Internet of Things: A Definition & Taxonomy

Conference Paper · September 2015

DOI: 10.1109/NGMAST.2015.71

5 authors, including:

Bruno Dorsemaine

5 PUBLICATIONS **340** CITATIONS

SEE PROFILE

jean-philippe Wary

France Télécom

45 PUBLICATIONS **738** CITATIONS

SEE PROFILE

Nizar Kheir

Thales Group

2 PUBLICATIONS **282** CITATIONS

SEE PROFILE

Pascal Urien

Institut Mines-Télécom

182 PUBLICATIONS **1,833** CITATIONS

SEE PROFILE

Internet of Things: a definition & taxonomy

Bruno Dorsemaine, Jean-Philippe Gaulier,
Jean-Philippe Wary and Nizar Kheir
Orange
Paris, France
{bruno.dorsemaine, jeanphilippe1.gaulier,
jeanphilippe.wary, nizar.kheir}@orange.com

Pascal Urien
Telecom ParisTech
Paris, France
pascal.urien@telecom-paristech.fr

***Abstract*—The Internet of Things (IoT) has various fields of application including health care, resource management, asset tracking, etc. Depending on the use case, various technologies like RFID, Wireless Sensor Network (WSN) or Smart Objects can be used. With each of these comes a specific vision of what the IoT and connected objects are and – to our knowledge – there is no global picture of the IoT. The issue with this approach is that specific problems have been addressed before global ones: what if something has been missed? We propose a definition and taxonomy for connected objects and the IoT.**

***Index Terms*—Internet of Things; connected objects; definition; taxonomy.**

I. Introduction

The IoT is – soon to be – everywhere. According to many companies like Gartner [1] and IBM [2], there will soon be billions of objects connected together and gathering data on everything they can in order to make predictions, improve processes, etc. However in their predictions, some include tablets and smartphones [3] whereas others do not [1]. As long as there is no common definition of the IoT, it is impossible to compare those evaluations or objects that uses *really* different technologies (like RFID [4], WSN [4] or Smart Objects [5], [6]).

The technology specific visions [7] come from the different requirements and needs of the use cases the IoT is applied to. A wireless access card will, in fact, have very different specifications when compared to a smart fridge: the access card will most likely use RFID, use hardware cryptography, harvest its electricity, be very constraint in terms of CPU, memory and RAM, etc. whereas the fridge will communicate over Ethernet or Wi-Fi with other machines on the Internet, be connected to the mains, have very little to no constraints in terms of CPU, memory and RAM, etc. Despite many differences between these technologies and the possible applications, they still make the connection of the physical world possible and have needs in common like security.

This work includes itself in an approach to define security policies for the IoT in a corporate environment. Due to the lack of literature on the topic, the first step is to propose a way to classify connected objects. Then derive classes of objects according to their needs and, in the end, security policies. Hence, to be able to easily classify various connected objects – *i.e.* even if they use different technologies – a precise and exhaustive definition is needed. This paper is organized according to the following plan. We first define what a connected object and the IoT are. Then, in Fig. III, we propose a taxonomy. Afterwards, we give a few examples of classification and finally conclude this paper.

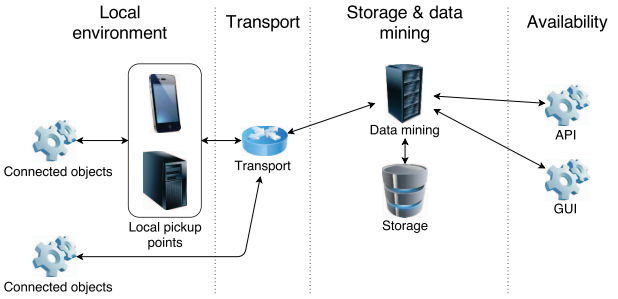

Fig. 1. Architecture related to the IoT

II. A definition

Before giving a definition of the IoT, an end-to-end vision is needed.

A. The architecture

Despite the number of possible fields of application the IoT and the associated technologies can be applied to, connected objects are always associated to the same kind of architecture: data needing to be transported, stored, processed and made available.

In [4], the authors described a generic four level architecture (names of the levels have been changed to take the definition we propose into account) that is being used for the IoT. It is shown in Fig. 1:

- The local environment contains the connected objects and the local pickup points. These elements communicate through wired technologies (Ethernet, optic fiber, etc.) or wireless links [8] (Bluetooth Low Energy, Wi-Fi, ZigBee, etc.). The local pickup points (optional) can be smartphones, small computers and other objects. They are used as gateways to reach the infrastructure by objects that are not powerful enough (battery, computing power,

978-1-4799-8660-6/15 $31.00 © 2015 IEEE
DOI 10.1109/NGMAST.2015.71

etc.). Sometimes, they allow direct user interaction with the objects (an application on a smartphone for example).
- The transport level allows the objects or local pickup points to communicate with the command servers.
- The storage and data mining generally take place in the cloud and allow the processing of the data.
- Finally, the user, or other systems, can access the data through APIs or GUIs.

