WoT Store: a Thing and Application Management Ecosystem for the W3C Web of Things

Luca Sciullo, Cristiano Aguzzi, Lorenzo Gigli, Luca Roffia, Angelo Trotta, Tullio Salmon Cinotti, Marco Di Felice
* Department of Computer Science and Engineering, University of Bologna, Italy
{luca.sciullo, cristiano.aguzzi, luca.roffia}@unibo.it, lorenzo.gigli@studio.unibo.it
{angelo.trotta5, tullio.salmoncinotti, marco.difelice3@unibo.it}@unibo.it

Abstract—In the next few years, the W3C Web of Things (WoT) platform will represent a reference solution toward the deployment of fully interoperable systems, hence unlocking the potential of the IoT paradigm on several use-cases characterized by the current fragmentation of devices and technologies. At the same time, the worlwide adoption of the W3C WoT architecture depends on many factors, including also the availability of support tools that might facilitate the deployment of novel WoT applications or the integration with traditional IoT systems. To this purpose, the paper presents the **WoT Store**, a complete software platform enabling the discovery and management of W3C Things, the monitoring of its properties and events, and the invoking of actions, all within the same dashboard. In addition, the platform leverages on the semantic description of each Thing with the goal of easing and automatizing the installation and execution of WoT applications, e.g. defining the behaviour of a Thing or implementing mash-up operations from multiple Things. We sketch the main features, the architecture and the prototypal implementation of the **WoT Store**. Moreover, we discuss the **WoT Store** capabilities on three IoT use-cases, i.e. industry 4.0, smart agriculture and home automation.

I. CONTEXT

Since the beginning, the Internet of Things (IoT) suffered from lacks of common interaction and communication paradigms that translated into the creation of several, independent technology silos, with severe concerns on the deployment of heterogeneous systems. The interoperability can be considered both a challenge and a market opportunity: according to [1], almost 40% of the value of IoT markets can be unlocked when achieving full interoperability among heterogeneous IoT systems. Among the several research projects and standardization initiatives proposed so far and addressing the IoT interoperability issue, a key role is played by the World Wide Web Consortium (W3C), i.e. the leader of Open Web platforms, and more specifically by the Web of Things (WoT) working group [2]; this latter has been/is working on the proposal of a platform-independent API and of service discovery mechanisms for the full IoT interoperability. Clearly, a worldwide acceptance of the W3C WoT platform is fundamental for the purpose of mitigating the fragmentation of WoT-related initiatives, and of the definition of a reference solution; on the opposite, there is always the risk of a new IoT technology silo. To this purpose, it is worth remarking that the software acceptance in industrial scenarios is largely influenced by the size and the quality of its Software Ecosystem (SECO). This latter can be defined as the set of software solutions that enable, support, and automatize the activities and transactions by the actors in an associated social or business scenario. As reported in [3], several studies in the literature investigate the success of a software solution and its SECO as vehicle to attract new players. The availability of supporting software modules is acknowledged as the key success of any platform in [4]. Furthermore, the availability of a large SECO can translate into several other benefits, like the cooperation and the knowledge sharing among multiple and independent software vendors and the cost reduction in the software development [5].

We believe that the presence of well-defined and complete support tools can become a turning point toward the large-scale adoption of W3C WoT solutions. This paper provides a significant contribution in this direction, by presenting the **WoT Store**, a software platform that ease the management of Things and the deployment of WoT applications on top. The **WoT Store** can be considered a component of the W3C WoT SECO, as well as an enabler of novel services and applications for the W3C WoT SECO, hence contributing to the further dissemination and utilization of the W3C WoT standards. We would like to highlight that the contribution presented in this study is the culmination of the research activities conducted by our research lab in the field of IoT interoperability platforms and the WoT. In 2017, we joint the W3C Face to Face meeting (Dusseldorf, Airport Conference Center) and discussed how to support the emerging WoT platforms via the Dynamic Linked Data and the SEPA technology [6]. This latter is a semantic broker enabling the detection and the notification of changes over Linked Data endpoints by means of content-based publish-subscribe mechanisms. The **WoT Store** exploits some functionalities of the SEPA tool for the Thing discovery, while adding a suite of novel sotware modules for the Thing management, the WoT application management and the IoT data management.

