WoT-AD: A Descriptive Language for Group of Things in Massive IoT

Kim-Hung Le[1], Soumya Kanti Datta[1], Christian Bonnet[1], and Francois Hamon[2]

[1]Communication Systems Dept., EURECOM, Sophia Antipolis, France.

[2]GREENCITYZEN, Marseille, France.

Emails: {lek, dattas, bonnet}@eurecom.fr, francois.hamon@greencityzen.fr.

Abstract—Recently, the *Massive Internet of Things* (IoT) and *Web of Things* (WoT) are remarkable research fields aiming to facilitate the connectivity, accessibility, and control of the Things by Web standards and technologies for large-scale deployment. In such context, the end-user is capable of simply creating, mashing-up, and presenting the multiple Things to gain high-level information. However, current research approaches much more pay attention to describe a single Thing. The modeling and building the application for the compound objects consisting of groups of Things namely "Asset" are still limited due to the lack of description and seamless integration mechanism. Moreover, the traditional IoT Device Description Language directly installed on device is highly restricted in Massive IoT scenario because of stringent requirements for power consumption and operation cost. In this paper, we introduce the WoT based Asset Description (WoT-AD), a descriptive language for the Asset aiming to mitigate such limitations. WoT-AD explicitly describes a group of Things as homogeneous object to enable mash-up, self-discovery, and simply access their resources, entities, and services. We also provide a lightweight framework that fully integrates with WoT-AD to enable WoT for Massive IoT scenario. Such integration not only effectively models the Asset but also simplifies the development of mash-up application to different-skilled users. Finally, we evaluate the performance of our proposal to demonstrate its effectiveness and scalability in real use-case.

I. INTRODUCTION

The Internet of Things (IoT) is aiming to connect every object in the physical world into the digital world [1]. With the exponential growth of the number of Things, IoT creates several business opportunities and estimates generating up to 11 trillion by 2025 [2]. As a result, Massive IoT (MIoT) is emerging as a novel technology referring to huge volume of constrained IoT devices which stringently require excellent coverage, cost-effective, and low-energy consumption. Among several novel connectivity solutions for MIoT, the LPWAN technologies such as Sigfox and LoRA have been considered the most potential candidates while cellular-based connectives such as 5G or NB-IoT are under developing and testing process [3]. Although these protocols support collecting data from Things through the Web standards, the access and mash-up the Things resources are still limited due to lack of global standardization and inter-operation. Thus, Things are usually formed into small and isolated silos for specific applications.

These considerations leverage to a novel concept, namely, the Web of Things (WoT), for directly collecting and accessing any Thing's resources as Web resources. However, to fulfill Massive IoT requirements, there are some existing drawbacks mainly related to constraint in power and computing [4], [5]. For example, the MIoT Things is unsuitable for WoT applications, which are costly to constrained devices. In a similar situation, other effective solutions based on pub/sub model suffer from the limitations of LPWAN downlink. Most of them require the bi-directional connection to manage the changes of Thing's resources whereas LPWAN connectivity only supports a limited number of down-link messages per day [6].

At a higher level, the WoT is expected to break-down the *Silo of Things* by combining the traditional web paradigms with unified standards to present the Things. To do so, logical interfaces of Things are used to present the Things' resources and services using Semantic Web languages and annotations, such as the Device Description Language (DDL) [7], IoT-DDL [8], WoT-TD [9], CoRE-TD [10]. However, all of them only describe a single Thing that has a limited number of sensors and services. In practice, the Things may be a compound object, namely *Asset*, including hierarchical group of Things and services. For example, in a smart building scenario, the Asset could be a floor that includes several rooms monitored and controlled by IoT devices. Each room is considered as an independent Thing producing different resources and services which are tightly linked to these IoT devices. Therefore, we need a descriptive language to effectively describe either single or compound Things. In this way, we could enhance the capability of access and managing the IoT Things.

