



The Limitations of the IoT and how the Web of Things Can Help

By Dominique D. Guinard and Vlad M. Trifa

In this article, we define the Internet of Things and its limitations and describe how the Web of Things can .

Capturing the essence of the Internet of Things in one sentence is nearly impossible. Although the term itself is relatively new, the concepts have been around for decades; therefore there are no clear boundaries to what the IoT is or isn't. Nevertheless, the broad definition of the Internet of Things vision is simply a world where the Internet becomes more than the collection of multimedia pages it is today, but extends into the physical, real-time world using a myriad of embedded devices. In short, the simplest definition we can offer for the Internet of Things is the following:

The **Internet of Things** (IoT) is a system of physical objects that can be discovered, monitored, controlled or interacted with by electronic devices which communicate over various networking interfaces, and eventually can be connected to the wider Internet.

Two decades ago, a world where everyday objects could *feel* the world through sensors and then analyze, store, or exchange information existed only in science-fiction novels. Today, such scenarios are increasingly becoming reality, thanks to the colossal progress in embedded devices that brought into the world a new class of objects: **Smart Things**. A Smart Thing (which we'll refer to as a **Thing** - with capital "T" - in the rest of this book) is a physical object that is digitally augmented with one or more of the following:

- **Sensors** (temperature, light, motion, etc.)
- **Actuators** (displays, sound, motors, etc.),
- **Computation** (can run programs and logic),
- **Communication interfaces** (wired or wireless).

Things extend the world we live in by enabling a whole new range of applications. Being able to put a bunch of powerful, tiny, and cheap computers everywhere around us makes it

For source code, sample chapters, the Online Author Forum, and other resources, go to

<http://www.manning.com/quinard/>

possible to monitor and interact with the physical world with a much finer spatial and temporal resolution than ever before.

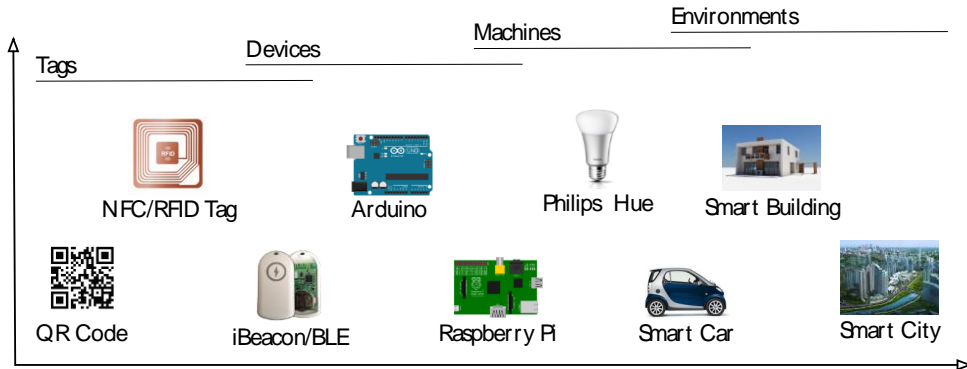


Figure 1. - The IoT landscape. The IoT is a network of Things, which are anything that can be connected in some form to the Internet. From a box of oranges with an RFID tag, to a smart city, and to every Thing in between, all these digitally augmented objects make up the IoT.

More concretely, the *Things* in the Internet of Things can range from very simple *tagged products* such as your Fedex package with an Auto-ID tag (AUTOMATIC IDENTIFICATION methods, such as bar codes, QR codes, or NFC and RFID tags) attached to it so can be tracked from the shipping center to your door; to more elaborate, complex, and wirelessly *connected products*, devices or machines such as security systems, your car or a factory assembly line; and all the way up to a building or even a city. The *Internet* part simply means that the Thing (its services or data about/from it) can be accessed and processed by other applications through the existing Internet infrastructure. This does not imply that the thing must be physically connected to the Internet. The communication network used can be an Auto-ID technology, short-range radio (e.g., Bluetooth, ZigBee, etc.) or a local Wi-Fi network in a building.

Unfortunately, building a single global ecosystem of Things communicating with each other seamlessly is virtually impossible today. There is no unique and universal application protocol for the Internet of Things that can work across the many networking interfaces available today. To put it bluntly, the IoT of today is essentially a collection of isolated *Intranets of Things* that can't really interact with each other.

