Author(s): Mineraud, Julien; Mazhelis, Oleksiy; Su, Xiang; Tarkoma, Sasu

Title: A gap analysis of Internet-of-Things platforms

Year: 2016

Version:

Please cite the original version:

Mineraud, J., Mazhelis, O., Su, X., & Tarkoma, S. (2016). A gap analysis of Internet-of-Things platforms. Computer Communications, 89-90, 5-16. https://doi.org/10.1016/j.comcom.2016.03.015

A gap analysis of Internet-of-Things platforms

Julien Mineraud, Oleksiy Mazhelis, Xiang Su, Sasu Tarkoma

PII: S0140-3664(16)30073-1
DOI: 10.1016/j.comcom.2016.03.015
Reference: COMCOM 5284

To appear in: *Computer Communications*

Received date: 9 July 2015
Revised date: 8 January 2016
Accepted date: 8 March 2016

Please cite this article as: Julien Mineraud, Oleksiy Mazhelis, Xiang Su, Sasu Tarkoma, A gap analysis of Internet-of-Things platforms, *Computer Communications* (2016), doi: 10.1016/j.comcom.2016.03.015

A gap analysis of Internet-of-Things platforms

Julien Mineraud[a,*], Oleksiy Mazhelis[b], Xiang Su[c], Sasu Tarkoma[a]

[a] *Department of Computer Science, University of Helsinki, Finland*
[b] *Department of Computer Science and Information Systems, University of Jyväskylä, Finland*
[c] *Center for Ubiquitous Computing, University of Oulu, Finland*

Abstract

We are experiencing an abundance of Internet-of-Things (IoT) middleware solutions that provide connectivity for sensors and actuators to the Internet. To gain a widespread adoption, these middleware solutions, referred to as platforms, have to meet the expectations of different players in the IoT ecosystem, including device providers, application developers, and end-users, among others.

In this article, we evaluate a representative sample of these platforms, both proprietary and open-source, on the basis of their ability to meet the expectations of different IoT users. The evaluation is thus more focused on how ready and usable these platforms are for IoT ecosystem players, rather than on the peculiarities of the underlying technological layers. The evaluation is carried out as a gap analysis of the current IoT landscape with respect to (i) the support for heterogeneous sensing and actuating technologies, (ii) the data ownership and its implications for security and privacy, (iii) data processing and data sharing capabilities, (iv) the support offered to application developers, (v) the completeness of an IoT ecosystem, and (vi) the availability of dedicated IoT marketplaces. The gap analysis aims to highlight the deficiencies of today's solutions to improve their integration to tomorrow's ecosystems. In order to strengthen the finding of our analysis, we conducted a survey among the partners of the Finnish IoT program, counting over 350 experts, to evaluate the most critical issues for the development of future IoT platforms. Based on the results of our analysis and our survey, we conclude this article with a list of recommendations for extending these IoT platforms in order to fill in the gaps.

Keywords: Internet of Things, IoT platforms, IoT marketplace, gap analysis, IoT ecosystem.

1. Introduction

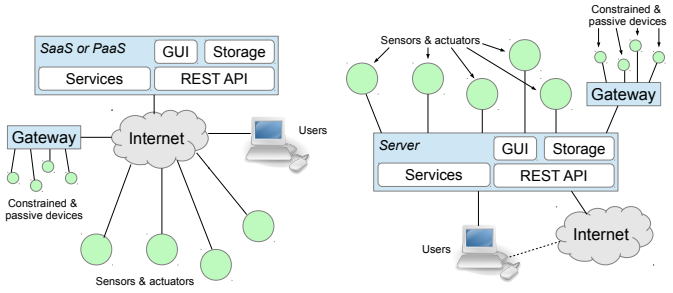
The Internet of Things (IoT) paradigm foresees the development of our current environment towards new enriched spaces, such as smart cities, smart homes, smart grid, digital health, and automated environmental pollution control [1, 2].

In recent years, an abundance of solutions has emerged to interconnect smart objects for systems with different scales and objectives. For instance, a lightweight platform can be deployed in one's home to orchestrate several connected objects, such as the fridge, the lights, and the heating system. On a broader scale, a smart city may benefit its development and management from new IoT solutions that can handle thousands of sensors, ease their maintenance, recalibration and, more importantly, analyze the data that they produce [3, 4].

In this article, we study today's IoT landscape with regard to the distribution of applications and services, as well as the platforms that connect the devices to the Internet. For the purposes of this paper, an IoT platform is defined as the middleware and the infrastructure that enables the end-users to interact with smart objects, as depicted in Figure 1. We frame our study as a gap analysis of these platforms with regard to their capacities in meeting the challenges emerging from the current development of the IoT technologies. In order to evaluate the limitations of the current IoT platform landscape and identify the gaps that need to be filled, we consider the viewpoints of different players of the IoT platform ecosystem, including device vendors, application developers, providers of platforms and related services, and the end-users. In order to strengthen the findings of the gap analysis, we conducted a survey among the experts of the national Finnish IoT program [5] to highlight the most critical gaps for the development of future IoT platforms. As a result of this evaluation, we propose a set of recommendations aimed at filling in the identified gaps.

The remainder of this article is organized as follows: Section 2 presents the review of a representative list of IoT platforms. This is followed by a thorough gap analysis of the solutions in Section 3. In Section 4, we present the results of the survey and in Section 5, we enumerate our

*Corresponding author
Email addresses: julien.mineraud@cs.helsinki.fi (Julien Mineraud), mazhelis@jyu.fi (Oleksiy Mazhelis), xiang.su@ee.oulu.fi (Xiang Su), sasu.tarkoma@cs.helsinki.fi (Sasu Tarkoma)



(a) Cloud-based platform (b) Local platform

Figure 1: **End-to-end interactions between users, smart devices and the platform.** *IoT platforms currently enables interaction between devices and users. However, interactions between IoT platforms are limited and costly.*

recommendations for filling in the gaps outlined in the previous sections. Finally, we will conclude this article in Section 6.

2. Review of today's IoT platforms

In this section, we survey available IoT platforms, both proprietary and open-source, that connect *smart objects* or *things* to the Internet. The list of the 39 platforms being surveyed, ordered alphabetically and numbered (e.g., [Platform 17], where 17 also refers to the "ref" column in Table 1), along with further details about these platforms, can be found in Appendix A. The list of the surveyed platforms shall by no means be seen as exhaustive though we believe that a representative sample of the available platforms has been included in the survey.

Table 1 lists the surveyed platforms and summarizes some characteristics which are seen by the authors as fundamental for meeting the expectations of the users and application developers. Hence, this table aims to provide quick visual information for those interested in selecting the most appropriate IoT platform to be deployed in their environment. To improve the clarity of the table, we applied a color code to the table cells. Specifically, the green color indicates that a particular platform's characteristic fits the expectations of the users of the platforms, while the red color indicates a mismatch between the characteristic of the platforms and the expectations of the users. An intermediate orange color has been added to indicate partial fitting.

In Table 1, Column *a)* enumerates the types of devices that are supported by the platform. Platforms that require a proprietary gateway to connect IoT devices are dependent on the platform providers to respond to emerging technologies, thus limiting the reactivity of the platform to adopt new protocols and support an increasing number of heterogeneous IoT devices.

Column *b)* describes the type of the IoT platform. In most cases, the platforms are provisioned from a cloud, as shown in Figure 1a, either in a form of a Platform-as-a-Service (PaaS) or a Sotfware-as-a-Service (SaaS). The PaaS refers to the platforms that provide cloud computing services for IoT devices and data. The services include, but are not restricted to storage facilities, devices management, device connectivity, backup mechanisms or online support. By contrast, SaaS focuses on the mashup of data using cloud computing capabilities. We added an additional Machine-to-Machine (M2M) tag if the platform targets primarily this part of IoT [6].

The type of architecture is shown in the Column *c)*. While the independent deployments are usually centrally controlled (see Figure 1b), the decentralized deployments (*LinkSmart*™ [Platform 17] or *OpenIoT* [Platform 23]) include multiple sub-networks of sensing and actuating devices (referred to as sites in *LinkSmart*™ and hubs in *OpenIoT*) that are independently controlled.

Note that no color code is used for columns *b)* and *c)* as we believe that different types of platforms and architectures are needed in different deployment environments. For example, a decentralized PaaS, such as *H.A.T.* [Platform 13], is ideal for a home environment while a cloud-based solution like *Xively* [Platform 39] is more appropriate for a large deployment of sensors and actuators (e.g., smart factory).

The table also includes information about the openness of the platforms, the availability of a Representational State Transfer (REST) API, as well as data access control and service discovery mechanisms.

A number of open-source platforms are considered more promising compared with the proprietary alternatives for the following reasons. First, the use of the open source is expected to enable the faster integration of new IoT solutions across the application domains. Second, the use of the open source has been reported to speed up the adoption of a software technology in a bottom-up fashion. Finally, when seen from the social surplus perspective, the industry based on the open-source platforms has been found to provide larger total welfare, compared with the industry structures based on proprietary platforms [7].