In this paper, we propose an approach enabling the WoT for compound objects namely Asset in Massive IoT scenario. Our method is based on a novel semantic description (WoT-AD) and a light-weight WoT framework, which are fully integrated together for enabling WoT. Such combination is not only capable of presenting, accessing and managing the Asset but also speed up the development of applications for Massive IoT. According to our solution, the Asset composing a group of Things is modeled as a uniform object that supports discovering, querying and visualized its resources through an unified interface. In addition, creating Asset description is facilitated to non-tech-savvy users by using a web-based graphic interface. This increases the usability of our proposal and make it suitable to a wide-range of Massive IoT use-cases. Our architecture consists of four primary layers as illustrated in Figure 4. The top of architecture is a composition layer that

directly interacts with users. This layer supports composing the user's desired Asset from the available templates. Then, the next layer named execution layer is used to (1) generate Asset API; (2) execute of Asses Model; (3) index the physical devices. The fundamental component of this layer is virtual sensor framework which receives, processes and updates the Asset's resources from collected data. The lowest layer is a connectivity layer which handles the connections from various IoT devices as well as external data sources (weather forecast, open IoT stream). The designed architecture is implemented as a framework following clustering model using the *Node.js language* to enhance scalability and performance. We experimentally evaluate the proposed architecture in a smart space scenario. In summary, our contribution is highlighted following:

1) Identifying the limitations of existing Things descriptions and WoT frameworks about describing and managing the Asset.
2) Proposing a novel semantic description for the Asset named WoT-AD.
3) Presenting a lightweight framework enabling WoT for the Massive IoT scenario. Such framework fully exploits WoT-AD for composing, querying and managing the Asset through a uniform interface. The prototype is implemented based on a clustering model along with various performance enhancement strategies to archive high scalability and performance.
4) A cross-platform development tool on the top of the framework speed up the Asset development for non-tech-savvy users.

The paper focuses on WoT-AD concepts and specifications. It also describes the detail of WoT framework that is used to exploit the WoT-AD. The paper is organized as follows. Section 2 summarizes the state-of-the-art related to solutions. The WoT-AD and the WoT framework is introduced in Section 3. Section 4 presents the implementation and evaluation of our approach. A conclusion is drawn in Section 5.

II. RELATED WORKS

In this section, we briefly review the Things description and existing Web of Things frameworks. Their limitations are also identified. At the end of the section, we present our motivations to bridge the gaps by delivering a lightweight WoT framework along with novel semantic description language for group of Things.

A. Things Description

To enable the IoT device integration into smart space scenarios, the Mobile and Pervasive Computing Lab at University of Florida presents the Device Description Language (DDL) based on the Service Oriented Architecture model. A IoT device in DDL is described as an entity including properties, internal mechanisms, and interfaces [7]. At first, the DDL is implemented and integrated within the Atlas sensor platform. Then, it is used to develop the Cloud-Edge-Beneath (CEB) architecture [11] to handle Massive scale of sensors and

devices in the smart city context. This CEB is based on Atlas architecture that allows end-user access directly to IoT devices or sensors through cloud interfaces [12][13]. Atlas framework and DDL are implemented at Edge as an intermediate layer between cloud and "Beneath" layer. The author uses the Open Service Gateway Initiative (OSGi) to connect to Atlas sensor platform and discovery service.

The World Wide Web Consortium (W3C) provides a WoT framework aiming to abstract the Things through a set of web services. To handle the digital presentation of the Things, W3C proposes the W3C Things Description (WoT-TD) consisting of (1) semantic description to present the general information of Things; (2) an interaction model with WoT properties, Action and Event to present the Thing's resources and services; (3) a semantic schema to express the data model [14]. The WoT-TD is used in [15] to describe in detail about how to control the different events and activities in specific domains.

