

BETaaS Platform – A Things as a Service Environment for Future M2M Marketplaces

Sofoklis Kyriazakos^{1(✉)}, Bayu Anggorojati¹, Neeli Prasad¹,
Carlo Vallati², Enzo Mingozzi², Giacomo Tanganelli²,
Novella Buonaccorsi³, Nicola Valdambri³, Nikolaos Zonidis⁴,
George Labropoulos⁴, Belen Martinez Rodriguez⁵,
Alessandro Mamelli⁶, and Davide Sommacampagna⁶

¹ Center for TeleInfrastructure, Aalborg University, Aalborg, Denmark
{sk, ba, np}@es.aau.dk

² Department of Information Engineering, University of Pisa, Pisa, Italy
{c.vallati, e.mingozzi, g.tanganelli}@iet.unipi.it
³ Intecs S.p.A., Roma, Italy

{novella.buonaccorsi, nicola.valdambri}@intecs.it

⁴ Converge ICT Solutions and Services S.A., Athens, Greece
{nzonidis, glabropoulos}@converge.gr

⁵ Tecnalía Research & Innovation, San Sebastian, Spain
belen.martinez@tecnalia.com

⁶ Hewlett-Packard Italiana S.r.l., Roma, Italy
{alessandro.mamelli, davide.sommacampagna}@hp.com

Abstract. Building the Environment for Things as a Service (BETaaS) is a novel platform for the deployment and execution of content-centric M2M applications, which relies on a local cloud of gateways. BETaaS platform provides a uniform interface and services to map content with things in a context-aware manner. Deployment of services for the execution of applications is dynamic and takes into account the computational resources of the low-end physical devices used. To this aim, BETaaS platform is based on a suitable defined IoT model, allowing the integration of the BETaaS components within the future Internet environment. In this paper we present the BETaaS concept, the high level platform architecture and application scenarios that extend the state-of-the-art in M2M communications and open the horizon for future M2M marketplaces.

Keywords: IoT · M2M communications · TaaS

1 Introduction

Marketplaces of applications became a trend over the past years within this decade and this trend is a game-changer by providing a different business model. Usually the app marketplaces are linked with smart phones, but are not limited to that. Recently operating systems make use of such models and even TV manufactures is using this ‘channel’ to differentiate from the competitors in the smart TVs world. M2M communications, which imply the elimination of the human in the loop, are not only a

trend, but is the natural evolution of the marketplaces. The IoT that integrate objects into the Internet, has opened new horizons in the M2M communications and we consider this as a unique opportunity to come up with a Things as a Service Environment that will increase the spectrum of M2M applications.

This paper is organized around 5 sections. After the first section, which is this introduction, we present in Sect. 2 the current approaches, the limitations and the constraints of M2M applications and IoT. The section finishes with the challenges that are addressed from BETaaS platform, that are described in more detail under Sect. 3, BETaaS platform. In this section the concept, the high level architecture and the Things as a Service model are described, which are the basis of the BETaaS project and the major innovation of this approach. In Sect. 4 we focus on M2M applications that can be available in future marketplaces, requiring the flexibility offered by BETaaS approach. Finally, in the conclusions section we focus on the next step and the vision of BETaaS in the IoT and M2M world.

2 M2M Applications and IoT

2.1 Current Approaches, Limitations and Ways to Address These Challenges

The main limitations coming from the world of M2M applications and IoT is certainly related to fragmentation of standards and technologies. A lot of them are available on the market and platforms are often designed to work with specific subsets, depending on the applications. BETaaS architecture includes a specific layer in charge of adapting to specific technologies and protocols. Furthermore the platform is based on a software framework that allows plug-ins to be dynamically added, making it easy to cope with the fragmentation of the M2M market referred above. Besides that, BETaaS focuses on the ETSI M2M standard that is going to be a promising solution to next M2M systems. The platform then opens to this class of sensor networks that is likely to soon become the most widespread M2M standard, partially representing a solution to fragmentation. Another class of limitations is related to the heterogeneity of ontologies that have to be harmonized and integrated when several applications and devices need to talk each other. BETaaS faces this issue through the construction of an ontologies network. Concerning the implementation patterns of M2M systems, a strong limitation often comes from their vertical and closed architecture. It commonly happens that each infrastructure serves a single M2M application, excluding any interoperability pattern between other systems, applications or things. Thus, when new M2M applications are needed, providers have to re-create M2M communication platforms and data representation formats. BETaaS differentiates itself from this kind of model, implementing a horizontal approach and defining a single common reference model that can be used by different applications and devices to build M2M services.

