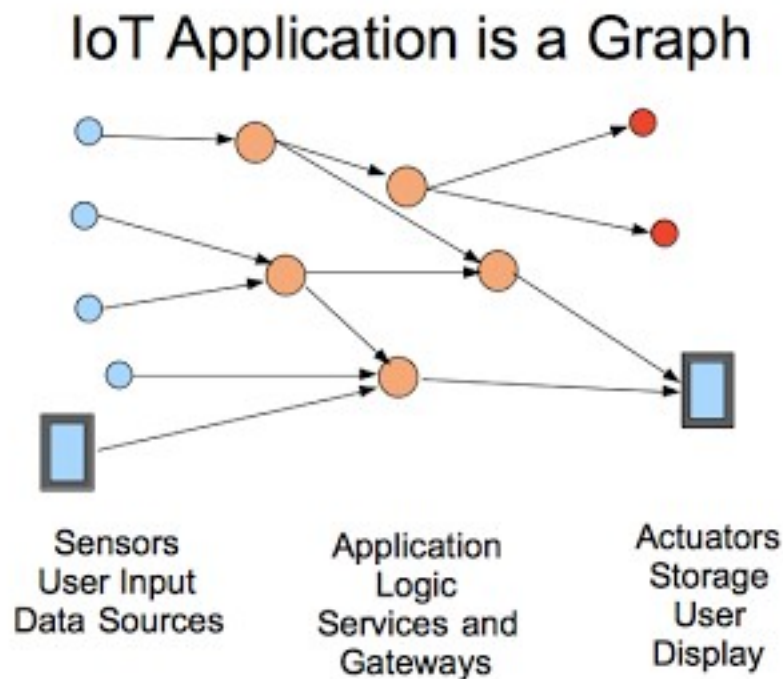


The Internet of Things needs an Open Infrastructure

Michael J Koster

December 30th, 2012

An Internet of Things application may be best represented by a graph, with sensors and data sources connected to filters, triggers, rule evaluators, and other processes, with these outputs connected to actuators, storage, visualizations, etc. The connections are made according to user intentions and are customized to the user's context.



It's often observed that one thing holding the Internet of Things back is the difficulty of connecting things together. Some are trying to solve this at the platform level, by providing connectivity within the platform e.g. SmartThings, and some are working to connect data between vendor's offerings, e.g. IFTTT.

A general solution to the problem would be a common connection and communication mechanism between the various sensors, actuators, gateways, services, and apps. It's generally realized that some degree of "open-ness" is needed for the systems to be connected. Various claims of open-ness are being made for different IoT systems and in different contexts. In this post I would like to discuss the meaning and impact of open participation in various aspects of Internet of Things technology and processes.

First, let me describe the meaning of some common concepts as I understand them for the purpose of this discussion:

What is a Platform?

A platform is a common set of resources used by developers to build and deploy applications. There are other types of platforms, but for this discussion there is a common set of resources used by IoT developers on which to build IoT applications.

For the common IoT deployment model of sensors, data sources, and actuators connected to services, the platform could consist of sensor nets, gateways, service enablement, APIs, servers, clients, application libraries, smartphone UX enablement, and other common resources. The platform would allow application developers to build or add sensors as needed and develop applications that can be hosted on the sensors, gateways, services, and smartphones. A minimum of customization of the platform should be necessary to support new applications.

Platforms allow developers to share resources and patterns, thus minimizing the new code needed to create new applications. Platforms are often specific to application domains; there may be many different platforms across the IoT just as there are many platforms on the WWW. A platform is often defined by a programming language or an operating system, e.g. "the Windows platform" or "the Java Platform". Applications built on a particular platform are often *interoperable* with other applications on the same platform, and *portable* from one instance or implementation of the platform to another.

Examples of platforms that are built on the WWW are WS, .NET, AJAX

We could say that the WWW is itself an application built on IP, where IP is a platform. Other IP applications include ftp, smtp, POP, IMAP.

What is Infrastructure?

Infrastructure consists of common resources used for deployment of platforms.

Physical infrastructure includes the copper, fiber, silicon, and wireless switching fabric of the internet.

DNS is an infrastructure service for the Internet.

Anytime some layer in the stack becomes ubiquitous and is used to build platforms on, it could be considered infrastructure. IP could be considered infrastructure on which the WWW runs.

