EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

A Taxonomy and Survey of IoT Cloud Applications

T. Pflanzner, A. Kertesz[1,*]

[1]Software Engineering Dept., University of Szeged, H-6720 Szeged, Dugonics ter 13, Hungary

Abstract

Internet of Things (IoT) systems are realized by dynamic global network infrastructure with self-configuring capabilities, in which things can interact and communicate in the environment through the Internet by exchanging sensor data, and react autonomously to events generally without direct human intervention. Such systems can be utilized in many application areas, thus they may have very different properties. There is a growing number of cloud providers offering IoT-specific services, since cloud computing has the potential to satisfy IoT needs such as standardizing the custom data structures of the devices, processing and visualization tasks. IoT application developers do not only have to decide which cloud provider to use, but they also have to choose which combination of protocols and data structures best fits their application. As a result, it is necessary to know what properties these systems have and to learn to what extent cloud providers support IoT capabilities. In this paper, we address these issues and investigate 23 IoT cloud use cases and perform a detailed classification of them in a survey, and introduce a taxonomy of IoT application properties based on this survey. We also compare current cloud providers supporting IoT capabilities and gather requirements for IoT device simulation to support further research on IoT application development.

Received on 08 December 2017; accepted on 12 February 2018; published on 06 April 2018

Keywords: internet of things, cloud computing, taxonomy, survey

doi:10.4108/eai.6-4-2018.154391

1. Introduction

Internet of Things (IoT) systems recently appeared as a dynamic global network infrastructure with self-configuring capabilities [1], in which things can interact and communicate among themselves and with the environment through the Internet by exchanging sensor data. They can react autonomously to events and influence them by triggering actions with or without direct human intervention. Such systems can be utilized in many application areas, thus they may have very different properties. According to recent reports in the IoT field (e.g. [2]), there will be 30 billion devices always online and more than 200 billion devices discontinuously online by 2020. Such estimations call for smart solutions that provide means to interconnect and control these devices in an efficient way. Cloud computing [3] enables flexible resource provisions that has become hugely popular for many businesses to take advantage of responding quickly to new demands from customers. There is a growing number of cloud providers offering IoT-specific services, since cloud computing has the potential to satisfy IoT needs such as hiding data generation, processing and visualization tasks. While each provider offers its own set of features, two critical features they all have in common are the ability to connect devices and to store the data generated by those devices. IoT application developers do not only have to decide which cloud provider to use, but they also have to choose which combination of protocols and data structures best fits their application. As a result it is necessary to know what properties these systems have, and to learn to what extent cloud providers support IoT capabilities. The main goal of this work is to provide an insight to these issues.

There are already some related work categorizing IoT systems and applications, but we found that there is still room for further investigations. To facilitate further research in IoT Cloud systems, we need to reveal current IoT application properties in more detail. Therefore in this paper we study and investigate 20 IoT cloud use cases in a survey, and perform a detailed classification of them in a taxonomy.

*Corresponding author. Email: tampfla@inf.u-szeged.hu

To aid the design, development and testing processes of these systems, simulating IoT devices or sensors could be useful to emulate real-world systems. Therefore we also gather the requirements for basic functionalities of such a simulator in the light of the previously presented taxonomy, i.e. to send and receive messages, generate sensor data (for one or more devices), and react to received messages.

The main contributions of this work are: (i) a survey and classification of IoT applications, (ii) a taxonomy of IoT application capabilities and properties represented by the previous IoT use cases, (iii) a survey of cloud providers supporting IoT capabilities, and finally (iv) a requirement analysis for an IoT device simulator to support further research on IoT application development.

The remainder of this paper is as follows: Section 3 introduces the overviewed IoT use cases and Section 4 presents their classification in a taxonomy. Section 5 discusses IoT capabilities of current cloud providers and Section 6 introduces the requirements for a general purpose IoT device simulator gathered from the survey. Finally Section 7 concludes the paper.

2. Related work

There are already some related work categorizing IoT systems and applications. Gubbi et al. [4] were one of the pioneers by describing IoT systems in 2013, and they also emphasised the importance of the joint utilization of IoT and cloud systems.

Atzori et al. [5] examined IoT systems in a survey. They identified many application scenarios, and classified them to five application domains: transportation and logistics, healthcare, smart environments (home, office, plant), personal and social, finally futuristic domains. They described these domains in detail, and defined open issues and challenges to all of them. Concerning privacy, they stated that a lot of information about a person can be collected without the person being aware, and control on all such information is impossible with current techniques.

In a recent paper, Ray presented a survey of IoT Cloud platforms in [7]. This survey identifies several service domains IoT cloud platforms should deal with. This domain specific survey argues that application development and monitoring management are the mostly served domains, while visualization capabilities and open source IoT APIs are less provided. Our research is more application-oriented, and real-world properties of IoT use cases are also revealed in our study.

Want et al. [6] set up the following three categories. The first one is the group of composable systems, meaning ad-hoc systems that can be built from a variety of nearby things by making connections among these possibly different kinds of devices. As these devices can discover each other over local wireless connections, they can be combined to provide higher-level capabilities. The second one is the category of smart cities, including utilities of modern cities that could be managed more efficiently with IoT technologies, e.g. traffic-light systems can be capable of sensing the location and density of cars in the area, and optimizing red and green lights to offer the best possible service for drivers and pedestrians. Finally, the third one is resource conservation, in which there are applications performing the extensive use of Internet-connected, networked sensors major improvements can be made in the monitoring and optimization of resources such as electricity and water. After studying these works we found that there is still room for further investigations, and to facilitate further research in IoT Cloud systems, we need to reveal current IoT application properties in more detail. Therefore in this paper we study and investigate 20 IoT cloud use cases and perform a detailed classification of them in a taxonomy.

3. A Survey of IoT Cloud Use Cases

In order to reveal properties of current IoT applications, we selected and investigated 20 IoT cloud use cases from various application areas coming from three main groups: smart home, smart city and smart region. In this section we introduce and describe these relevant applications, and in the following section we study and gather real-world IoT properties of these cases, and compile them into a taxonomy.

3.1. Smart home applications

The Green Plug [18] is a smart home project that enables users turn on and off appliances remotely, they also enable real time energy consumption by measuring current and voltage. A magnetic sensor is placed near to the outlet, and the changes in the field can be used to conclude the electric usage.

The Mimo [8] project develops smart home products that are created for remote monitoring of babies. It measures baby body functions such as breathing, temperature, body position and activity level. It can send occasional alerts and nightly reports to a smartphone by using ultra low-power Bluetooth connection. Caretakers can see sensor information in real-time. It also supports special features to interconnect other devices, e.g. if the baby temperature is not optimal a smart thermostat can change the room temperature, or if the baby is moving a webcam can be switched on to visually monitor the baby.

The Vitality GlowPack [9] solutions offer health-care smart home capabilities. They propose a smart pill bottle top as a replacement for the standard one. It connects via Bluetooth to the users smartphone and

an application reminds the patient to take the pills at the right time. There is a lamp unit feature of the product: the lamp and the bottle top starts to flash, if the bottle is not opened when the patient needs to take the medicine. If still unnoticed, it is able to play music or send an SMS, even initiating a phone call. Since pill taking is not so frequent activity, there is no need for a high speed network connection for operation. There are additional features upon request: patients can be alerted if some pills cannot be taken at the same time, and the local pharmacy can also be interconnected to order additional medicine, when the user runs out of them.

