Cloud Technology Partners

How developers can prepare for IoT Standards

Jeff Maynard

February 2017

Agenda

1. Introduction
2. Defining the Internet of Things
3. IoT Past and Present
4. What IoT standards look like now?
5. The problem with standards
6. Navigating the standards soup
7. How do we build good stuff?

Introduction

I work there -->

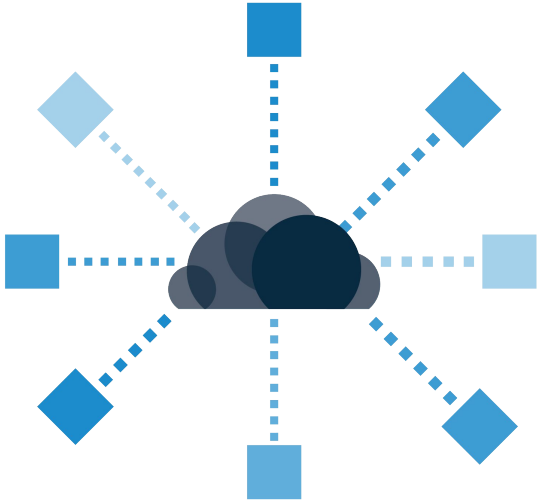
Abridged introduction to me

- **Principal Architect** at Cloud Technology Partners
- **Specialities:** Geospatial-temporal data, forward deployed sensor networks
- **Things I've built:** Nanosatellites, autonomous machine* platforms, connected health devices and platforms, behavioral analytics platforms**
- **Things I like:** Bulleted lists, beer, whisk(e)y, motorcycles, ice climbing, and things that go boom***
- **Things I don't like:** "..., and non-standard use of the oxford comma"

*Fancy abstraction for drone
**Less sketchy way of saying big brother-like platforms
***What engineer doesn't?

The coolest photo of me...

Defining IoT



Defining IoT: It's harder than you think...

Oxford Dictionary

"A proposed development of the Internet in which everyday objects have network connectivity, allowing them to send and receive data."

ETSI[2]

"Machine-to-Machine (M2M) communications is the communication between two or more entities that do not necessarily need any direct human intervention. M2M services intend to automate decision and communication processes."

IEEE[1]

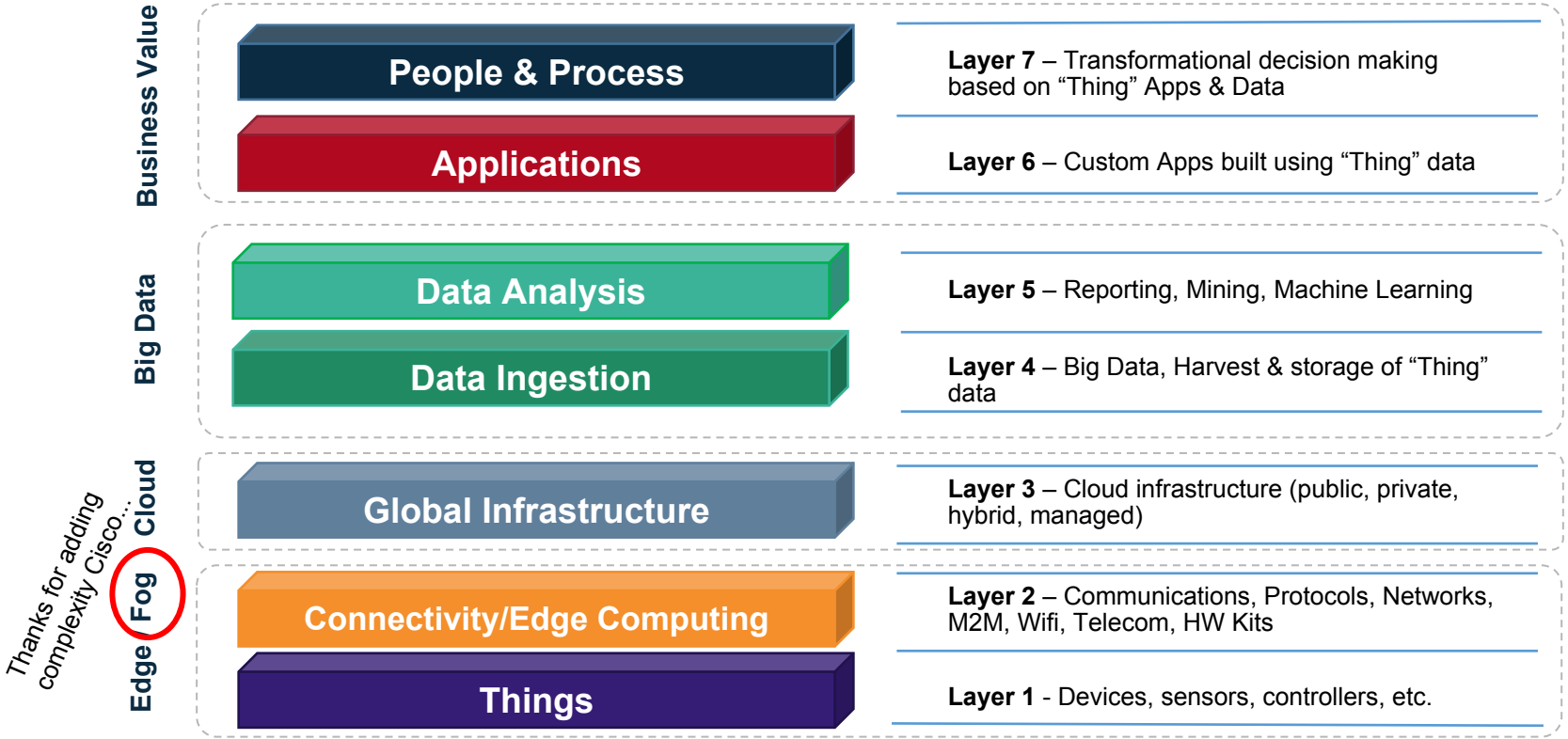
"A network of items-each embedded with sensors- which are connected to the Internet."