We can notice, that only the first level – the local environment – is specific to the IoT, the other three can be found anywhere massive amount of data are being treated.

B. A definition of the IoT

Taking into account the previous elements, a definition for a connected object could be: "Sensor(s) and/or actuator(s) carrying out a specific function and that are able to communicate with other equipment. It is part of an infrastructure allowing the transport, storage, processing and access to the generated data by users or other systems."

Then, a definition for the IoT would be: "Group of infrastructures interconnecting connected objects and allowing their management, data mining and the access to the data they generate."

III. A TAXONOMY FOR CONNECTED OBJECTS

Following from the definitions, we described a taxonomy for connected objects as shown in Fig. 2. It revolves around the following categories: energy, communication, functional attributes, local user interface, hardware and software resources and cost. As the IoT is a relatively new and evolving field, this taxonomy is extensible in order to take new cases into account.

A. Energy

For many objects, energy is critical. In some cases, it can even determine the lifespan of an object. This characteristic can be divided in two parts:

- *Source*. The way the object will use to get its power is extremely important: it will determine many other characteristics like the ability to work continuously, the computational power of the CPU, etc. In [9], the authors provide with four types of power source for a connected object: *harvesting* (the energy is gathered from the environment, *e.g.* solar panels), *periodically recharged or replaced*, *non-replaceable primary source* (the power source determines the lifespan of the object) and *mains-powered* (the power source is virtually unlimited).
- *Management*. The management of the power source can be summed up to how the power source will be used and, in the end to how long the object can operate with a given amount of power. In [9], three types of power management are described for the communication interfaces, they can also be applied to the management of the power source in general:
  - *Normally-off*. The object is off most of the time and wakes up periodically or on a given signal from the environment (power available through harvesting, for example).
  - *Low power*. The object has to preserve its battery to last over time; hence it has to be able to consume the lowest amount of power possible.
  - *Always-on*. There is no reason for the object to implement specific measures in order to limit power consumption.

B. Communication

As some objects provide several communication interfaces, it is possible to give one occurrence of this section for each of these interfaces.

1) Type: The object can communicate with the pickup points using various types of interfaces that can be divided in two categories: *wired* and *wireless*. Wired interfaces can include Ethernet, buses... whereas light, Wi-Fi, etc. are wireless ones.

2) Local pickup point: In Fig. 1, the local pickup point is part of the local environment level. It is a gateway to the other parts of the architecture for the objects that are connected to it. The use of a local pickup point may have implications on the battery life, computational power and cost of an object because the object may not have to use heavy protocols/addressing systems (it does not need to communicate with machines on the Internet thanks to the local pickup point) and so does not need a big computational power for this purpose.

3) Total disconnection: Sometimes, it is possible to disable the "connected" feature of an object (or at least not to use it) but to keep its main functionalities (*e.g.* counting and displaying the number of steps for a pedometer). It is sometimes called "chip silence". In other cases, the object cannot function properly without a connection to the pickup point and the rest of the architecture.

4) Initiation of the communications: Either the object or the pickup point can initiate the communication, it has an impact on how the communication interfaces are being used. There are three possibilities:

- *Object*. In this case, as the object is the one that starts the communication its interfaces does not have to be turned on all the time. It means that the object will be able to save power while it is not communicating by turning its communication interfaces off.
- *Pickup point*. In this mode, this object needs to wait for the pickup point to initiate the communication. It means that, contrary to the previous mode, the object cannot turn its communication interfaces off and that it has to be registered with the pickup point.
- *Object and pickup point*. Either the object or the pickup point can start a communication. As previously, the object has to constantly listen of its interfaces and register itself to the pickup point.