To describe the Things in constrained environment, the Constrained RESTful environment (CoRE) based on Representation State Tranfer (REST) architecture is a common approach. CoRE applies Universal resource identifier (URI) to identify and discover the resources hosted by constrained nodes. The authors in [10] replace CoRE link format by a semantic-based description named JSON-LD to enable seamless thing to thing interaction in the constrained environment. A lightweight Things management framework residing in an M2M gateway is also proposed.

Fig. 1: Overview Asset Model

B. Web of Things Framework

Guinard *et al.* propose a WoT framework generating a set of REST services to exploit and hierarchically link the sensor node resources via web interface[16]. In that way, their solution speeds up creating ad-hoc applications for end-users. Their work is later applied in the AutoWoT project [17] aiming to rapidly integrate IoT devices into the WoT context by automatically building the web services to expose IoT device resources and functionality. These web services are based on a hierarchical service description which is created by

the end-user for the specific device. In addition, a graphical user interface is also proposed to facilitate creating description process for developers and tech-savvy users.

Another academic framework involving REST principle for integrating IoT devices into the Web is WebPlug [18], a WoT framework including several functional blocks allows to represent and manage sensor data as the web resources. The users can compose their personal services based on physical objects. These services and resources accessibility is enhanced by the Meta-URL that is proved to bring considerable benefits about resource discovery services and creating mash-up WoT applications. In the same approach, Christophe *et al.* [19] proposes a framework for creating and handling IoT devices as virtual objects according to the event-based rule schemas. The author in [20] presents a service-oriented framework supporting multiple real-time services (such as storage, sharing, discovery) for Wireless Sensor Networks (WSN) and smart objects via the Web. SemSense framework [21] aims to collect, process, equip with semantic meta-data and publish on the Web according the linked data standard. SPITFIRE [22] targets to unlock the uni-modal closed systems in WoT by generating the semantic sensor description and an effective searching mechanism. After abstraction, the sensors and Things are integrated into Linked Open Data (LoD) cloud. This effort makes the collected data easily accessible on the Web and accelerates the development of WoT applications. With the convergence of academic and commercial worlds, several Web platforms are emerging with the aim to abstract the heterogeneity of the physical devices by REST web services. There are well-known framework such as Xively [23], ThingSpeak [24], and ThingWorx [25]. These frameworks support accessing and visualizing the device data from cloud. Moreover, a RESTful API is offered for developer to build the custom applications on the collected data.

In the light of this state-of-the-art, there are still missing the semantic description and a lightweight IoT architecture seamlessly presenting and managing the compound objects, which are monitored by multiple IoT devices.

III. PROPOSED FRAMEWORK ARCHITECTURE

A. WoT Asset Description

WoT Asset Description is considered as the major part enabling Web of Things in our propose. Before describing in detail, we have to identify the required components inside such Asset: (1) The Asset must be able to briefly introduce and discovery itself via self-description meta-data;(2) The existing resources and interaction method are also effectively described to be fully accessible via Web; (3) The devices along with their services belonging to the Assess must be presented and managed via defined actions.

Based on the structure and requirements outlined above, we propose the WoT Asses Description (WoT-AD), a semantic description considering as an abstract entry point of a group of Things enabling the effective discovering, accessing and management process. WoT-AD is an extended concept of the W3c Things description (W3C-TD), which uses a JSON-LD

schema to describe the single Things. This allows WoT-AD could co-operate with the IoT devices described by W3C-TD. The WoT-AD structure consists of three primary sections as illustrated in Figure 1: (1) The Descriptive meta-data describes the general Asset information. (2) The Interaction model presents the Asset Resources (ARs) under the semantic schemes. (3) The Entities express the Things and services constituted of Asset and their relations. More details of each element are described below:

- **Descriptive Meta-data Section:** This section characterizes the Asset by presenting the unique identifier (URI) along with general information, such as name, model, description, link.
- **Interaction Section:** This section presents the available resources and services on the Asset and interaction methods. Each resource is considered as an interaction pattern including two separated parts:
  1) Descriptive meta-data describes the general information and configuration of resources such as type, description.
  2) Interaction information presents the information to connect to available resources such as connection address, method.