Only a few platforms do not have a REST API. This demonstrates that the current IoT services will tend to become more and more like traditional web services (i.e., Web of Things [8]). In particular, IoT service mashups and data analytics will be key integrators for the future of IoT technologies [9, 10, 4]. We noted that only a few platforms have integrated some type of service discovery mechanisms, even in a very simplified fashion. A comprehensive survey on discovery protocols for constrained M2M communications can be found in [11].

Security and privacy of IoT platforms

One of the fundamental criteria for IoT platforms is the need to include efficient and reliable privacy and security mechanisms [12, 13, 14, 15, 16]. In [16], Satyadevan *et al.* survey five IoT platforms (including platforms 4, 17,

Table 1: Available IoT platforms

Ref	Platforms	a) Support of heterogeneous devices	b) Type	c) Architecture	d) Open source	e) REST	f) Data access control	g) Service discovery
1	AirVantage™	Needs gateway	M2M PaaS	Cloud-based	Libraries only (Apache v2, MIT and Eclipse v1.0	Yes	OAuth2	No
2	Arkessa	Yes	M2M PaaS	Cloud-based	No	n.a.	Facebook like privacy settings	No
3	ARM mbed	Embedded devices	M2M PaaS	Centralized/ Cloud-based	No	CoAP	User's choice	No
4	Carriots®	Yes	PaaS	Cloud-based	No	Yes	Secured access	No
5	DeviceCloud	Yes	PaaS	Cloud-based	No	Yes	n.a.	No
7	EveryAware	Yes	Server	Centralized	No	Yes	4 levels	No
8	Everyware	Needs gateway	PaaS	Cloud-based	No	Yes	n.a.	No
9	EvryThng	Yes	M2M SaaS	Centralized	No	Yes	Fine-grained	No
10	Exosite	Yes	PaaS	Cloud-based	Libraries only (BSD license)	Yes	n.a.	No
11	Fosstrack	RFID	Server	Centralized	No	No	Locally stored	No
12	GroveStreams	No	PaaS	Cloud-based	No	Yes	Role-based	No
13	H.A.T.	Home devices	PaaS	Decentralized	Yes	Yes	Locally stored	Yes
14	IoT-framework	Yes	Server	Centralized	Apache license 2.0	Yes	Locally stored	Yes
15	IFTTT	Yes	SaaS	Centralized	No	No	No storage	Limited
16	Kahvihub	Yes	Server	Centralized	Apache license 2.0	Yes	Locally stored	Yes
17	LinkSmart™	Embedded devices	P2P	Decentralized	LGPLv3	No	Locally stored	Yes
18	MyRobots	Robots	Robots PaaS	Cloud-based	No	Yes	2 levels	No
19	Niagara^{AX}	Yes	M2M SaaS	Distributed	No	n.a.	n.a.	n.a.
20	Nimbits	Yes	Server	Centralized/ Cloud-based	Apache license 2.0	Yes	3 levels	No
21	NinjaPlatform	Needs gateway	PaaS	Cloud-based	Open source hardware and Operating System	Yes	OAuth2	No
22	Node-RED	Yes	Server	Centralized	Apache license 2.0	No	User-based privileges	No
23	OpenIoT	Yes	Hub	Decentralized	LGPLv3	No	User-based privileges	Yes
24	OpenMTC	Yes	M2M client/ Server	Centralized/ Cloud-based	No	Yes	Secured access	No
25	OpenRemote	Home devices	Server	Centralized	Affero GNU Public License	Yes	Locally stored	No
26	Open.Sen.se	Ethernet enabled	PaaS/SaaS	Cloud-based	No	Yes	2 levels	Limited
27	realTime.io	Needs gateway	PaaS	Cloud-based	No	Yes	Secured access	No
28	SensorCloud™	No	PaaS	Cloud-based	No	Yes	n.a.	No
29	SkySpark	No	SaaS	Centralized/ Cloud-based	No	Yes	n.a.	No
30	Swarm	Yes	PaaS	Cloud-based	Client is open source (unknown license)	Yes	n.a.	n.a.
31	TempoDB	No	PaaS	Cloud-based	No	Yes	Secured access	No
32	TerraSwarm	Yes	OS	Decentralized	n.a.	n.a.	n.a.	Yes
33	The thing system	Home devices	Server	Centralized	M.I.T.	Yes	User's choice	No
34	Thing Broker	Yes	Server	Centralized	Yes	Yes	Locally stored	No
35	ThingSpeak	Yes	Server	Centralized/ Cloud-based	GNU GPLv3	Yes	2 levels	Limited
36	ThingSquare	Embedded devices	Mesh	Cloud-based	Gateway firmware is open source	Yes	No	No
37	ThingWorx	Yes	M2M PaaS	Cloud-based	No	Yes	User-based privileges	Yes
38	WoTkit	Yes	PaaS	Cloud-based	No	Yes	Secured access	Yes
39	Xively	Yes	PaaS	Cloud-based	Libraries are open source (BSD 3-clause), platform is not	Yes	Secured access	Yes

35 and 39 of our survey) with respect to security and trust management. The survey suggests that cloud-based IoT platforms are prone to traditional web and network security attacks such as Denial of Service (DoS), man-in-the-middle, eavesdropping, spoofing and controlling attacks. A survey of low-level protocols for ensuring security and privacy in both centralized and distributed IoT scenarios is presented in [15], and the research community constantly aims to improve protocols to address these security challenges. An example is the work proposed by Asokan *et al.* [17] to secure large swarms of embedded devices while overcoming the limitations of these constrained devices (i.e., memory, computation, communication, latency and energy consumption constrains). A number of areas are critical for the widespread adoption of IoT but not yet fully addressed by IoT platforms; many of these have been listed and analyzed in the above surveys, including: (i) device authentication, (ii) communication and physical privacy, (iii) data storage protection, (iv) device protection, (v) trust management, (vi) governance and (vii) fault tolerance.

In this article, we limit our analysis of the security and privacy issues to the protection mechanisms for data storage and data access available on the IoT platforms. Meanwhile, for more comprehensive discussions on the other security, privacy and trust challenges pertaining to IoT platforms, we invite the interested readers to refer to [14, 15, 16, 17].

To authenticate users, most of the cloud-based platforms use the standard protocol OAuth 2.0 [18] while centralized servers required only a local access to the machine. We evaluated the expectations in term of privacy as the flexibility of the access control offered by the platform. Throughout our evaluation, we noted four types of access granularity from the basic private or public choice (i.e., 2-level for *MyRobots* [Platform 18] or *Open.Sen.se* [Platform 26]) to a fine-grained access control where the data could be either private, protected, public or anonymous (i.e., 4-level for *EveryAware* [Platform 7]). In our opinion, the latter is the only one having the necessary flexibility to maximize the re-usability of the data by remote third-party services.

3. Gap analysis

In the previous section, we presented the characteristics of IoT platforms that are the most important to users and application developers. However, multiple gaps can be identified in the functionality offered by these platforms. Therefore, we present in this section a gap analysis, summarized in Table 2, that aims to evaluate the maturity of the current solutions by assessing their shortcomings along several dimensions. The dimensions covered by the analysis include (i) the extensibility of the platform in terms of supporting heterogeneous sensing and actuating technologies, (ii) the data ownership and its implications for security and privacy, (iii) the data processing and sharing for supporting new services, (iv) the support of application developers, and (v) the completeness of an IoT ecosystem. Then, we extend the gap analysis to dedicated IoT marketplaces that (vi) support the deployment of IoT applications and services.

3.1. Integration of sensing and actuating technologies

The essence of an IoT platform is to enable the secure connection of a multitude of heterogeneous sensing and actuating devices, having different constraints and capabilities, to the Internet. In the absence of de-facto communication standard(s), the sensing and actuating devices by different vendors may subscribe to different interaction patterns, and may implement different subsets of available communication protocols. As a result, arguably, the value of an IoT platform grows proportionally with the number and the versatility of the supported devices. An ideal IoT platform would offer a pool of standardized communication protocols where the device manufacturer may select the appropriate protocols (e.g., CoAP for constrained devices [19]). In the case of passive devices (e.g., RFID-enabled) or constrained devices, the connectivity relies on a mediating gateway (see Figure 1) that must be fully controlled by the platform user, alike the *NinjaBlock* [Platform 21], which provides open-source hardware and firmware for the gateway.

For a smooth integration with sensing and actuating devices, it is essential that the IoT communities establish standardized protocols for all devices, as it is currently done for highly constrained devices by the IETF [20] or for M2M communications by IEEE1888, ETSI M2M and 3GPP [21][1]. Presently, protocols for constrained devices are supported by *OpenRemote* [Platform 25] (KNX, Z-Wave, X10, etc.), *LinkSmart*™ (ZigBee), *ARM mbed* [Platform 3] (MQTT, CoAP) and *ThingWorx* (MQTT); the others assume the use of relatively powerful devices capable of supporting traditional web protocols. It shall be noted that for some platforms, such as *LinkSmart*™, the support for constrained devices protocols is implied though the publicly available documentation is insufficient for judging the extent of such support. Meanwhile, *SensorCloud*™ [Platform 28], *SkySpark* [Platform 29] or *TempoDB* [Platform 31] require full-fledged HTTP end-point to upload the data, assuming powerful devices capable of supporting traditional web protocols and do not integrate device communication protocols into their solutions. Finally, *IFTTT* [Platform 15], which communicates with both device manufacturers and web service providers, "adjusts" to the vendors' needs (e.g., to the needs of Belkin) to extend the platform to new technologies. It shall be emphasized that there is no de-facto communication protocol suit, and this makes the task of interfacing heterogeneous devices more challenging and hence

[1] More details on standardization bodies and protocols can be found in [1].

more expensive. In addition, as previously mentioned in Section 2, IoT platforms do not integrate sufficient security and privacy protocols to satisfy the integrity of the data and the management of connected devices [15, 16].