2.2 Challenges

BETaaS is providing a new vision about the way to expose and manage things in the Internet of Everything environment. The BETaaS platform is using semantic technologies and stays close to the M2M nodes, through the creation of local clouds among nodes in concrete contexts. That ease the development and execution of content centric user applications, thus allowing the definition of new M2M applications whose scope spans across different domains. Evaluating the advances of BETaaS compared to existing approaches, there are significant improvement in aspects like QoS, interoperability aspects, discovery, security and trust, and the integration of Big Data analytics capabilities. At QoS level, the BETaaS improvements include QoS negotiation support, support to real-time applications, energy-efficient selection of things and QoS assurances in a distributed environment. BETaaS defines a comprehensive QoS management framework that comprises a general QoS model in order to differentiate IoT/M2M services based on appropriate QoS measures, and optimized resource reservation and allocation to services. BETaaS extends the current state-of-art of SLA creation and negotiation for enabling the BETaaS QoS model in the relevant M2M application scenarios. Regarding interoperability, BETaaS integrates different IoT/M2M systems and exploits different capabilities of different IoT/M2M systems. As far as discovery is concerned, BETaaS supports distributed discovery of thing services and look-up of thing services based on context and with regards to security and trust, BETaaS focuses on the secure access to thing services and trust establishment in the overall system. BETaaS also enables virtualization in ARM-based devices, so it is possible to perform more complex tasks in the local environment. In addition, BETaaS leverages innovative Big Data capabilities to build a highly scalable and efficient architecture that is capable of handling the large amount of highly distributed information generated by things, and then enabling to query data with low complexity.

3 BETaaS Platform

3.1 BETaaS Approach

BETaaS platform provides a runtime environment relying on a local cloud of gateways to support the deployment and execution of content-centric M2M applications. BETaaS platform seamlessly integrates existing heterogeneous M2M systems by means of a network of gateways. Each gateway runs the BETaaS run-time environment that forms a logical overlay. The logical federation of networks forms a local cloud in which each gateway shares the functionalities offered by the smart objects of its M2M systems with the rest of the network. The run-time platform running on each gateway provides M2M applications connected to any of the BETaaS gateways a common interface to access their respective things, regardless of location and technology. In order to account for heterogeneity, the BETaaS platform builds upon a baseline reference model and architecture, Things-as-a-Service (TaaS), providing an abstract and uniform description of the underlying M2M systems. The TaaS paradigm defines for each thing connected to the platform, a *thing service* is created to represent the *basic service(s)* that can be provided to each application. The BETaaS platform is content-centric in the sense that

it provides services that depend on the type of data that they provide and on the context in which that data is used. The circumstances that have not been considered in BETaaS as part of the context of a thing are its battery level, its available computing capacity, the communication protocol used and its location. BETaaS uses semantic technologies and natural language processing to unify the information that comes from heterogeneous things and to deal with the dynamic nature of their context. Following this purpose, we have built the BETaaS ontology, a network of ontologies created reusing ontologies that are relevant in their domains and that model the BETaaS scenarios. In particular, we have used the following ontologies: SSN [1], Time [2], CF [3], Phenonet [4], MUO [5], FIPA [6] and GeoNames [7]. To promote standardization, the contextual information related to things (type of thing, location, etc.) is translated to WordNet [8] synsets whenever possible, and stored in the ontology. Wordnet is a lexical database that groups English words into sets of synonyms (synsets), and whose organization is based on the semantic relationships between synsets (hypernymy, hyponymy, holonymy and meronymy). All synsets inserted in the BETaaS ontology are stored following these relationships, through SKOS, [9] which offers a common data model to organize classifications in a hierarchical way. The relationships between the terms are used as a mechanism of knowledge inference. Inference can be applied at the time of the execution of applications: e.g. if an application demands the temperature at home, a temperature sensor installed in the kitchen is valid (*kitchen* is meronym of *home*). Inference can also be applied when registering things in the BETaaS ontology: e.g. a new thing described as moistness sensor, would be added to a family in the ontology described as humidity sensors (*moistness* and *humidity* belong to the same WordNet family). The contextual information associated to each of the things connected to a gateway allows the platform to create a thing service for each of those things, following the nomenclature *setLocationType/getLocationType* (e.g. *getKitchenTemperature*).