For the IoT to become a reality, we will need a single universal application layer protocol (think "language") for devices and applications to talk to each other, regardless of how they are physically connected. Instead of creating yet another protocol from scratch (as many IoT projects have been – and keep – doing), why not simply reuse something that is already

widely used for building scalable and interactive applications, such as the Web itself? This is what the **Web of Things** is all about: reuse and leverage readily available and widely popular Web protocols, standards and blueprints to make data and services offered by objects more accessible to a larger pool of (Web) developers.

Enter the Web of Things

The limitations of the IoT become obvious as soon as one wants to integrate devices from various manufacturers into a single application or system. To illustrate how the Web of Things can deal with these limitations, let’s consider the life of Johnny B., the owner of a famous hotel chain in several cities around the world. Johnny would like to digitally connect all the appliances in all the rooms of his hotels, so that he is able monitor, control, and improve the management of his hotels from the deck of his yacht in the Bahamas, using a single Web-based “hotel control center” application. At the same time, this system could also offer a more pleasant and personalized experience to each guest in his hotels, as shown in Figure 2.

WEB OF THINGS SCENARIO: CONNECTED HOTEL

Building this smart hotel system will likely require electronic door locks made by company *Alpha*, security cameras from company *Beta*, and a control application to manage all of this made by company *Gamma*. Making these devices and systems talk and work with each other will require lots of custom system integration work to be done. Johnny could contract a specialized company and spend his well-earned economies on a substantial project that will take months to complete. Such a complex project will have the robustness of a Jenga tower (touch the wrong piece and all hell breaks loose): it will be swamped with bugs and hacks, therefore will be a nightmare to maintain and to extend. In this scenario, there is little doubt that Johnny will run out of money before he gets the system he wants.

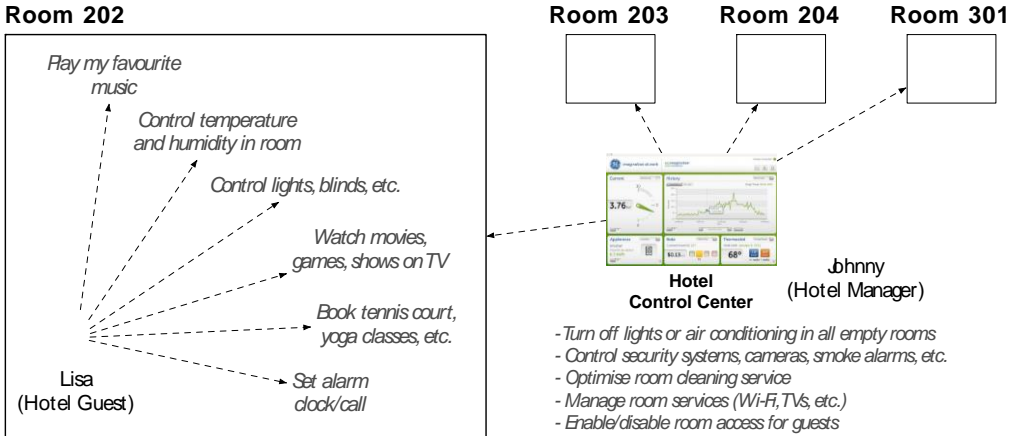
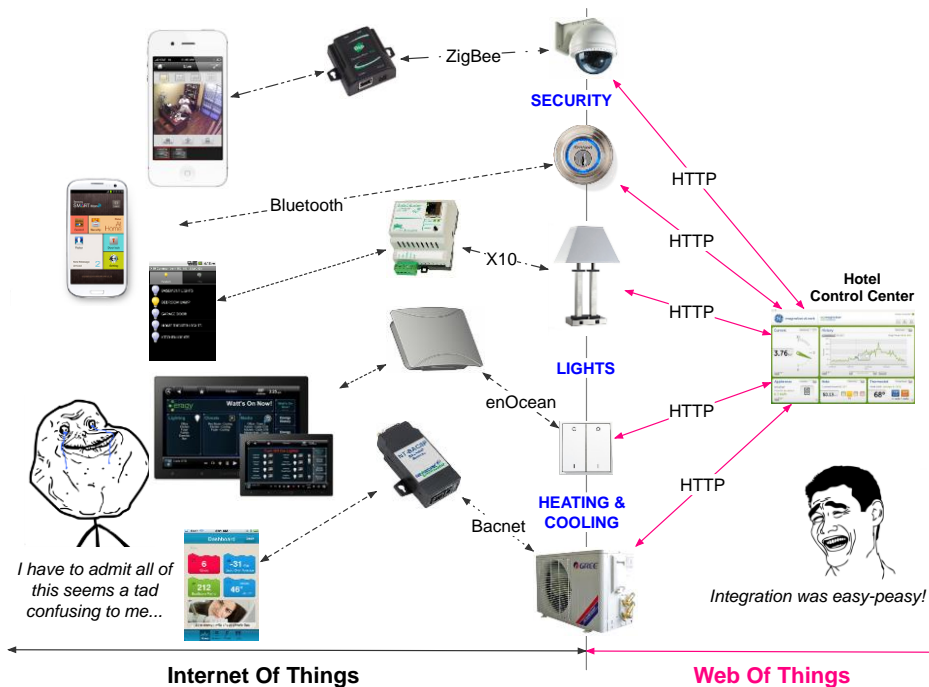


