An Internet of Things platform to facilitate the development of context aware applications. Overview, challenges and experiments.

FERNANDO RIBEIRO[1,2], JOSÉ METRÔLHO[1]
Informatics Department
[1]Instituto Politécnico de Castelo Branco
[2]ALGORITMI Research Centre, University of Minho
PORTUGAL
fribeiro@ipcb.pt, metrolho@ipcb.pt

Abstract: - There are several Internet of Things (IoT) platforms developed to integrate sensors and actuators in the Internet. To be successful, these platforms have to meet the expectations of different stakeholders, namely device manufactures, application developers and end-users. They should support distinct data formats provided from many different objects and be adaptable to the real characteristics and needs of each specific context and developer. They should also be provided with appropriate tools and mechanisms to facilitate the task of developing context aware applications, including support to easily integrate new objects and different types of data, as well as providing appropriate API to control and access objects and get contextual information in the appropriate way. This work describes a platform that provides support for application developers to integrate, manage and use information about standard and user customized objects. It allows for easy integration of IoT objects, to manage them and make information about them available to be used by context-aware applications. Things can be registered on the platform by its owner or they can register by themselves, sending the information using simple protocols, and referring the type of information they represent and also the accessibility level to their information. The platform processes the information about Things, stores it, and provides tools to manage Things and to make them available to be used by external applications. Examples of context-aware applications are described to illustrate the usefulness of the platform and its viability making the development of context aware applications easier.

Key-Words: - ambient intelligence, context-awareness, internet of things, smart spaces, ubiquitous computing.

1 Introduction

The Internet of Things (IoT) has been defined as a global infrastructure for the information society, enabling advanced services by interconnecting physical and virtual things based on existing and evolving interoperable information and communication technologies [1] or, in other words, an infrastructure that allows people and things to be connected at anytime, anywhere, with anything and anyone [2]. In this infrastructure, objects of the physical world or the information world, which are capable of being identified and integrated into communication networks, embedded in smart environments will provide support for people in their everyday life activities. This will facilitate collaboration and communication between people and things, and between things themselves, and create new opportunities for several innovative applications/services but, at the same time, they also create additional research and development challenges. Wearables, smart devices, mobile devices, and a huge diversity of sensors, with computational and communicational capabilities, can be used to support the needs of citizens. However, to succeed, it is important to facilitate the development of context-aware applications and it is necessary to develop interfaces between the physical world and people to gain new knowledge about the environment, people and their activities and interests and thus set up innovative, useful services and applications. While computers, smartphones, or other computer devices, provide all the technical resources for developing applications for the IoT, the deployment of smart and context aware applications poses several challenges that need to be addressed, namely: discovering and connecting to smart objects and external information resources; developing interfaces to connect with the heterogeneity of real world smart objects; and other issues related to security and privacy. In these scenarios, IoT platforms can play an important role, interconnecting devices onto the Internet as web resources, using traditional Web standards (such as HTTP, REST, etc.), managing them, facilitating access to the objects' capabilities and facilitating the design and implementation of context aware applications, allowing developers to easily access and use information available on the IoT and providing them with a set of tools to enrich their knowledge about the context and thus making their applications smarter and more useful.

In this paper we present an IoT platform that allows easy integration and management of IoT objects, making information about them available to be used by context-aware applications. At this stage, the objective is not to provide a complete middleware solution for managing context. Instead the objective is to provide support for developers to integrate, manage and use information about standard and user customized objects. Objects may self-register on the platform sending their information using simple protocols and identifying the object, the type of information they represent and also the accessibility level to their information. The platform processes the information about things, it stores it, and provides tools to manage things and to make them available to be used by external applications. Examples of context-aware applications are also described to illustrate the usefulness of the platform and its viability, making the development of context aware applications easier.

The rest of this paper explores the key challenges involved, some alternatives available and provides an overview of some systems that are developed to support context aware applications for the IoT. It also presents the framework we developed to facilitate the integration and the usage of the Things over the internet and describes some context aware applications we developed that make use of the platform to access contextual information, and thus, adapt its behavior accordingly. Initial experiments suggest that the proposed platform can be easily used to associate and integrate Things on the Internet, to manage them and to make them available to be easily used by context aware applications developers, without the burden of managing and handling with the particularities of each IoT device.