Nest Thermostat [10] is one of the Nest products for smart homes. With this samer thermostat one can control room temperatures in a house. The use of a single thermostat does not require high-speed network connection. Their concept is that the IoT devices can interact without the need of a configuration. Supported devices like smart lamps can use the already used Nest Thermostat and use its data and learn from it. There are more and more new products with Nest support, this can be a good example of more devices working together. The Nest API helps the interaction between different kind of network capabilities to work together.

Smart Outlets [11] are designed to implement the smart electrical outlet concept. A user can remotely control the appliances, or set timers from a smartphone. The number of IoT devices can be varied, but in average we can say the application needs a medium size environment. It can communicate with different networks, but WIFI is the default setting. Its monitoring and energy saving capabilities are very useful.

Key Finder Tags [12] are location sensors that can be attached to utilities as key fobs or stick-on tags. The default version has Bluetooth connection and can make a beeping sound or light signals to warn users. Other versions have own cellular data connection and GPS sensors, so that they can also report exact localization information. Using this feature, smartphones can request devices to show their positions. Advanced tags support a useful reverse (actuator) function: if the user has the smart key fob, but his/her smartphone is missing, when pushing the fob, the phone will signal its position.

The Samsung company offers many smart home solutions. Their smart refrigerator [25] is a good example for them. It has a big touchscreen, where the family calendar can be shown, note taking and photo displays are aslo supported. It has three built-in cameras, therefore every time the door closes, newly taken photos can be sent to registered smartphones.

3.2. Smart city applications

The SmartPile [26] system is an interesting solutions for monitoring the condition and quality of the concrete in building constuctions. It offers wireless devices to be built in the concrete and later gives information during curing (e.g. core and skin temperature and compression strain), transport and installation (e.g. compression, strain and load capacity).

Wireless Plant Sensors [13] help to take better care of plants. They can be composed to build indoor or outdoor sensor systems using WIFI connection to send status information about the monitored plants. They provide different algorithms for different plant types to water them. They also offer timers not to forget to water the plants. The size of the system may vary from few home plants to an industry size system.

The Bigbelly [14] is a smart waste and recycling system that helps to figure out if a particular trash can needs to be emptied. It is a solar powered system, so no electricity is needed for fullness level sensing. It offers communication with the Bigbelly cloud. The system is designed to provide smart trash cans from home to city scale.

The Smart Parking [19] system can detect if the parking spot is reserved with a magnetic sensor. The cellular network is an option to send the data to the Save9 cloud and use it to provide smart parking solutions.

Echelon [15] offers an open standards-based approach to lighting control, hence outdoor lighting is an important function for enterprises and cities to increase the efficiency and safety of their lighting systems.

The Optoi chemical sensors [20] are silicon devices mainly based on microelectrodes and specific sensing layers, such as silicon nitride for pH measurement. These devices for water monitoring can be integrated into a multiparametric Microsystem together with Conductivity, Red-Ox potential, Temperature and other sensors for water monitoring applications.

3.3. Smart region applications

The project called Assessment of Landslides using Acoustic Real-time Monitoring Systems (ALARMS) [16] utilizes low cost, miniaturised acoustic sensors integrated with wireless networking capability (GSM) to provide early warning of slope instability. If the acoustic soil activity reaches a pre-defined threshold, the system sends an SMS message. ALARMS uses a steel tube called a wave guide to conduct the signals out of the ground. The steel tube is placed in a borehole in the ground that is filled with sand or gravel. The sand and gravel produces more energy when moved than the surrounding soil, which makes the signal easier to detect.

An example application for wildlife tracking is the Open Source Lion Tracking Collars (OSLTC) [17], which is an open source collar system to help conservationists protect the last lions living in the wild in Southern Kenya. It is also used to safeguard the Maasai herders cattle, with the aim to restore Maasai-land to a working ecosystem.

With the Phenonet Project [21], plant breeders can evaluate the performance of different wheat varieties using measurements taken from remote sensors. These sensors monitor things like soil temperature, humidity, and air temperature and are often used for crop variety trials. This allows farmers to forecast harvest time, improve plant health, plan irrigation time, and determine frost and heat events.

Companies like Govtech [22], AquamatiX [23] and i2O [24] provide solutions for cities to better control the flow of water by embedding sensors in water pipes throughout the distribution network and connecting them to pump control systems. These sensors monitor water flow, feeding the data back to facilitate optimized water pumping throughout the system. By minimizing the amount of water in the pipes, cities can reduce the amount lost to leakage and prevent the formation of new leaks. In the process, the system also saves energy by reducing the need for pumping. Moreover, by distributing water monitoring throughout the network, these technologies can detect abrupt events, like bursts, facilitating faster response and minimizing water loss.

The latest GE Evolution Series Tier 4 Locomotive [27] is equipped with 250 sensors to measure staggering 150,000 data points in a minute. This data combined with other incoming streams of data from informational and operating systems to monitor anticipating events and to support taking driving decisions in real time.

Caterpillar [28] is helping its dealers to succeed with help of IoT for industrial analytics. Company is harnessing data it collects from its industrial locomotives like engines, machines and tools and shares the analyzed data insights with its customers. It helps them to anticipate problems, manage fleets and schedule maintenance proactively.

Based in Ireland the CleanGrowâĂŹs [29] project helps with monitoring the crop nutrients making use of a carbon nanotube-based sensor system. This information helps farmers to alter maturity rate or color of the crop production. As opposed to analog devices used conventionally the CleanGrow device uses a nanotube sensor that detects quantity and presence of specific ion in the production.

OpenWeatherMap [30] is a service, which collects Weather data from thousands of remote and ground sensors distributed all around the world and is expanding permanently. The collected data can be accessed with an API with more than 200,000 cities. A weather forecast service is implemented based on the sensor data.

4. A Taxonomy of IoT Application Properties

Based on the survey presented in the previous section, we gathered the most relevant properties of IoT applications and classified them into categories of our proposed taxonomy. We revealed five top-level categories: Context, Participants, Sensors, Network and Other.

4.1. Context

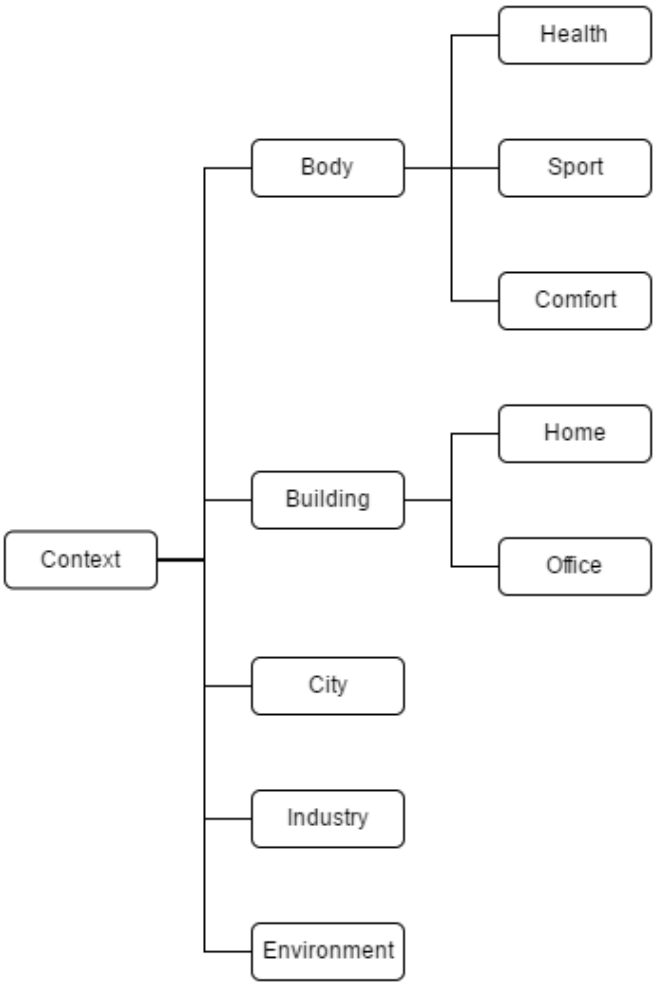

Figure 1. The Context category of the taxonomy

We named the first top-level category as *Context*, its structure is depicted in Figure 1. This is also the most usual viewpoint in earlier related works. Its categorization is based on the types and fields of the overviewed applications. From this category one can determine key features of an application, considering its scale and area of utilization. One of its sub-categories is the *Body*, which contains use cases providing IoT

