementation Under Test (IUT) to the server. The orchestrator has administrative rights on the overall server. It monitors the connected users, provisions message broker, and updates firewall rules. Since F-Interop provides a testing environment, there is a test execution script. It contains a machine understandable code about the test configurations and testing steps. The test analysis tool performs verification of traces during a test. To support wide range of protocols for testing, the server provides such tool for several protocols. The third component of the testing tools is a packet generator. It can be used to generate packets of a desired protocol (e.g. CoAP) for an IUT. Finally, the web interface element allows users to select a test, start the test execution and obtain a report. Currently the F-Interop server supports CoAP based interoperability tests.

III. SemTest Implementation and Integration

In this section, we describe the proposed tool architecture, its components and their integration within the F-Interop platform environment.

A. Conformance Testing Implementation

The SemTest Module is deployed in the F-Interop platform which communicates with the user via the F-Interop GUI. It is the interface responsible for testing tool module to interact with F-Interop platform. Inside the SemTest module, a GUI Enabler is in charge of the communication with the F-Interop GUI which consists of receiving the semantic annotation to be validated and of sending the validation report once the validation finishes. The semantic data is sent to the ontology validator to perform the validation. The architecture for conformance testing is displayed in Fig. 2.

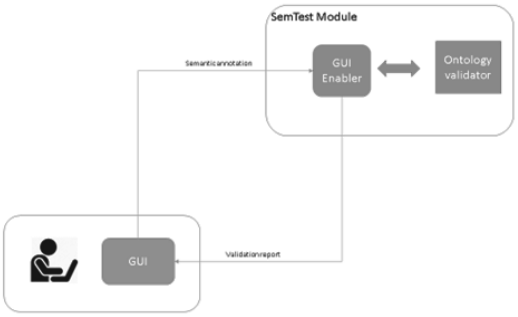

Fig. 2. Conformance testing architecture.

In the scenario of conformance testing, at first, the user needs to create a session and configure it. He/she will be guided during the all creation process. The operational steps for this procedure is depicted in Fig. 3 and described below.

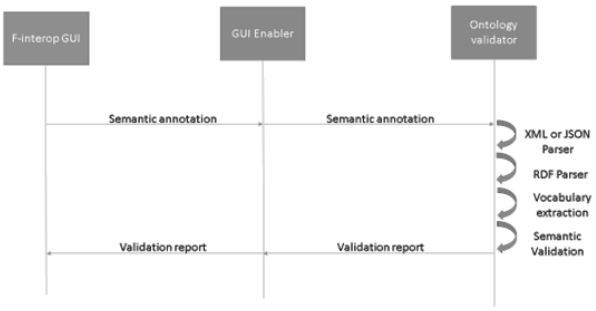

Fig. 3. Conformance testing sequence diagram.

1) **GUI Enabler**: A Python script is responsible for connecting the F-Interop GUI and the ontology validator

module. The GUI mainly asks the user to upload the semantic annotation which will be stored in a temporary folder and sent to Ontology Validator web service for validation. Following that, it captures the validation report and send it back to the user F-Interop GUI.

2) The Ontology Validator web service is composed of four main modules shown in Fig. 4.
   a) **XML and JSON Parsers**: If the IoT data annotation is indicated as XML or JSON along with the extension of the two formats (e.g. RDF/XML, OWL/XML, JSON-LD), the XML and JSON parser check if the syntax of XML or JSON is respected. If it is not validated, the validation process will not proceed further.
   b) **RDF Parser**: This module takes either a validated XML or JSON file or a file in another supported format as an input document. Then it verifies if the IoT data represents a valid RDF model. If it respects the specification of the RDF model, triples in this model are extracted to serve as the input for the next validation step in the validation module.
   c) **Vocabulary Extraction**: This module takes the semantic description as input and extract the vocabulary/prefixes contained in the semantic description.
   d) **Validation**: This module takes the reference ontology constructed from the different checked ontologies as an input of the validation and the triples extracted before. Then, according to the predefined reference ontology, it checks for syntactic errors in the testing document which is based on the functionalities implemented in Eyeball, an Apache Jena ontology validator. A reasoner is also used to enable logical level verification of the RDF document. The validation results sent to a reporting server shows a list of errors and explanations regarding the ontology affected elements. The error report is sent back to the Python script. It converts the report in a table format and shows that to F-Interop GUI.