2 Context-aware applications and IoT: Background and challenges

Deploying context-aware information systems for the IoT poses many challenges that are not usually found when developing traditional information systems and that may limit their applicability and success. The singularities and specificities of these environments make them very rich and heterogeneous in terms of devices (sensors and actuators) and interaction mechanisms and create new opportunities for several innovative applications. However, these also simultaneously create additional challenges. When large numbers of sensors and actuators are required, the traditional approaches (i.e. handle with sensors and actuators directly with applications individually and manually) become infeasible. In these scenarios, IoT platforms may represent an important role and they can be used and configured for multiple and different applications.

In this section, we consider some of the major research challenges when developing IoT platforms to support the development of context-aware applications. In addition to the hardware and technical challenges, related to communication and networking, latency, power and storage, there are a number of other challenges, that may influence the development of platforms to support the development of context-aware applications, and they should be considered:

– Security and access control: platforms should provide mechanisms to ensure that only authorized users, devices and applications can access the platform and its data. IoT enables a constant transfer and sharing of data between Things and users in order to achieve particular goals. In these environments, authentication, authorization and access control are important to ensure secure communication and information manipulation [3-6].
– Trust and privacy: in the domain of trust, some concerns are related to the surveillance of users without knowledge or consent, the exchange of sensitive data, the control of access and ownership of data and sensitive context [5-7]. IoT introduces new ways of collecting and processing information from objects and from different sources and very often it reveals information about the individuals, namely information related to their location, activities, habits or interactions with others. For all of these situations, platforms should provide users with mechanisms to control what personal data may be collected or accessed and who may collect and access their personal information.
– Processing and mining large volumes of data: platforms should be able to process and mine large volumes of data from many different sources to provide useful services [8].
– Managing heterogeneity: managing the high diversity of technologies, devices, environments and applications is a significant challenge [8].

- Discover and integrate sensors and services: there is the need for efficient mechanisms to support sensor discovering, and integrating them into the platform. Thus, platforms should provide mechanisms to easily and efficiently integrate objects.
- Publish/subscribe: software platforms should provide mechanisms to easily support publishing/subscribing services facilitating the use of data by external applications.
- Facilitate software development: the availability of efficient tools to be used by external application developers is an important issue. Designing and building context aware applications for IoT will require enormous effort and device particularities, networking and embedded software knowledge. Thus, tools to manage devices and services, configure permissions and access historical data provided by IoT platforms represent important resources and may contribute to facilitate software developers to build their applications quickly and with higher quality.

3 Related work

Here, we review some systems that, in one sense or another, integrate and manage objects in the IoT and make them available to be used by external consumers, allowing developers to focus on the development of the context aware application and release them from the complexity of sensor deployment, namely, the tasks of processing and dealing with sensors, handling multiple devices and multiple sources and dealing with security and privacy concerns which need to be solved adequately when targeting real-world deployments.

COMPOSE [9] (Collaborative Open Market to Place Objects at your Service) is a framework that uses cloud computing infrastructures and IoT technologies allowing for the integration of smart objects and external services as well as the provision of scalable resources for data and application management. It enables the end-to-end development and deployment of context-awareness by providing a set of tools and methods for the collection of contextual information on smart devices (smartphones, sensors, actuators), tools for communication with external resources, infrastructure for hosting the data storage and processing, an open and scalable marketplace, and an environment where not only applications, but also services are derived by interacting with objects, which can be shared and traded. EVRYTHNG [10] is a IoT cloud platform that connects consumer products to the Web to make them interactive and trackable, and manages real-time data to drive applications. It provides a number of toolkits and support resources that can be used by front-end mobile and Web developers. ThingSpeak [11] is an API and web service for the IoT, to store and retrieve data from things using HTTP. Using custom HTTP headers, GETs, POSTs, PUTs, and DELETEs, ThingSpeak can trigger several types of responses and actions on remote individual hardware, storing numeric and alphanumeric data, numeric data processing, location tracking, and status updates. It supports real-time data collection and storage, MATLAB analytics and visualizations, alerts, scheduling and Geolocation data. It works with Arduino, raspberry PI and Electric Imp (a connectivity platform for connecting Wi-Fi devices to cloud services), mobile and web apps and Twitter. Carriots [12] is an application hosting and development platform specially designed for projects related to the IoT. This platform makes it possible to connect devices (e.g. Arduino, Raspberry Pi, Nanode), collect data from connected objects, store it, and build applications. It provides an API, hosting and a development environment to support application development through the HTTP RESTful API to push and pull XML or JSON data and it provides support to deploy, interact, enable or disable devices from Carriots. Zetta [13] is an open source platform built on Node.js to create IoT servers that run across geo-distributed computers and the cloud. It combines REST APIs, WebSockets and reactive programming for assembling devices into real-time applications and it makes possible to connect Raspberry Pi, BeagleBones and PCs together with cloud platforms. CA4IOT [14] is a middleware that is conceived to help users by automating the task of selecting the sensors according to the problems/tasks at hand. Its focus is on automated configuration of filtering, fusion and reasoning mechanisms that can be applied to the collected sensor data streams using selected sensors.