The current IoT solutions address the issue of interfacing heterogeneous devices differently. Generally, the interoperability with devices is ensured either by implementing a gateway that can be expanded, e.g., with the help of plug-ins, to support new types of devices whenever needed, or by mandating the device vendors to use protocols from a limited set of supported ones. For example, the *realTime.io* [Platform 27] platform proposed a connection of sensors via a proprietary gateway (*ioBridge* which even requires the use of a proprietary transport protocol, *ioDP*), while the *ThingWorx* [Platform 37] and *OpenMTC* [Platform 24] platforms use web sockets, MQTT or other standard communication protocols to interconnect devices to the platform. Some other platforms, such as *the thing system* [Platform 33] or *H.A.T.* targets the integration of devices present in "smart homes" and "smart places" environments. Other platforms, such as *Fosstrack* [Platform 11], only enable one type of technology which, in the case of *Fosstrack*, is RFID. Note however that either the heterogeneity of supported devices is limited, or the use of a gateway is necessary (Gap G1.1). We believe that, in order to streamline the integration of new device types, standard object models for IoT devices, such as the models recommended in the recent IPSO Smart Objects guidelines [22] based on the Lightweight M2M (LwM2M 1.0) specifications [23], should be integrated widely by IoT platforms (Gap 1.2). Furthermore, security mechanisms, such as in [17], should be integrated to IoT platforms to provide secure management of IoT devices (Gap 1.3).

3.2. Data ownership

In our opinion, the enormous volume of data that would be generated by the devices in the IoT mandates the data management to be at the core of IoT paradigm, and it amplifies the need to maintain a certain degree of privacy and security [13][2]. The owner of the data can thus be expected to have a full control over the placement of the data, as well as over who has the access rights to which portions of this data.

Based on the information collected during our gap analysis of today's IoT platforms, the data ownership has been a major concern for all the platforms. For instance, the cloud-based platforms (e.g., *Swarm* [Platform 30]) ensure that the data collected and stored by the platform remains the property of the customers. However, the full ownership of the data is rarely guaranteed. In most cases, rather than storing and manipulating the data locally at the edge, the data is sent to the platform in a raw format, stored unencrypted and very little information is presented on the security measures taken to secure the data (Gap G2.1).

The majority of the listed platforms requires the use of access keys or other access control mechanisms to get *read* or *write* permissions. The access rights are either determined by the end users of the devices, through a web interface (*ThingSpeak* [Platform 35], *Nimbits* [Platform 20]), or are left for the application provider to define when implementing the applications (*OpenRemote*, *Swarm*, *LinkSmart*™, *Thing Broker* [Platform 34]). Furthermore, the *EveryAware* platform provides access to public data feeds to anonymous users, who do not require access keys. Such overly strict or too relaxed privacy settings do not provide enough control over the data (Gap G2.2).

Only solutions, where the data is stored locally (e.g., H.A.T. or OpenRemote), truly offer the full ownership of the data to the end-user. We suggest that future IoT solutions must have algorithms and mechanisms for the data owner to give access only to a predefined set of the resources, and that the raw data must remain under control of the end-user. For instance, if the data owner is willing to archive data using a service offered by a PaaS, he must be able to encrypt the data or process it before sending it to the cloud. Further, since too strict or too relaxed privacy settings do not provide enough control over the data, in future IoT solutions, fine-grained data visibility must be coupled to local storage functionalities to re-attribute the ownership of the data to the users.

3.3. Data processing and data sharing

IoT data can be large in terms of volume and the applications typically have real-time requirements. IoT data streams are unbounded sequences of time-varying data elements. This data could often be unreliable, incomplete, and have different qualities and out-of-order arrival problem, and communication loss [25]. Furthermore, this data is represented in different formats and various models. For example, it is a challenge to directly utilize low-level data provided by sensors without a well-defined knowledge model.

Data and knowledge behind data are the core of the wealth produced by the IoT. Data processing and sharing mechanisms should be developed to ensure that IoT data can be utilized in applications to its best. Today's IoT solutions either do not support, or have limited support for the processing and sharing of data streams. Yet, it remains possible to combine multiple streams into a single application if one knows the URI to the desired sources of information, but this represents a technical challenge for application developers. The *Ericsson's IoT-Framework* [Platform 14] provides mechanisms to integrate virtual streams (e.g., from external data sources) that can be combined with local streams for visualization or statistical analysis

[2]ETSI Partnership Project oneM2M has issued a technical report for the specification of security architecture for M2M communications [24]. The standardization body has classified the security solutions as (i) identification and authentication, (ii) authorization and (iii) identity management. In this study, we only considered the first two classes.

and data predictions. Moreover, different data processing techniques are adapted for IoT. For example, Tsai *et al.* [4] survey data mining technologies for IoT. Su *et al.* [26] study how to embed semantics on IoT devices and Maarala *et al.* [27] extend this research with processing large IoT data in city traffic scenarios. Nevertheless, the aggregation of these available data processing techniques within IoT platforms is still limited (Gap G3.1).

The principle of data fusion has been addressed by the *Node-RED* tool [Platform 22], which enables the composition of IoT data and devices with the concept of *data flows*. In [28], Blackstock and Lea integrated the *Node-RED* composer to the WoTKit processor [Platform 38] to enable the creation of distributed data flows. Hence, such mechanisms support the creation of innovative and enriched web-of-things contents. We suggest that such mechanisms should be integrated into IoT middleware systems to perform similar operations on data streams. The current gap is in processing these streams efficiently and handling different formats and models (Gap G3.2). Here, efficient processing means (i) processing IoT data considering computing, storage, communication, and energy limitations of IoT environments; and (ii) the timely generation of useful knowledge for IoT applications before it becomes outdated. Meanwhile, to cope with big IoT data, most IoT platforms shall have a high processing throughput.

To mitigate the gap above, edge analytics solutions (i.e., closer to where the data is being produced), such as *cloudlets* [29], are now available for constrained deployments. We believe that future IoT platforms should include *cloudlets*-like technologies to enable local IoT networks to perform edge analytics. Edge analytics contributes to maximize energy efficiency, reduce privacy threats and minimize latencies. The *Kahvihub* platform [Platform 16] envisions to support this for constrained devices, by providing sandboxed execution platforms for IoT services. As a result, networks of heterogeneous devices can collaboratively analyze the data that they produce. Sandboxing IoT application has also been addressed in [30].

IoT devices produces low-level data, which is often unreliable, incomplete, disordered, and even lost. Therefore, fault management is essential for IoT platforms. Availability of input data streams for IoT platforms is often undetermined. Hence, additional challenges are introduced to guarantee the complete data processing result (Gap G3.4). Moreover, intrusion detection, prevention, and recovery mechanisms should be developed in IoT platforms, which will help IoT entities to protect their data and services [15].

Finally, in order to find the relevant data streams that are available, these streams should be listed in dedicated data catalogs where context information may be used to provide efficient discovery mechanisms. Semantic indexing can be used on these catalogs and other metadata available on the IoT devices [31]. The efficient processing of IoT data from multiple external sources is still an open issue. From all the platforms reviewed in this article, only four (i.e., *IoT-Framework*, *Kahvihub*, *ThingWorx* and *Xively*) integrated a search mechanism for data streams. In the case of *IoT-Framework*, the search mechanism is performed through geolocalization, tagging or data types. However, the search was limited to the streams available on the platform, whereas the search through multiple platforms is hitherto unavailable (gap G3.3). Recent research efforts have been invested towards this direction with *HyperCat*, a lightweight JSON-based URI catalog that references services provided by IoT platforms [3].

3.4. Support of application developers

In order to foster an expedited development of applications, the IoT platforms are expected to provide the developers with streamlined application programming interfaces (APIs) to their functionality, preferably with the help of higher abstraction level primitives. Further, to enable an efficient development of cross-IoT-platform applications, these APIs shall be uniform across the platforms, to the extent possible.

Today's IoT platforms almost all provide a public API to access the services. The APIs are usually based on RESTful principles, and allow common operations such as `PUT`, `GET`, `PUSH` or `DELETE`. These operations support the interaction with the connected devices on the platform, as well as the management of these devices. Only four of the studied platforms did not include a REST API for easing the development of web services (i.e. *Fosstrack*, *LinkSmart*™, *IFTTT* and *OpenIoT*), but use different interaction means. Nonetheless, the other platforms uses nonuniform[3] REST APIs and data models which complicates the mashing up of data across multiple platforms (Gap 4.1; see Section 3.3).