sensors measuring body functions. The purpose of measuring can be related to health, e.g. when patients are examined by doctors, these cases are covered by the element *Health*. The *Sport* element is supposed to cover the training-related applications. Various sporting communities utilize these smart devices to optimize the trainings and track performance, but it is also very popular among amateurs, because of the social network sharing habits. *Sport* and *Health* use cases are close to each-other, but usually the precision is not critical in the first case, it is rather the durability that matters. The third element of the *Body* sub-category is *Comfort*, where corresponding applications help the everyday life to be more relaxed and efficient.

The next sub-category is *Building*, where the sensors are placed inside or attached on buildings. These applications generally utilize sensors, but there are cases using actuators to help the automation of smart homes. Their main purpose is to improve personal life (represented by element *Home*) – since these applications are targeting our homes –, or to improve professional life by supporting a home office environment (element *Office*). Smart homes are one of the first use cases of the IoT world, therefore nowadays we can find some mature solutions in this field.

There are many use cases related to cities and municipalities, where data from a local geographical area is collected and analyzed – denoted by the *City* element. The main challenge is the growth of the distance between the sensors and gateways (which is also a decision point while planning IoT systems). The *Industry* sub-category is usually covered by factories to optimize resources and operation costs. The roots of IoT are in Wireless Sensor Networks, which solutions have been highly utilized in factories. The *Environment* sub-category is to understand and manage natural resources like water, to monitor wildlife or to operate wider-scale agricultural systems.

4.2. Participants

Besides geographical coverage, the number of potential users and devices (or sensors) is also an important property of IoT applications – we named this category as *Participants* (see Figure 2). By analyzing these numbers we can differentiate three category elements: *small-scale* applications, where there are generally less than 10 sensors utilized. *Medium-scale* applications are considering the sensor range of 10 to 100, where the handling of devices require special care, e.g. device groupping. Finally, *large-scale* applications contain more than 100 IoT components. This category covers use cases with a large number of devices, therefore system scalability becomes critical here.

The data handling approaches depend on the number of users, which is covered by the element: *Number of Users*. Access control is usually not a critical issue nowadays, but the way the managed data to be published/provided requires special attention. Similarly to the previous element, we can identify three main groups: the *small-scale* with less than 5 users, the *medium-scale* with the range of 5 to 50 users, and if the data have to be available for more than 50 users, we are talking about *large-scale* applications.

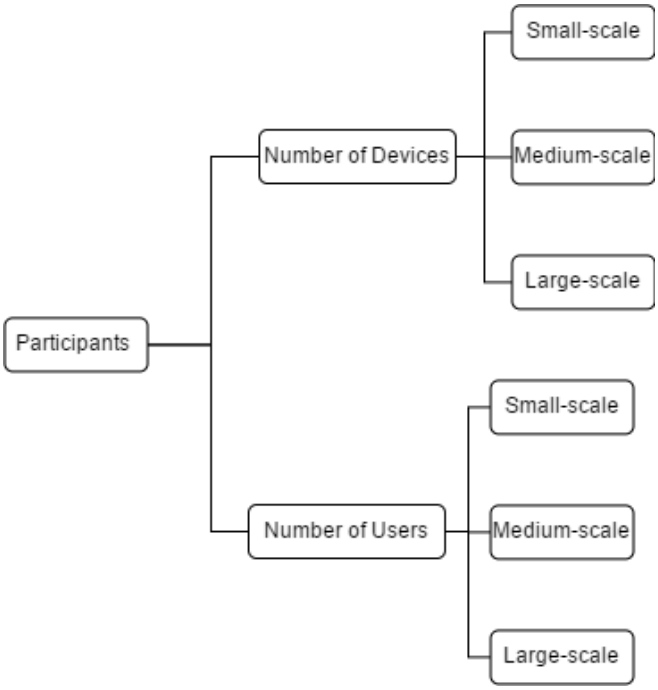

Figure 2. The Participants category of the taxonomy

4.3. Sensors

The biggest top-level category is the one called *Sensors*, which is used to define all sensor-related properties as shown in Figure 3.

The main purpose of sensors to measure something to work with. We revealed four sub-categories: *Vision*, *Position*, *Physical* and *Other*. *Vision* contains sensors capable of monitoring visual effects, hence we have the following elements: *Photo*, *Video* and *Light* sensors. The *Position* sub-category refers to sensor placement and physical location changes having the elements: *State*, *Local* and *Global*. The open/close or tilt sensors are represented by *State* sensors. *Local* sensors provide information about position changes within a local environment, denoted by elements: *Motion*, *Velocity* and *Displacement*. The *Global* sensor positioning contains the widely applied GPS sensor. The *Physical* sub-category covers sensors monitoring the physical world. It has the following elements: *Force*, *Load*, *Torque*, *Strain* and *Pressure* sensors. Finally, the *Other* sub-category aims to cover the remaining, widely used sensor areas