There are several IoT platforms to support integration of objects into the Internet, provide access of their data, manage them and make their information available to be used by external applications. Some of them integrate objects associated to devices like Arduino, Raspberry pi or smartphones. However, these platforms should also support many types of information and distinct data formats provided from many different sensors, and they should be adaptable to the real characteristics and needs of each specific context and developer. They should be provided with appropriate tools and mechanisms to facilitate the task of developing context aware applications, including support to easily integrate new sensors and different types of data and appropriate API to control and access objects and to obtain contextual information in the appropriate way. Additionally, these platforms should also provide support for users to integrate, manage and use information about standard and user customized objects.

4 Platforms for the IoT

IoT platforms are very common and they are usually developed to support the integration of data from several sensors and actuators into the Internet. Frequently they are focused on the communication and networking aspects between devices that are used for sensing and acting over real world objects. However, these platforms can also play an important role in helping context aware application developers to deal with the particularities of each device and with a variety of protocols to support most representative technologies, and also to provide the mechanisms to deal with data persistence, security, privacy and provide a uniform and open interface to access data through well-known web technologies. In other words, they may help context-aware application developers to focus on the development of their applications and release them from dealing with the problems of context information acquisition, modeling, and management. To achieve these goals IoT platforms should provide support to easily integrate sensors and actuators allowing their owners to configure them and allowing them to define new types of information. Moreover, platforms should provide support for processing and storing the information and should provide mechanisms to manage this information and to make it available to be used by external applications. Additionally, the platform should have an open architecture and should provide developers with appropriate tools to allow them to access and to use this information in their applications. Specifically, platforms should combine two different perspectives: *i)* information producers and *ii)* information consumers and, among others, they should be designed to:

– Facilitate connectivity among heterogeneous IoT-based networks.
– Be flexible to support new types of objects and data.
– Provide open API to manage different objects and to appropriately answer to the needs of consumer applications. For this purpose, a set of dynamic interfaces, based on the REST architectural style, should be provided.
– Adequately manage data according to their private or public nature.

Fig. 1 presents a general architecture of the IoT platform.

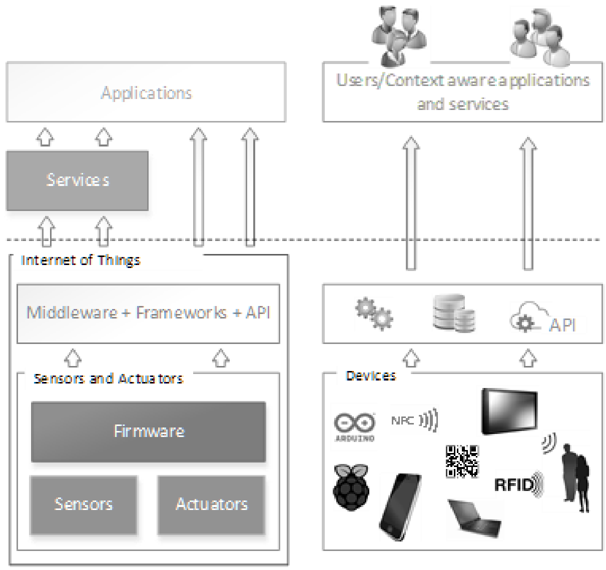

Fig. 1. General system architecture and technologies.

5 A platform to facilitate the development of context-aware applications

As part of our work on this topic, we have developed a platform that is able to easily integrate, manage and make available information about standard and user customized objects on the IoT.

The proposed architecture includes five main modules: sensors and actuators, data processing and storage, API, web portal and context-aware applications (see Fig. 2).

Sensors and actuators: This module includes sensors, actuators and software that is responsible for controlling, monitoring and manipulating the data provided by the objects, and then sends this data to the data processing and storage module. Each object should be able to communicate its own data to the data processing and storage module. Objects are registered in the middleware by the owner through the web portal or, autonomously, including in their message to the middleware the owner ID, location, type of data and correspondent value. It is also possible to define the level of access to the data of each object. In this case the message should include a specific field to indicate if the data is private or public. This information is sent to the

Data Processing and Storage module using specific formats.