Many platforms also offer libraries, which are in some cases open-source (e.g. *AirVantage*™ [Platform 1], *Exosite* [Platform 10], *IoT-Framework* or *Xively*), that are bindings for different programming languages to the REST API available on the platforms. However, these bindings libraries do not greatly improve the support to application developers in using the services provided by the platforms as they only include basic functionalities, e.g., connection to the platform with access keys (Gap 4.2). To some extent, some platforms such as *ThingSpeak* enable the creation of widgets written in Javascript, HTML and CSS that may be distributed on the platform to other users. Alternatively, the *Carriots*® platform [Platform 4] provides a full Software Development Kit (SDK) written in Groovy for application developers. We believe that this approach should be more generalized within IoT solutions to maximize usability of the services provided by the IoT platforms.

[3]Nonuniform in this context means that every platform provides custom APIs and data models as standards such as HyperCat [32] are not yet widely adopted.

Table 2: Summary of the gap analysis

Category	Current status	Expectations	Gaps	Problems	Recommendations
Support of heterogeneous devices	Platforms assume smart objects to talk HTTP or require gateway	•Devices must be easily and securely integrable to the IoT platform without a gateway • Unified resources and simplify usability	**G1.1** Support of constrained devices **G1.2** Standardized IoT devices models **G1.3** Secure authentication, identification of management of IoT devices	• Heterogeneous interactions • Protocol standardization	• Relying on standard protocols (e.g., CoAP, LwM2M, MQTT) • Integration of state-of-the-art security and privacy protocols
Data ownership	Mainly given to the end-user but with very simple privacy policies	• Full control given to the owner of the data • Local storage • Fine-grained data visibility model	**G2.1** Manipulation of data in edge devices **G2.2** Self-storage	• Security of the data storage • Device constrains to store data and provide secure access control	Algorithms and mechanisms available to the data owner to limit the access only to a predefined set of the resources
Data processing & sharing	• Nonuniform data sharing format • Sharing is performed via nonuniform REST API	• Uniform data format across multiple platforms. • Pub/Sub mechanism and data catalogs • Edge analytics	**G3.1** Data processing is not well integrated in IoT platforms **G3.2** Efficient processing for data formats and models **G3.3** Data analytics is only available in cloud-based solutions **G3.4** Data catalogs are missing	• Complex identification system to access data • Fusion efficiently data streams from multiple data catalogs • IoT devices have limited computing capabilities	• Data catalogs with semantic indexes • Uniform and interoperable data models • Integration of data processing technologies in platforms • Cloudlet-like solution for edge analytics
Developer support	• REST API to access the data or devices handled by the platform • Applications are for internal use rather than for sharing (except IFTTT)	• Use of a common API to ease the development of cross-platform applications • Domain Specific Language (DSL) dedicated to cross-platform application development	**G4.1** Application mashup APIs **G4.2** Limited presence of SDKs **G4.3** Absence of DSL with higher abstraction level primitives	• Require standardization of application interactions dedicated to the IoT • IoT app store are missing	IoT platforms must provide SDKs and APIs that maximize the re-usability of the services provided by their platform
Ecosystem formation	Platforms provide useful building blocks, storage and run-time environment for application developers	• Platform easily expandable by the developers and offering them incentives to contribute • Cross-platform sharing of applications and services • Local composition of services	**G5.1** Low platform expandability **G5.2** Limited monetizing possibilities **G5.3** Limited support for cross-platform integration	• Silos of platform-specific solutions • User's using multiple platforms may not be able to aggregate the whole data into a single application	• Financial incentives for developers shall be offered • A broker is needed to ease cross-platform integration • Models to contextually define IoT applications to simplify their discovery by the end-users
IoT marketplace	• Limited applications sharing • Limited (usage-based) charging of the end users of these applications	• Dedicated IoT data catalogs, IoT app store and IoT device store • Ability to advertise, deliver and charge for the use of applications and data • Validate applications against policies	**G6.1** Application, data and device catalogs dedicated to the IoT are generally missing **G6.2** The billing (based on fixed fees, usage, or other metrics) of the end-users of the data is generally missing	An ecosystem of independent application developers, device manufacturers, and end-users all supporting the platform is needed for the demand for marketplace to appear and sustain	The marketplace functionality shall be provided by future IoT platforms

In addition to APIs, a Domain Specific Language (DSL) could be defined to simplify the development of IoT applications, also by offering functional primitives describing the problem and solution space at a higher abstraction level. For instance, primitives for querying the data stream catalogs, fusing and aggregating data should be available to the developers in order to speed up the process of developing cross-platform data-centric applications; such DSL, however, are largely non-existent at the time of writing (Gap 4.3).

3.5. Toward IoT ecosystem formation

The success of an IoT platform is dependent on the existence of a business ecosystem of firms where the buyers, suppliers and makers of related products or services, as well as their socio-economic environment, collectively provide a variety of applications, products, and services to the end-users of IoT [33]. By offering a common set of assets, that are shared by the ecosystem members and are essential for their products and services, such platform shapes a core of its ecosystem.

To prosper, the platform, besides performing an essential IoT function or solving an essential IoT problem, should be easily expandable by the developers of the complementing products or applications based on it, and should provide them with incentives to innovate and contribute to the platform [34]. In other words, the platform shall attract the developers of add-ons and applications, thereby enabling a bottom-up formation of the ecosystem around it.

Today's IoT platforms claim to solve some of the essen-

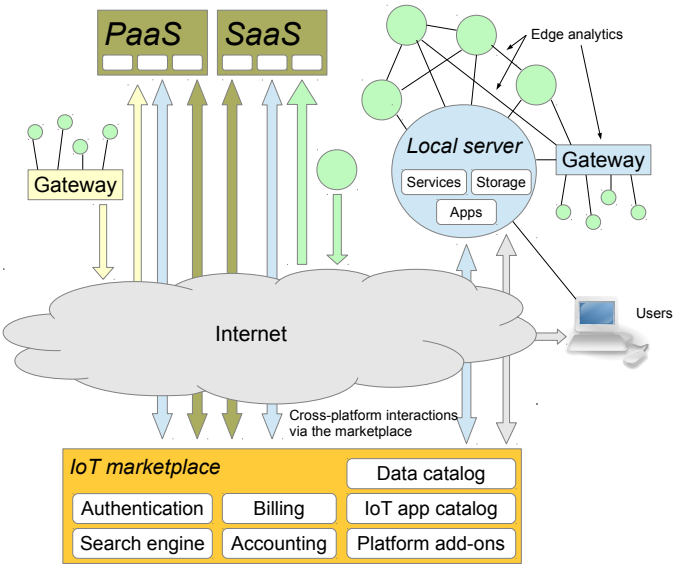


Figure 2: **End-to-end interactions between users, smart devices, the platforms via the marketplace.** *The IoT marketplace allows cross-platform interactions and drives the development of new business opportunities (e.g., billing of IoT data). White rectangles within platforms represents functional elements (e.g., search engine or apps) but text was omitted in cloud-based architecture to improve visibility. Check Figure 1a for more details.*

tial problems of application developers, and are generally open for third-party application creators. However, only open-source platforms can be expanded rapidly to cope with the emergence of new technologies. Proprietary platforms do not allow to add reusable components or add-ons to the platform, except recipes in *IFTTT* and third-party tools integration for *ThingWorx* (Gap 5.1), and monetizing possibilities for platform complementers are absent or limited to integration services, e.g., *OpenRemote* (Gap 5.2).

In order to allow to treat the IoT domains as a single converging ecosystem that provides innovative products and services and permits an economy of scale, an IoT platform broker is needed. Such a broker will facilitate the sharing of applications and services across space and time, and across platform-specific IoT sub-ecosystems. However, the possibility of multi-platform brokerage has not been investigated in depth and the resulting IoT ecosystem represents a multitude of fragmented IoT vertical silos (Gap 5.3).

Still, this vision of new IoT ecosystem formation is shared by the Terra Swarm Research Center for the *TerraSwarm* [Platform 32] and by the Technology Strategy Board[4] with the specification of *HyperCat* to solve interoperability issues among IoT solutions.

3.6. Dedicated IoT marketplaces

Software application marketplaces are aimed at facilitating the discovery, purchase, and distribution of the applications. These marketplaces can be exemplified with hardware-specific and centrally-controlled solutions, such as *Apple App Store* or *Google Play*, or hardware-agnostic marketplaces, such as *Good*, *Handster*, *Nexva*, and *SlideMe*. The availability of such marketplaces is crucial for the dissemination of software innovations in general, and IoT innovations in particular [35].

These marketplaces address the needs of the application providers and users, and alternatively, the needs of the platform vendors and platform operators. However, the traditional application stores are seemed to have limitations as far as IoT applications are concerned. Namely, to the best of our knowledge, none of the contemporary application stores support the delivery of purchased software to the connected devices other than the mobile terminals (e.g. smartphones and tablets) supported by the platform (Gap 6.1). Among IoT platforms, some platforms have dedicated application stores (e.g. ThingWorx) but only some (*IFTTT*) allow the applications to be publicly shared, and only some (*OpenIoT*) promise to enable the (usage-based) charging of the end users of these applications (Gap 6.2). Moreover, one of the key challenges of IoT is to exploit all the data that is currently being produced by businesses. According to McKinsey[5], businesses already collect tremendous volume of sensor data but the data is only used for anomaly detection and control. However, data should also be used for optimization and prediction which provide the greater value, but businesses may lack the expertise to analyze and process their data. This justifies the need for the development of new marketplaces for IoT data that will thrive new business interactions (i.e., business-to-business).