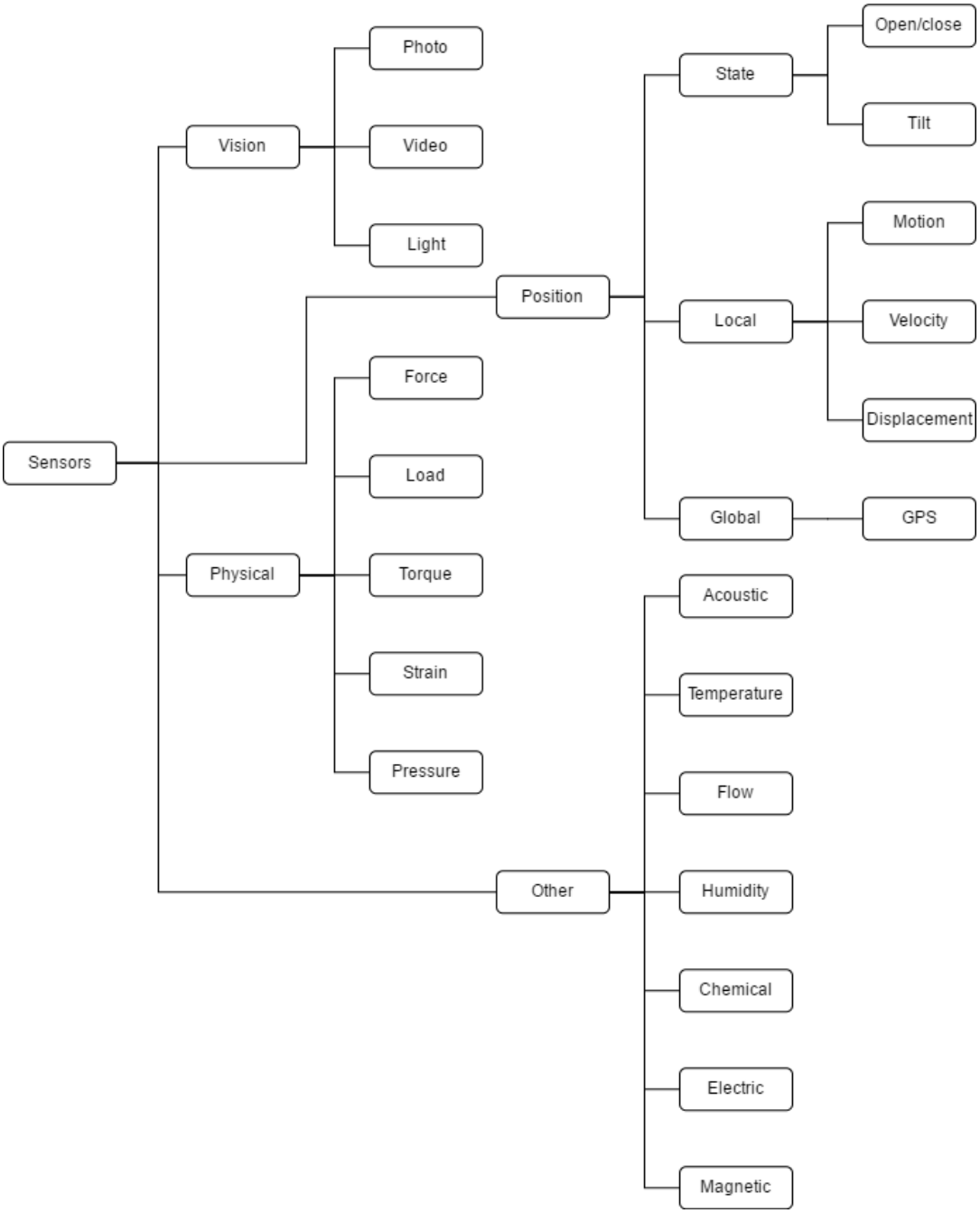

Figure 3. The Sensors category of the taxonomy

by the *Acoustic*, *Temperature*, *Flow*, *Humidity*, *Chemical*, *Electric* and *Magnetic* elements.

It is typical to combine more sensor types to create special-purpose devices for solving a problem. In the comparison table of use cases (to be shown later in Figures 1 and 2), we only highlight the most relevant sensors, but in the application descriptions additional (composed) sensor types can also be found where applicable.

The type of the sensor and additional configuration informations can be used in the cloud application for validating the data, notice hardware errors or just filter a few damaged data packet. For example there are sensors which sending data periodically and sensors are sending data if some event happens. If there is no data from a sensor where should be data in every sec, the application can suspect some network error. Further more the cloud application have to know the type of the sensor to know what kind of commands can handle.

4.4. Network requirements

Next to *Sensors* the other most important category is *Network requirements*. This category has five sub-categories to cover network-related properties. The *Network Speed* requirement differentiates applications to three groups: *Slow*, *Medium* and *Fast*. To be more precise on network requirements we can further categorize the applications based on *Message Size* and data generation *Frequency*. The *Message Size* sub-category can be *Small*, *Medium* or *Large* based on the size of one sensor data sample. The *Frequency* can be *Rare*, *Medium* or *Often*, referring to the sampling timings. The *Network Speed* requirement usually correlates to the *Message Size* and the *Frequency* properties, but there are examples for unexpected setups. For example, if a medium size message is rarely sent to the cloud, the network speed requirement could be slow or medium, but if the data is important to refreshed for a trigger, for example in a security device, the network speed should be fast. The *Connection type* sub-category contains many protocols, and they can be separated into two groups: *Short range* and *Large range* connections. The *RFID*, *NFC*, *Bluetooth*, *Zigbee*, and *Z-wave* are considered as short range types, while *Cellular network*, *Wifi* and *TCP/IP* belong to the large range group. The final sub-category related to network requirements is the *Sensor availability*. We can talk about *Fixed devices*, where the location of the sensors are constant, or *Moving devices*, where the devices can change positions relative to each other. The other possibility is an *Ad-hoc* network, where the devices can connect and disconnect to/from the network: in this case the number of active sensors can vary in time.

4.5. Other

The previously unclassified properties are groupped to the *Other* category shown in Figure 5. Its elements differentiate IoT applications with three sub-categories: the *Connect to* sub-category defines where the IoT application sends its data. It can be transfered directly to a smartphone or other *Mobile* device, to a *Local gateway* service (possibly running on a local, physical device), or to cloud gateway running in a *Private cloud* or *Public cloud*. The *Energy consumption* sub-category reveals information on power utilization, which can be *Low* or *High*. This is an important factor for cost efficient systems, and it reminds managers to calculate with battery lifetime for moving sensors or large-scale networks. The Security sub-category refers to the security-level of data handling. Here the data management can be *Secret*, or *Protected* or *Public*. If the data is Secret, it means that only the owner can access it, for example if its a personal thing, like a medicine handling device. The validity of the data is important too. The Protected data means the data can be shared with other people, but these people must identify themselves. The Public data does not require authentication to access it.

4.6. Categorizing the Overviewed IoT Applications

Finally we categorized the previously seen IoT applications to the presented taxonomy. In Figures 1 and 2 we map the IoT use cases introduced in the survey to the categories of the taxonomy.

It can be seen from the tables that the number of users are usually small-scale and the number of devices are usually medium-scale. This can be explained with the complexity of deploying large-scale systems. The sensor types and the context of the use cases are really diverse, this comes from nature of these systems to be everywhere and help our life. Usually we talking about devices with low energy consumption, because usually they work with a battery. Regarding to the network we can say the frequency and size of the data is not too big, but if we take into account the type of the networks, it can be seen that the wireless networks are the majority, and the bandwidth and error rate of these network is not as good as in wired networks.

5. Solutions for Cloud-assisted IoT Applications

The Cloud Computing has many benefits, such as good resource management, scalability, the capability of handling big data, and that it can be accessed from anywhere. Due to these capabilities, clouds can be useful when we try to store and process data coming from IoT environments. To operate such systems, there are already some cloud providers supporting IoT features. These basic features are to connect the devices to the cloud, store, process and/or visualize data. In this section we overview and categorize providers supporting some of these IoT capabilities.

The IBM Bluemix Platform [31] is an IoT-enabled cloud solution offered by IBM. It can be used for the development of cloud-based applications managing data generated by various sensors and devices. Products of several major device manufacturers are supported, such as ARM, the Electronics B&B, Intel, Multi-Tech Systems and Texas Instruments, but other individual cases can also be connected to the platform. Data generated by an IoT device is received by the popular and lightweight MQTT protocol connected to the cloud. The service allows users to configure and manage devices, to store the history of generated data, or to stream real-time data to a cloud application. The Bluemix platform offers several specialized services to support the development of cloud applications, i.e. push for messaging, Cloudant NoSQL DB to manage NoSQL databases, Geospatial Analytics for location tracking, and IBM Analytics for Hadoop computations. Secure data transfers are supported