At this stage the platform is able to support Arduino and Raspberry based objects (e.g. RFID, Bluetooth, Temperature, Humidity, display control). The platform is able to accept objects according to owner specifications, i.e. the owner may define its own type of objects, defining the data type, format and access level.

Data processing and storage: This module receives the information provided by the objects. It acts as a context repository and it stores information in a way that queries and information can be efficiently handled when needed. It is also responsible for processing and storing the information and for controlling the access to this information through the API.

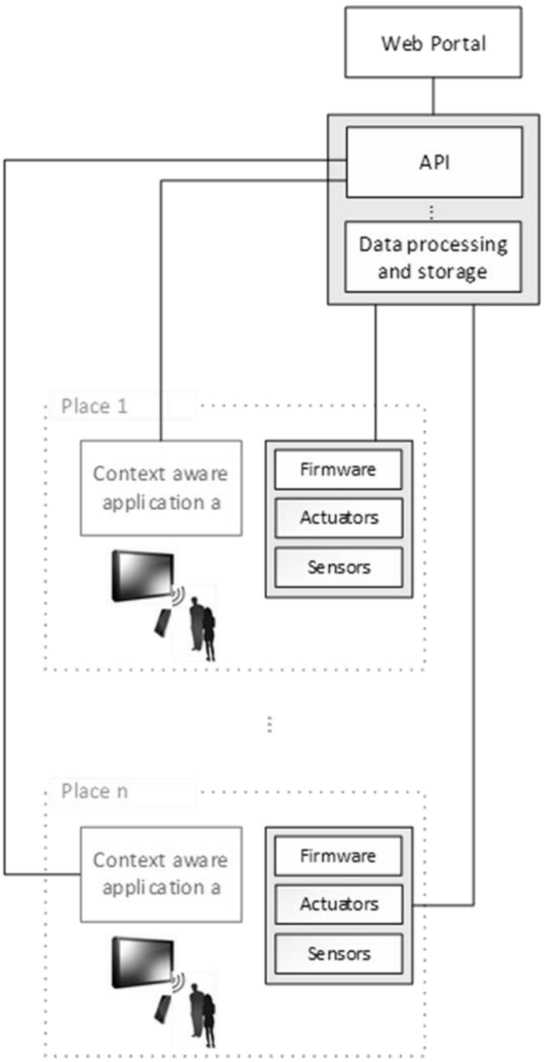

Fig. 2. System architecture and technologies.

API: This module includes a set of functions that can be accessed by external applications and services to get information about objects on the Internet. It includes a set of functions to access information about objects, locations, and about historical data related to each object. This module also provides a set of functions to control some actuators.

Web Portal: It provides the interface to access the data of the objects to manage them and control the accessibility to their data. This module, besides acting as a portal to visualize the information of each sensor or location, it also allows sensor configuration and provides support to manage access to the information of each sensor or location, namely the access to the API functions.

Context-aware applications: Application developers are able to access the API functions according to their permissions and to integrate them into their applications. Through the API they are able to easily connect to external objects and to access contextual information according to the specified object or location. Additionally, they are able to ask the platform about historical data and this information may be used to discover patterns or habits.

6 Experimental work

The experimental work comprised three experiments to assess its overall operation. The goal was to validate the concepts theoretically described, mainly aspects related to the working issues of the core of this system, namely:

- How easy is it to integrate different objects in the platform.
- How easy is it to consume information from the IoT platform in the development of new context aware applications.
- How it helps in monitoring and configuring objects.

6.1 Integrate Things in the platform

As referred before, Things can be integrated in the platform using two different methods. One method is for, the owner to register their Things through the Web Portal. To do this the owner needs to fill in the form with the object information: location, type of data and the time period for sending readings. After this register, the system provides a ThingID to be used by the owner. Alternatively, the Thing may register itself in the platform. In this case, Things should send the platform the data to identify the correspondent owner, location, type of data and level of access. Arduino based sensor (e.g. Arduino Yún and DHT22 digital temperature and humidity sensor DHT22) runs the standard code for reading DHT22 and the user adjusts the message fields to correctly identify the sensor type, location and access level. By utilizing this method users are able to define and use customized objects.