5) Security:

- *Authentication*. This process allows the objects and pickup points to recognize each other, to know *who*

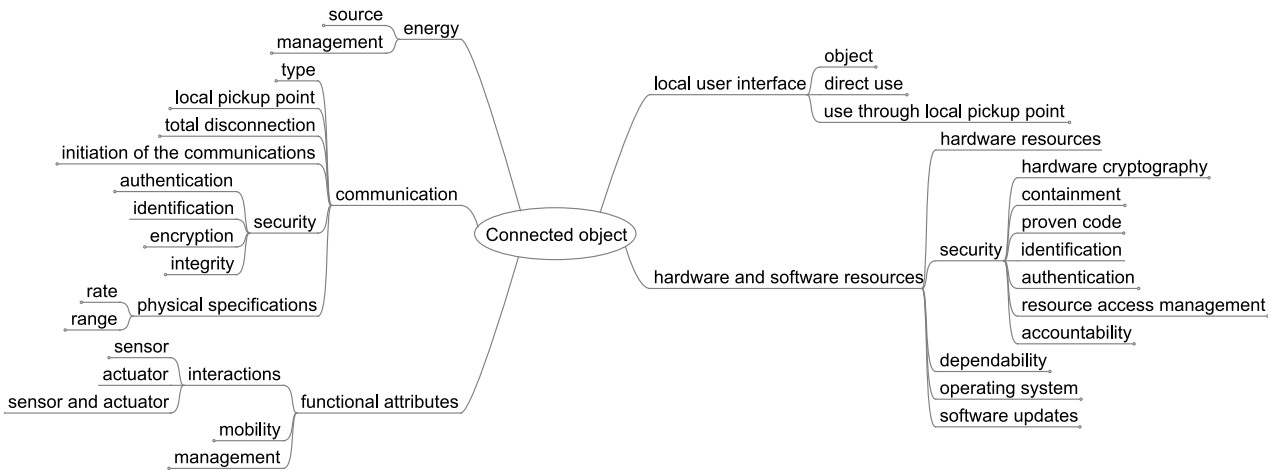

Fig. 2. Taxonomy for connected objects

they are *talking* to. Several types of authentication are possible:

- *Mutual*. Both the object and the pickup point prove their identity. The main problem it raises is that there is a need for configuration on both sides. The most problematic one is the objects': they might not be easily reachable physically, management might not be possible, etc. It could be an issue if there is a new pickup point to configure for example.
- *One way*. There are two possibilities here: the object or the pickup point authenticates itself. In either of these possibilities, the other component can be a fake, there is no way for the authenticated one to be sure. The case of the authentication of the pickup point raises the same issues as in the mutual authentication case.
- *None*. There is no authentication between the two components. Either of them could be a fake.

- *Identification*. Whereas the authentication allows objects to know that they are part of a known group, identification makes it possible to differentiate two object that belong to the the same group. As with authentication, three types of identification are possible: *mutual*, *one way* and *none*.
- *Encryption*. Encryption is what makes it *theoretically* impossible for an attacker to eavesdrop a communication if it is implemented correctly with state of the art algorithms. Not using encryption is not necessarily a problem if the transmitted data is not sensitive (*e.g.* a temperature reading emitted from a weather station). A simple classification could take the form of a *yes*/*no* answer or sort objects according to the algorithms they use to encrypt their communications.
- *Integrity*. It is possible for a communication to be disturbed by external elements. In this case, insuring the integrity of the data is possible but at a certain cost: information has to be added to the initial data in order to insure integrity. This additional information has to be calculated/verified (computational power) and sent/received with the rest of the data (communication). Hence, in some cases, it is more interesting to accept transmitting data without integrity controls (*e.g.* with many sensors in the same area gathering the same kind of data, it might be simpler not to use integrity checks but to discriminate some results by correlation). Objects could be classified with a *yes*/*no* answer or based the algorithms that are being used.

6) Physical specifications: The following characteristics are intimately linked and also play a big role in the choice of the "energy" specifications: the lower the rate and the range are, the lower the power consumption will be for a given amount of data to transmit.

- *Rate*. It is the *theoretical* maximum amount of data the communication interface can send in a given amount of time. It is generally measured in Kb/s or Mb/s. A good way to classify objects for this criteria could be the technologies that are being used or ranges of rates.
- *Range*. It is the *theoretical* maximum distance the data can reach while remaining understandable with a given technology. To classify objects according to this characteristic, the technologies or ranges of distances in meters or types of networks (PAN, LAN, building scale, city scale, etc.) can be used.

C. Functional attributes

1) Interactions:

- *Sensor*. The object is able to extract various data from its environment like temperature, light exposure or just give its position (*e.g.* RFID tag). There are two types of sensors:
  - *Sensor with memory*. These sensors can store the data they gather in order to send it periodically. This can

be used to save power or if the pickup point is not always reachable.
  - *Sensor without memory*. With this kind of sensors, if the data is not send when gathered, it is lost. It means that the object has to emit continuously and that a pickup point must be reachable.
- *Actuator*. The object is able to act within its environment: make a movement, produce cold, emit light, etc.
- *Sensor and actuator*. The object is a hybrid of the two previous categories: it can gather data and act within its environment. As such, it inherits the specificities of the sensors and can have memory or not.