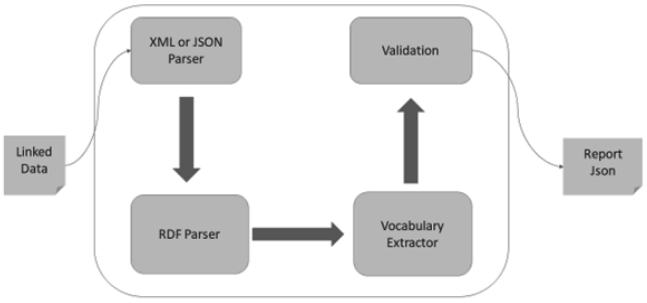

Fig. 4. Ontology validator web service architecture.

B. Interoperability Testing (User-to-User) Implementation

In this scenario, the user will test the interoperability of its semantic annotation against another semantic description provided by another user. As we have described in [3], we address this scenario by considering if two systems share the same vocabulary/prefixes, so they are then interoperable.

Regarding the integration, both users are invited to be connected to the same shared session (but they still have a different authentication credentials). A script inside the SemTest module will then ask the user via F-Interop platform to upload their annotations. These annotations will be stored in a temporary folder of the module and sent to the Ontology Validator web service. The web service will retrieve all the reference ontologies from each annotation and compare them. If the two annotations dont share any vocabulary (prefixes), the validation process stops and sends back a report to the script which will report in its turn the user that the two annotations are not interoperable. Otherwise, the ontology validator will perform a conformance validation (in the same manner as the conformance test scenario). The second step, a comparison server will take as an input the list of prefixes used in each validated annotation (A and B) and compare the vocabularies, and based on the percentage of shared vocabulary, the comparison server will decide if the two annotations are interoperable or not. The scenario is depicted in Fig. 5.

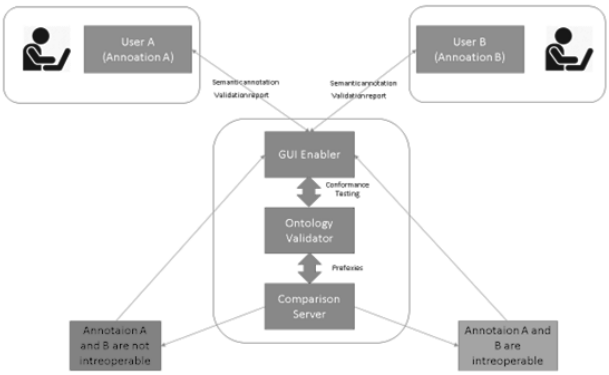

Fig. 5. User to user interoperability testing.

C. Interoperability Testing (SUT-to-SUT) Implementation

In this scenario, if two IoT systems respond in the same manner to a SPARQL query, they are interoperable. In our implementation, we choose Mobius servers (which provide oneM2M based IoT implementations able to interpret semantic queries) as the SUTs. As shown in Fig. 6, a query server (hosted in SemTest Module) will send a query to both oneM2M servers. The two SUTs which contain same semantic dataset return their responses to the query (R1 and R2) and the comparison server compares them to determine if they are the same. If R1 and R2 are the same, the two SUTs are considered to process the data in the same way, so they are interoperable.

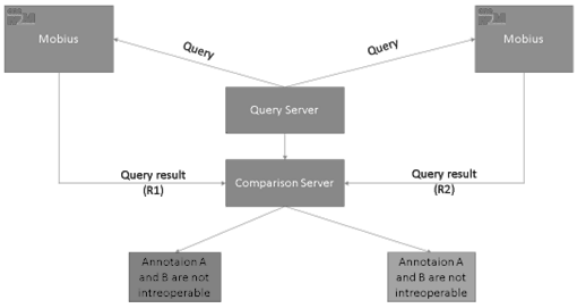

Fig. 6. SUT to SUT interoperability testing.