The paper is organized as follows. Section II describes the features, architecture and implementation technologies of the **WoT Store**, while section III discusses its suitability on three different application domains, i.e. the Industry 4.0, Smart Agriculture and Home Automation. Finally, open issues and current extensions are discussed in Section IV.

II. The WoT Store

As previously mentioned, the W3C Web of Things (WoT) standards constitute a novel and effective approach to enable the seamless integration of heterogeneous IoT components, abstracting from their implementation [2]. Several IoT markets, traditionally characterized by the fragmentation of the protocols and communication technologies, might greatly benefit from the definition of a reference architecture providing a way to describe, in a non-ambiguous way, the interfaces and the interaction patterns of the IoT components [7] [8] [9] [10]. At the same time, the goal of our research was not to target a novel application scenario of the W3C WoT, rather to leverage on the novel opportunities offered by a world of W3C Web Things, being them native (i.e. implementing the architecture defined in [2] from scratch) or adapted from traditional deployments (e.g. via a proxy). More specifically, we addressed the following key questions: (*i*) how to ease the discovery and the management of W3C Things, by supporting both private (i.e. only visible within a local network environment) and public (i.e. reachable by the whole Internet) environments? The WoT management means general-purpose functionalities like for instance: find the Things satisfying specific requirements (e.g. location), perform actions on them, display property values, etc that we expect to be present in any W3C WoT deployment, regardless of the use-case. (*ii*) given that the Thing Descriptor (TD) provides a computer-readable interface of the behaviour of a Thing and of the way to interact with it, how can we ease and even automatize the deployment and execution on WoT applications composed by multiple, heterogeneous Things?

We addressed the questions above in the WoT Store, a generic software platform for the management of W3C-compliant Things and applications of the WoT SECO. A prototype version of the WoT Store has been presented in [11]. The overall framework is designed to be highly modular; different modules can be loaded/unloaded or customized for specific application use-cases, as we discuss further in Section III. We present the main modules in the Section below, while the architectural and implementation details are sketched in Section II-B and II-C, respectively.

A. Features

The features offered by the WoT Store platform can be grouped in three main modules:

1) Things Manager: The first module allow users to handle and manage all the Things deployed in the environment. It works as a dashboard, i.e. users can visualize and monitor the Things status and properties. Moreover, it provides basic mechanisms to interact with the Things, by invoking, for instance, actions with custom inputs. Finally, it integrates discovery mechanisms so that new Things can dynamically be loaded on the WoT Store as soon as they are deployed.

2) Applications manager: The second module supports the installation and execution of IoT Applications, by considering two classes of software, i.e.: Things Applications (TA) and Mash-up Applications (MA), as better detailed in [11]. The WoT Store mimics the operations of a mobile application markets, since users can download and install the applications on their Things with minimal manual efforts; in addition, there are sharing mechanisms that allow developers to upload the code, revise it and leave line comments. Each application is provided with a semantic description, hence users can issue semantic queries in order to find the perfect match between their needs, the available Things exposed by the Thing Manager and the applications available on the store. Finally, the module offers the possibility to run MAs directly on the cloud, so that no computational resources are requested on the client side.

3) Data manager: The third module includes functionalities for the visualization, processing and manipulation of the data produced by the deployed Things. More in details, the WoT Store provides data filtering mechanisms according to the data origin, i.e, the producer Things, and additional components to plot the data on the dashboard. Different data flows can be aggregated by installing the proper MA. We plan to include data-analytics sub-modules as future work.