ITU[3]

"An ubiquitous network." *BUT ALSO* "A network that is available anywhere, anytime, by anything or anyone."

And the different definitions go on and on…

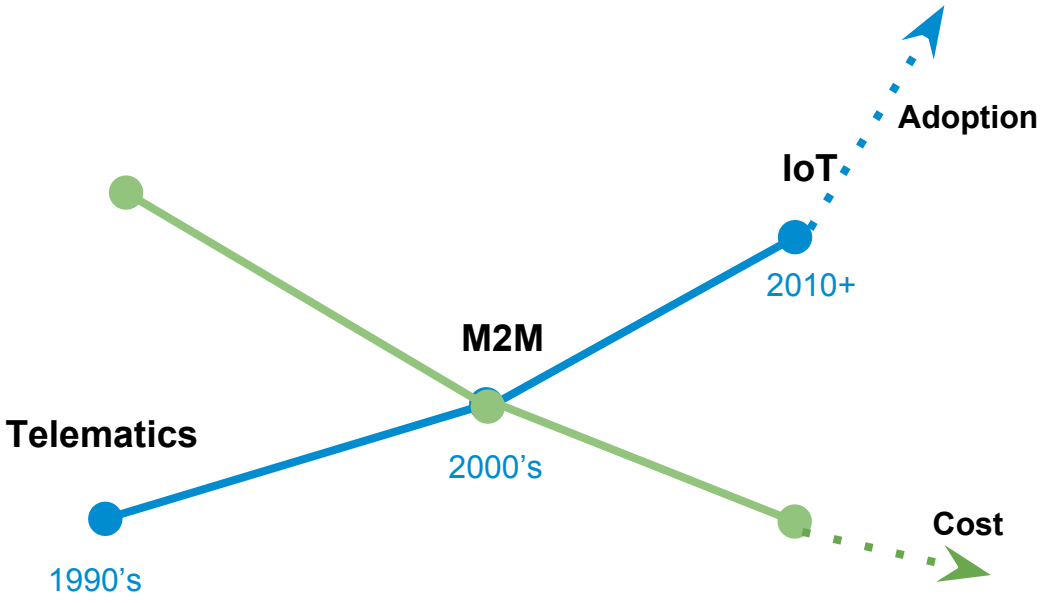
What's involved in an IoT solution?



