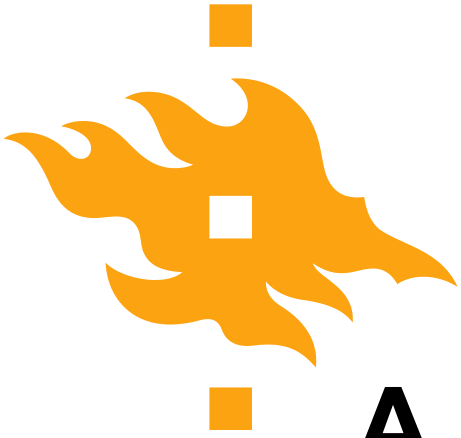
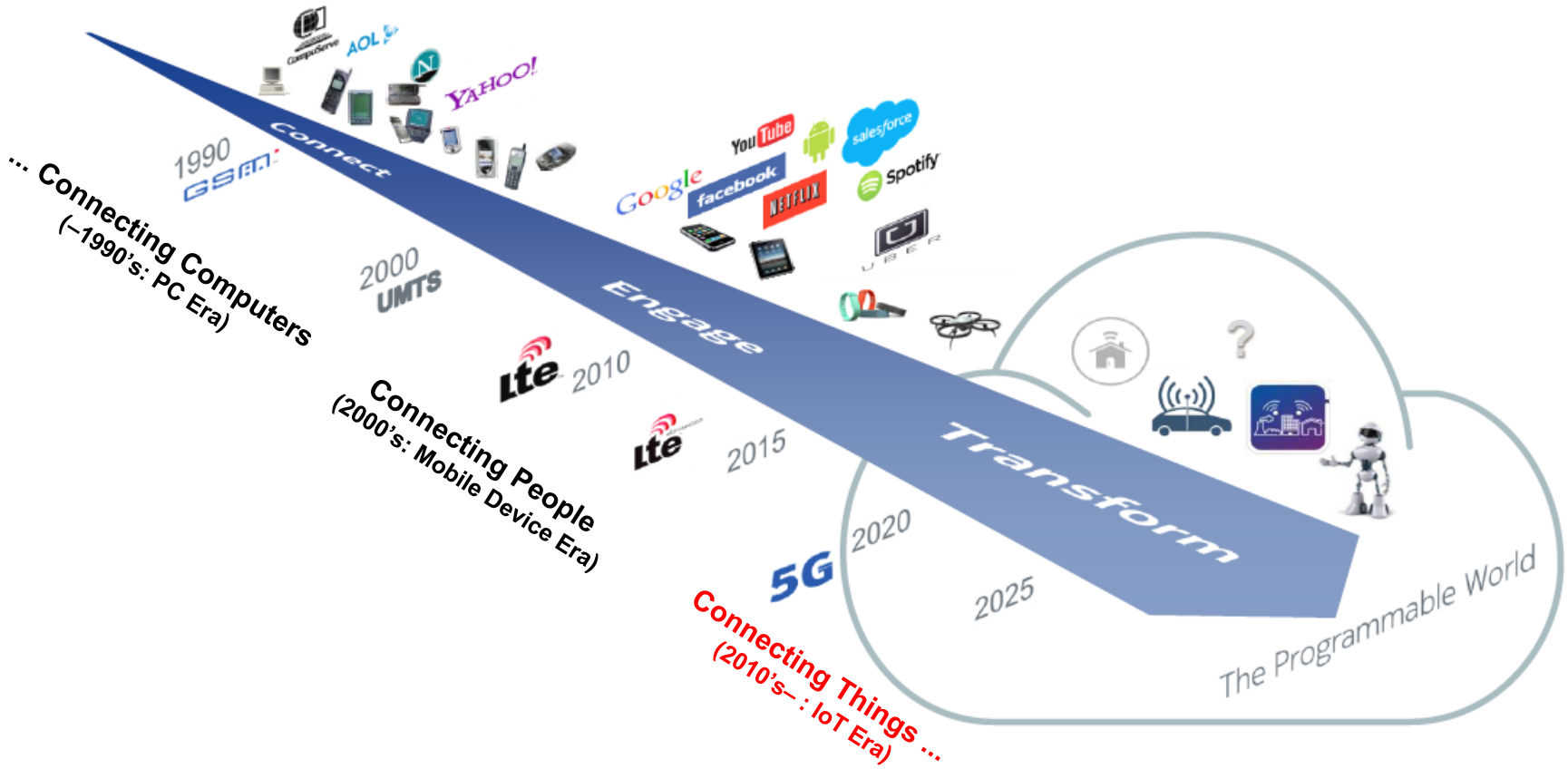