The *Windows Azure Data Market*[6] platform provides an example of a successful business model that could emerge from IoT data. For instance, the platform allows businesses to publish data streams to the platform in order to make them available to a large number of application developers. The platform offers the possibility of charging for the data consumption either by a time-defined subscription or by the amount of data to be consumed. The platform also allows publishing data streams free of charge. The catalogs of data sources published on the platform is also browsable. We believe that the development of IoT dedicated platforms, similar to the *Windows Azure Data Market*, is a requisite to the sustainability of IoT solutions. Figure 2 illustrates the opportunities that emerge from the availability of dedicated IoT marketplaces. Unlike in Figure 1, where interactions between IoT platforms is limited and difficult, the IoT marketplace allows the flow of IoT data across platforms. Marketplaces should include authentication, billing, accounting, as well as catalogs for IoT data and applications. Marketplaces could also be ex-

[4] `https://www.innovateuk.org/`

[5] `http://www.mckinsey.com/insights/business_technology/the_internet_of_things_the_value_of_digitizing_the_physical_world`

[6] `http://datamarket.azure.com/`

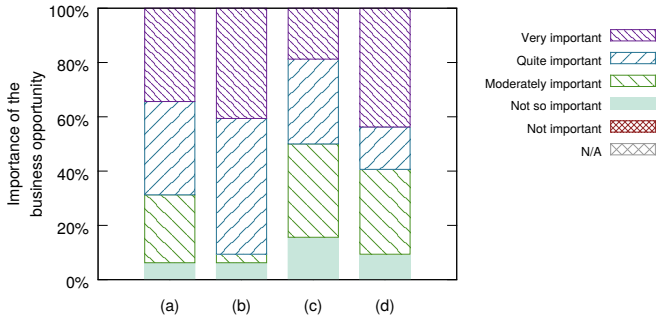


Figure 3: **Importance of the business opportunities** *for (a) sharing and selling data and/or applications in a controlled manner, (b) Maximizing re-usability of data to increase profit, (c) Searching for data/applications in an ad-hoc fashion and (d) reducing transaction costs of data/application acquisition.*

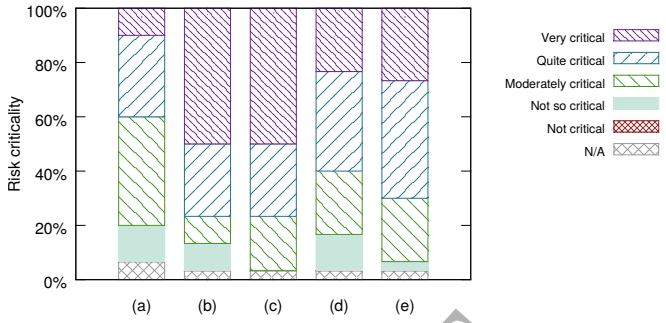


Figure 4: **Risk criticality** *of (a) direct sells preferred, (b) lack of data suppliers and application providers, (c) small customer base, (d) challenge of making generic applications and (e) fragmentation of the IoT landscape.*

tended with an additional catalog for communication protocols (platform-specific) and for IoT devices/components to provide a complete solution for the IoT users.

4. The perspective of the national Finnish IoT program

In this section, we present the results of a survey conducted among the partners of the Finnish IoT program [5] on the importance of various key points for the future development of IoT platforms, including the IoT dedicated market places. Table 3 lists the number of survey participants.

Table 3: Organization distribution

Type	Count	Percentage
Academia	19	54.29%
SME	7	20%
Large company	9	25.71%
Total	35	100%

Figure 3 summarizes the results of the survey regarding the possible business opportunities that could emerge from filling out the gaps presented in the previous section. As can be seen from the figure, survey respondents have designated the business opportunity of maximizing re-usability of data as the most important (at 40% very important, as well as 50% quite important). On the other hand, searching for data or applications in an ad-hoc fashion raised less interest as less than 50% of considered it quite/very important, and since 15% of the project's experts declared it of little importance. Finally, the sharing and selling of data/applications as well as reducing the cost of data/applications acquisition have raised moderate interests.

We also asked our experts to evaluate the risks that may emerge from developing the next generation of IoT platforms. As shown in Figure 4, the most critical risks are i) the lack of suppliers and application providers as well as ii) having a too small customer base. In fact, these two risks are going hand in hand as a large customer base attracts application developers and data suppliers, while the latter attracts more customers. Noteworthy, possible negative impact of the introduction of IoT marketplace onto the traditional way of selling directly has not been defined as critical risk by our panel of experts (see Figure 4(a)). However, the risks coming out from the current verticality of the IoT landscape have been found moderately critical, thus showing the readiness of the IoT for more horizontal interactions between IoT solutions (see Figure 4(d) and Figure 4(e)).

In the final stage of the survey, we asked our experts to evaluate the most important features that must be integrated to IoT platforms with regard to the gaps underlined in Section 3. The features are grouped by four different viewpoints; (i) application provider viewpoint, (ii) data publisher viewpoint, (iii) platform provider viewpoint and lastly (iv) the customer viewpoint. The results of this evaluation has produced the following list of features in a descending order of importance:

1. Publishing applications: register and upload the applications, make applications discoverable and available for external parties (Application provider viewpoint).
2. Available description or detailed information about the application or the data on the marketplace (Customer viewpoint).
3. Purchasing the right to use the application or the data (Customer viewpoint).
4. Publishing data: make the data discoverable and available for external parties through predefined interfaces (Data publisher viewpoint).
5. Gathering information about resources usage by customers, as well as summarizing it into accounting records, e.g., for the purpose of charging and billing (Platform provider viewpoint).
6. Setting or modifying the access rights separately to different views or portions of the data in order to maximize re-usability (Data publisher viewpoint).

7. Gathering information about the sells and downloads of the applications (Application provider viewpoint).
8. Searching for the applications based on type, payment details, rating (Customer viewpoint).
9. Registering, unregistering, uploading and validating an application (Platform provider viewpoint).
10. Managing platform subscriptions of customers (create, read, update, delete) (Platform provider viewpoint).

This listing shows that eight out of the ten most important features selected by our experts are related to the IoT marketplace and generate cross-platform interaction as depicted in Figure 2, thus comforting our view on the necessity of developing this type of platform. The sixth most important feature is, on the other hand, related to increasing the re-usability of the data by setting multiple role-based views on the data or on selected portions of the data (e.g., for only a time period of 24 hours).

5. Recommendations for the development of IoT middleware

In the previous sections, we evaluated the current IoT platform landscape with a thorough gap analysis, that is summarized in Table 2, and complemented the gap analysis with a survey conducted among the experts of the national Finnish IoT program. As a result, numerous gaps have been identified; furthermore, several recommendations were made in section 3 for the IoT platform vendors to expand their offerings so as to address these gaps. These recommendations included, among others,

- leaning on standardized communication protocols to interface heterogeneous devices (Subsection 3.1),
- adding the provisions for handling and processing data locally (Subsection 3.2),
- adding uniform data models, data catalogs, and the edge analytics capabilities (Subsection 3.3),
- offering streamlined APIs (Subsection 3.4),
- introducing cross-platform brokers and financial incentives for ecosystem players (Subsection 3.5), and
- developing dedicated IoT makerplace(s) (Subsection 3.6).

In this section, we return to these recommendations and complement them with further recommendations both concerning the short-term (easier to implement) and long-term (harder to implement) perspectives. For the reader's convenience, these recommendations are shown in the right-most column of Table 2.

In the short-term perspective, the development of a basic IoT marketplace, as shown in Figure 2, serves as a repository for data streams and applications, should boost tremendously the ability of the IoT landscape to fill partially in some of the gaps. For instance, the immediate benefits would be:

- **Data processing & sharing**: the ability to request numerous external data streams to enrich local content. It would also enable users to publish some of their streams to third-parties;
- **Developer support**: the possibility for application developers to publish their products and reach a wide range of customers;
- **Ecosystem formation**: the increasing awareness about new innovations and possibility of creating new business models;
- **Market & billing**: the ability to market/search for data and applications and sell/purchase the rights to use them.

From the viewpoint of middleware solutions, fine-grained access control must be implemented first to re-provision the user with the full ownership of his data. Finally, SDKs should be provided to application developers to facilitate the creation of the applications based on the platform.

In the long-term perspective, the marketplace would drive the uniformity for the REST APIs and the data models. It would also contribute to the standardization of popular communication protocols as IoT device manufacturers will be encouraged to comply to these open standards (e.g., ETSI, IETF, etc.) in order to improve their visibility on the marketplace. Accounting functionalities must be implemented next to strengthen ecosystems and permit a large scale economy. Additionally, efficient search engines for data streams must be developed to maximize the quality of services of IoT applications. From the viewpoint of the middleware solutions, the development of a cross-platform DSL would provide massive support to application developers. Moreover, performing edge analytics (see Figure 2) would help reduce the latency, the volume of data transported across the network and reduce threats on privacy and security (e.g., raw and risk-critical data may be pre-analyzed locally).