Credit: CTP's Mike Kavis

IoT Past and Present

Past and Present

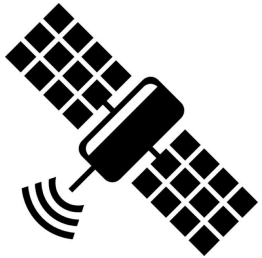


The Internet of Things has been around for awhile, just with different enabling technologies.

- Costs have decreased
- Capabilities have increased
- Adoption has started to dramatically increase

Where did IoT come from?

Vehicle telematics and **connected manufacturing facilities** are the foundational industries that helped pave the way for IoT.



It’s true that there were other industries that had connected solutions in parallel and helped push the technology forward, but they were more isolated, with high costs, and minimal return.

What are we doing with IoT now?

Smart Buildings/Cities

Smart Contracts

Automated Logistics

Connected Car

Wearables

Smart Manufacturing

Retail

Smart Railroad

Smart Ports

What’s driving investment

New Business Models

Innovation and new product development

Compliance

Regulation and legislation support

Operational Efficiency

Process optimization & automation

Customer Satisfaction

Improvements tailored to customer needs

Sustainability

Efficient business practices

IoT Standards Now
Complexity turned up to 11!

I’m not scared of hardware and communications and industry standards, oh my!

Internet of Things Consortium

IoT-A

IBM SIEMENS SAP NXP

IOT-i

ERICSSON HITACHI NEC Telefonica

IEEE IoT

THREAD GROUP

WiFi ALLIANCE

industrial internet CONSORTIUM

Bluetooth SPECIAL INTEREST GROUP

OSGP ALLIANCE

ZigBee Alliance

buildingSMART International Alliance for Interoperability

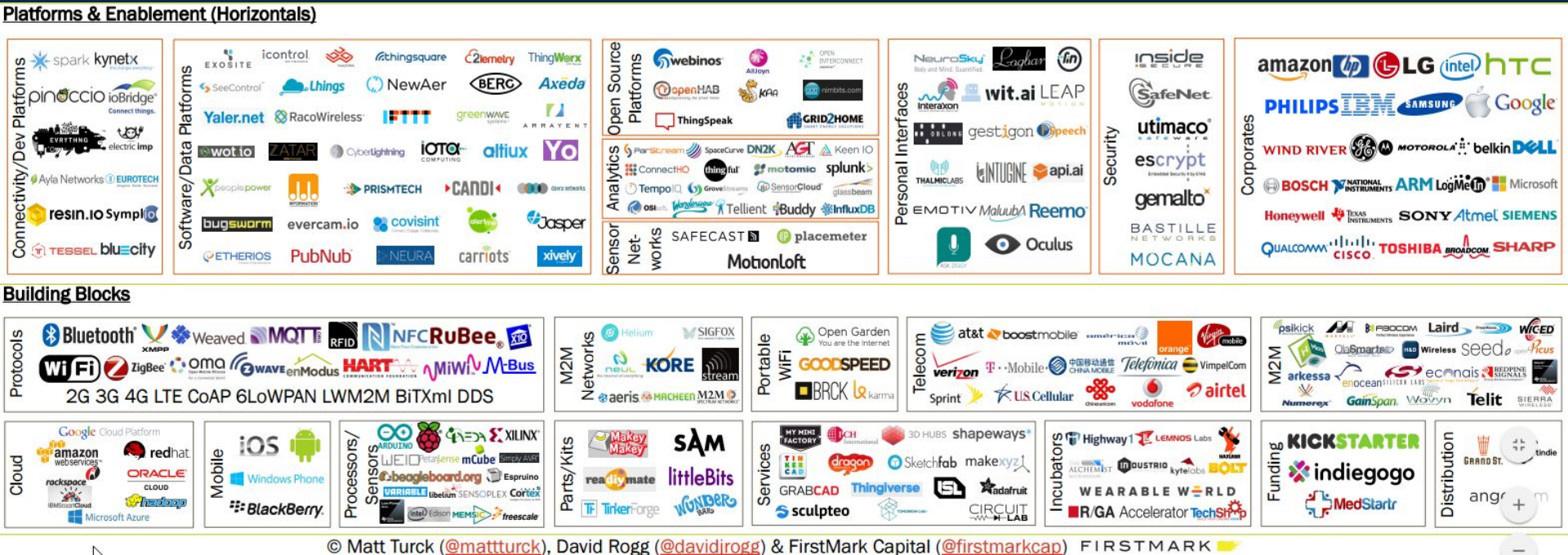
ALLSEEN ALLIANCE

HOMEPLUG ALLIANCE

Well, I'm a little scared...