A Research Agenda for the Programmable World: Software Challenges for IoT Era

Tommi Mikkonen
Dept. Computer Science
University of Helsinki, Helsinki. Finland
tommi.mikkonen@helsinki.fi

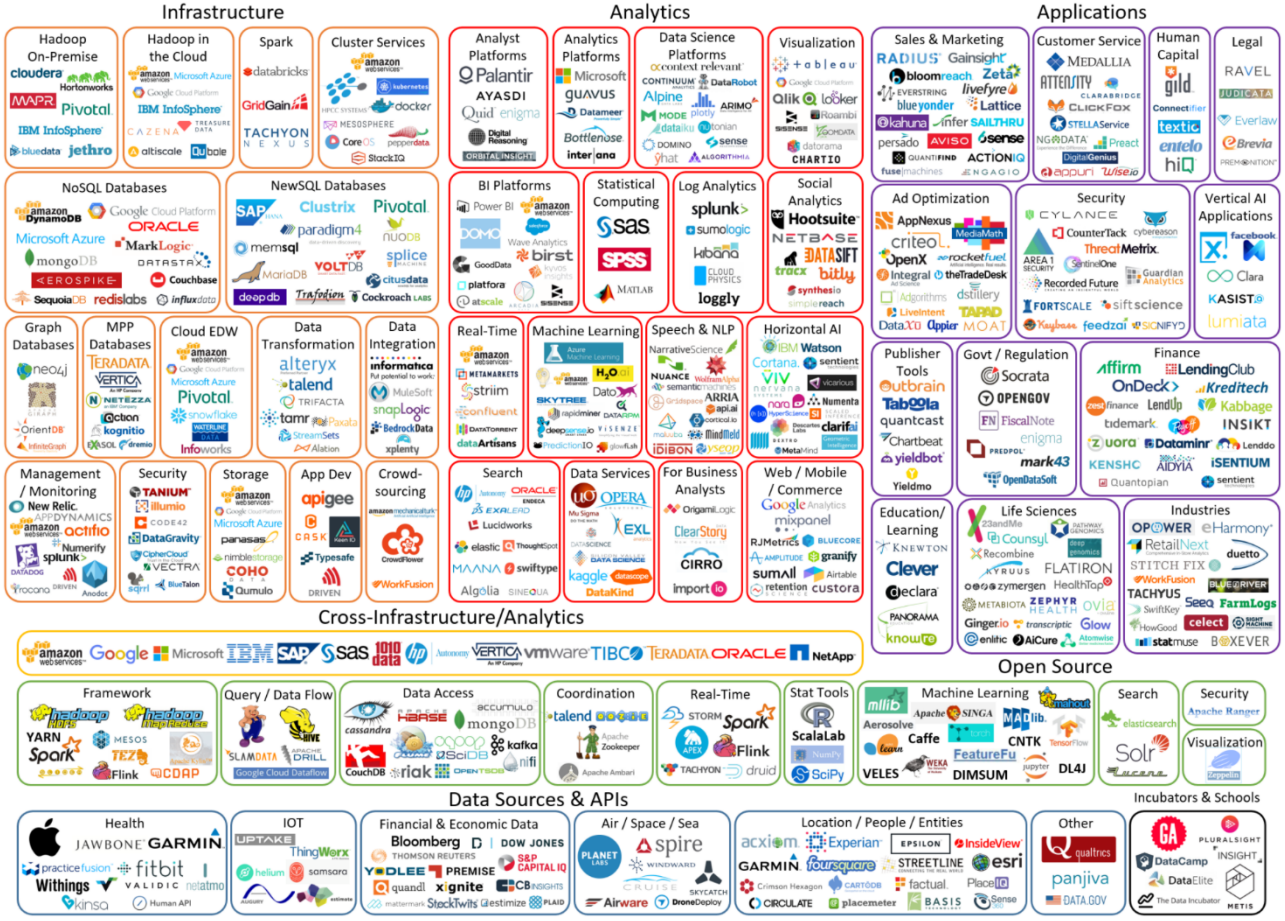
Evolution of the Internet – Moving to the Next Era!