B. Architecture

The overall architecture of the WoT Store is shown in Figure 1. We briefly introduce each component in the following. On the left part of the Figure there are the interfaces allowing the user to interact with the platform:

- *Market Interface* (MI): this consists of a Web application offering the main features of the store previously listed in Section II-A. The component communicates directly with: the Market Service (MS) for all the resource operations and the real-time notifications; the available Things (rendering them and displaying the property values); the MQTT Broker, which is used as collector of messages related to events triggered by the Things.
- *Thing CLI* (TC): this is constituted by a command line tool that helps the configuration and then the online publication of a Thing, or of a group of Things. The users are guided during the deployment of their application scripts and the definition of the semantic metadata. At the end of the process, each new Thing is registered on the Discovery Service and on the Market Service.

The back-end of the WoT Store platform consists of two main APIs services:

- *Market Service* (MS): this consists of a REST API enabling the interaction with the main system resources, such as the Applications, Things, Users and so on. CRUD operations are implemented for each category of resource (and several other resource-specific calls). It also exposes WebSockets for real-time data retrieval and updates.
- *Discovery Service* (DS): this service monitors the status of each Thing. Whenever a change occurs (voluntary or not), the DS keeps track of the new state, then notifies the MS which updates the Thing Description (TD) correspondingly.
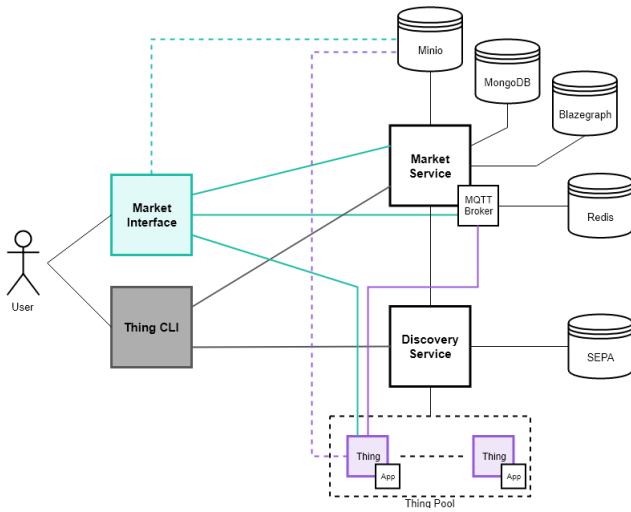
Fig. 1. The WoT Store Architecture

Fig. 2. The WoT Store MI - view of a Thing (actions)

C. Implementation

The MS has been implemented as a `Node.js` v10.x application using the `LoopBack` v3 framework and a MQTT Broker built with the `Mosca` library. The current deployment includes four types of databases/storage:

- `Minio`: this is an object storage server used to store the applications source code.
- `MongoDB`: this database contains user data, relationships between resources and other utility models.
- `Blazegraph`: this triplestore contains the application metadata and the TDs.
- `Redis`: this is a high performance in-memory database, required for the real-time processing of MQTT messages.

The MI has been implemented as a an `Angular` v6.x Web application composed of several modules. Among the various libraries, we find `@angular/material` for quality Material Design elements, `ngx-mqtt` and `socket.io-client` for communication. TC is another Node.js application that uses packages like `chalk`, `clui` and `inquirer` to build the interactive terminal; `request` and `shelljs` modules are employed in order to manage the HTTP calls.

The WoT Store assumes a Servient to be installed on devices. Currently the only W3C-supported and guaranteed implementation of the Servient is the one by the Eclipse Foundation (Eclipse `Thingweb node-wot` [12]). The Node.js implementation requires some additional dependencies to be executed: `Python` 2.7, `make` and a `C/C++` compiler. Our TC downloads the Servient directly from the repository, configures it according to user preferences and runs it by passing the application scripts as arguments.