Figure 1 – Johnny would like to digitally connect the appliances in all rooms of his hotel. First, guests could

have access to a variety of services from controlling their room (lights, air conditioning, entertainment, etc.), to booking hotel facilities, to ordering food and drinks – all of this on their mobile phone. Second, this system would allow Johnny to coordinate and optimize all aspects of his hotel in a centralized and efficient manner, without having to use a variety of siloed applications and tools.

If Johnny is keen on DIY (“Do it yourself”), he could also decide to build the whole system himself. He could buy all the pieces from the same company, so they are designed to fit and work well together. Unfortunately, it’s unlikely any solution on the market will have all the sensors and equipment he would like to control. Even if he could find this perfect system, it’s not likely that the user interface will be on par, especially when it comes to customizing access control, and configurability. If he also wants the system to be scalable, reliable, and secure, you can easily double - if not triple - the time it will take to build it. Should we also talk about the mobile apps that need to be built for the hotel guests? You get the idea...

Johnny’s life may seem surreal. Sadly, it is pretty much how the IoT looks like today. We know this because we had the chance to work with many Johnnies over the last decade. Ranging from shop managers wanting to combine their existing security cameras with RFID gates to create smarter security systems, to LED manufacturers wanting their lights to be controlled from the Web, we’ve experienced this scenario over and over.



For source code, sample chapters, the Online Author Forum, and other resources, go to <http://www.manning.com/quinard/>

Figure 2 – In the IoT, hundreds of incompatible protocols co-exist today. This makes the integration of data and services from various devices extremely complex and costly. In the Web of Things, any device can be accessed using standard Web protocols. Connecting heterogeneous devices to the Web makes the integration across systems and applications much simpler.

Wouldn't it be great if any device could be easily integrated and consumed by any application, regardless of the networking protocols used by those devices? This is exactly what the Web of Things enables you to do, as illustrated in Figure 3.

COMPARING IOT AND WOT

As more everyday objects will be digitally augmented, the next logical step is to use the World Wide Web ecosystem and infrastructure to build applications for the IoT, effectively breaking this ongoing "one device, one protocol, one app" pattern. It would be particularly interesting to push down to each of those tiny devices the exact same technology that helped modern Web sites such as Facebook or Google scale to millions of concurrent users, without compromising on security or performance. The idea of maximizing existing and emerging tools and techniques used on the Web and apply them to the development of IoT scenarios is what we call the Web of Things.

While the IoT has been busy resolving networking problems, the Web of Things relies exclusively on application level protocols and tools. Mapping any device into a Web mindset makes the Web of Things agnostic to the physical and transport layer protocols used by devices. As you will learn to do in the next chapters, pretty much any custom protocol or standard can be linked to the Web thanks to software or hardware "bridges" (via proxies or gateways).