Fundamentally, the Internet of Things is all about *transforming physical objects into digital data product and services.*

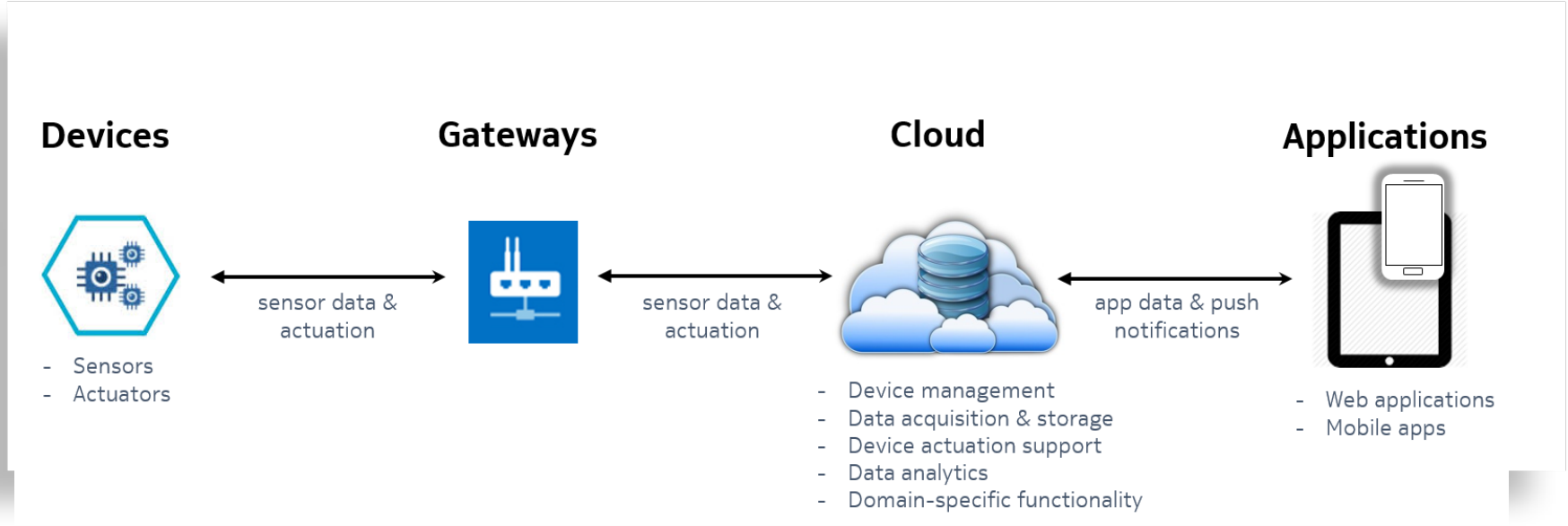
Thing “X” + Internet + Service -> Smart “X”