As a result, the marketplace plays a central role in connecting IoT actors and thus, allowing cross-platform interactions for the IoT (different interactions are represented with separated colors in Figure 2), and consequently creating more opportunities for data exchanges and business operations. The marketplace will also allow the distribution of IoT-specific applications to a large number of IoT users as we currently experience with smartphone application stores, and thrive the economical growth of IoT which is expected to reach as much as $19 billions (Cisco's forecast for 2020[7]).

[7] https://agenda.weforum.org/2014/01/are-you-ready-for-the-internet-of-everything/

6. Conclusions

In this article, we have evaluated a number of available IoT platforms, both proprietary and open-source, that together form a representative sample of the IoT platform landscape. The IoT platforms were evaluated via a gap analysis that outlined their capability to (i) support the integration of heterogeneous hardware, (ii) provide sufficient data management mechanisms, (iii) support application developers, (iv) support the formation of ecosystems, as well as (v) provide the dedicated marketplaces for the IoT. Collectively, these capabilities reflect the needs of different players of the emerging IoT ecosystem, including the device vendors, the application developers, the providers of platforms and related services, and the end-users.

We complemented the gap analysis with a survey conducted among the experts of the Finnish IoT program to evaluate the business opportunities, risks and the most important features that may emerge from filling in the highlighted gaps. Based on the results of the gap analysis and the survey, we compiled a list of recommendations, both for short and long term perspectives. Our recommendations are aimed at filling in the identified gaps in contemporary IoT platforms and include, among others, the development of a dedicated IoT marketplace, the availability of SDKs and open APIs, and the possibility to analyze data locally, flexibly control access to the platform and its data, as well as providing data processing and sharing mechanisms.

Appendix A. Reviewed IoT Platforms

Platform 1: **AirVantage™** (`https://airvantage.net/`)

AirVantage™ is a proprietary cloud-based M2M dedicated platform that provides end-to-end solutions to connect wireless-enabled devices to their platform. From an user viewpoint, the platform proposes interactive dashboards for device management, and big data storage. The platform uses open-source M2M dedicated development tools such as the framework *m2m.eclipse.org*[8]. The platform also integrates the standard protocol MQTT.

Platform 2: **Arkessa** (`http://www.arkessa.com/`)

Arkessa is a proprietary cloud-based M2M management architecture and IoT platform. It includes the MOSAIC platform that enables devices to be easily connected to many applications. Privacy with third-party applications is done in similar way than Facebook or Linkedin. Ownership of the data remains to the end-user. Arkessa provides an ecosystem of devices and applications giving high flexibility to the end-user.

Platform 3: **ARM mbed** (`https://mbed.org/`)

ARM mbed® provides a device server, that is proprietary, to connect constrained devices to the IoT. The platform proposes security solutions for embedded devices, such as embedded Transport Layer Security (TLS). It uses CoAP and RESTful API for creating M2M networks of constrained devices.

Platform 4: **Carriots®** (`https://www.carriots.com/`)

Carriots® is a proprietary cloud-based platform (PaaS). REST API and Groovy SDK are available for web application development. Data format supported are JSON and XML. The data is stored on the platform and access keys are required to access it.

Platform 5: **DeviceCloud** (`http://www.etherios.com/products/devicecloud/`)

DeviceCloud is a proprietary and cloud-based device management platform (PaaS). The platform provides access the devices connected to the platform via a REST API.

Platform 6: **Devicehub.net** (`http://www.devicehub.net/`)

Devicehub.net is a proprietary cloud-based platform which does not provide a true REST API (using GET method to PUT data). Currently, the documentation of the platform is too limited to provide more information.

Platform 7: **EveryAware** (`http://www.everyaware.eu/`)

The EveryAware platform [36] provides an extendable data concept that could be use to enhance the possibilities of sharing and processing data feeds. The platform is running on a centralized server. This platform was the one providing the finer-granularity of data visibility with four different levels (details, statistics, anonymous, none). A REST API has been integrated to access the data (extendable data models).

Platform 8: **EveryWare Device Cloud™** (`http://www.eurotech.com/en/products/software+services/everyware+device+cloud`)

EveryWare Device Cloud™ is a proprietary cloud-based platform (PaaS) using a pay-as-you-go business model. A RESTful API supporting JSON and XML data formats, is integrated for communication with the devices. The sensors required to be connected to Eurotech gateway to be connected to the cloud. A variety of applications and tools is available within the platform to provide full end-to-end solution.

Platform 9: **EvryThng** (`http://www.evrythng.com/`)

EvryThng is a proprietary centralized platform (SaaS) that provides a persistent presence on the Web of identifiable objects (RFID, NFC, connected objects, etc.). It allows via RESTful API to store and retrieve metadata as well as real-time data for these objects. The API allows fine-access grained control to easy sharing of products information. No search tools are available to find data feeds. Billing is done on-demand. The EvryThng platform in-

[8] `http://m2m.eclipse.org`

cludes standard protocols MQTT and CoAP.

Platform 10: **Exosite** (http://exosite.com/)

Proprietary cloud-based solution (PaaS) enabling vertical markets (from devices to IoT solution). Libraries for binding of the REST API with the Exosite platform are open-source, available under the BSD license.

Platform 11: **Fosstrack** (https://code.google.com/p/fosstrak/)

Fosstrack is a closed-source SaaS platform to handle RFID devices. Electronic Product Code (EPC) cloud have been developed on top of the Fosstrack for fast deployments of RFID systems. Fosstrack shows that the fragmentation of the IoT landscape is high. However, the users stores RFID data on their own database accessed via a Tomcat server.

Platform 12: **GroveStreams** (https://grovestreams.com/)

GroveStreams proprietary cloud-based solution for analytics of data from multiple sources. It uses a REST API and JSON data format. GroveStreams is an open platform, in the cloud, that any organization, user or device can take advantage of. GroveStreams is free for small users. Large users will only be billed for what they use.

Platform 13: **Hub of All Things** (http://hubofallthings.wordpress.com/)

The Hub-of-All-Things (H.A.T.) platform has as primary objective the creation of multi-sided market platform to generate new economic and business opportunities using IoT data generated by a "smart home". An important feature of the H.A.T. is that the data belongs to the individual. It enables the end-users to get control of their data, and thus maintaining their expectations about privacy and other issues. In particular, the H.A.T architecture defines different kind of applications (in-apps and out-apps). The "in-apps" (owned by either residents, landlords or building managers) have their content enriched by local data available on the private H.A.T, while "out-apps" may be used by external platforms.

Platform 14: **Ericsson IoT-Framework** (https://github.com/EricssonResearch/iot-framework-engine)

The Ericsson IoT-Framework is a PaaS that accumulates sensor data from IP networks and focuses on the analytics and the mashing up of the data. The PaaS includes a REST API, data storage functionalities and OpenId access control for the data. The strength of this platform is the publish/subscribe mechanism, and querying of data streams, both from local and external data sources) to perform analytical tasks.

Platform 15: **IFTTT** (https://ifttt.com/)

("if this then that") is a SaaS offering, allowing a rapid composition of services called "recipes" by applying simple if-then rules to external service building blocks, such as emails, Facebook events, or Belkin's WeMo switch, that either play the role of a trigger (if) or an action (then, do). Though the service is free to use, the APIs to the service are not open at the time of writing. The recipes can be personal or shared at the discrepancy of the user; otherwise, the service building blocks rather than IFTTT deal with the user generated data.

Platform 16: **Kahvihub** (http://github.com/uh-cs-iotlab/kahvihub)

The Kahvihub platform is open-source and designed to be extremely extendable, as all components in the Kahvihub are delivered by third-parties, in the form of plugins or applications. These components are preferably scripted, to ensure a high re-usability of the platform's operations on a different platform implementation (the platform is expected to be deployed on various and heterogeneous hardware). The Kahvihub prototype is aiming to enable edge analytics by creating local networks of IoT devices that can collaboratively and autonomously analyze the data that they produce.

Platform 17: **LinkSmart™** (http://www.hydramiddleware.eu/news.php)

The LinkSmart™ middleware platform, formerly Hydra, is an open-source platform licensed under the LGPLv3. The platform enable the creation of a network for embedded systems, using semantics to discover the devices connected to the network. The middleware is based on a service-oriented architecture. The platform provides a SDK for application development and a DDK for device development.

Platform 18: **MyRobots** (http://www.myrobots.com/)

MyRobots is a proprietary cloud-based platform to connect robots to the IoT. Data format supported are JSON, XML, CSV and the web services are buildable using REST API. By default, the privacy of robots is set to public, but can be changed to private. The platform enables robots to be controlled over the Internet. The platform also includes an application store.

Platform 19: **Niagara^{AX}** (http://www.niagaraax.com/)

Niagara^{AX} [37] is a proprietary M2M dedicated software development framework that is fully distributed. It interconnect heterogeneous devices. However, details are missing about the nature of the open API.