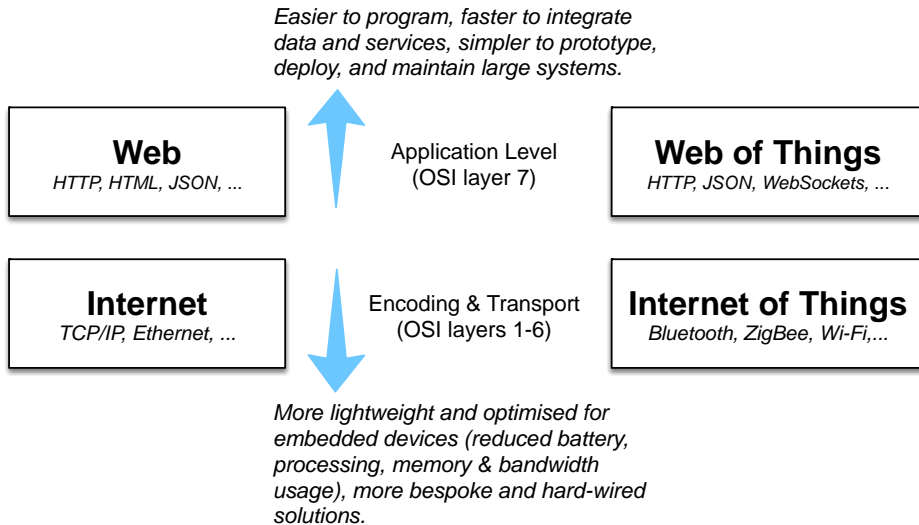


Figure 3 – The Web of Things relies only on the highest OSI Layer (7), which handles applications, services and data. Working on such a high level of abstraction makes it possible to connect data and services from many devices regardless of the actual transport protocols used. In contrast, the Internet of Things does not advocate a particular application level protocol and is usually focusing on the lower layers of the OSI stack.

Abstracting the complexity and variety of the lower-level protocols behind the simple model of the Web offers many advantages. Just like the Web has become the global integration platform for distributed applications over the Internet, the Web of Things facilitates the integration of all sorts of devices and the applications that interact with them. In other words, by hiding the complexity and differences between various transport protocols used in the IoT, the Web of Things allows developers to focus on the logic of their applications without having to bother about how this or that protocol or devices actually work.

Coming back to our smart hotel scenario, if all devices (regardless of their manufacturer) could offer a standard Web API, integration of data across device and applications will pretty much come out of the box, as *all devices would speak the same language*. In this case, the hotel owner (or system integrator) would need only to worry about building the “hotel control center” application that is likely going to be a Web mashup – a single Web application that combines data and services from various sources. He would not have to bother learning the

specifics of each protocol used by the various devices he wants to use¹. This will not only take significantly less time to build, but also minimize the effort required to maintain the system each time a device or service is added, removed, or updated.

Making this vision a reality has been the objective of the Web of Things community we started in 2007². Using HTTP and other Web standards or tools to interact with embedded devices made perfect sense to us. At the time, this idea seemed unrealistic and even pointless, and we got our fair share of criticism. However, recent embedded Web servers with advanced features can be implemented with only 8 KB of memory. Thanks to efficient cross-layer TCP/HTTP optimizations they can run on tiny embedded systems or even smart cards such as the chip on your credit card! Since embedded Web servers in an IoT generally have more limited resources than the clients who access them (such as browsers or mobile phones) and thanks to the massive developments in the JavaScript community, it has become increasingly easy to shift a lot of the workload from devices to client applications and even to the cloud.

¹ A "short" list of automation protocols: https://en.wikipedia.org/wiki/List_of_automation_protocols

² <http://webofthings.org>

The Pillars of the Web of Things



Figure 4 – The Web of Things is the ability to use modern Web standards directly on embedded devices. By leveraging all these standards for IoT scenarios, we both enable new types of interactive applications to be built, and also make sure that devices can be integrated with modern Web applications and services with minimal effort.

In the Web of Things, devices and their services are fully integrated in the Web because they use the same standards and techniques as traditional Web sites. This means that you can write applications that interact with embedded devices in exactly the same way as you would interact with any other Web service that uses Web APIs and in particular using *RESTful architectures*.