IoT Technology Landscape – a Cornucopia of Choices

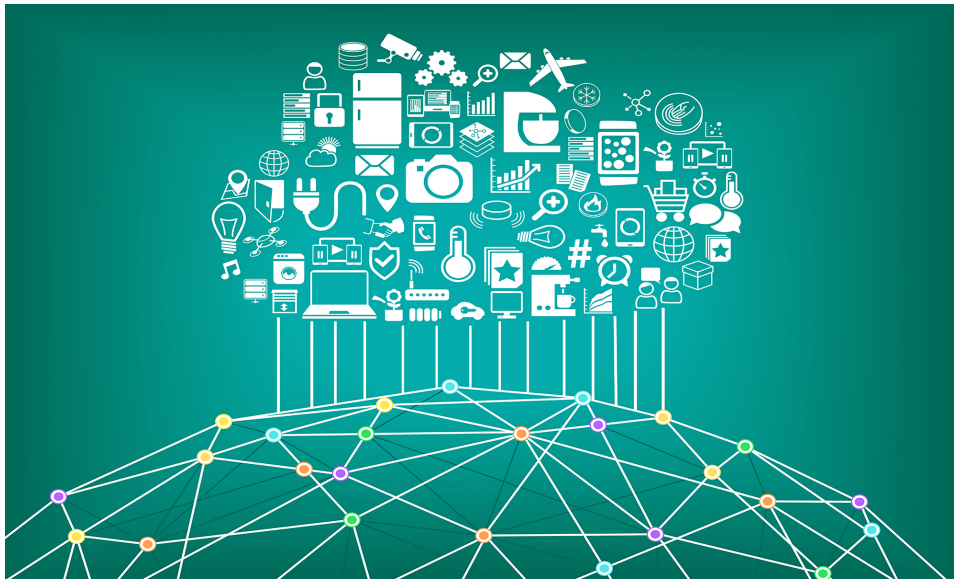


- There are numerous commercial and open source technologies available for nearly all the imaginable areas in IoT system development.
- Today, it is **difficult to win by technical differentiation or platform-level work alone**.
- Ultimately, this will be an ***ecosystem play***, as it simply won't be feasible to have thousands of incompatible everyday things and services.

Still, at the high level, all IoT platforms and solutions are nearly identical



Today, the Majority of Focus in the IoT Area is on Data Acquisition & Data Analytics



DATA COLLECTED IN 00:00:60 SECONDS

694,445 GOOGLE SEARCH QUERIES

168 MILLION EMAILS SENT

13,000+ iPhone Applications Downloaded

320+ NEW TWITTER ACCOUNTS

98,000+ TWEETS

20,000+ NEW POSTS ON TUMBLR.

70+ DOMAINS ARE REGISTERED

1,600+ READS ON Scribd.

60+ NEW VIDEOS

25+ HOURS TOTAL DURATION

1,500+ BLOG POSTS

60+ NEW BLOGS

695,000+ facebook STATUS UPDATES

79,364 WALL POSTS

510,040 COMMENTS

50+ WORDPRESS DOWNLOADS

125+ PLUGIN DOWNLOADS

12,000+ NEW ADS POSTED ON CRAIGSLIST

1700+ FIREFOX DOWNLOADS

Meanwhile, a More Subtle Revolution is Taking Place…

Hardware improvements are enabling dynamic programming capabilities in unprecedented form factors and price points.

This makes it possible to turn everyday objects into connected devices that can be programmed dynamically.

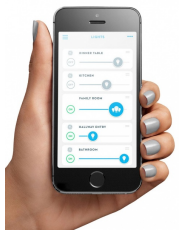
This is truly revolutionary from commercial perspective – the impact will be at least as significant as that of the emergence of virtual machines in mobile phones 20 years ago!

Where Will This Lead Us? Programmable World

Literally:

“Every *thing* in my realm programmable remotely”