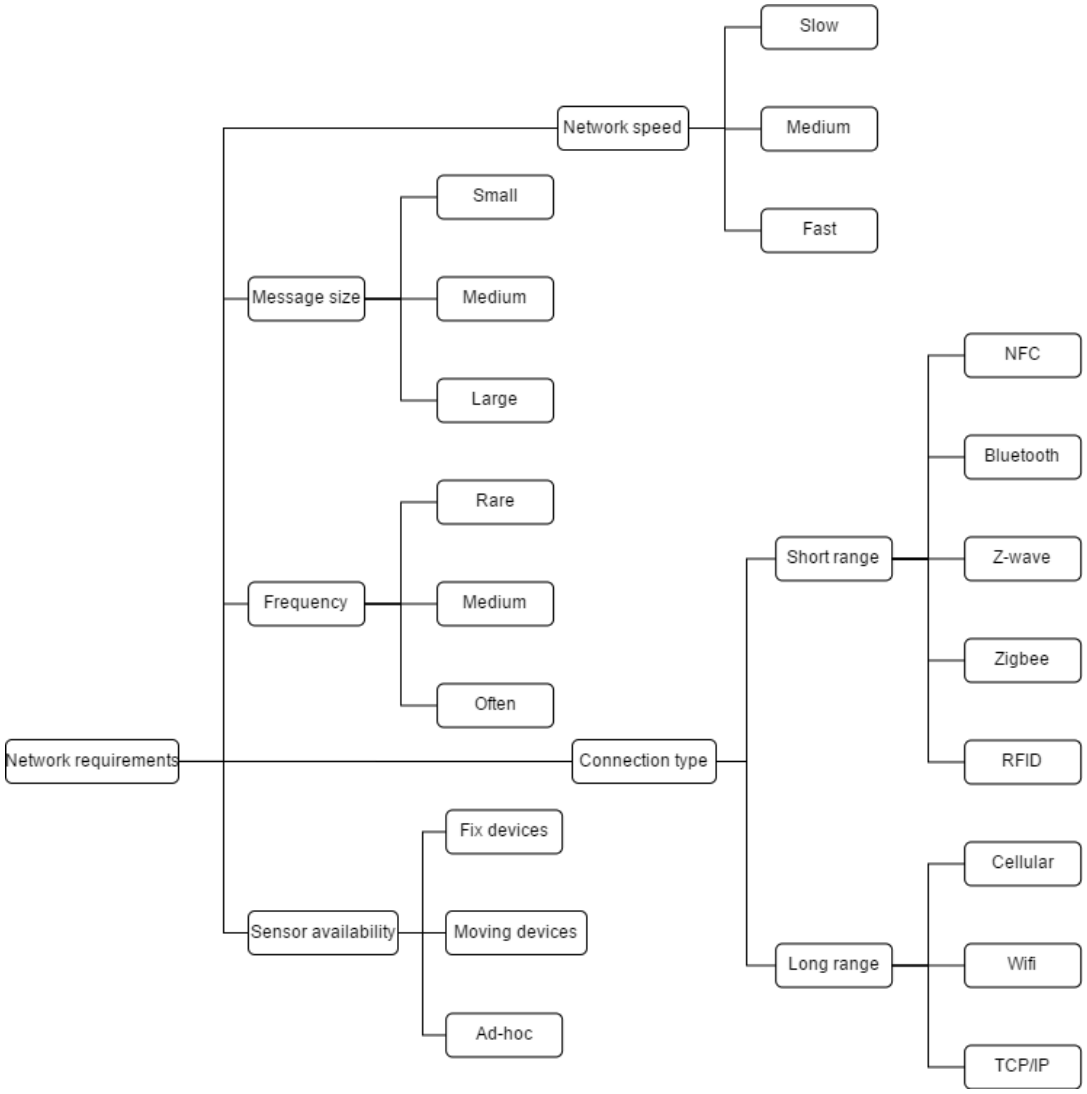

Figure 4. The Network requirements category of the taxonomy

by APIs. The supported languages for application development are Java, JavaScript, GO, PHP, Python and Ruby. Considering costs, a price calculator helps to determine a monthly fee for a 30-day trial period. To illustrate the inner workings of the platform, a real-time data visualization demo is also provided. To use it, first a data provider should be configured, which is in the simplest case a smartphone, but it is possible to use a TI SensorTag, ARM Mbed, Raspberry Pi, Intel or other devices. The opened browser page of the demo on a smartphone can send real-time data of the phone's movement to the cloud application. The framework also provides a pre-defined web-based sensor simulator [32] that is able to act as simulated sensors (sending temperature and humidity values through websockets).

The Parse [40] platform also has IoT support, and it was developed by Facebook. It promises quick and easy application development with the support for mobile devices (through its Mobile-Backend as a Service (MBaaS) solution). It supports C SDK for Linux (Raspberry Pi) and real time systems (RTOS) (TI CC3200), and some more specific devices are also managed, such as Arduino. Some partner companies have SDKs for data sending and push notifications, such as Atmel, Broadcom, Intel and Texas Instruments, and they have the great benefit that all of these SDKs are open source. Many sample applications are available to demonstrate its usage, including farming, music and cooking scenarios. JavaScript applications are supported, and the Parse Webhook service makes it able to link applications from remote clouds. The supported mobile platforms are the iOS, Windows Phone, Android, Unity and Xamarin. Web and desktop SDKs are for OSX, Windows, JavaScript, Unity, PHP, and .NET. The SDK supports offline data storing, social media connections can be properly handled.

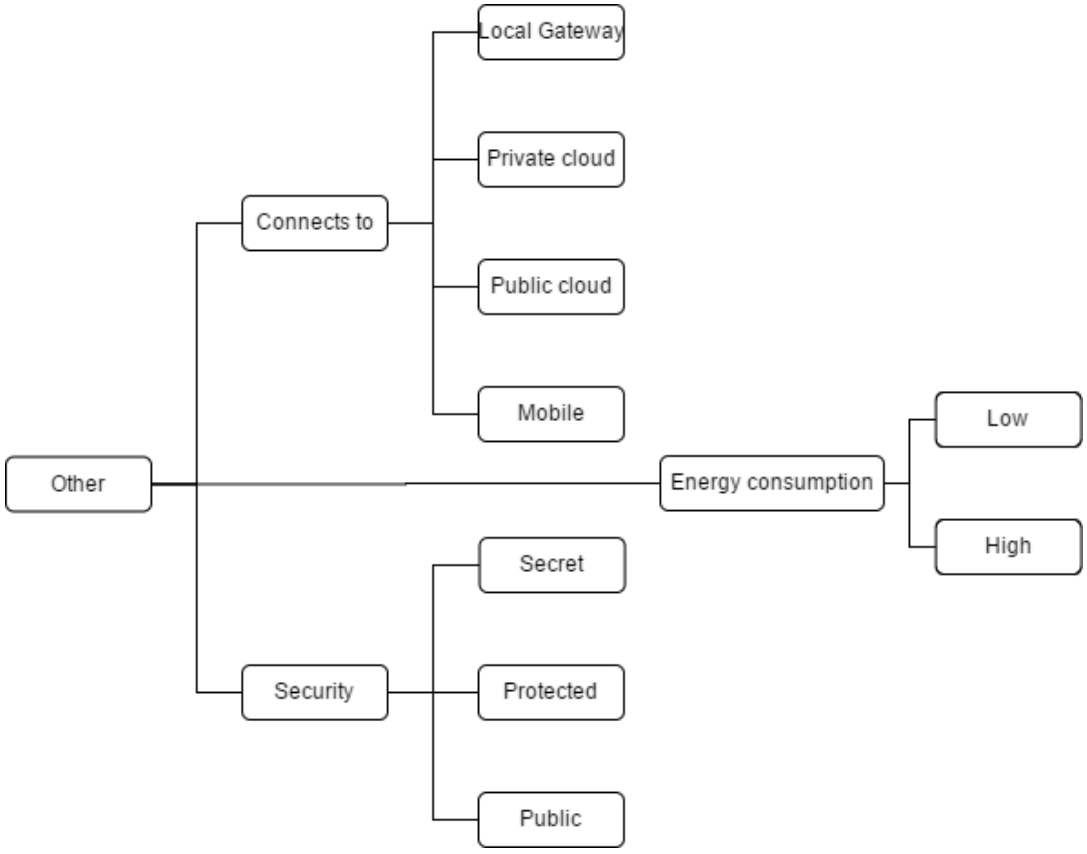

Figure 5. The Other category of the taxonomy

It is possible to schedule jobs, as well as a high quality Dashboard is also available that supports data modification, statistical testing, push notification, and storing logs. Objects can be managed in the system after specifying their schema. Their schema supports many field formats, e.g. arbitrary objects, array, blobs and geopoints. Unfortumately, their dashboard provides no tools for data visualization. Blobs can be imported from a public URLs, even directly from the Base64 encoded Data URI format.