The IoT infrastructure consists of wireless networks, the Internet, routers and switches, and in many ways the basic protocols used. For example, HTTP and CoAP enable IoT platforms to be built using these web protocols as APIs to the common internet infrastructure.

What is an Ecosystem?

An ecosystem is a system of systems that all share resources and protocols, and interact and influence the outcome of each other. The WWW can be considered an ecosystem. The Internet and supporting protocols HTTP, etc. are it's infrastructure. There are a number of platforms that all interoperate, more or less, within the WWW as an ecosystem.

In a healthy ecosystem, it's natural for platforms and protocols to come and go. There is a natural interdependence that is balanced against competition. The concept of "all boats must rise" is important to an ecosystem.

In this context, it's difficult for an ecosystem which has multiple competing platforms to be contained within a single company or institution. The sustainable ecosystem seems to be based on open participation at all levels.

What is Open?

There may be many definitions of "open", but for this purpose it should refer to an open avenue of participation. By this I mean that any capable participant is allowed to participate on an equitable basis. Participation can occur at many levels in an Internet based system; here are a few example categories:

Open participation for Users

Participants at this level trade for goods and services that benefit them in their life or business. Open participation for users has to do with access to goods and services at fair prices with fair policies, through accessible channels and outlets, and enabling user participation across languages, etc. without any obligation or limitation on the part of the user as to their fair use of the goods or services, including sharing what they own and control with others.

Internet of Things users will participate in creating their own applications and managing their own information in ways not previously possible. The role of user as composer of IoT application graphs gives the user agency in customizing their own experience. At the same time, it requires a new level of compose-ability in services and data sources. It will also drive new solutions to today's problems of password based security and user control of information.

Open platform for Developers

Participants at this level add value to goods and services by creating new offerings based customizing, programming, combining, or otherwise building on top of the offering as a platform. Open participation here requires equitable access to specifications and design information without requiring intellectual property licenses, regardless of institutional affiliation or lack thereof. It should be possible for a developer to create a new application on an open platform without encountering barriers to entry.

Internet of Things application developers will need to provide high level compose-able services and data sources, in a way that users can easily manage. There will need to be standard, frictionless, transparent mechanisms to pass user authentication and provenance along with commands and data, built into platforms. This will require a level of cooperation between developers and standards bodies throughout the process.

Providers (Service Provider, Hardware Manufacturer)

Open access for service providers enables multiple sources. Open participation at this level requires open access to specifications and access to the standards process, without any licensing restrictions, and regardless of institutional affiliation or lack thereof. It should be possible for a start-up company to enter the business as an open platform service provider or manufacturer without barriers.

Standard infrastructure services, APIs, and platforms will enable a service provider business model, where someone with domain expertise and an idea to solve a problem can easily build a new service business focused on solving the domain problem. Open source reference implementations of infrastructure components will enable IaaS and PaaS providers to support rapid prototype deployment of new platforms and services.

Standards Body

Standards bodies are responsible for the common specifications of a platform or system of platforms. They are usually composed of representatives from corporations and public institutions, and experts in the field. An open standards process is one which allows equitable participation of any qualified person, without any licensing restrictions, regardless of institutional affiliation or lack thereof.

There are currently a number of different standards processes being used to develop IoT standards, each resulting in subtly different qualities of open participation at the different levels. Here are a few examples of broad categories.

Types of Standards

Examples of standards process include de facto, Industry SIG, and the Internet standards process (IETF).

An example of an industry de facto standard is Microsoft Windows. In this case, avenues for participation are limited to a single vendor provider and single corporation develops the specification, i.e. only one vendor provides the Windows platform. Participation by software application programmers is most open, with practically no barriers. User participation is limited by a restrictive end user license.

Many current standards are based on Industry Special Interest Groups (SIG). A common SIG purpose is to share and cross-license Intellectual Property between corporate and institutional members. The specification is often not publicly available, but can often be obtained by joining the SIG and executing the license and NDA. An example of this type of SIG is the Weightless TV whitespace M2M standard. Participation as a developer or vendor is limited to SIG members. Participation as a user is usually based on a hardware purchase or in the case of software a binary license.