The IoT ecosystem is vast and complex and each player has their own agenda, making life difficult for us.

Here's just a bit of the ocean we have to navigate.



Just a few standards to start

IEEE 802.11ah (Wi-Fi HaLow)

Wi-Fi will remain at the mainstay of home and corporate office networks. IoT continues to crave low power communications technology for devices that are not primarily hardwired. Enter IEEE 802.11ah, a sub-1GHz version of Wi-Fi with lower power consumption that was amended as a standard in 2016.

Bluetooth Smart/BLE

Bluetooth is a popular protocol for many consumer devices. The next evolution of Bluetooth for IoT is a low power version called Bluetooth Smart (or Low Energy). It provides longer range and support for mesh networking, where each IoT device acts as a repeater node that relays communication from itself and other nodes to a compute/gateway device.

Z-Wave

Z-Wave, another IoT communications standard, is a low-power mesh networking technology licensed by Sigma Designs. Z-Wave operates at 908.42 MHz in the US (868.42 MHz in Europe), and enables a single mesh network to support up to 232 nodes.

A couple more

ZigBee

ZigBee, developed by the ZigBee industry alliance, is another mesh network based on the IEEE 802.15.4 standard that's designed to be used in low-power devices. The technology defined in the ZigBee spec was purpose-built for the IoT to connect lower cost devices.

6LoWPAN

An IPv6-only version of IEEE 802.15.4 mesh networking. It's purpose is to enable low power devices with limited processing capabilities to communicate and play well with others.

ULE

The ULE (Ultra Low Energy) alliance introduced ULE, a low-power version of the Digital Enhanced Cordless Telecommunications cordless telephone network technology that's currently enabling most cordless phones in your home and work.

Thread

Thread is one of the youngest of the standards groups and spun out of Nest. Thread is a wireless-centric standard that covers networking, power conservation, security, and product compatibility. ***A value add:*** Thread-certified devices get issued an IPv6 address, which may be useful as we get closer to that 20 Billion, 30 Billion, whatever Billion number of devices that folks are touting these days.

And don’t forget

- **Hardware** standards
- **Cloud** standards
- **Security** standards
- **Industry** standards
- **Data** standards
- **Privacy** standards
- **Regulatory** standards

The list goes on...

The problem with standards

The problem with standards

The biggest challenge is that there are simply **too many competing IoT standards**. Ultimately, the creation of IoT standards with **redundant purposes offers very limited benefits**.

This mire of standards can cause some anxiety until you **start to view them as selectable building plans** to frame your own requirements for building and deploying IoT systems.

Evaluating standards: To use or not to use?

A few questions that you need to ask yourself when evaluating standards

- **What value** does a given standard bring to my IoT application?
- What are the **risks of not using** a specific standard?
- What are the **risks of leveraging a standard that ultimately fails**?
- **Can I have input into the standard**?

How to navigate the standard soup

Open Standard vs. Proprietary

- Open standards provide you more flexibility and options versus utilizing a proprietary standard.

- **Open standards*** *normally* end up being evolved in a proactive and *more* thoughtful manner in hopes to solve pain points that hinder adoption and to simply make the standard better.

- On the other side of the fence, **proprietary standards**** are built in a more reactive way and are designed to be sticky, making it more challenging to move away from down the road. You see this with software all the time and IoT is no different.

*Full disclosure: I have been bias due to my startup experience
**Full disclosure: I'm still biased. So are my colleagues. And many of my customers.
There are a few exceptions such as aircraft networking (an important note).

What problems do these standards solve?

Big Fear

With so many standards today all jockeying for position, many of them aren't going to make it, and **no one wants to bet on the wrong horse.**

No one wants to be the one who chose Betamax or MiniDisk these days.

Do we wait for an answer?

Word of wisdom: When in doubt you might as well pick a standard communications protocol rather than a proprietary one. This holds true for most other standards types.

- The IoT FOMO is too great at this point for companies, we can't pump the brakes and wait for folks to come to a consensus.
- Standards shouldn't lead you, but should be viewed as trail guides and you should examine each and the value it brings for your application.