REST is an architectural style for developing distributed applications and is the basis upon which the modern Web is built. The essence of REST is to focus on creating loosely coupled services that can be easily reused, which is implemented using URIs, HTTP, and standardized media types. Abstracting the services from their application-specific semantics thanks to a uniform interface (HTTP verbs and response codes) makes it easy to build loosely coupled services as it provides a simple mechanism for clients to select the best possible representations for interactions. This makes the Web the ideal substrate for building a “universal” architecture and Application Programming Interface (API) to interact with Things.

In practice, this means you can start interacting with Things via Web browsers and explore the Web of Things just as you would surf the Web (via links to other related things). Real-time

data collected from distributed sensors can then be easily retrieved, processed, and displayed on Web pages using HTML, CSS and JavaScript.

In contrast to many protocols and standards existing in the IoT, the programming model behind the Web is significantly easier to learn and to use. This is particularly interesting because it enables anyone with basic Web programming skills to build Web sites and applications, not only around multimedia content, but also with real-time data from the physical world.

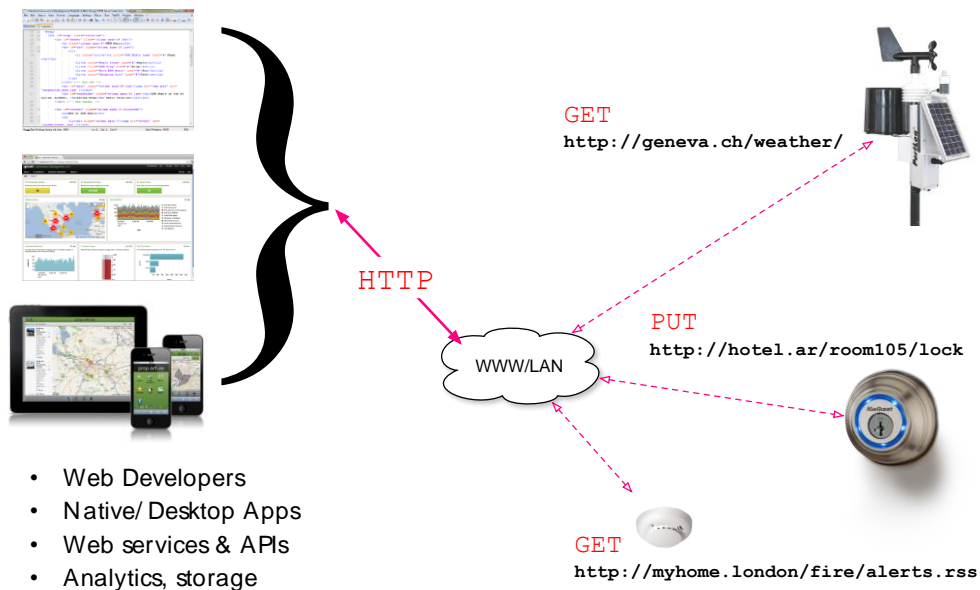


Figure 5 – The Web of Things allows developers and applications to exchange data with any device using standard HTTP requests, regardless of how the device is connected.

Although the Web of Things emphasizes the usage of Web standards to exchange data between devices, it does not imply anything about how the devices should be physically connected to each other. In other words, devices can (but don't have to) be openly connected to the Web and publicly accessible by anyone just like Web sites. The Web of Things can work equally well in local networks (e.g., the Intranet of your company or your home network).

In some cases, it makes sense for Things to have a public URL and be openly accessible over the Web. For example, traffic or pollution sensors in a city operated by the public authorities. In this case, devices could also be crawled and indexed by search engines like any other Web page and allow users to literally Google the physical world, or bookmark the URL of

a smart object and share it with friends. Web-connected objects can also become active and participate to the Web just other users, by publishing their own blogs or talk to each other using the APIs of services such as Twitter.

Using services such as IFTTT³, users can create small logical rules that mix real-world devices, such as sensors in their home with virtual services in the cloud for example an SMS gateway or a weather forecast service. Such applications are called *physical mashups*. You can learn the principles and tools that will allow you to create physical mashups on top of your Things in my book, [Building the Web of Things](#).

To really understand why the Web of Things represents an interesting next stage in the evolution of the IoT, we first need to look at the history of this field until today. Why did the idea of connected devices emerge in the first place? And if the vision of a global network of connected device is so promising, how come it doesn't exist yet? These are the questions we attempt to answer in this book.

³ <https://ifttt.com/>