The interaction pattern is divided into three types: Property, Event and Action to present several resources. The Figure 2 illustrates the interaction section of a simple WoT-AD that abstracts a building floor as an Asset.

  1) The Property section expresses the state of Asset that is collected or calculated from the attached Things. The property values are timely updated by a mathematical formula complying with the *Virtual Sensor Framework*. The end-user can directly access or observe such values via interaction information defined via URI. As shown in the example, the temperature of the meeting room in the floor is declared as an Asset property and updated from the average of environment temperature collecting by *device 1* and *device 2*. An HTTP access API is also provided to obtain the values.
  2) The Event section presents the special situation detected by a set of specific conditions on Property. When the condition is satisfied, the corresponding actions are triggered. For example, in Figure 2, we identify an overheating event with the condition that the value of temperature property is higher than 30 degree and trigger the action to turn on an air conditioner.
  3) The Action section presents the supported functions of the Asset which could be triggered by events. The user could also directly access and control the actuator via available services belonging the Asset. In the presented example, the turning on air conditioner action is trigger by the overheating event and the command is directly sent to actuator of device 3.

- **Entities and Attachment Section:** This section describes entities belonging to an Asset. In our proposal, the entity may be a Things or service, namely attachment. Each Things contains the descriptive meta-data and the address to interact with Things resources and services. The attachment could be the cloud-based services such as open data servers, repositories, device management servers. Such attachment is treated as the Things which has meta-data and the interactive address.

```
"interaction": {
  "properties": {
    "temperature": {
      "@id": "meeting-room-temp",
      "description": "The temperature of all meeting room",
      "type": "int",
      "input": "avg(device_1.temp + device_2.temp)",
      "forms": [
        {
          "href": "https://buiding.example.com/floor1/temperature",
          "method": "GET"
        }
      ]
    }
  },
  "events": {
    "overheating": {
      "@id": "overheating_meetingRoom",
      "description": "Room reaches a high temperature",
      "condition": "self.meeting-room-temp > 30",
      "action": "self.air_condtioner_on"
    }
  },
  "actions": {
    "toggle": {
      "@id": "air_condtioner_on",
      "description": "Turn on the conditioner",
      "trigger": "self.overheating_meetingRoom",
      "output": "device_3.actuator_1",
      "forms": [
        {
          "href": "https://buiding.example.com/floor1/air_condtioner_on",
          "method": "PATCH",
          "mediaType": "application/json"
        }
      ]
    }
  }
},
```

Fig. 2: Overview the WoT-AD interaction section

B. WoT Architecture Principles and Design

Our goal is to enable WoT for Asset in Massive IoT by providing a novel concept for Asset description along with an WoT framework. In addition, the framework facilitates the development of IoT applications on the Asset by providing a composition editor. For this reason, the designed architecture complies following principles:

- High performance and low computational load for the server to adapt to Massive IoT scenario that must handle a large amount of connections.
- Effectively discovery, manage and access the Asset to fully aware of the real-world scenario.
- Accelerating the development of IoT applications on the Asset regardless of the user's skills.
- Handling the heterogeneity of constrained devices in the LPWAN context. This enables the connection from a wide range of devices to framework effortlessly regardless of their characteristics.