D. SemTest Integration in F-Interop Platform

The developed components of SemTest are packaged in Docker container images making them independent of the underlying infrastructure of deployment. SemTest can therefore run on its own as a microservice within the overall F-Interop Platform. The environment variables related to the F-Interop user session for testing are required. We provide a makefile to make the integration process easy and seamless. The makefile contains commands to automatically build the docker container, run the SemTest module at any desired port (e.g. 8001), and launch the implemented tests. The GUI Enabler will launch a different script for each of the three test scenario.

IV. Executing Test Scenarios and Validation of SemTest Integration

At first, the Semtest module needs to connect to the user GUI (using the session parameters available on session info section) and ask the user to upload an ontology or semantic description (shown in Fig. 7). To validate the above mentioned test scenarios, we take the example of smart parking semantic description. The parking.owl file is available on F-Interop github repository.

A. Conformance Testing

For this testing, the user needs to check Semantic testing checkbox while creating the session. Then, the user is redirected to the Web page (shown in Fig. 7) to upload the semantic description (e.g. parking.owl in this case). The user needs to push the button ”Send” for the validation. The SemTest module will then receive the user ontology and save it in temporary file during the validation process shown in Fig. 8.

The GUI enabler component takes the users input and send it to the ontology validator web service for the conformance testing. A report will be generated and sent back to the user (shown in Fig. 9).

If the validation results in errors, SemTest logs the errors in its backend as depicted in Fig. 10. In this example, the conformance test detects and lists many errors such as bad URI and class not declared.

Fig. 7. Uploading semantic description.

Fig. 8. SemTest backend logs for conformance testing.

B. Interoperability Testing

It consists of both the user to user and SUT to SUT testing scenarios. To execute such a test, the users need to check the user-user semantic testing checkbox while creating the session. Lets consider two users hamza_baqapuic and tt7874498hbkc running the interoperability testing (we assume both the users have authentication credentials for the F-Interop platform). The user hamza_baqapuic needs first to create a user-user interoperability session and both the users need to share the session for the execution of the test scenario.

After both users have submitted their semantic data, a validation report will be send back to both users in the end of the interoperability validation process. It shows vocabularies used in the semantic description from each of the users. Thus, a percentage of the interoperability is calculated by SemTest and shown as a part of the report (Fig. 11).

C. Novel Aspects

The primary novel aspect of this work is the SemTest tool itself. During our extensive literature review on semantic interoperability presented in [3], we found few tools providing the such testing scopes. Conformance testing along with two interoperability testing provisions are considered as progress beyond the current state-of-the-art. In addition to that, we successfully integrate the SemTest module on F-Interop platform and validate the integration. This extends the current capabilities of the platform.

V. Community Feedback on Semantic Interoperability Testing

We have disseminated the proposed conformance and interoperability testing requirements and methodologies during IETF 101 and W3C Web of Things PlugFest event in March 2018. It involved distributing a questionnaire[5]. The collected feedbacks have been analyzed and presented in this section.

A. Prior Semantic Interoperability Testbeds

55% of 22 responses in Fig. 12 point to the lack of testbeds providing semantic interoperability while two responses cite FIESTA-IoT project. While W3C WoT has been mentioned in the survey, but the WoT community still lack a sophisticated

[5]https://bit.ly/2wkyW8b

Fig. 9. Validation report for parking semantic description.

Fig. 10. Conformance testing errors log in SemTest backend.

testbed supporting the testings proposed in this project. This point highlights the high importance, timeliness, and novel aspects of the presented semantic interoperability testing tool.

B. Feedback on Conformance Test

On being queried where proposed lexical, syntactical and semantic checks are necessary for conformance testing, most of the responses were in favor of the checks (shown in Fig. 13). However, some additional checks were also suggested by the responders:

1) If the SUTs do not support RDF, other checks should be brainstormed.
2) Inference check - to check if two semantic systems have no rules excluding each other.
3) Checking that object APIs are consistent with the semantics they claim, verifying access to semantic mappings between data in different vocabularies.
4) Security parameters related checks should be a part of conformance test.
5) Tests that verify that a system doesn't break semantically/functionally when the underlying ontologies are modified.