3.2 BETaaS High Level Architecture

BETaaS high level architecture is depicted in Fig. 1. A layered structure is adopted to guarantee at the top the proper level of abstraction to applications and at the bottom the flexibility necessary to integrate different system characterized by different technologies.

The core component of the BETaaS architecture is the Thing as a Service layer (TaaS) that is implemented in a distributed fashion this layer exposes a service oriented interface to access the things connected to the platform, regardless of their technology and physical location. Build on top of the TaaS layer BETaaS platform exposes the service layer to applications. A basic service that exposes directly one or more thing services, or can be an extended service installed on the platform to provide to custom services.

Transparent integration of existing systems is achieved through the definition of adaptation layers. An adaptation layer implements a common interface towards TaaS layer to access the functionalities of the physical system in a uniform manner regardless of its technology. The definition of an adaptation layer allows the integration without requiring any modifications to the original system which is BETaaS-unaware, though at

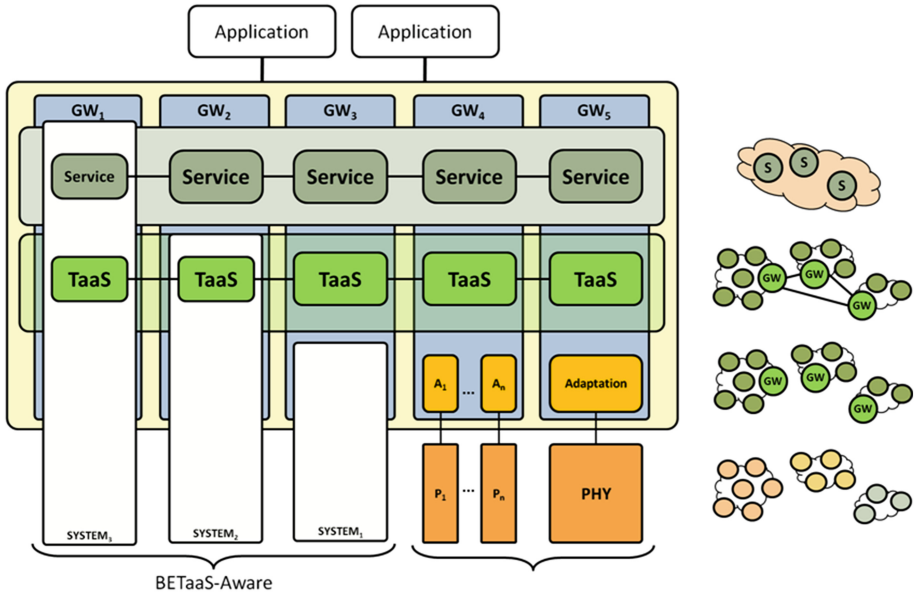


Fig. 1. BETaaS architecture.

the cost of possible replication of some functionalities. When modifications to the original systems are allowed, a more efficient integration can be achieved introducing a set of modifications which make the original system BETaaS-aware, through the implementation of some of the BETaaS layers. Some or all of the BETaaS functionalities are implemented by the integrated system in a fully interoperable manner, i.e. compliant with BETaaS internal interfaces.

3.3 Things as a Service

The major impediment of the current IoT galaxy is its fragmentation. Current IoT is made of vertical isolated systems characterized by different technologies and communication protocols, often proprietary and not interoperable. In this context the BETaaS goal is to build the environment for the integration of these systems into a horizontal architecture on which the next generation of applications will leverage. At this aim, one of most important feature of the platform is providing a unified interface to applications in order to enable seamless access to smart objects abstracting their technology and location. At this aim, the BETaaS platform defines at its core a Service Oriented interface, the Things as a Service model, *TaaS* for short. Goal of the TaaS is to abstract all the details behind each thing, e.g. location, technology, communication protocol, through a common representation, the *Thing Service*. Defined in [11], the Thing Service is the logical representation of a basic functionality that can be offered by a physical smart object. The role of the TaaS model within the BETaaS platform is illustrated in Fig. 2. As can be seen the TaaS model is implemented in the platform by