6.2 Using the platform to obtain contextual information

Information about Things can be used as contextual information associated to the environment where they are situated. Through the platform, this information can be easily accessed by information consumers (e.g. context-aware application developers), without the burden of dealing with particularities of each sensor or each communication technology. The API, provided by the platform, allows information consumers to get the information they want according to specific customization. Using these functions, it is possible to get information about a specific location and/or types of information and about some historical statistical data. Below are some examples of functions to get information about Things:

– Get information about the user's Things.
  Return: List of Things (ThingID, type of Thing, location). Empty if none.

– Get information about all Things in the location ID "Classroom".
  Return: List of Things (ThingID, type of Thing). Empty if none.

– Get information about "Temperature" on the location ID "Classroom".
  Return: List of Temperatures, if more than one Temperature Thing in this location. Empty if none.

– Get information about all Bluetooth Things in the location ID "Classroom".
  Return: List of Bluetooth devices (Bluetooth ID). Empty if none.

6.3 Monitoring and configuring Things

The Web Portal provides the user with an interface to configure and monitor their things. Through this portal users are able to add or remove Things to their accounts, configure parameters of their things (e.g. time period between readings for humidity sensors) and visualize data and historical data about each Thing.

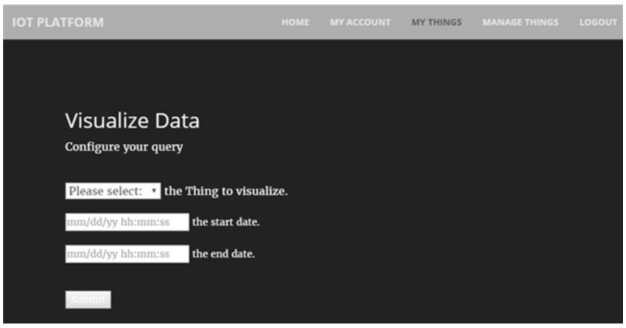

Fig. 3. Associate and configure an Arduino based sensor.

The web portal allows users to personalize their queries about objects, namely, selecting the type of Things or location. Fig. 4, shows the result of a query to list the user Arduino based sensors (humidity and temperature).

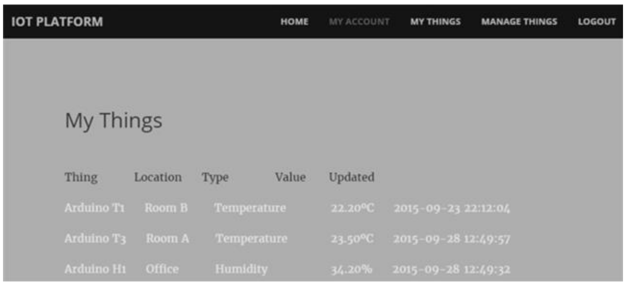

Fig. 4. List of Arduino based sensors.

Additionally, it is possible to analyze historical data related to each Thing and personalize the time period that the user is intending to visualize (see Fig. 5).

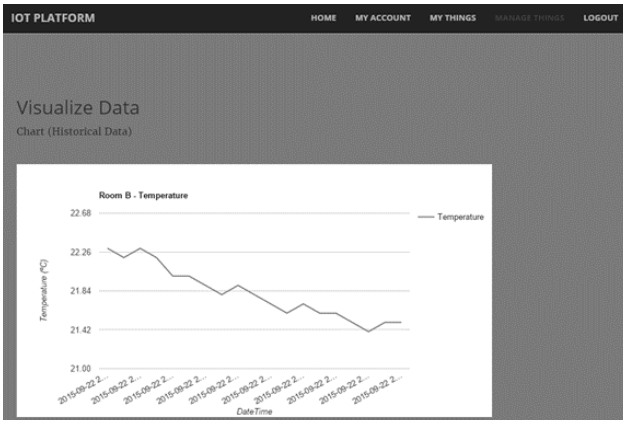

Fig. 5. Graph with historical data related to a humidity sensor.

7 Using the platform to facilitate the development of context-aware applications

In this section, we describe two sample scenarios where context-aware application developers may use the platform to handle with contextual information.

7.1 Scenario 1

An application was developed to be used in a common area of a company that is visited by their workers. The system includes an Arduino based Bluetooth Scanner and a large public display presenting news and advice. The display is connected to a Raspberry pi that receives an URL and displays on the screen the correspondent content. The goal is to present on the display the content that is more appropriate for the place visitors at each moment.