C. Feedback on Semantic Interoperability Test

Similar to the previous aspect, our query on semantic interoperability testing highlights that most responses agree on communication level check, lexical/format level check and data processing level check (shown in Fig. 14). This aspect also received some additional suggestions from the responders:

1) For data processing level check, the result may not need to be **exactly** the same. Sometimes you may lose information / accuracy and be OK with it.
2) For data interpretation check semantic reasoning on the IoT data should be made possible as a part of this test.
3) Application level "reference system" goals can be achieved across diverse device and application ecosystems.

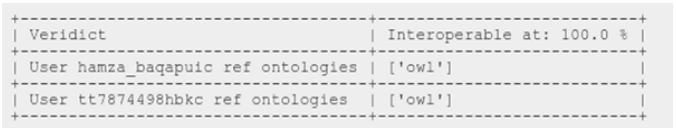

Fig. 11. Semantic Interoperability testing report.

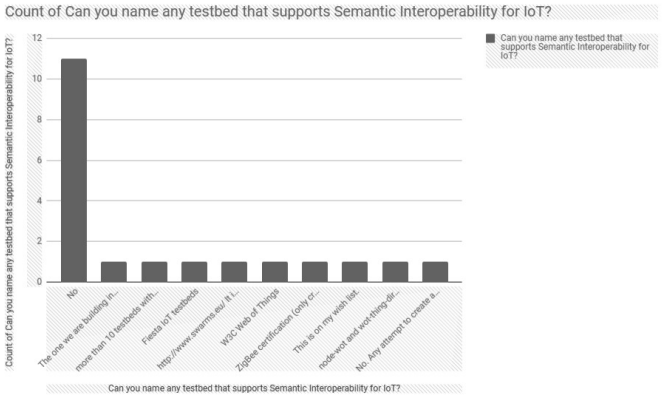

Fig. 12. Responses on prior semantic interoperability testbeds.

4) Other semantic data and runtime data are separated (values not in RDF), so the correct correllation between the two must be checked.
5) Checking that object APIs are consistent with the semantics they claim, verifying access to semantic mappings between data in different vocabularies.
6) Semantic check in light of adapting ontologies (Tests that verify that a system doesn't break semantically/-functionally when the underlying ontologies are modified).

D. Suggested Additional Requirements

Several responders suggested additional requirements that would broaden the scope of our proposed testing tool. These suggestions are briefed below:

1) Software Versions and code updates should be taken into account while performing the semantic interoperability testing.
2) With the introduction of General Data Protection Regulation[6], the EU has taken a significant step to ensure consumer privacy and protection of data (including IoT data). Therefore the tool should consider privacy and semantic interoperability on anonymized IoT data.
3) Individual systems may keep their own data format and model depending on business requirements. In this case, such IoT systems should introduce a middleware in order to harmonize formats and relationships with the testing tool.
4) Non-RDF based systems should also be able to benefit from the proposed testing tool.

[6]https://www.eugdpr.org/

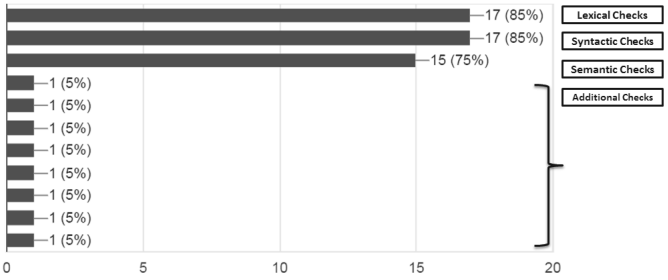

Fig. 13. Feedback on conformance test.

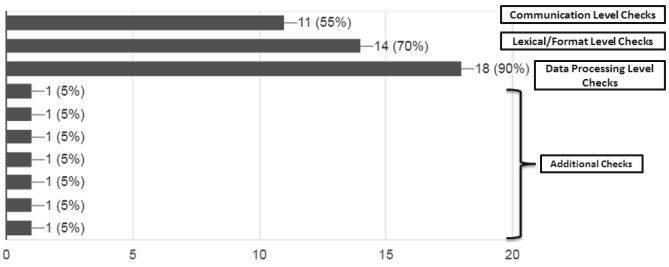

Fig. 14. Feedback on semantic interoperability testing.