The Google IoT solution [33] is part of the Google Cloud Platform, which includes various Google services. Scalability is an excellent feature of this platform. It allows devices to be connected, and it collects data and visualizes them. The data sent from the devices are received by the Google Load Balancer and forwards to instances of the AppEngine applications. In general, the main part of the application is the AppEngine, which may use other services. Compute intensive tasks are supported by the Compute Engine. The Cloud Storage and the Cloud SQL manages data. It is possible to send data with streams to the BigQuery service, which is ideal if we want to work with real-time data. Visualization is supported in real time using Google Charts. Google is also strong in managing a large amount of data processing, which is important for IoT cloud systems, since there might be many devices generating a huge amount of data. Google FireBase plays an important role in the management of the devices. It was originally designed to assist mobile devices (like MBaaS). It provides synchronized real-time database, authentication and capable of offline operations.

Amazon Web Services [34] is a collection of services that make up a cloud computing platform, which are based on 11 geographical regions across the world. Though Amazon mostly offer services at the IaaS level, it can also be regarded as a cloud platform provider, since it has many "blocks" to build applications with. The most central and well-known services are Amazon EC2 (Elastic Compute Cloud) and Amazon S3 (Simple Storage Service). The products are offered to large and small companies as a service to provide large computing capacity faster and cheaper than the client company building and maintaining an actual physical server farm. AWS automatically handles the details such as resource provisioning, load balancing, scaling and monitoring. One can create applications in PHP, Java, Python, Ruby, node.js, .NET, Go or in a Docker container that runs on an application server with a database. An environment using the default settings will run a single Amazon EC2 micro instance and an Elastic Load Balancer. Additional instances will be

Table 1. Comparison table of IoT Use Cases - Part I

Name	Context	Nr. of Devices (scale)	Nr. of Users (scale)	Sensors	Connects to	Energy cons.
Mimo [8]	Body / Health	Small	Small	Motion	Mobile	High
Vitality GlowPack [9]	Body / Health	Small	Small	Open/Close	Mobile	Low
Nest Thermostat [10]	Building / Home	Medium	Small	Temperature	Private cloud	Low
Smart Outlets [11]	Building / Office	Medium	Medium	Electric	Private cloud	Low
Key Finder Tags [12]	Building / Home	Small	Small	GPS	Mobile	Low
Wireless Plant Sensors [13]	Industry	Medium	Medium	Photo	Public cloud	High
Bigbelly [14]	City	Medium	Small	Load	Private cloud	Low
Echelon [15]	City	Large	Small	Light	Private cloud	High
ALARMS [16]	Environment	Large	Large	Acoustic	Public cloud	Low
OSLTC [17]	Environment	Medium	Small	GPS	Private cloud	Low
Green Plug [18]	Building / Home	Medium	Small	Magnetic	Mobile	Low
Smart Parking Sensor [19]	City	Large	Medium	Magnetic	Private cloud	Low
Optoi [20]	City	Small	Small	Chemical	Private cloud	Low
Phenonet Project [21]	Environment	Medium	Small	Humidity	Public cloud	Low
GovTech [22]	City	Medium	Small	Flow	Private cloud	Low
AquamatiX [23]	City	Medium	Small	Flow	Private cloud	Low
i2O [24]	City	Medium	Small	Flow	Private cloud	Low
Smart Refrigerator [25]	Building / Home	Small	Small	Photo	Private cloud	Low
SmartPile [26]	Building / Office	Large	Small	Temperature	Private cloud	Low
Locomotive [27]	Industry	Large	Small	Electric	Private cloud	High
Caterpillar [28]	Industry	Medium	Medium	Electric	Private cloud	High
CleanGrowâĂŹs [29]	Environment	Large	Medium	Chemical	Private cloud	Low
OpenWeatherMap [30]	Environment	Large	Large	Temperature	Public cloud	Low

Table 2. Comparison table of IoT Use Cases - Part II

Name	Security	Network speed	Message size	Frequency	Connection type	Sensor availability
Mimo [8]	Protected	Medium	Medium	Often	Short range/Bluetooth	Moving devices
Vitality GlowPack [9]	Protected	Slow	Small	Rare	Short range/Bluetooth	Moving devices
Nest Thermostat [10]	Protected	Slow	Small	Rare	Short range/Zigbee	Fix devices
Smart Outlets [11]	Protected	Slow	Small	Rare	Long range/TCP/IP	Fix devices
Key Finder Tags [12]	Secret	Slow	Small	Rare	Short range/Bluetooth	Moving devices
Wireless Plant Sensors [13]	Public	Medium	Medium	Medium	Long range/Wifi	Fix devices
Bigbelly [14]	Public	Slow	Small	Rare	Long range/Cellular	Fix devices
Echelon [15]	Protected	Slow	Small	Medium	Long range/Wifi	Fix devices
ALARMS [16]	Public	Medium	Small	Often	Long range/TCP/IP	Fix devices
OSLTC [17]	Protected	Slow	Small	Medium	Long range/Cellular	Moving devices
Green Plug [18]	Protected	Slow	Small	Medium	Short range/Bluetooth	Fix devices
Smart Parking Sensor [19]	Public	Slow	Small	Rare	Long range/Cellular	Fix devices
Optoi [20]	Protected	Slow	Small	Rare	Long range/Cellular	Fix devices
Phenonet Project [21]	Public	Slow	Small	Rare	Long range/Cellular	Fix devices
GovTech [22]	Protected	Medium	Small	Medium	Long range/Cellular	Fix devices
AquamatiX [23]	Protected	Medium	Small	Medium	Long range/Cellular	Fix devices
i2O [24]	Protected	Medium	Small	Medium	Long range/Cellular	Fix devices
Smart Refrigerator [25]	Protected	Medium	Medium	Rare	Long range/Wifi	Fix devices
SmartPile [26]	Protected	Slow	Small	Rare	Short range/Bluetooth	Moving devices
Locomotive [27]	Protected	Medium	Medium	Often	Long range/Cellular	Moving devices
Caterpillar [28]	Protected	Medium	Medium	Medium	Long range/cellular	Moving devices
CleanGrowâĂŹs [29]	Protected	Slow	Small	Rare	Long range/Cellular	Moving devices
OpenWeatherMap [30]	Public	Slow	Small	Rare	Long range/Cellular	Fix devices

added if needed, to handle any peaks in workload or traffic. Each Amazon EC2 instance is built from an Amazon Machine Image which can be an Amazon Linux AMI or an Amazon Windows Server 2008 R2 AMI by default. Concerning MBaaS, it has three main components: Cognito for user management, Mobile Analytics and Simple Notification Service for mobile-related solutions. Amazon IoT connects devices to services and other devices securely. The device state is synchronized, so messages can be sent even if the device is offline. The Rule Engine helps to convert the data for more specific services.