Platform 20: **Nimbits** (http://www.nimbits.com/)

Similarly to *ARM mbed* [Platform 3], the Nimbits server has been made cloud architecture compatible, hence it scales from a single private server to a cloud architecture. Nimbits includes three levels of privacy for the data: (i) private, (ii) protected (read-only is public) and (iii) public. Control over the data and its ownership is to the user. The data is transmitted via XMPP messaging protocol. Web services access the data with HTML POST request and JSON data format. The platform is open source licensed

under the Apache License 2.0.

Platform 21: **NinjaBlock** (http://ninjablocks.com/)

NinjaBlock provides open-source hardware and open-source software to facilitate the development of sensors. However, the Ninja platform is proprietary and cloud-based. A RESTful API is avaiblable to connect NinjaBlock hardware to the cloud. NinjaBlock is open-hardware and serves as a gateway between the sensors and the Ninja platform. JSON data format is used by the platform and access is granted via the OAuth2 authentication protocol.

Platform 22: **Node-RED** (http://nodered.org/)

Node-RED is an open-source *Node.js* tool that aims to simplify the connection between IoT devices and web services. It incorporates the concept of flow for IoT devices and data that allows complex interactions between objects and services. The flow can be published on the Node-RED website for sharing. Node-RED is a creation of IBM Emerging Technology. Some cloud-based services, such as FRED[9], provide front-end for Node-RED and others [28] integrate Node-RED to their own platform (e.g., WotKit) for added values.

Platform 23: **OpenIoT** (http://openiot.eu/)

OpenIoT platform is an open-source platform, fully decentralized, that provides connectivity with constrained devices such as sensors. The platform provides a billing mechanism for the use of services.

Platform 24: **OpenMTC** (http://www.open-mtc.org/)

Cloud-based solution for M2M that aims to integrate all the standards defined by the ETSI M2M, oneM2M and 3GPP.

Platform 25: **OpenRemote** (http://www.openremote.org)

OpenRemote is a centralized open-source platform, licensed under the Affero GNU Public License. The platform supports home and domotic automation spaces using a top-down approach.

Platform 26: **Open.Sen.se** (http://open.sen.se/)

Open.Sen.se is closed-source PaaS/SaaS. A tool called *Funnel* can be used to aggregate data, but only on data feeds that are within our dashboard. It is possible to get the data from different source and mash it up. The platform uses the JSON data format and REST API for web services development. The privacy of data visualization is either public or private, data is always private (needs private keys at all times to use the API).

Platform 27: **realTime.io** (https://www.realtime.io/)

IoBridge realTime.io provides a proprietary cloud-based platform (PaaS) to connect devices to the Internet and build applications upon the data. As realTime.io uses a proprietary transport protocol for data, *ioDP*, the physical devices need to be connected to the realTime.io cloud service via a proprietary gateway. Once these gateways are connected to the service, public API (requiring realTime.io keys) enables the connection to the device to pull or push data to the devices.

Platform 28: **SensorCloud™** (http://www.sensorcloud.com/)

SensorCloud™ is a proprietary cloud-based sensor data storage and visualization platform (PaaS). It provides a fully REST compliant API and the CSV and XDR data formats are supported. It also provides tools for visualization and data mashup (MathEngine).

Platform 29: **SkySpark** (http://skyfoundry.com/skyspark/)

SkySpark is a proprietary software that can be locally installed on a private server or on a cloud and enable analytic tools for big data processing. The software does not require the connection of devices to the cloud. The software includes a REST API for connection with third-party applications and web services. The SkySpark software does not include direct management of connected devices.

Platform 30: **Swarm** (http://buglabs.net/products/swarm)

Bug's Swarm cloud-based platform (PaaS) is not open-source but provides an open-source client and some tools (unknown license). It creates swarm of resources to consume data, produce data or both among actors connected to the swarm. There is limited information on how the swarm data is stored, and who had its ownership. A RESTful API and JSON data format are usable to communicate with the devices. The platforms also provide GUI tools, such an interactive dashboard with data visualization capabilities.

Platform 31: **TempoDB** (https://tempo-db.com/)

TempoDB is a proprietary, cloud-based PaaS that enables the users to upload their data on the cloud via a REST API. The service enables to store, retrieve, and query the data, while ensuring data security, multiple back-ups and providing visualization tools, etc. This service offers billing offers depending on the user need.

Platform 32: **TerraSwarm** (http://www.terraswarm.org/)

The TerraSwarm project [38] envision the development of a new kind of operating system, the SwarmOS, to natively support the heterogeneous nature of the devices and solutions existing in the IoT and enable the infrastructure with the ability to aggregate information from a variety of data sources. The architecture relies heavily on the power of cloud computing. The operating system will be also open-source to improve its reliability and efficiency, while maximizing the potential of innovative development of "swarm-apps" build upon the system.

Platform 33: **The thing system** (http://thethingsystem.com/)

[9] https://fred.sensetecnic.com/

The thing system is a software using *Node.js* that enables discovery of smart things in the home environment. The project is open-source and licensed under the M.I.T license. The software does not provide storage functionalities and must be coupled with a PaaS to enable storage outside the home area. The software intends only to provide access remotely to smart devices of smart homes.

Platform 34: **Thing Broker** (http://www.magic.ubc.ca/wiki/pmwiki.php/ThingBroker/ThingBroker)

The Thing Broker [39] is a centralized platform that provides a Twitter-based abstraction model for *Things* and *Events*, that could be used to create local ecosystems such as smart homes. A REST API is provided by the platform to access the data and devices.

Platform 35: **ThingSpeak** (https://www.thingspeak.com/)

ThingSpeak is decentralized, open-source and copyrighted by ioBridge under the licence GPLv3. Commercial software or hardware using ThingSpeak requires a commercial agreement with IoBridge Inc. ThingSpeak provides a server that may be used to store and retrieve IoT data. It allows opening of the channels (data flows, support the JSON, XML, CSV data formats) to the public but do not provide extensive configuration of the data flows. The platform also provides visualization tools and enables the creation of widgets in Javascript/HTML/CSS to visualize the data in a more personified fashion.

Platform 36: **ThingSquare** (http://thingsquare.com/)

ThingSquare is a proprietary cloud-based platform specialized on connecting constrained devices. It require a gateway, but its firmware is open source. The gateway creates a wireless mesh networks of sensors and connect it to the Internet. The devices can access the Internet, but the devices are invisible from outside the mesh. The platform also includes a protocol for constrained devices.

Platform 37: **ThingWorx** (http://www.thingworx.com/)

ThingWorx is a proprietary cloud-based M2M dedicated platform (PaaS). It provides a variety of tools and services to support end-to-end solutions. The devices and data are accessible via a REST API. Due to acquisition of Axeda[10], the platform will likely be expanded with the IoT connectivity services, software agents and toolkits of the latter, Axeda being a proprietary cloud-based solution for M2M communication of businesses and one of the key player in the current IoT landscape.

Platform 38: **Sense Tecnic WoTkit** (http://sensetecnic.com/)

The WoTkit [3] is a proprietary cloud-based platform that offers an interesting search tool for public sensor. Public sensors do not require an account to be used.

Platform 39: **Xively** (https://xively.com/)

Xively (formerly Pachube) is a proprietary cloud-based platform (PaaS). Ownership of the data remains to the user, but the data is stored on the Xively server. Xively provides open-source APIs (in various programming languages) mostly with the BSD 3-clause license. Xively provides an extensive RESTful API including a search tool in order to retrieve feeds (flow of data) depending on selected characteristics (location radius, name, type of data stored, etc.)

[10] http://www.thingworx.com/news/\ptc-to-acquire-axeda-to-expand-internet-of-things-technology-portfolio

References

[1] E. Borgia, The internet of things vision: Key features, applications and open issues, Computer Communications 54 (0) (2014) 1 – 31. doi:http://dx.doi.org/10.1016/j.comcom.2014.09.008.
URL http://www.sciencedirect.com/science/article/pii/S0140366414003168

[2] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer Networks 54 (15) (2010) 2787–2805. doi:http://dx.doi.org/10.1016/j.comnet.2010.05.010.
URL http://www.sciencedirect.com/science/article/pii/S1389128610001568

[3] R. Lea, M. Blackstock, CityHub: a cloud based IoT platform for smart cities, in: IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom), 2014, pp. 799 – 804. doi:10.1109/CloudCom.2014.65.
URL http://eprints.lancs.ac.uk/71554/1/CityHub_CloudCom2014_pre_final.pdf

[4] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, L. Yang, Data mining for internet of things: A survey, Communications Surveys Tutorials, IEEE 16 (1) (2014) 77–97. doi:10.1109/SURV.2013.103013.00206.

[5] S. Tarkoma, H. Ailisto, The Internet of Things program: the Finnish perspective, IEEE Communications Magazine 51 (3) (2013) 10–11. doi:10.1109/MCOM.2013.6476854.

[6] J. Kim, J. Lee, J. Kim, J. Yun, M2M service platforms: Survey, issues, and enabling technologies, IEEE Communications Surveys Tutorials 16 (1) (2014) 61–76. doi:10.1109/SURV.2013.100713.00203.