5) As a best-practice guideline, suggestions should be given for IoT data models (ontologies), data types conversion, and normalization for global interoperability.
6) IoT data models should be interchangeable between the systems, meaning one can use a model from system A in system B and vice versa.
7) Inclusion of a human validation of common data shared across two SUTs to make sure they have the same meaning and not just the same name.
8) One of the biggest challenge is to relate different vocabularies with overlapping semantics. Without a means to map IoT data between such vocabularies, semantic interoperability won't be possible. Such mappings may be lossy, resulting in partial interoperability. Upper ontologies aren't practical except for certain restricted contexts. We thus need more work on direct transformations that can be conditioned upon the data.
9) Semantic interoperability needs to be tested in a full-stack scenario set where diverse applications are tested against IoT devices/sources with diverse data models and protocols.
10) It is important to highlight, given some premises, are two IoT systems drawing the same conclusion. This is a design requirement for the proposed tool.
11) Complexity of the overall testing scenario should be taken into account. Here, complexity is calculated in terms of how many vocabularies and ontologies must an IoT system load when inter-operating with another IoT system.

VI. Conclusion

This paper described the implementation and integration of the semantic interoperability testing tool (called SemTest) onto the F-Interop platform. The tool considers both conformance and interoperability tests. The architecture for one conformance test and two interoperability tests are presented and developed. The implementations and their integrations on F-Interop platform were tested, and validated by executing the semantic test scenarios introduced [3]. SemTest has been proven successful to determine whether a piece of semantic data is compliant with a reference ontology or two pieces of semantic data are interoperable. We have also collected feedbacks from semantic experts (in IETF and W3C) regarding to various aspects of SemTest. The result has turned to be positive on our proposal and established the timeliness, high impact and innovations outlined for the tool. We received additional suggestions as well which will be utilized to broaden the requirements, test scenarios and the testing capabilities of SemTest and F-Interop Platform.

Acknowledgment

This work is supported by F-Interop Project which has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No. 687884. EURECOM acknowledges the support of its industrial members, namely, BMW Group, IABG, Monaco Telecom, Orange, SAP, and Symantec. All authors have equal contribution to this paper.

References

[1] J. Strassner and W. W. Diab, "A semantic interoperability architecture for internet of things data sharing and computing," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 609–614, Dec 2016.

[2] M. I. Ali, P. Patel, S. K. Datta, and A. Gyrard, "Multi-layer cross domain reasoning over distributed autonomous iot applications," *Open Journal of Internet Of Things (OJIOT)*, vol. 3, no. 1, pp. 75–90, 2017. Special Issue: Proceedings of the International Workshop on Very Large Internet of Things (VLIoT 2017) in conjunction with the VLDB 2017 Conference in Munich, Germany.

[3] S. K. Datta, C. Bonnet, H. Baqa, M. Zhao, and F. Le Gall, "Approach for Semantic Interoperability Testing in Internet of Things," in *GIoTS 2018, Global IoT Summit 2018, 4-7 June 2018, Bilbao, Spain*, 2018.

[4] S. K. Datta and C. Bonnet, "Advances in web of things for IoT interoperability," in *ICCE-TW 2018, International Conference on Consumer Electronics, May 19-21, 2018, Taichung, Taiwan*, (Taichung, TAIWAN, PROVINCE OF CHINA), 05 2018.

[5] S. Ziegler, S. Fdida, T. Watteyne, and C. Viho, "F-Interop - Online Conformance, Interoperability and Performance Tests for the IoT," in *Conference on Interoperability in IoT (InterIoT)*, (Paris, France), Oct. 2016.

[6] E. E. Kim and S. Ziegler, "Towards an open framework of online interoperability and performance tests for the internet of things," in *2017 Global Internet of Things Summit (GIoTS)*, pp. 1–6, June 2017.

[7] R. Leone, F. Sismondi, T. Watteyne, and C. Viho, "Technical Overview of F-Interop," in *Conference on Interoperability in IoT (InterIoT)*, (Paris, France), Oct. 2016.