2) Mobility: Some objects have been made with the idea of it moving (*e.g.* a pedometer) and others have not (*e.g.* a thermometer). We considered two categories of objects for this criteria:

- *Fixed*. The object is fixed or is in a static / very constraint (in terms of mobility) environment, a room for example. It has not been made with it moving in mind.
- *Mobile*. The object has been made to move.

3) Management: Depending on the use case, objects can be manageable or not. In the case of a manageable object, another question could be weather it is possible physically or over one of the communication interfaces.

D. Local user interface

1) Object: Sometimes, the object has components that are dedicated to a direct interaction with its user so that they can configure it, use its basic functions, etc. There are four types of interfaces:

- *Active*. Parts of the object are dedicated to the interaction with the user: buttons, etc.
- *Passive*. The user cannot interact directly with the object but the it can communicate with them through various components: screen, lights, sounds, vibrations, etc.
- *Active and passive*. The object is a hybrid of the two previous categories and both the object and the user can interact with each other.
- *None*. Nor the object or the user can interact directly with each other.

2) Direct use: If the object has an active or active/passive user interface, it sometimes is possible for the user to – at least – use its basic functionalities.

3) Use through local pickup point: If the object communicates with the rest of the architecture through a local pickup point, it might be possible for the user to – at least – use the basic functions of the object through the local pickup point.

E. Hardware and software resources

1) Hardware resources: The amount of RAM, memory and CPU the object has at its disposal conditions many things like the "intelligence" that can be embedded in the object, the type of power source it will rely on, etc. In [9], the authors defined a three ranges classification of devices using RAM and memory but it only addresses "small" (in terms of RAM and memory) devices.

2) Security:

- *Hardware cryptography*. Hardware cryptography can be provided by an object in order to make encryption faster (than with the same algorithm implemented with software). The classification can be made with a *yes/no* answer or by using the names of the algorithms.
- *Containment*. Whether processes can be executed in a contained manner or not.
- *Proven code*. In some cases, the user needs to be sure that the object will work as it is expected to, that its behavior is deterministic. Then, proving the object's code is needed.
- *Identification*. When the user can physically interact with the object, the identification process can ensure – if correctly implemented – that the user and the object are who/what they *say* they are. As with the identification for the communication, there are three possibilities: *mutual*, *one way* and *none*.
- *Authentication*. When the user can physically interact with the object, the authentication process can ensure – if correctly implemented – that the user and the object have the right to interact with each other. As with the authentication for the communication, there are three possibilities: *mutual*, *one way* and *none*.
- *Resource access management*. It is possible to access the object's resources (every component of the object is seen as a resource) through the infrastructure (including local pickup points if possible) and the object if a suitable local interface is provided. Resource access management gives the possibility to authorize or not access to some of the resources *e.g.* a measurement, a specific process.
- *Accountability*. Whether the object keeps traces of its activity or not.

3) Dependability: The object can provide several means to ensure that its information is correct (*e.g.* an open/close sensor that has two different ways of knowing if what it monitors is open or closed) is more dependable than an object that only provides one.

4) Operating system:

- *Software and hardware*. The object runs with an operating system that is both software and hardware.
- *Kernel only and hardware*. The object only runs a kernel on a hardware platform.
- *Pure hardware*. The operating system that the object runs with is pure hardware.

5) Software updates: Whether it is possible or not for the object's software to be updated.

IV. Examples of classification

Fig. 3 describes the classification of a few objects according to the previously defined taxonomy. Those have been selected because of the variety of use cases they represent. Their classification highlights the interdependencies between the characteristics an object can have, like the type of power source, how it is being managed, the rate and range provided by the communication interface, etc.

To explicit the interdependencies, we can take the example of the RFID access card from Fig. 3. This device is only powered by its local collector when at the right distance. Hence, it does not include a power source and does not use lots of power to function. Contrary to an object that is connected to the mains, power consumption is critical here and is impacted by many characteristics. As most of the access card's *work* consists in cryptography and communication, hardware cryptography (it also is faster than software cryptography and does not require a *strong* and energy consumer processor as it uses dedicated components) is being and so is a very low rate and range wireless communication interface.

In comparison to the access card, the smart washing machine has completely different specifications (only using the previously defined taxonomy, not the *really* use cases). The fact that it is not mobile has a huge impact on other characteristics: a high rate wired connection is available and the device is powered through the mains. Among the other differences are the user interface (it is possible to use the object directly) and the security measures (the type of resources the object needs to function is impacted).