the TaaS layer, which is at the core of the BETaaS architecture, between the lower layers (Adaptation and Physical layers) and the higher layers (Service layer and applications). TaaS main responsibility is to implement the bridge between the “*service providers*” (things) and the “*service consumers*” (services and applications). Smart devices from different physical IoT systems are integrated within the platform through the Adaptation layer: for each IoT system a different Adaptation instance is installed to implement a common interface that is used by the TaaS to discover and access things. For each device, one or more Thing Services are created to be exposed to Service Layer. TaaS layer represents also the foundation for the BETaaS distributed architecture. Implemented in a distributed manner over the local cloud of gateways that characterizes BETaaS deployments, the TaaS allows services, and indirectly applications, to access the Thing Services not only regardless of their specific interface or communication protocol, but also independently of their physical location. Retrieval and exploitation of context information is of paramount importance for TaaS operations. Semantic capabilities, in particular, are included in the TaaS implementation to process the context associated with each thing to derive the set of Thing Services. At this aim the TaaS Context information are also exploited to provide Look-up functionalities to services which can request a Thing Service through its contextual requirements. In order to support deployment of real-time applications the TaaS layer includes Quality of Service (QoS) functionalities that can be used by the Service Layer to request a certain QoS for invoking a Thing Service. In order to support heterogeneous requirements that characterize IoT deployments, the QoS is negotiated to stipulate a SLA. For this reason, a negotiation procedure is implemented in the TaaS to allow services to negotiated the QoS level necessary to fulfill application needs. QoS is enforced through an efficient managing of resources, which is performed exploiting the

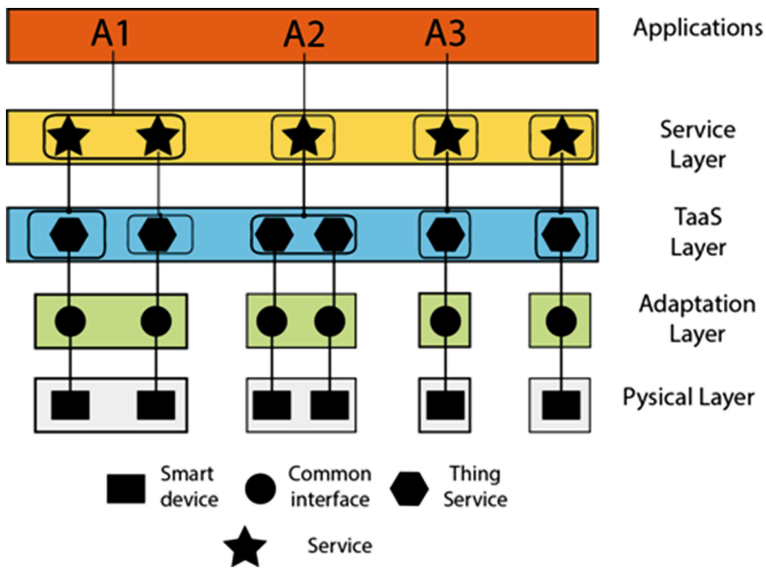


Fig. 2. Thing as a service model.

concept of Equivalent Thing Services, Thing Services that can provide the same functionality equivalently.

4 Application scenarios

4.1 Smart City

The Smart City scenario is mainly focused on the integration of Smart Things belonging to different systems. This scenario shows how BETaaS can adapt to different data sources, how it can be used to build a complete system and how it can provide an added value to existing ones.

4.2 Building Blocks

The scenario is built upon:

- A smart lighting system that controls public lamp posts in a parking lot
- A system to manage cars access to a restricted area of the city based on the car's positions
- A traffic system that receives data from traffic sensors installed along a network of roads.

The smart lighting system is completely implemented through a single BETaaS gateway connected to ETSI-enabled devices: light intensity tunable lamp posts and infrared presence sensors. A second BETaaS gateway is in charge of receiving users' position, matching them to the city map. It is intended to implement a service to check the access to Low Emission Zones (LEZ), making users to pay a fee as they enter. In this case Things are represented by on-board cars' GPS receivers. The traffic system is then a completely independent system called SIMIS (Sustainable and Intelligent Mobility Integrated System). It allows to check the current status of roads and parking lots, also providing traffic data to external applications. In this scenario it is shown how BETaaS can be easily extended to receive data from it.