Azure [35] is a Cloud computing platform, which allows developers to publish web applications running on different frameworks, written in different programming languages such as any .NET language, node.js, php, Python and Java. Azure Web Sites supports a website creation wizard that can be used to create a blank site or use one of the several pre-configured sites. Developers can add or modify content of the website via multiple deployment methods: TFS, FTP, CodePlex, GitHub, Dropbox, Bitbucket, Mercurial or git. Developers can select the place where their website will be hosted from several Microsoft data centers around the globe. Azure Traffic Manager routes traffic manually or automatically between websites in different regions. Web sites are hosted on IIS 8.0, running on a custom version of Windows Server 2012. The component regarding IoT called IoT Hub, which can communicate with devices with protocols like MQTT, AMQP and HTTP, but it's possible to implement other protocols too.

Heroku [36] first appeared in 2007, starting with support for Ruby, and it has added support for many languages till then, such as Java, Node.js, Scala, Clojure, Python, PHP and Perl. Heroku was acquired by Salesforce.com in 2010, as a subsidiary. HerokuâĂŹs services run on the Amazon cloud systems. From the Developer Experience point-of-view, HerokuâĂŹs interface is well-polished, intuitive and easy to use. Many times Heroku has been seen as an example by other PaaS providers, for their ease of use, features and reliability. One such example is Deis, which uses a Heroku inspired buildpack system in their deployments. The basic units of computing power in the Heroku ecosystem are the Dynos. A Dyno is a lightweight, isolated container that runs an instance of the application. The IoT capability of Heroku is only an MQTT broker, which can be added as a 3rd party add-on, called CloudMQTT.

CloudFoundry [37] is an open source PaaS service, originally developed by VMware, later owned by Pivotal Software, primarily written in Ruby and Go. CloudFoundry is available in three flavors, Cloud Foundry OSS, an open source project available to everybody, which uses the developers own infrastructure and the BOSH shell to interact with it, Pivotal Cloud Foundry, a commercial product from Pivotal, which includes extra tools for installation and administration, and Pivotal Web services, which is an instance of Pivotal Cloud Foundry hosted on Amazon Web Services. Applications deployed to CloudFoundry access external resources via Services. All external dependencies such as databases, messaging systems, file system, etc. are Services. When releasing an application, the developer must specify the Services it should use. Many pre-defined services are available via an administration console, such as MySQL, PostgreSQL, MongoDB, etc. as database services, RabbitMQ as a messaging service, Jenkins for continuous integration, and API Gateway, Data Sync, Push Notifications for mobile development.

DreamFactory [38] has many install guides for IaaS providers (Docker, Amazon Web Services, Microsoft Azure, Google Cloud Platform, VMware Marketplace, Bitnami Cloud Hosting), for PaaS providers (Red Hat OpenShift, Pivotal Web Services, IBM Bluemix, Heroku) for Desktop Computer (LAMP or WAMP) (Linux, OS X, Windows) IBM SoftLayer, Rackspace Marketplace. For push notification it uses the Amazon SNS. The SDK is moderately good, too general, requires too much coding. On the free hosted version the login window pops up for almost every click.

The Kinvey [39] platform has 3 main parts, the client, the context and the aggregation part. The client part, called the Mobile Client Tier, is a consistent set of mobile client features like offline caching and encryption that stay current and enable any use case. The context part, called the Mobile Context Tier, has on-demand services for data, push notifications, location, analytics and business logic to support any use case, versioning and deployment of new applications and backend infrastructure for multiple projects. The aggregation part, called the Mobile Aggregation Tier, is responsible for the consistent and secure data and identity integrations to deliver the right information for the mobile experience.

The main IoT-related properties of these cloud providers are shown in Table 3. Summarizing this comparison, Google, Amazon, Azure and Bluemix has the highest variety of IoT-related services. The MQTT (or other IoT protocol) should be a basic functionality to an IoT cloud platform, but many provider has just a REST interface. Parse has its high quality MBaaS service, while its IoT feature is a limited version of it. IoT applications in Google can be composed of many connected services, which makes it complex providing more freedom for the developers. They are also very good at scaling and performance, and this complexity is compensated by the simplicity of FireBase.

Table 3. IoT features of cloud providers.

Provider	Protocols	Data store	BLOB	GEO	Push not.	Trigger	Visualization
Bluemix	MQTT	yes	no	yes	yes	yes	yes
Parse	REST	yes	yes	yes	yes	yes	no
Google	REST	yes	yes	yes	yes	yes	yes
Amazon	MQTT, REST	yes	yes	yes	yes	yes	yes
Azure	MQTT, AMQP, REST	yes	yes	yes	yes	yes	N/A
Heroku	MQTT	yes	yes	N/A	no	no	N/A
CloudFoundry	REST	yes	yes	N/A	no	no	no
Kinvey	REST	yes	yes	N/A	yes	yes	N/A
DreamFactory	REST	yes	yes	N/A	yes	yes	N/A

6. Requirements of an IoT Device Simulator

After introducing the survey of IoT use cases and examining the most popular IoT Cloud providers, we identified the following challenges related to IoT networks by examining the currently available solutions:

- IoT devices are battery powered;
- They communicate using low-power wireless technologies (e.g., IEEE 802.11, IEEE 802.15.4, Bluetooth);
- There are different resource constraints of devices (e.g., on CPU, memory, connectivity);
- IoT networks are very dynamic as network conditions can change rapidly;
- They are heterogeneous: there is a large spread on device capabilities (e.g., powerful cameras, low cost temperature sensors), and there are sources (sensors) and sinks of information (actuators);
- They are very dynamic: the networking environment in a IoT environment is largely unstructured can vary rapidly.

In order to enhance the design, development and testing processes of IoT systems and networks, an IoT device simulator would be useful that can emulate real devices and sensors instead of real resources.

The requirements for basic functionalities of an IoT device simulator are to send and receive messages, generate sensor data (for one or more devices), and react to received messages. These capabilities are sufficient to use the simulator in IoT system analysis. Requirements for advanced functionalities such as simulating network errors, recording and replaying concrete simulation cases, and connecting real IoT devices to the simulator can contribute to the analysis of more realistic system.

There are different kind of IoT environments, hence their static or dynamic properties, and the number of utilized devices can affect the design of such a simulator. For example, a connected house can be regarded as a static environment, because its devices are usually in one place, possibly with wired connection, providing reliable network stability. The dynamic environment is more complex to simulate, in such cases we would like to simulate a broader part of the environment considering WiFi interference, battery lifetime and locations of the devices.

Device templates can be useful to create more lifelike IoT environment simulations faster and easier. There are some typical IoT devices with different behavior regarding the frequency, data structure, meta-data, value range and generalization. The user can almost instantly create a heterogeneous environment, and the capability of fine tuning the devices created from a template is still possible with further configurations.

The data generalization is an important part of a simulator. The most simple solution is to generate random numbers in a range. The range should be defined based on the possible values of a device type. In some cases the value does not change in big jumps, just small steps, so if the simulator generates the next value based on the previous and randomly adds or subtracts a small amount, the result is closer to the reality.

The capability of recording a session of simulation so it can be replayed multiple time with exactly the same values can be a great feature for fine tuning, testing, debugging and validating the cloud application. The replay function can be used to work with data which is not recorded by the simulator, but from other sources, like real life IoT environment trace files.