III. USE CASES

The WoT Store can be considered a component and an enabler of the W3C WoT SECO; the main goal is to ease the deployment of heterogeneous systems, with advantages in terms of time and costs reduction. The application scenarios
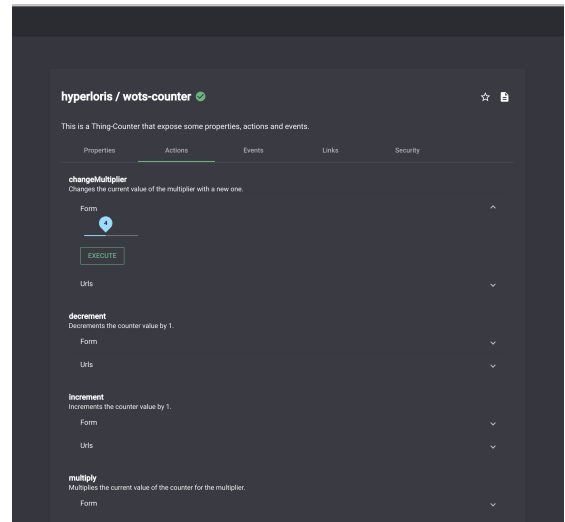
of the WoT Store are numerous, and comprises several IoT markets active so far. We discuss three of them, with explicit references to the research projects which see the participation of the authors [13].

A. Industry 4.0

The WoT Store supports the management of large number of heterogeneous devices via a common dashboard. For this reason, one of the main application scenario is constituted by Industry 4.0 environments characterized by the large-scale installation of sensors, and by the need to process vast amount of sensor data in order to build the digital twin of a physical component. We can assume that each sensor is represented by a Thing and runs a default factory application that publishes itself on the DS. Once Things have been deployed, it is possible to display them on the main console of the WoT Store thanks to the Discovery module, and take advantage of the functionalities provided by our tool, like for instance:

- *Update-all*: this feature allows to replace the TA currently deployed on a set of Things with a single action. Let us imagine a situation where a bug is discovered on the current TA, and hence a patch must be applied on all the Things. Without the WoT Store, this would require a manual re-configuration of all the devices, with a clear impact in terms of time and cost. The WoT Store includes a remote, secure mechanism to dynamically change the TA code currently executed on a Thing; moreover, the users can issue a semantic query in order to update a set of Things all satisfying the same query conditions (e.g. "update all sensor devices of type temperature available in room 262").
- *Mashup Application*: after Things have started their TAs, data should be collected and aggregated according to user-defined policies. For this reason, directly from the console of WoT Store, technicians can write and run a custom MA for Things querying and data processing.

- *Data analysis*: the WoT Store dashboard includes built-in mechanisms for plotting the data produced by set of Things and analyzing the aggregated results.

B. Smart Agriculture

Smart Agriculture leverages on IoT solutions to implement the so called Third Green Revolution. In this context, ICT technologies face unique challenges like, among others: the lack of stable power supply, the need of calibration procedures customized for every type of soil, the security concerns from farmers, just to name a few [13]. The role of the WoT Store is similar to the Industry 4.0 use case, since our platform can be considered an open ecosystem where different stakeholders can interact to effectually collect and process data from crops. Let us consider for example the digitization of wheat production in farms. Commonly, various sensors/actuators are deployed to enhance the productivity and the economical gain of the farm: soil moisture sensors, drones, automatic sprinklers and cameras. The deployment of the technological infrastructure consists in the major effort from both technicians and farmers. Moreover, in most of the crop types, the process should be repeated every year. In this scenario, the Thing Manager of the WoT store can be used to ease the installation process and automatize most of the configuration phases. Furthermore, using the data flow analysis, agronomists or soil experts can create a calibration models for soil moisture sensors to better fit the real volumetric water content. Later, the model could be uploaded to the system as a MA or the Thing Manager can be used to modify directly the sensor parameters (i.e. invoking an action with the new calibration curve or changing a conversion parameter). The open platform can be also exploited by IT companies specialized in data mining techniques, IA or agronomic models. Such software applications can be offered as MA applications and installed in a farm system with a simple add on. Other functionalities of the WoT Store can be used, similarly to the previous case:

- *Update-all*: Industry farmers need to keep TA up-to-date in order to avoid data losses and discontinuous services.
- *MA deployment*: customization is the key element of every Smart Agriculture application, since crops are not always the same and several variables should be taken in to account, like soil, weather, geographical position, irrigation and cultivation methods. MAs can be deployed and instantiated hence helping the customization process even by non ICT experts.
- *Data analysis*: Using data analysis tools embedded in the platform we can support the decision makers in the customization process described above.