A few things to consider

1. Some providers' products will become so popular that these companies can define a standard all by themselves. While you might be choosing a "standard" that they create and control, it may very well be the way in which standards evolve in this next phase.

2. Historically the winning standards have been the ones backed or selected by licensed spectrum owners (i.e., Vodafone, Verizon, AT&T, Sprint, etc.) - they control the infrastructure and they can steer folks to their preferred standards across the board (hardware and comms). Keep an eye out for which ones they are backing.

3. In addition, consortiums have rarely yielded the winner - the exercise becomes too academic and the cat herding is too great.

How do we build good stuff?

Start with the right approach - “right” is relative here

1. Identify or understand your customer’s problem
2. Figure out what data is required to solve your customer’s problem
3. Identify the right tools for the job
4. Architect the solution
5. Build the solution
6. Deploy the solution

(7...) Iterate

1

Identify or understand your customer's problem

Why is it a problem?

- Does the business see it as a problem that needs to be solved? (read: is there a market and can the solution make money?)
- In the history of building successful things, technology driving technology has one of the worst track records. I know that money is money*, but I prefer that the solutions I build get used and solve some meaningful issue**.

*Everyone is in sales after all **But I'm still a engineer

2

Figure out what data is required to solve your customer's problem

Interview the business.

Interview IT folks.

Interview OT folks.

Interview the customers of your customers if possible.

The goal here is to derive the type, frequency, volume, etc. of the data required to solve the identified problem. This very much falls into the 'What' category of solutioning.

3

Identify the right tools for the job

What technologies enable you to get the data required in the method required?

From sensors to action: IoT solutions often require you to understand each piece of the puzzle and how they will connect to pieces up and down, left and right, forward and backward.

Standards can play a huge part here and **can help provide that trail guide** to help you down-select technology options **so you have less to evaluate**.

Effort should be made to keep things somewhat modular*, but don't overdo it! In the IoT world, time to value is the critical driver.

*Open Standards vs. Proprietary

4

Architect the solution

Future-proof where possible, but go with your ***educated gut and focus on time to value****.

Be confident and educated in what path you begin to go down, but keep your ears and eyes open so you can learn from other folks' mistakes in technology choice, design, or standards adopted.

If you make too many reactive course corrections too early you will stall your progress.

Key takeaway: Good enough is a good start**.

* I get it already, time to value is important!
**Does not imply incompleteness of necessary requirements is acceptable, just that it doesn't have to be perfect. Good enough for automated locomotives is very different than that for a IoT doorbell.

5

Build the Solution

As you are building the solution talk to your clients often so you can re-prioritize their feature requests and needs. Respectfully challenge them by asking what level of completeness does this need to be right now?

It might seem like I'm trying to sell you on Agile methodology, but in reality I'm not.

Why? Because I have yet to see a customer able to effectively keep up with Agile and do their part. You normally end up with a psuedo-Agile approach*.

It is possible to build IoT solutions in an Agile method, but if you are influencing hardware it will be a challenge**.

*Fear not! Many large corp tech groups are beginning to really embrace the idea, but now the hurdle is Finance

**But not impossible! Engineers built a road-worthy car in an Agile method in just 3 months! YMMV

6

Deploy the Solution

This is super complicated. Far more complicated than many think.

The **type of solution** and the **industry** you are deploying in will dictate the level of complexity and time required for this step.

If there are Federal or State regulations that are applicable to your solution then the deployment effort might take significantly longer than the engineering effort. Being able to point to the number of standards (HW, Comms, Industry, etc.) you have adopted/adhered to and why along with detailed documentation of your development may help expedite this, but no promises.

7

Iterate

Build new features, adopt new technologies, adapt and conform to leading standards, etc.

This step is rather self-explanatory for those who run or work in Agile-ish shops.

A few words about mistakes

Don’t be afraid to make mistakes* while embarking on your IoT journey, we all have and will continue to do so.

To be a bit cliche and soapbox-y:

1. Make sure you can **learn from them**.
2. **Don’t let your mistakes become heroic obstacles** you have to overcome later.
3. Good stuff is built to be at least somewhat **future proof** - don’t build yourself into a corner if you can help it!

*I’ve made a lot...

For those of you who build solutions to solve hard problems I wish you brilliance, wisdom, and clarity.

Thanks for your time.

CLOUD WITH
CONFIDENCE®