The Internet standards, for example the Internet Engineering Task Force (IETF) are based on open participation and industry consensus. Avenues for participation are mostly open to all, at all levels. Tim Berners-Lee didn't have to ask permission from a TCP/IP SIG to develop the WWW application and HTTP.

In Europe, there is a broad consortium of Industry, government, and educational institutions working with standards bodies to develop a comprehensive framework for the IoT. This is an open process that allows many avenues of participation. There isn't enough space here to break it down, but there are some excellent participatory models being used.

What's the license?

When it comes to answering the question "How open is it?", the final word is the license. Any formal work these days, whether software, hardware, or process, e.g. a standard or benchmark, comes with a license. It's the details and requirements of the license that ultimately determine how open your avenues of participation are. It's worth noting that the term "open" is used by different providers to describe a wide range of license restrictions, and it's very important to understand exactly what the license requires and how it limits or enhances your value in participating.

Examples of licenses include the Microsoft Windows EULA, the GNU General Public License (GPL), The Apache 2.0 License, The Facebook Terms of Service agreement, and basically every long document you need to scroll through and check "agree" on before participating in a service or downloading software or activating a hardware.

Open Source software

Open Source software is generally recognized as software which is available in human-readable and maintainable source code form, to any developer that follow the provisions of the license. Open Source software has both copyright and license. The concept of "copyleft" is sometimes used to describe open source license terms. Some open source licences require derivative work to be published with the license intact, e.g. the GPL. Other licenses do not restrict the use of the code, and allow a developer to produce closed source derivative works.

Sometimes source code is published for informational purposes, for example to write interface code to an API, but is not available under an open source license, that is the work may not be reproduced or used for derivatives. This is not open source.

It's generally agreed that infrastructure services and platforms benefit from being Open Source, where multiple contributors are involved in revisions and bug fixes. This provides a level of transparency that has been proven to result in very robust code.

The infrastructure of the Internet, that is the DNS servers, web servers, programming languages, and client components, is built from Open Source software. Since the Internet is an ecosystem, or system of systems, there are compatible closed license source alternatives for many of the services.

Open Source Hardware

Lately the notion of Open Source Hardware has become popular. The basic idea is the same as Open Source software, in that the design specifications, schematics, board layout, and firmware are available for anyone "skilled in the art" to reproduce the hardware for any purpose as long as the terms of the license are followed. A good example of this is the Arduino.

Again, it's the license that matters, even if the design specifications are published for reference. Arduino is available under an open hardware license that allows anyone to manufacture and sell the designs. Note that this is not the case with Raspberry Pi. The Pi hardware license is controlled by the Pi foundation, and is only made available on a case-by-case basis to their manufacturing partners.

Open API

An Application Programming Interface allows developers to build applications on a platform or to communicate with other applications and services. The idea of an open API is to open the system at a particular point to allow developers to build onto the system without signing a license

or NDA. This gives the platform provider control over how developers can use the platform, while allowing anyone to develop on, and add value to, the platform.

The API a platform provides becomes essentially an extension to the application programming language, baking the API into application programs developed on the platform. The unfortunate consequence of this is that applications developed on one platform and API can not be used, or sometimes even ported easily, to other platforms that use different a different API.

So very often, an open API is a tool for a platform provider to attract developers and lock them into the platform. Sometimes the lock-in is not intentionally done to exploit the developers, but rather an unintended consequence of simply developing a unique API for a new platform. That is, there is no well known API that exists, no a new one is created.

Open Source vs. Single Source

Another impact of closed development is the creation of a single source infrastructure. For example, a number of the current IoT system deployments include sensors, a gateway, and a cloud service. For the system to perform it's intended function, all components must be functional. If any component fails or if the provider of a component goes out of business, the entire system fails. Many of the vendors have addressed the sensor/actuator part of this equation by providing an open API or open source client code, etc. but still present a single point of failure in the cloud service.

In this sense, then, the opposite of Open Source is Single Source, and the attendant single point of failure. Of course, we need to provide value in unique services that, if open sourced, would allow competitors to offer equivalent services... Where is the business model?

One solution is an open source service enablement layer that allows the creation of a service ecosystem. Vendors can offer different, competing services that are targeted to user groups and use cases. Users can have a choice of differentiated services to perform a particular function in the application graph. The Internet of Things is going to be big enough to not only enable this, but soon to require it.