In order to archive these principles, we deal with several challenges listing below:

- The connectivity layer as the lowest part of the architecture should be able to handle the connections from not only IoT devices but also the open data sources (open weather, open MQTT broker). In addition, this layer needs to deal with the heterogeneity in term of LPWAN callback configurations. For example, the SIG-FOX forwards the data via HTTP GET method whereas Lora Objenous uses HTTP POST method with different syntax [26].
- The core of architecture must automatically discover and index the available resources in the Things which constitutes of the Asset. These resources should be discovered in mash-up applications to facilitate the Asset creation procedure. This part also takes responsibility for real-time updating the Assess resources from collected data.
- The upper part of architecture must provide simple APIs to allow the user effectively discover, access and manage the Asset. A graphics editor is also provided to facilitate the Asset creation process.
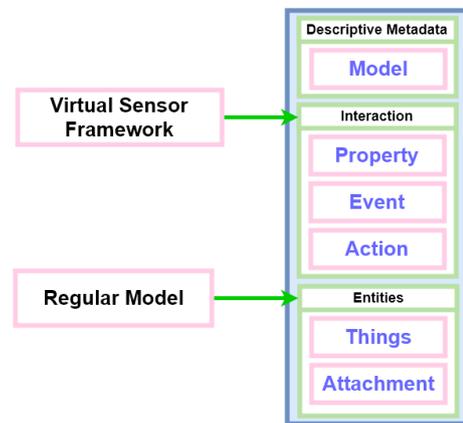
C. Overall Architecture

Fig. 3: Overview Operation Model

Based on the listed principles and challenges, we propose a WoT framework fully exploit the WoT-AD concept in Massive IoT scenarios. The initial implementation of such architecture is represented by three main components spreading on horizontal layers, whose features properly fit the mentioned principles. The connector framework is used at lowest layer [26] to handle the heterogeneous connections from both IoT devices and external data sources. The second component is the virtual sensor framework [27], an IoT framework supporting to create the logical data flow by visualizing the resources. The last component is a Graphic Editor facilitating the building and configuring process of the Asset by offering the drag-drop actions on HTML5 web interface. The overall of the framework is illustrated in Figure 4. The details of each layer are described below:

- **Connection Layer:** This layer takes responsibility to handle the connection from LPWAN devices to the framework through dedicated connectors that are simply

created and managed by connector framework. At this layer, collected data from physical devices is aggregated and pre-processed before conveying to upper layers. In case a new device connects to framework, the resources of this devices are registered and tracked by sensor tracking services.

- **Processing Layer** This layer is considered as a primary part of the architecture consisted of four main components: (1) Virtual sensor platform is used to create the Asset by abstracting and linking the device resources together. This platform also has responsibility to update the Asset resources from collected data (2) Indexing: After successfully creating, the Asset resources are indexed by the indexing system. This guarantees all the resources are discovered and synchronized with physical devices in real-time. (3) Asset scripting is used to generate the Asset descriptions and APIs based on defined resources. (4) Asset model execution coverts the Asset model to logical data flow [27] which is suitable with virtual sensor framework.

- **Presentation layer** This layer contains an interactive HTML 5 web interface, namely Asset Composer. This graphical interface provides the user several IoT components such as sensors, actuators, Assets, and symbolic links. The end-user could simply create their own Asset by drag-drop actions. A dashboard to manage the existing Asset also proposed.
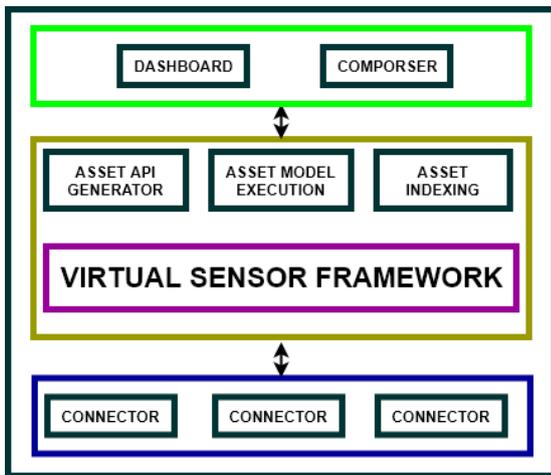