Near-Term: A Lot of Incompatible Systems and APIs



There are 47 apps for that…

THREAD GROUP

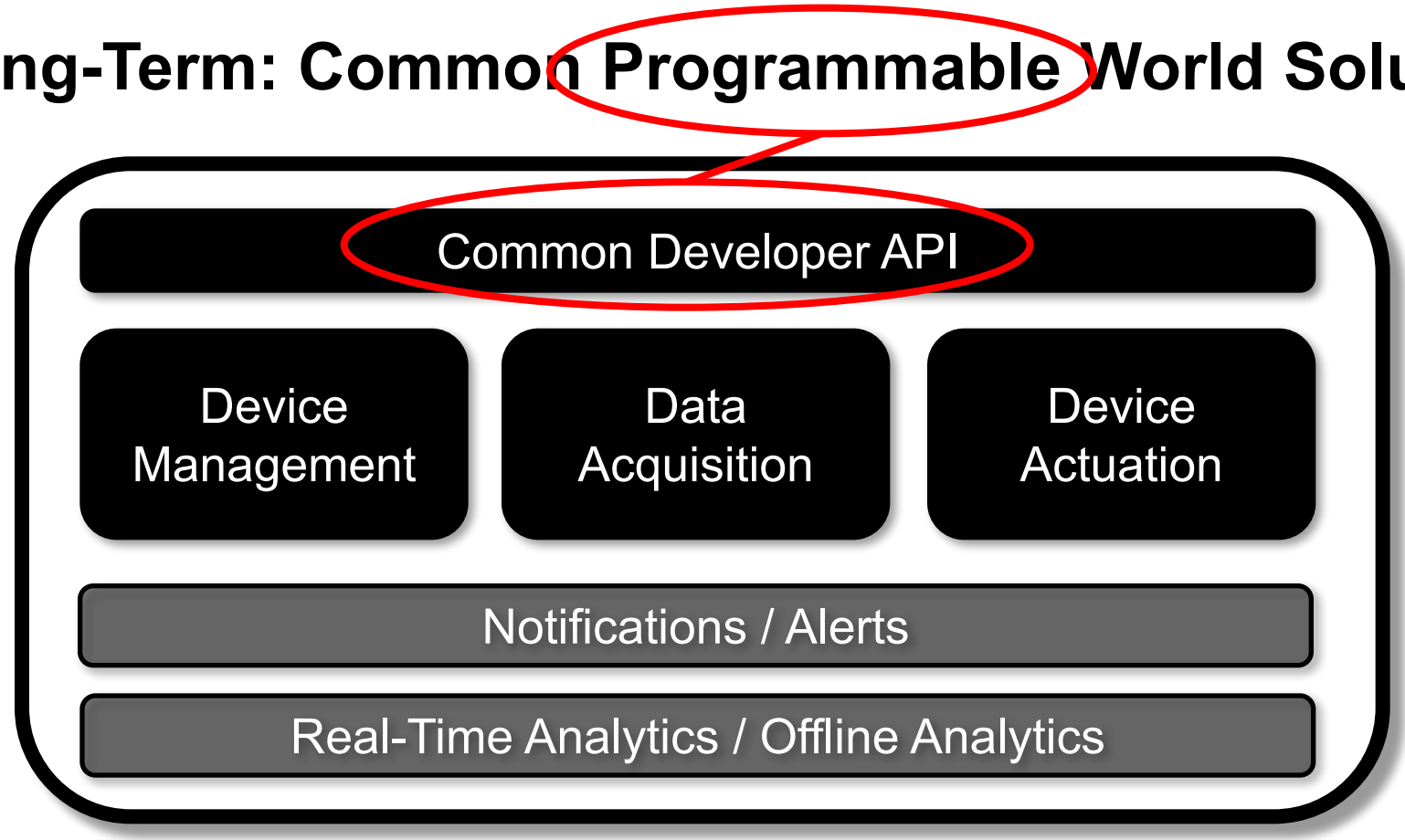
IPSO Alliance

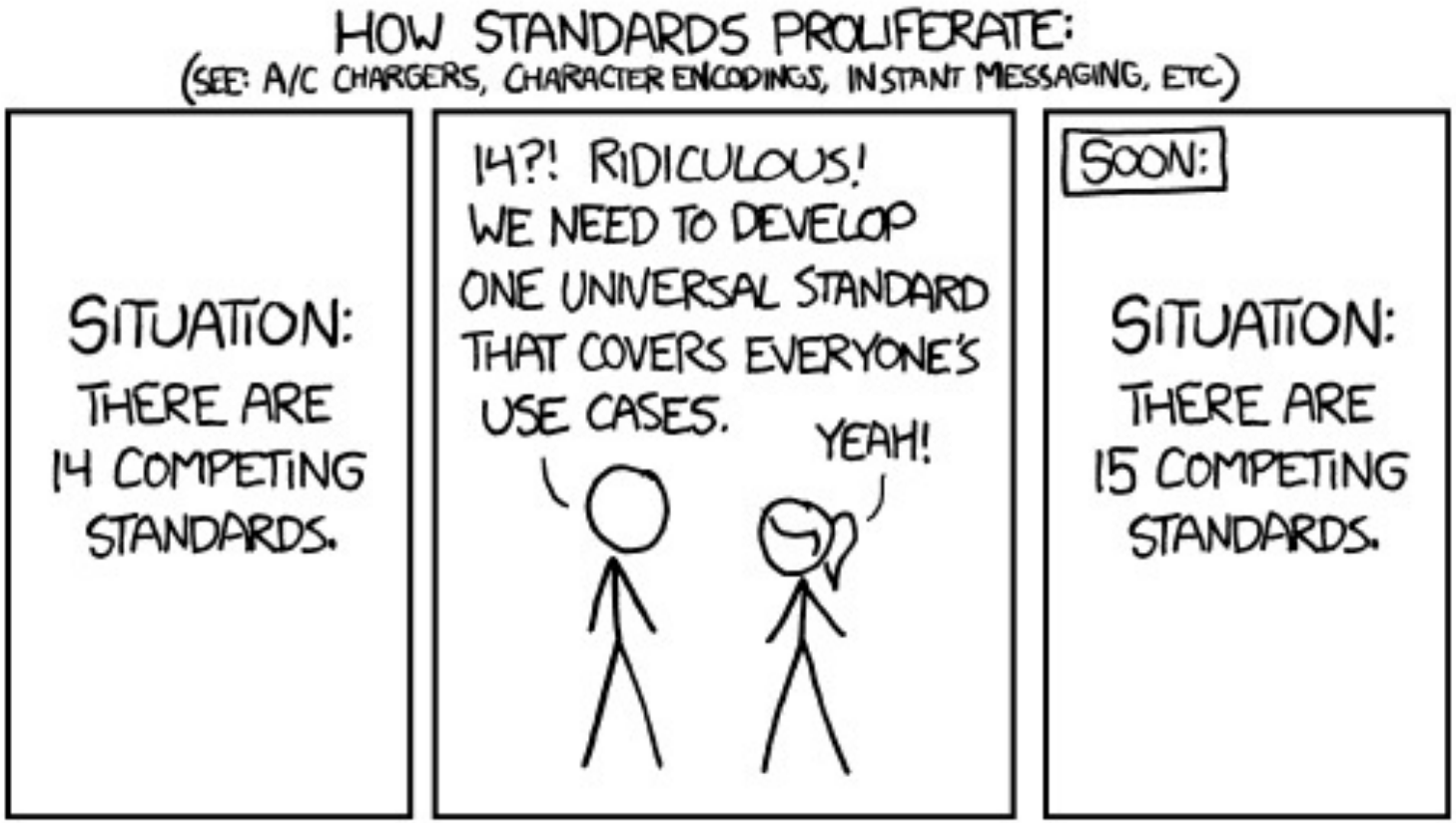
ALLSEEN ALLIANCE

industrial internet CONSORTIUM

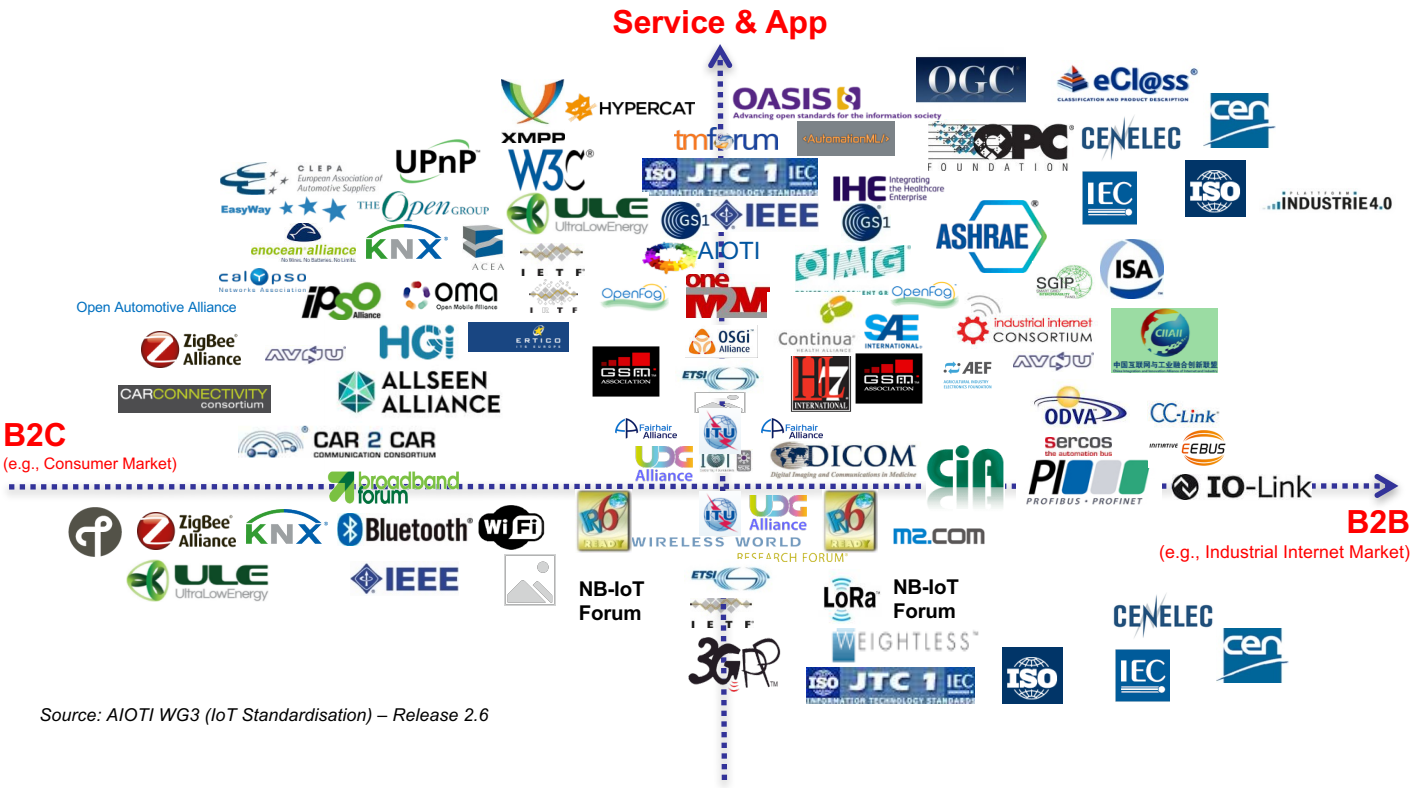
OPEN CONNECTIVITY FOUNDATION™

Long-Term: Common Programmable World Solution




HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION:
THERE ARE
14 COMPETING
STANDARDS.
14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.
YEAH!
SOON:
SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

Currently: Very Complex IoT Standards Landscape



Source: AIOTI WG3 (IoT Standardisation) – Release 2.6

Desired Basic Programmable World API Functionality

1) Effortless discovery and management of things, based on various metadata such as associated sensors, actuators and their current topology and location.
2) Acquisition and observation the state of the physical world through tapping into the sensory values of the things, either by (1) reading the latest known values, (2) reading historical values, or by (3) creating listeners to receive streams of latest sensor readings.