The network is a critical part of the IoT environments. Unfortunately the network errors can be common in IoT networks, especially because of the wireless technologies and the changing environment. With a mobile simulator the used network can be changed, for example the wifi and 3G, 4G networks can be used.

To validate the whole system, a gateway can be helpful, to visualize all the data came from the simulated IoT environment. This way the general

operation of the cloud application can be checked based on the received values, and if the simulator is capable of receiving commands and react to them, the applications controlling capabilities can be seen. If the simulator uses MQTT, a common IoT protocol, the MQTT broker of the gateway receives the messages, and forwards them to the application and the visualization part too.

Our future work will address these needs, and we have already made a proposal for such a tool called MobIoTSim [44].

7. Conclusion

The Internet of Things is a new trend that composes communicating sensors in a dynamic global network infrastructure. Though this field is fast evolving, there is already a large number of application areas having very different properties. To support the development of IoT systems, it is necessary to know what properties these systems have, and to learn to what extent cloud providers support IoT capabilities.

In this paper we addressed these goals and investigated 23 IoT cloud use cases and introduced them in a survey. We also proposed a taxonomy of these IoT applications composed of 12 categories and 67 properties based on the survey. We also compared current cloud providers having the potential to support IoT capabilities, and gathered further requirements for IoT device simulation to support further research on IoT application development.

Our future work will address the development of a generic IoT device simulator to aid IoT application development with the surveyed providers, based on the identified use cases and requirements. Such a simulator can be used in the future to help cloud application developers to learn IoT device handling and to evaluate test IoT environments without buying real sensors.

8. Acknowledgment

The research leading to these results was supported by the UNKP-UNKP-16-4 New National Excellence Program of the Ministry of Human Capacities of Hungary, and by the European COST programme under Action identifier IC1304 (ACROSS). This paper is a revised and extended version of a conference paper [45].

References

[1] H. Sundmaeker, P. Guillemin, P. Friess, S. Woelffle. Vision and challenges for realising the Internet of Things. CERP IoT report. CN: KK-31-10-323-EN-C, March 2010.

[2] J. Mahoney and H. LeHong, The Internet of Things is coming, Gartner report. Online: https://www.gartner.com/doc/1799626/internet-things-coming, September 2011.

[3] Buyya B, Yeo C S, Venugopal S, Broberg J, and Brandic I, Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems, vol. 25, no. 6, pp. 599-616, June 2009.

[4] Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M., Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29 (7), pp. 1645-1660, 2013. DOI: 10.1016/j.future.2013.01.010

[5] L. Atzori, A. Iera, and G. Morabito, The Internet of Things: A survey. Comput. Netw. 54, pp. 2787-2805, 2010. DOI: 10.1016/j.comnet.2010.05.010

[6] R. Want, S. Dustdar, Activating the Internet of Things. Computer, Vol. 48, No. 9, pp. 16–20, 2015.

[7] Ray PP, A Survey of IoT Cloud Platforms. Future Computing and Informatics Journal (2017), doi: 10.1016/j.fcij.2017.02.001.

[8] Mimo, http://mimobaby.com/. Accessed in December, 2016.

[9] Vitality GlowPack, http://www.vitality.net/products.html. Accessed in December, 2016.

[10] Nest Thermostat, https://nest.com/works-with-nest/. Accessed in December, 2016.

[11] Smart Outlets, http://www.postscapes.com/smart-outlets/. Accessed in December, 2016.

[12] Key Finder Tags, http://postscapes.com/wireless-key-locators. Accessed in December, 2016.

[13] Wireless Plant Sensors, http://postscapes.com/wireless-plant-sensors. Accessed in December, 2016.

[14] Bigbelly, http://bigbelly.com/solutions/stations/. Accessed in December, 2016.

[15] Echelon, http://www.echelon.com/applications/pl-rf-outdoor-lighting. Accessed in December, 2016.

[16] Assessment of Landslides using Acoustic Real-time Monitoring Systems (ALARMS), http://www.bgs.ac.uk/research/tomography/alarms.html. Accessed in December, 2016.

[17] Open Source Lion Tracking Collars, http://home.groundlab.cc/lioncollars.html. Accessed in December, 2016.

[18] Green Plug, http://www.crocus-technology.com/crocus_applications_sensor-internet-of-things.php. Accessed in December, 2016.

[19] Smart Parking Sensor, http://www.save9.com/home/products-and-services/internet-and-wireless-networks/wireless-sensor-networks/. Accessed in December, 2016.

[20] Optoi Chemical Sensors, http://www.optoi.com/en/products/details/chemical-physical-sensors-mems. Accessed in December, 2016.

[21] Phenonet Project, http://www.csiro.au/en/Research/D-61/Areas/Robotics-and-autonomous-systems/Internet-of-Things/Phenonet. Accessed in December, 2016.

[22] Govtech, http://www.govtech.com/fs/Internet-of-Things-Helps-Cities-Manage-Water.html. Accessed in December, 2016.

[23] AquamatiX, http://www.aquamatix.net/. Accessed in December, 2016.

[24] i2O, http://en.i2owater.com/. Accessed in December, 2016.
[25] Samsung refrigerator, http://www.samsung.com/us/explore/family-hub-refrigerator/. Accessed in December, 2016.
[26] SmartPile, http://smart-structures.com/technology/EDC-embedded-data-collector/. Accessed in December, 2016.
[27] GE Evolution Series Tier 4 Locomotive, http://www.getransportation.com/locomotives. Accessed in December, 2016.
[28] Caterpillar, http://www.cat.com/en_US.html. Accessed in December, 2016.
[29] CleanGrow, http://www.cleangrow.com/. Accessed in December, 2016.
[30] OpenWeatherMap, http://www.openweathermap.org. Accessed in December, 2016.
[31] IBM Bluemix Platform. Online: https://console.ng.bluemix.net/. Accessed in December, 2016.
[32] IBM Bluemix IoT Sensor. Online: https://developer.ibm.com/recipes/tutorials/use-the-simulated-device-to-experience-the-iot-foundation/. Accessed in October, 2016.
[33] Google Cloud Platform. Online: https://cloud.google.com/solutions/iot/. Accessed in December, 2016.
[34] Amazon Web Services. Online: http://aws.amazon.com/. Accessed in December, 2016.
[35] Microsoft Azure. Online: https://azure.microsoft.com/. Accessed in December, 2016.
[36] Heroku. Online: https://www.heroku.com/. Accessed in January, 2017.
[37] CloudFoundry. Online:http://cloudinary.com/. Accessed in April, 2016.
[38] DreamFactory. Online:http://www.dreamfactory.com/. Accessed in December, 2016.
[39] Kinvey. Online:http://www.kinvey.com/. Accessed in March, 2016.
[40] Parse. Online: https://parse.com/products/iot. Accessed in March, 2016.
[41] IBM IoT Foundation message format. Online: https://docs.internetofthings.ibmcloud.com/gateways/mqtt.html#/managed-gateways#managed-gateways. Accessed in December, 2016.
[42] IBM IoT Foundation Quickstart application. Online: https://quickstart.internetofthings.ibmcloud.com. Accessed in December, 2016.
[43] IOT Visualization application. Online: https://github.com/ibm-messaging/iot-visualization/. Accessed in December, 2016.
[44] T. Pflanzner, A. Kertesz, B. Spinnewyn and S. Latre, MobIoTSim: Towards a Mobile IoT Device Simulator. 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Vienna, pp. 21-27, 2016.
[45] T. Pflanzner and A. Kertesz, A survey of IoT cloud providers. 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, pp. 730-735, 2016.