4.3 Integration through BETaaS

In this smart city scenario the two BETaaS gateways described above are then used to make up a single BETaaS instance. Once they are configured to join each other, they start sharing their resources. Gateways resources are mainly represented by the Thing Services they created on top of the Things: lamp posts, presence sensors, traffic sensors and cars' GPS receivers. BETaaS allows to add an application logic inside the instance through extended services. In this scenario two extended services are included:

- The smart lighting extended service on the first gateway. It not only exploits the presence sensors directly connected to the gateway itself but it also considers the traffic data provided by the other gateway. Lamp light is then intensified or dimmed also based on the current traffic density.

- The LEZ extended service not only uses the cars' position to make users pay a fee once they enter the restricted zone. It also exploits traffic data coming from SIMIS to compute dynamic fees based on the current roads congestion.

Extended services may also be accessed by BETaaS users from their external applications. So in this case users access the platform through mobile applications from their cars, being notified about the current fee that is currently applied, depending on their position and traffic intensity.

4.4 Home domain

The Home Domain scenario is focusing on a typical need of Smart Homes and Building Management Systems, which is the exploitation of existing infrastructures. BETaaS concept aims to prove through this scenario, that it can extend proprietary systems in order to result in integrated solutions with multiple services and capabilities. The first sub-scenario of the Home Domain presents the steps that a user would do in order to cover a need, which is the deployment of an application that is managing the garden watering. In this scenario, the main actor downloads the app from a marketplace and is exploiting a number of proprietary systems that can be upgraded into BETaaS enabled gateways. Therefore devices such as thermometer, humidity sensor, etc. can be exploited from the home BMS. Additional devices that are required, e.g. electronic watering valve, has to be installed and controlled by another BETaaS gateway. The second sub-scenario of the Home Domain is focusing on the case of a user who already has an alarm system installed for his home, but still he would be interested for an extension of the alarm to cover the entry points of the building, i.e. front door of the building and garage automatic door. In this scenario, again he downloads (or writes) an application that exploits the capabilities of the home alarm, which is upgraded to become BETaaS enabled, and extends its services by managing some additional devices (Things) that are required to extend the coverage of the alarm.

5 Conclusions

The paper presents the Things as a Service environment for future M2M applications, proposed by BETaaS. The challenges and the implementation of BETaaS are described, as well as the indicative scenarios and applications. Unlike many IoT and M2M approaches, BETaaS is a tangible platform that aims become an open-source community to further develop the source code and achieve the highest impact. We strongly believe the clear advances of BETaaS and in the future work we are considering to prepare the community and all stakeholders for the evolution of M2M applications.

Acknowledgment. This work has been carried out within the activities of the project "Building the Environment for the Things as a Service (BETaaS)", which is co-funded by the European Commission under the Seventh Framework Programme (grant no. 317674).

References

1. SSN. <http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628>
2. Time. <http://www.w3.org/2006/time>
3. CF. <http://www.w3.org/2005/Incubator/ssn/ssnx/cf/cf-property>
4. Phenonet. <http://www.w3.org/2005/Incubator/ssn/ssnx/meteo/phenonet>
5. MUO. <http://purl.oclc.org/NET/muo/muo>
6. FIPA. <http://www.fipa.org/specs/fipa00091/PC00091A.html>
7. GeoNames. http://www.geonames.org/ontology/ontology_v3.1.rdf
8. WordNet. <http://wordnetweb.princeton.edu>
9. SKOS. <http://www.w3.org/2004/02/skos/intro>
10. Teixeira, T., Hachem, S., Issarny, V., Georgantas, N.: Service oriented middleware for the internet of things: a perspective. In: Abramowicz, W., Llorente, I.M., Surridge, M., Zisman, A., Vayssière, J. (eds.) ServiceWave 2011. LNCS, vol. 6994, pp. 220–229. Springer, Heidelberg (2011)
11. Thoma, M., et al.: On iot-services: survey, classification and enterprise integration. In: 2012 IEEE International Conference on Green Computing and Communications (GreenCom). IEEE (2012)