The scheduling algorithm is quite simple. Using the user Bluetooth device identification and exploring the history of information associated to the contents that have been shown to each user in the past, the system should present the content that maximizes the exposure to the current audience, i.e. the system firstly presents the content that a higher number of place visitors have not seen.

Using the platform, the application developer only needs to integrate the Arduino based sensor indicating the platform address and the sensor configuration. Afterward, they get the information about the users that are visiting the place from the platform, using the appropriate API function. Once the algorithm decides which is the next content to be presented the application sends the platform the information to identify the display (Raspberry pi) and the url that identifies the content. Of course it also need to register the information about users and items presented in the display.

7.2 Scenario 2

In the first scenario the system adapts its behavior according to the users in the audience. However, unless the system knows the association between the Bluetooth device and its owner, its decisions are based on the fact that the system knows the user' identifier, but it does not know who they really are.

Consider the following scenario:

A company office that has installed two Arduino based sensors. One is an Arduino based digital temperature and humidity sensor, installed within the office. The other is an Arduino based RFID Reader installed near the door.

The application developer integrates the Arduino based sensors in the platform using standard code for these sensors but includes the correspondent configuration for each sensor, i.e. the platform address, the sensor type, the access level and the time period for readings. In this case, there is an identification of the person who carries the identifier. Thus, it is possible to make the association to the worker profile and use this information to select the content (e.g. news) that is according to their interests. Additionally, the developer may ask the platform what the temperature and humidity of the office is and use this information control the air conditioning.

8 Conclusions and future work

This work has discussed the development of an Internet of Things platform to facilitate the development of context aware applications. We have analyzed some previous works with similar goals and we have discussed some challenges that need to be considered when developing an IoT platform. Based on these assumptions, we have proposed and developed an IoT platform that allows to easily integrate IoT objects, manage them and make information about them available to be used by context-aware applications using appropriate API. We have performed some experiments to evaluate the proposed platform from different perspectives, namely: how easy is it to integrate different objects in the platform; how easy is it to use information from the IoT platform and how it helps in monitoring and configuring Things. Overall, experiments suggest that this is a viable approach to manage Things on the Internet and to facilitate the development of context aware applications, allowing developers to concentrate on the development of their applications and release them from the burden of dealing with the particularities of each sensor and the difficulties of storing and processing context.

For future work we intend to extend support for more sources of information (e.g. content from web sources in the Internet), different actuators and also extend the API (e.g. statistical functions, exploit the historic data in order to derive new information). Additionally, we need to perform more experiments with different sensors and actuators and some evaluations performed by context-aware application developers.

References:

[1] *Global information infrastructure, internet protocol aspects and nextgeneration networks*, ITU-T, 2012.

[2] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, *et al.*, "Internet of things strategic research roadmap," European research cluster on the internet of things2009.

[3] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146-164, 1/15/ 2015.

[4] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *Communications Surveys & Tutorials, IEEE*, vol. 16, pp. 414-454, 2014.

[5] X. Li, M. Eckert, J.-F. Martinez, and G. Rubio, "Context Aware Middleware Architectures: Survey and Challenges," *Sensors*, vol. 15, p. 20570, 2015.

[6] A. Whitmore, A. Agarwal, and L. Da Xu, "The Internet of Things—A survey of topics and trends," *Information Systems Frontiers*, vol. 17, pp. 261-274, 2015/04/01 2015.

[7] D. Bandyopadhyay and J. Sen, "Internet of Things: Applications and Challenges in Technology and Standardization," *Wireless Personal Communications*, vol. 58, pp. 49-69, 2011/05/01 2011.

[8] C. Yen-Kuang, "Challenges and opportunities of internet of things," in *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, 2012, pp. 383-388.

[9] C. Doukas and F. Antonelli, "COMPOSE: Building smart & context-aware mobile applications utilizing IoT technologies," in *Global Information Infrastructure Symposium, 2013*, 2013, pp. 1-6.

[10] (2015, november 3). *The Internet of Things Smart Products Platform*. Available: https://evrythng.com/

[11] (2015, november 3). *Thingspeak*. Available: https://thingspeak.com/

[12] (2015, november 3). *Carriots*. Available: https://www.carriots.com/

[13] (2015, november 4). *Zetta. An API-First Internet of Things Platform*. Available: http://www.zettajs.org/

[14] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "CA4IOT: Context Awareness for Internet of Things," presented at the Proceedings of the 2012 IEEE International Conference on Green Computing and Communications, 2012.