Fig. 4: Overview the Asset architecture

The primary requirement of WoT framework is that it must effectively and timely update and present the available resources. For this reason, we integrate our framework with a virtual sensor framework [27] which provides self-discovery mechanism and supports generating the logical data flow from existing IoT devices. In general, an Asset is a self-operation object updating its property values via an assigned logical data flow. The end-user can configure the updating frequency in the descriptive meta-data section of the Asset. The Asset's properties operate as a virtual sensor to retrieve and present the collected information from IoT sensors or devices. The elements in the Interaction section are build based on complex event processing (CEP) engine [28]. Therefore, if the current Asset properties reach a certain condition, the corresponding events are inferred and triggered the actions. For example, we present the building floor as an Asset with the properties are the C02 and Humility of each room on the floor. If the CO2 degree of 75% room is higher than 1000 ppm, the "suffocating" event is inferred and the system sends instruction command to relevant actuators to open window.

IV. PROOF-OF-CONCEPT

In order to validate the concept of WoT-AD and WoT architecture from the functional point of view, we have utilized the smart space scenario including a Raspberry Pi model B, which is considered as smart gateway be installed our WoT framework. In such context, the floors consisting of a set of rooms are considered as the Assets which are monitored by various IoT devices. Each device provides different resources and services. For example, a monitoring device has multiple sensors including Co2, humility, PIR (motion detection), temperature, light. All collected information is used to control the air conditioning and light system for saving energy consumption. The control decisions are based on the whole floor status instead of the single room. Therefore, this context is the best practice to apply Asset.

Following the described scenario, a remote user can visually create and control an Asset via the interface of smart gateway. If the users use the existing Asset template, they only need to assign the IoT devices identification to the template. Based on this identification, these devices resources will be discovered and added to the interaction model of WoT-AD. Then, the complete WoT-AD is conveyed to processing layer where (1) Asset Script component creates the access API for the Asset resources; (2) Indexing component assigns a unique identity to the Asset for further self-discovery. Based on the composed Asset configure file, the logical-data flow is created to update the Asset properties from the collected information. This ensures the Asset data timely updated and reacted with the environment changes via configured events and actions in Asset's properties. The software environment along with a light-weight local database for storing Asset configuration takes around to 100MB of spaces on the Raspberry Pi comparing with 32GB memory card. The most consumed operation of the framework is collecting, processing and updating the Asset information in processing layer. During the operational cycle mentioned in the scenario, the CPU load from 20 to 30 percent. The high performance and scalability of such operation are ensured by Virtual Sensor Framework [27].

V. CONCLUSION

This paper presents a semantic description for the physical Asset for enabling Web of Things in Massive IoT scenario. A WoT architecture is also defined and implemented to fully exploited the proposed concept. The architecture supports the interaction with various IoT devices and sensors based on

the connector model implemented at lowest layer. The core element of the architecture is a framework named Virtual Sensor Framework that executes all the Logical data flow converted from Asset description. At the highest layer, we implemented a graphics editor fascinating the Asset description composing. The effectiveness of designed solution was ensured by choosing and combining on-going technologies and IoT frameworks that have been practically demonstrated in real use-cases. To improve and extend the current works, we will evaluate and optimize WoT-AD in various IoT scenarios.

ACKNOWLEDGMENT

REFERENCES

[1] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[2] J. Manyika, M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin, and D. Aharon, "Unlocking the potential of the internet of things," *McKinsey Global Institute*, 2015.

[3] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[4] NorthStream, "White paper: Massive iot: different technologies for different needs," tech. rep., 2017.

[5] S. K. Datta, C. Bonnet, and N. Nikaein, "An iot gateway centric architecture to provide novel m2m services," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 514–519, March 2014.

[6] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on lpwa technology: Lora and nb-iot," *Ict Express*, vol. 3, no. 1, pp. 14–21, 2017.

[7] C. Chen and A. Helal, "Device integration in soda using the device description language," in *Applications and the Internet, 2009. SAINT'09. Ninth Annual International Symposium on*, pp. 100–106, IEEE, 2009.