V. Conclusion

In this paper, we proposed a generic definition for connected objects as well as the Internet of Things. Following on from these definitions, we were able to define a taxonomy for connected objects.

With these elements, we defined a common vocabulary that covers existing and forthcoming equipment. It can be used to easily classify and compare objects on a common scale.

The next step will consist in extracting object classes – based on security requirements – from the proposed taxonomy, providing a generic end-to-end risk analysis for each of these classes and defining security measures in order to reduce the risks to an acceptable level.

References

[1] J. Tully. (2015) Mass adoption of the internet of things will create new opportunities and challenges for enterprises.

[2] P. Brody and V. Pureswaran. (2014) Device democracy – saving the future of the internet of things.

[3] D. Evans. (2011) The internet of things how the next evolution of the internet is changing everything.

[4] L. D. X. Shancang Li and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, pp. 243–259, 2015.

[5] F. K. Gerd Kortuem, Dan Fitton and V. Sundramoorthy. (2010) Smart objects as building blocks for the internet of things.

[6] J.-P. Vasseur and A. Dunkels, *Interconnecting Smart Objects with IP – The Next Internet*, 2010.

[7] A. I. Luigi Atzori and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[8] L. P. Luca Mainetti and A. Vilei, "Evolution of wireless sensor networks towards the internet of things: a survey," in *Proc. Software, Telecommunications and Computer Networks (SoftCOM), 2011*, Dubrovnik, Croatia, 2011, pp. 1–6.

[9] M. E. C. Bormann and A. Keranen, "Terminology for constrained-node networks," RFC 7228, May 2014.

			Wireless sensor from a WSN *	RFID access card	Smart washing machine	Smart car	Parrot Flower Power	FitBit One
Energy	Source		harvesting	harvesting	mains-powered	periodically recharged or replaced	periodically recharged or replaced	periodically recharged or replaced
	Management		low-power	normally-off	normally-off	normally-off	low-power	low-power
Communication **	Type		wireless	wireless	wired	wireless	wireless	wireless
	Local pickup point		yes	yes	no	no	yes	yes
	Total disconnection		no	no	yes	yes	no	yes
	Initiation of the communications		object	pickup point	object and pickup point	object and pickup point	pickup point	object
	Security	Authentication	mutual	mutual	mutual	mutual	mutual	mutual
		Identification	mutual	mutual	mutual	mutual	mutual	mutual
		Encryption	no	yes	yes	yes	yes	yes
		Integrity	yes	yes	no	yes	no	no
	Physical specifications	Rate	< 100Kb/s	< 100Kb/s	> 1Mb/s	> 1Mb/s	from 100Kb/s to 1Mb/s	from 100Kb/s to 1Mb/s
		Range	city scale	PAN	LAN	city scale	LAN	LAN
Functional	Interactions		sensor without memory	sensor with memory	sensor with memory and actuator	sensor with memory and actuator	sensor with memory	sensor with memory
	Mobility		fixed	mobile	fixed	mobile	fixed	mobile
	Management		yes	no	yes	yes	no	no
Local user	Object		none	none	active and passive	active and passive	none	active and passive
	Direct use		no	no	yes	yes	no	yes
	Use through local		no	no	no	no	yes	yes
Hardware and software resources	Hardware resources		< 10Kio RAM / < 100Kio memory	< 10Kio RAM / < 100Kio memory	> 50Kio RAM / > 250Kio memory	> 50Kio RAM / > 250Kio memory	from 10 to 50Kio RAM / from 100 to 250Kio memory	from 10 to 50Kio RAM / from 100 to 250Kio memory
	Security	Hardware cryptography	no	yes	no	yes	no	no
		Containment	no	yes	no	yes	no	no
		Proven code	no	yes	no	no	no	no
		Authentication	none	none	none	user → object	none	none
		Identification	none	none	none	user → object	none	none
		Resource access management	no	yes	no	yes	no	no
		Accountability	no	yes	no	yes	no	no
	Dependability		yes	no	yes	yes	no	no
	Operating system		software and hardware	software and hardware	software and hardware	software and hardware	software and hardware	software and hardware
	Software updates		yes	yes	yes	yes	yes	yes

* Wireless Sensor from a WSN: With the example of a WSN dedicated to forest fire detection.

** Communication: in order to simplify the classification, only one communication interface has been considered for each object. For example, a smart washing machine and a smart car may also provide Bluetooth and WiFi interfaces.

Fig. 3. Classification of a few connected objects