Impact on innovation cycle

When everything was developed under the closed IP based system of the industrial age, innovation proceeded at a pace regulated by the patent system and the pace of product development cycles. We had to wait to see what the competitive field produced in secret and then released in public to drive each innovation cycle. These cycles often got out of phase, resulting in some spectacular self-indulgent technology spirals like the iAPX432 or, more recently, the CPU pipeline MIPS blowout ca. 2000 where, out of competitive pressure, all the vendors produced designs that were way too expensive and couldn't be cooled. Upon further

thought, it may be engineer turnover between companies that becomes the rate limiting factor in the closed IP development world...

On the other hand, the opening up of software engineering that came with Open Source and Open coding platforms like Python, PHP, etc, has resulted in sweeping innovation of the software process, with agile development cycles measured in days and continuous release processes. Use of common patterns like RESTful APIs and platform support for common patterns have resulted in most software effort focused on building an implementation of the idea, instead of building scaffolding and plumbing.

Access to one's own inventions

I hold a number of patents in computer architecture. Actually, I don't hold them, I assigned them to my employers who sold them to their buyers etc. and as a result are now owned by completely different people than when I made the inventions. I could expect to be sued by these people if I try to advance my work in these areas for another company, or at the very least my prior patents could be used against me.

What I'm getting at is the current system of IP assignment and value based on secrecy is a limiting factor in my development of any technology I worked on while under the IP agreements imposed by the corporations that employed me. In reality there is a concept of fair use but no one knows exactly what it is without a legal test. All this creates a bias in me to want to work on open source systems where I can maintain access to my own work.

Impact on scalability and single point of failure

Open systems have a distinct advantage in the ability to rapidly scale up and to tolerate single points of failure. IN an open Source ecosystem, if one company fails or is unable or unwilling to continue maintaining a component, there will be another to take over or sometimes provide resources to keep the project going. In a closed system, the single points of supply become the weak links.

Consider the Raspberry Pi. The single supplier situation combined with unexpected demand has resulted in continuous shortages, to the point of being a business risk for anyone considering using the Pi as a component. If the RPi design were licensed as Open Hardware, there may have been an opportunity for an second source supplier to come up to speed and meet the demand. At least the business risk would be mitigated by the possibility of an independent second source.

Business models

One item on every "must have" list for outside capital is something no one else has, preferably with a lock on the Intellectual Property. The requirement for exclusivity should be balanced against the advantages to everyone of a healthy sharing of infrastructure.

It's often assumed that a business must possess some secret in order to be competitive. It's also assumed that Open Source programmers don't get paid. Both these notions should be challenged. Plenty of businesses thrive without needing to keep secrets. Businesses based on Intellectual Property are relatively new. It's worth noting that most of the modern infrastructure around software patents was created during the ascent of the Microsoft corporation, with much help from a certain Mr. Gates Sr. in the legal department.

The Open Internet of Things provides clear opportunities in two areas. The first is in providing new hardware for sensors, actuators, gateways, and data centers plus the required hardware connectivity in sensor nets and public infrastructure. The second is in providing services that add value based on the rich connectivity afforded by an open infrastructure.

These devices and services can and should certainly be unique and don't need to be Open Source. The important thing is to not create lock-in or lock-out situations which result in unfair access or single points of failure. If there is good use of a common set of protocols and infrastructure, the ecosystem should support plenty of redundancy in heterogeneous solutions and open source alternatives.

I believe the best business models will be the ones that take advantage of connectivity and standards, rather than the ones that create vertical silos, walled gardens, and lock-in, regardless of the exclusivity of the Intellectual Property. Besides, recent legal battles illustrate the futility of pretending to own something everyone uses or does.

The Emerging Internet of Things needs an Open Infrastructure

To realize the Internet of Things application graph shown in the introduction, a level of open-ness is needed at all levels of participation. Open standards are needed to enable broad industry participation without any legal obligations. Open platforms are needed to allow users to compose, share, and migrate their application graphs across multiple providers.

Open standards and platforms enable developers to focus on the business model, domain oriented, value added, areas of devices and services, and allow the cost of developing and maintaining shared infrastructure to be shared across the industry. This is facilitated by the adoption of Open Source platforms, both hardware and software, for the emerging Internet of Things infrastructure.