[8] A. E. Khaled, A. Helal, W. Lindquist, and C. Lee, "Iot-ddl–device description language for the t in iot," *IEEE Access*, vol. 6, pp. 24048–24063, 2018.

[9] "Experimental Implementation of the Web of Things Framework." https://github.com/w3c/web-of-things-framework, 2016.

[10] S. K. Datta and C. Bonnet, "Describing things in the internet of things: From core link format to semantic based descriptions," in *2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pp. 1–2, May 2016.

[11] Y. Xu and A. Helal, "Scalable cloud-sensor architecture for the internet of things.," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 285–298, 2016.

[12] R. Bose, J. King, H. El-Zabadani, S. Pickles, and A. Helal, "Building plug-and-play smart homes using the atlas platform," in *Proceedings of the 4th International Conference on Smart Homes and Health Telematic (ICOST), Belfast, the Northern Islands*, pp. 265–272, Citeseer, 2006.

[13] C. Chen, R. Bose, and A. Helal, "Atlas: An open model for automatic integration and tele-programming of smart objects," in *Proceedings of the 3rd International Workshop on Design and Integration Principles for Smart Objects (DIPSO 2009), In conjunction with Ubicomp*, 2009.

[14] "W3C Web of Things (WoT) Thing Description." https://github.com/w3c/wot-thing-description.

[15] S. Kaebisch and D. Anicic, "Thing description as enabler of semantic interoperability on the web of things," in *Proc. IoT Semantic Interoperability Workshop*, pp. 1–3, 2016.

[16] D. Guinard, V. Trifa, T. Pham, and O. Liechti, "Towards physical mashups in the web of things," in *Networked Sensing Systems (INSS), 2009 Sixth International Conference on*, pp. 1–4, IEEE, 2009.

[17] S. Mayer, D. Guinard, and V. Trifa, "Facilitating the integration and interaction of real-world services for the web of things," *Proceedings of Urban Internet of Things–Towards Programmable Real-time Cities (UrbanIOT)*, 2010.

[18] B. Ostermaier, F. Schlup, and K. Römer, "Webplug: A framework for the web of things," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pp. 690–695, IEEE, 2010.

[19] B. Christophe, M. Boussard, M. Lu, A. Pastor, and V. Toubiana, "The web of things vision: Things as a service and interaction patterns," *Bell labs technical journal*, vol. 16, no. 1, pp. 55–61, 2011.

[20] A. Orestis, A. Dimitrios, and C. Ioannis, "Towards integrating iot devices with the web," in *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*, pp. 1–4, IEEE, 2012.

[21] A. Moraru, D. Mladenic, M. Vucnik, M. Porcius, C. Fortuna, and M. Mohorcic, "Exposing real world information for the web of things," in *Proceedings of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011*, p. 6, ACM, 2011.

[22] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kröller, M. Pagel, M. Hauswirth, *et al.*, "Spitfire: toward a semantic web of things," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 40–48, 2011.

[23] xively by LogMeln, "Public Cloud for the Internet of Things," 2014.

[24] MathWorks, "Internet of ThingsThingSpeak," 2014.

[25] PTC, "ThingWorx the 1st application platform for the connected world," 2014.

[26] L. Kim-Hung, S. K. Datta, C. Bonnet, F. Hamon, and A. Boudonne, "An industrial iot framework to simplify connection process using system-generated connector," in *Research and Technologies for Society and Industry (RTSI), 2017 IEEE 3rd International Forum on*, pp. 1–6, IEEE, 2017.

[27] L. Kim-Hung, S. K. Datta, C. Bonnet, F. Hamon, and A. Boudonne, "A scalable iot framework to design logical data flow using virtual sensor," in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–7, IEEE, 2017.

[28] C. Y. Chen, J. H. Fu, T. Sung, P.-F. Wang, E. Jou, and M.-W. Feng, "Complex event processing for the internet of things and its applications," in *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, pp. 1144–1149, IEEE, 2014.