3) Responding to the external stimuli and imposing changes on the physical world through actuators, either (1) immediately or (2) by defining rules that will trigger state changes when given conditions are met.

All of the above should be doable without complex setup issues or a lot of “boilerplate” code.

- without having to specify protocols or ports,
- without having to know the physical location of things,
- without having to know the specific type or manufacturer of device, etc.

Complex IoT Landscape – A Lot of

Because of the large number of verticals, it can still be debated whether there will ever be a common API set covering all types of domains.

Programmable World – Software Engineering Challenges

- How to discover, manage and visualize large, complex, dynamic topologies of IoT devices?
- How to dynamically program IoT systems that consist of hundreds or thousands or millions of devices?
  - Mindset shift: from computers as “pets” to “cattle”, or “swarms” of devices.
- IoT systems are distributed systems, with intermittent, potentially unreliable connectivity => How to reduce the programming overhead (boilerplate code) that arises from having to prepare for various kinds of error conditions?
- How to flexibly migrate computation and data between the cloud and the edge (devices, gateways) in order to balance computation, latency and power consumption requirements?
- Ultimately: How to establish a common programmable world API set that would work across devices and systems from various different manufacturers?

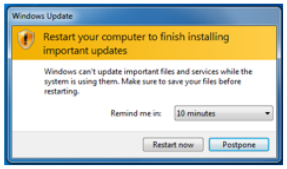
Research Topic Areas Taking Us to Programmable World API

Beyond Data Analytics

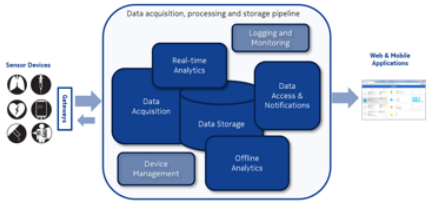
A. From Rebootables to Systems that Never Sleep
B. Thing Management
C. Liquid Multi-Device Software
D. Edge Computing and Local Connectivity
E. Rethinking the Scale of Software Development & Deployment
F. Development Stack Considerations
G. Security
H. Tools

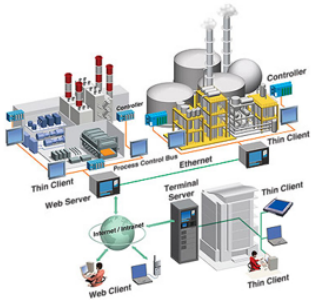
From Rebootables to Systems that Never Sleep

"Rebootables"



"Systems that Never Sleep"





- The vast majority of applications today are written for rebootables!
- IoT and cloud backend development require a different mindset!