C. Home automation

The current IoT market for home automation is characterized by the proliferation of devices mapped on different hardware/software technologies, often adopting proprietary solutions. This has lead to the emergence of the technology silos previously mentioned. Thanks to the W3C WoT, different silos might be integrated by adding a semantic description to each interface, hence without requiring changes to the original deployments. Again, Things can be displayed and managed through the WoT Store, which can provide the following benefits in this use-case:

- *New Thing integration*: when users need to add a new device to their home automation system, they can take advantage of the self-publishing service of Things on the WoT Store, if already W3C WoT compliant. Otherwise, through the `Thing CLI` they can easily set-up and configure a new Thing for the current device.
- *User Interface*: the WoT Store provides an easy and customizable configuration panel through which users can monitor the behaviour of their home devices.
- *Mashup Application*; after having configured their Things, even not-technical users can set-up their MAs providing user-defined automatized actions.

IV. Conclusion and Future Works

In this study we presented the WoT Store, a novel platform for the management of W3C-compliant Things and Applications. We presented a preliminary version in [11]; the platform is still under deployment since there are plenty of additional features that can be integrated. Among the others, we mention two current works. First, we plan to extend the platform with the integration of data analytics features, based on machine learning and data mining techniques. Second, we are working on a control access mechanism in order to manage all the access requests made by and toward each Things directly from the WoT Store.

References

[1] McKInsey Global Institute. The Internet of Things: Mapping the value beyond the hype. Executive Summary. 2015.

[2] WoT W3C Architecture. http://www.w3.org/TR/wot-architecture/

[3] O. Barbosa12, C. Alves A systematic mapping study on software ecosystems 2011 Citeseer

[4] B. Iyer, C.-H. Lee, N. Venkatramen and D. Vesset. Monitoring Platform Emergence: Guidelines from Software Networks. *Proc. of CAIS*, 2017

[5] J.V. Joshua, D.O. Alao, S.O. Okolie, O. Awodele Software Ecosystem: Features, Benefits and Challenges International Journal of Advanced Computer Science and Applications(IJACSA), 4(8), 2013

[6] L. Roffia, P. Azzoni, C. Aguzzi, F. Viola, F. Antoniazzi and T. Salmon Cinotti Dynamic linked data: A SPARQL event processing architecture. Future Internet, 10(4), 2018.

[7] Y. Ji, K. Ok and W. Suk Choi. Demo Abstract: Web of Things based IoT standard interworking test case. *Proc. of ACM BuildSys*, Shenzen, China, 2018.

[8] B. Klotz, S. K. Datta, D. Wilms, et al. A car as a semantic Web Thing: motivation and demonstration. *Proc. of IEEE GIoTS*, Bilbao, Spain, 2018.

[9] M. McCool and E. Reshetova. Distributed Security risks and opportunities in the W3C Web of Things. *Proc. of IEEE DISS*, San Diego, USA, 2018.

[10] M. Blank, S. Kaebisch, H. Lahbaiel, and H. Kosch Role models and lifecycles in IoT and their impact on the W3C WoT Thing Description. *Proc. of IEEE IoT*, Santa Barbara, USA, 2018.

[11] L. Sciullo, C. Aguzzi, M. Di Felice, T. S. Cinotti WoT Store: Enabling things and applications discovery for the W3C Web of Things. *Proc. of IEEE CCNC*, Las Vegas, USA, 2019.

[12] Eclipse ThingWeb https://projects.eclipse.org/proposals/eclipse-thingweb

[13] Swamp EU-BR H2020 Project http://swamp-project.org