[7] N. Economides, E. Katsamakas, Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry, Management Science 52 (7) (2006) 1057–1071. doi:10.1287/mnsc.1060.0549.
URL http://pubsonline.informs.org/doi/abs/10.1287/mnsc.1060.0549

[8] J. L. Pérez, A. Villalba, D. Carrera, I. Larizgoitia, V. Trifa, The COMPOSE API for the internet of things, in: Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion, 2014, pp. 971–976. doi:10.1145/2567948.2579226.
URL http://dx.doi.org/10.1145/2567948.2579226

[9] X. Qin, Y. Gu, Data fusion in the internet of things, Procedia Engineering 15 (0) (2011) 3023 – 3026, {CEIS} 2011. doi:http://dx.doi.org/10.1016/j.proeng.2011.08.567.
URL http://www.sciencedirect.com/science/article/pii/S1877705811020686

[10] M. Ma, P. Wang, C.-H. Chu, Data management for internet of things: Challenges, approaches and opportunities, in: IEEE Green Computing and Communications (GreenCom), 2013, pp. 1144–1151. doi:10.1109/GreenCom-iThings-CPSCom.2013.199.

[11] B. Villaverde, R. De Paz Alberola, A. Jara, S. Fedor, S. Das, D. Pesch, Service discovery protocols for constrained machine-to-machine communications, IEEE Communications Surveys Tutorials 16 (1) (2014) 41–60. doi:10.1109/SURV.2013.102213.00229.

[12] Z. Yan, P. Zhang, A. V. Vasilakos, A survey on trust management for internet of things, Journal of Network and Computer Applications 42 (0) (2014) 120 – 134. doi:http://dx.doi.org/10.1016/j.jnca.2014.01.014.
URL http://www.sciencedirect.com/science/article/pii/S1084804514000575

[13] R. Roman, P. Najera, J. Lopez, Securing the Internet of Things, Computer 44 (9) (2011) 51–58. doi:10.1109/MC.2011.291.

[14] Z.-K. Zhang, M. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, S. Shieh, Iot security: Ongoing challenges and research opportunities, in: IEEE 7th International Conference on Service-Oriented Computing and Applications (SOCA), 2014, pp. 230–234. doi:10.1109/SOCA.2014.58.

[15] R. Roman, J. Zhou, J. Lopez, On the features and challenges of security and privacy in distributed internet of things, Computer Networks 57 (10) (2013) 2266–2279, towards a Science of Cyber Security and Identity Architecture for the Future Internet. doi:http://dx.doi.org/10.1016/j.comnet.2012.12.018.
URL http://www.sciencedirect.com/science/article/pii/S1389128613000054

[16] S. Satyadevan, B. Kalarickal, M. Jinesh, Security, trust and implementation limitations of prominent iot platforms, in: S. C. Satapathy, B. N. Biswal, S. K. Udgata, J. K. Mandal (Eds.), Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014, Vol. 328 of Advances in Intelligent Systems and Computing, Springer International Publishing, 2015, pp. 85–95. doi:10.1007/978-3-319-12012-6_10.

[17] N. Asokan, F. Brasser, A. Ibrahim, A.-R. Sadeghi, M. Schunter, G. Tsudik, C. Wachsmann, Seda: Scalable embedded device attestation, in: 22nd ACM Conference on Computer and Communications Security (CCS), 2015.

[18] Internet Engineering Task Force, RFC 6749 (October 2012).
URL https://tools.ietf.org/html/rfc6749

[19] C. Bormann, A. Castellani, Z. Shelby, CoAP: An application protocol for billions of tiny internet nodes, IEEE Internet Computing 16 (2) (2012) 62–67. doi:10.1109/MIC.2012.29.

[20] I. Ishaq, D. Carels, G. K. Teklemariam, J. Hoebeke, F. V. d. Abeele, E. D. Poorter, I. Moerman, P. Demeester, IETF standardization in the field of the Internet of Things (IoT): A survey, Journal of Sensor and Actuator Networks 2 (2) (2013) 235–287. doi:10.3390/jsan2020235.
URL http://www.mdpi.com/2224-2708/2/2/235

[21] T. Klinpratum, C. Saivichit, A. Elmangoush, T. Magedanz, Toward interconnecting M2M/IoT standards: interworking proxy for IEEE1888 standard at ETSI M2M platform, in: The 29th International Technical Conference on Circuit/Systems Computers and Communications, 2014.

[22] IPSO Alliance, Ipso smartobject guideline, Tech. rep., Internet Protocol for Smart Objects (IPSO) Alliance (2014).
URL http://www.ipso-alliance.org/technical-information/ipso-guidelines

[23] Open Mobile Alliance, Lightweight machine to machine technical specification, draft version 1.0, Tech. rep., Open Mobile Alliance (2013).
URL http://openmobilealliance.hs-sites.com/lightweight-m2m-specification-from-oma

[24] oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TIA, TTA, TTC), Ts-0003-v1.0.1: Security solutions, Tech. rep., oneM2M (January 2015).
URL http://www.onem2m.org/images/files/deliverables/TS-0003-Security_Solutions-V1_0_1.pdf

[25] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems 29 (7) (2013) 1645–1660. doi:10.1016/j.future.2013.01.010.
URL http://dx.doi.org/10.1016/j.future.2013.01.010

[26] X. Su, J. Riekki, J. K. Nurminen, J. Nieminen, M. Koskimies, Adding semantics to internet of things, Concurrency and Computation: Practice and Experience 27 (8) (2015) 1844–1860. doi:10.1002/cpe.3203.
URL http://dx.doi.org/10.1002/cpe.3203

[27] A. Maarala, X. Su, J. Riekki, Semantic data provisioning and reasoning for the internet of things, in: Internet of Things (IOT), 2014 International Conference on the, 2014, pp. 67–72. doi:10.1109/IOT.2014.7030117.

[28] M. Blackstock, R. Lea, Toward a distributed data flow platform for the web of things (distributed node-red), in: Proceedings of the 5th International Workshop on Web of Things, WoT '14, ACM, New York, NY, USA, 2014, pp. 34–39. doi:10.1145/2684432.2684439.
URL http://doi.acm.org/10.1145/2684432.2684439

[29] G. Lewis, S. Echeverria, S. Simanta, B. Bradshaw, J. Root, Tactical cloudlets: Moving cloud computing to the edge, in: IEEE Military Communications Conference (MILCOM), 2014, pp. 1440–1446. doi:10.1109/MILCOM.2014.238.

[30] M. Kovatsch, M. Lanter, S. Duquennoy, Actinium: A restful runtime container for scriptable internet of things applications, in: 3rd International Conference on the Internet of Things (IOT), 2012, pp. 135–142. doi:10.1109/IOT.2012.6402315.

[31] J. He, Y. Zhang, G. Huang, J. Cao, A smart web service based on the context of things, ACM Transactions on Internet Technology (TOIT) 11 (3) (2012) 13:1–13:23. doi:10.1145/2078316.2078321.
URL http://doi.acm.org/10.1145/2078316.2078321

[32] J. Burt, HyperCat Spec Aims for Internet of Things Interoperability, EWeek (n.a.) (2014) n.a.

[33] O. Mazhelis, E. Luoma, H. Warma, Defining an Internet-of-Things ecosystem, in: S. Andreev, S. Balandin, Y. Koucheryavy (Eds.), Internet of Things, Smart Spaces, and Next Generation Networking, Vol. 7469 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 1–14. doi:10.1007/978-3-642-32686-8_1.
URL http://dx.doi.org/10.1007/978-3-642-32686-8_1

[34] A. Gawer, M. A. Cusumano, How companies become platform leaders, MIT/Sloan Management Review 49 (2) (2012) n.a.

[35] G. Kortuem, F. Kawsar, Market-based user innovation in the Internet of Things, in: Internet of Things (IOT), 2010, pp. 1–8. doi:10.1109/IOT.2010.5678434.

[36] M. Becker, J. Mueller, A. Hotho, G. Stumme, A generic platform for ubiquitous and subjective data, in: Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication, UbiComp '13 Adjunct, ACM, New York, NY, USA, 2013, pp. 1175–1182. doi:10.1145/2494091.2499776.
URL http://doi.acm.org/10.1145/2494091.2499776

[37] T. Samad, B. Frank, Leveraging the web: A universal framework for building automation, in: American Control Conference, 2007. ACC '07, 2007, pp. 4382–4387. doi:10.1109/ACC.2007.4282471.

[38] E. A. Lee, J. D. Kubiatowicz, J. Rabaey, A. Sangiovanni-Vincentelli, S. A. Seshia, J. Wawrzynek, D. Blaauw, P. Dutta, K. Fu, C. Guestrin, R. Jafari, D. L. Jones, V. Kumar, R. Murray, G. Pappas, A. Rowe, C. Sechen, T. S. Rosing, B. Taskar, The terraswarm research center (TSRC) (a white paper), Tech. Rep. UCB/EECS-2012-207, EECS Department, University of California, Berkeley (November 2012).
URL http://terraswarm.org/pubs/2.html

[39] R. A. Perez de Almeida, M. Blackstock, R. Lea, R. Calderon, A. F. do Prado, H. C. Guardia, Thing broker: a twitter for things, in: Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication, UbiComp '13 Adjunct, ACM, New York, NY, USA, 2013, pp. 1545–1554. doi:10.1145/2494091.2497588.
URL http://doi.acm.org/10.1145/2494091.2497588