Thing Management & Thing Management Systems

TMS environments will make it possible to remotely

- manage,
- monitor and
- visualize complex topologies of devices,

Providing notifications and alerts on abnormal system conditions

Enabling remote reconfiguration and management of the overall system.

The foundation and enabler for remote programming of IoT devices

- Enabler to discover and reach large numbers of devices that are under management in the system.

Liquid Multi-Device Software

- By *liquid software*, we refer to a multi-device software experience that can seamlessly "flow" from one device to another.

- Virtualized but personal computing experience that is independent of any particular device, OS platform, or vendor ecosystem.

- Liquid software allows the users to seamlessly roam and continue their activities on any available device or any "piece of glass".

Corning, Inc., A Day Made of Glass 2, 2012; http://www.youtube.com/watch?v=jZkHpNnXLB0

Edge Computing and Local Connectivity

Eight Fallacies of Distributed Computing

(L. Peter Deutsch, Sun Microsystems, 1994)

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology does not change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

~75% of the effort in developing distributed systems is spent coping with these issues [Laprie 1995]

For details, read: http://www.rgoarchitects.com/Files/fallacies.pdf

Rethinking the Scale of Software Development & Deployment: From Pets to Cattle

Developing for a ”pet” device (e.g. a toy car) is not that different from programming an individual computer

The first big "a-ha" moment: Develop a system that consists of fifty or more devices.

You can no longer …

- hook up a physical cable to each and every device,
- upload programs manually,
- tweak settings
- test application behavior individually on each device.

All the stages and device management to be automated, and run in parallel for hundreds or thousands of devices.

Development Stack Considerations

Various setups are possible (and with increasing amount of memory also viable)

- Hardware-only solutions, RTOS (e.g. FreeRTOS)
- Small-footprint language-specific virtual machines (e.g. Tessel JavaScript device)
- Generic operating system (e.g. Linux)
- App specific operating systems (e.g. Android)
- Web server centric systems (e.g. Node.js)
- Container-based systems (e.g. Docker + Linux)

Security: The Final Frontier

Many design parameters that drive the development to different directions

- Price (development, maintenance, bill-of-material)
- Functions
- Compatibility
- Innovation capabilities
- Market share and demand
- …
- …
- …
- (many more features)

Security: The Final Frontier

Many design parameters that drive the development to different directions

- Price (development, maintenance, bill-of-material)
- Functions
- Compatibility
- Innovation capabilities
- Market share and demand
- …
- …
- …
- (many more features)
- Security (which is eventually turned off when there is even slightest problem)

Security: The Final Frontier

1990: Every thing in your home
has a clock & it is blinking 12:00

2020: Every thing in your home has
an IP address & the password is "admin"

Tools to Match Development Practices and IoT Needs

Staging systems to as close as possible to the final system calls for creating digital replicas in virtual reality (“digital twins”)

- DevOps, constant deployment & delivery, etc.

Various needs

- Training AI systems (e.g. https://goo.gl/2ocVLq)
- Designing distributed algorithms
- Running simulations
- Rapid feedback loop

Summary and Key Takeaways

- There is more to IoT than just big data acquisition, analytics and visualization.
- Hardware advances will make things around us connected and programmable, thus leading us to a *Programmable World*.
- IoT development is different from PC, mobile or web application development.
- A generic end-to-end IoT architecture has already emerged, but today's IoT development APIs are still rather vendor- or hardware-specific.
- There is a need for a common Programmable World API set that would support device discovery, device management, data acquisition and device actuation in a universal, vendor-independent fashion.
- There are very interesting research topics in this area!

Thank you! Gracias! Kiitos!

Any questions?