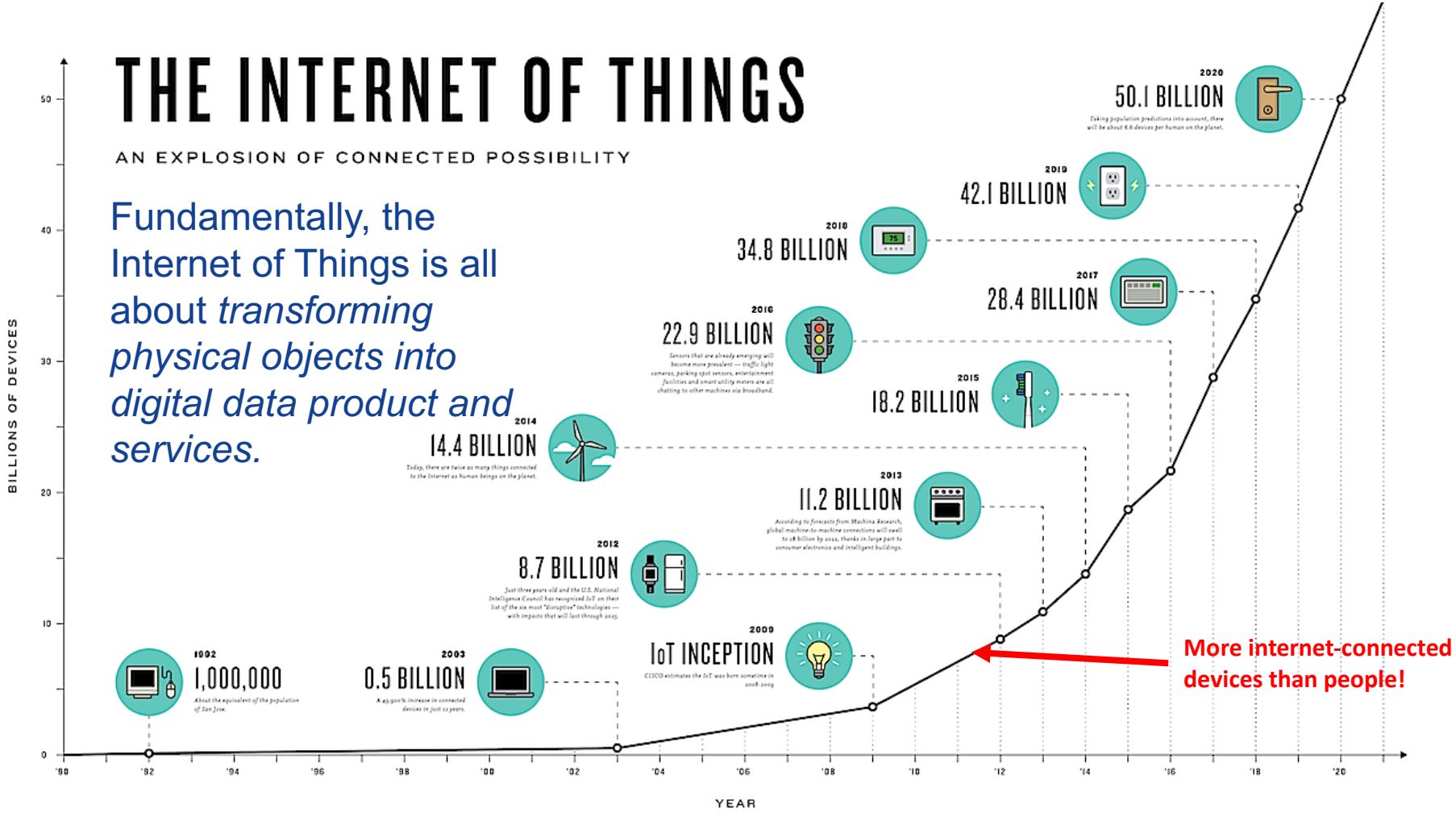
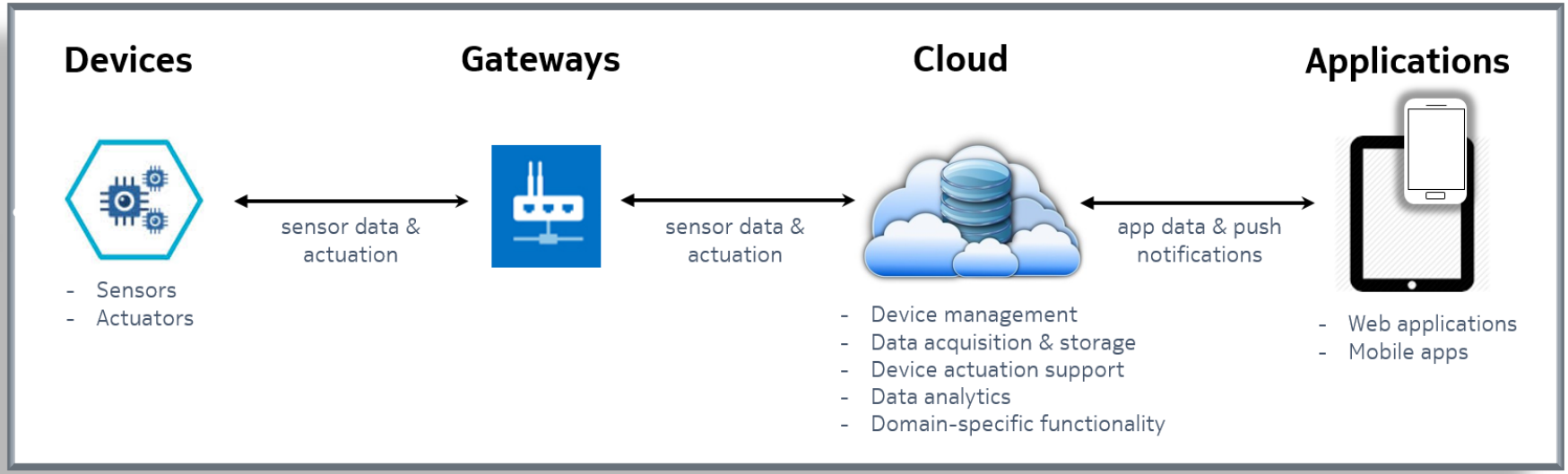
Järjestelmien yhteensopivuus ja asioiden internet

Tommi Mikkonen
University of Helsinki
Finland


THE INTERNET OF THINGS
AN EXPLOSION OF CONNECTED POSSIBILITY
Fundamentally, the Internet of Things is all about *transforming physical objects into digital data product and services.*
1992
1,000,000
About the equivalent of the population of San Jose.
2003
0.5 BILLION
A 49,900% increase in connected devices in just 11 years.
2009
IoT INCEPTION
CISCO estimates the IoT was born sometime in 2008-2009.
2010
34.8 BILLION
2012
8.7 BILLION
Just three years old and the U.S. National Intelligence Council has recognized IoT on their list of the six most "disruptive" technologies — with impacts that will last through 2025.
2013
11.2 BILLION
2014
14.4 BILLION
Today, there are twice as many things connected to the Internet as human beings on the planet.
2015
18.2 BILLION
2016
22.9 BILLION
2017
28.4 BILLION
2019
42.1 BILLION
2020
50.1 BILLION
Taking population predictions into account, there will be about 6.6 devices per human on the planet.
BILLIONS OF DEVICES
0
10
20
30
40
50
'90
'92
'94
'96
'98
'00
'02
'04
'06
'08
'10
'12
'14
'16
'18
'20
YEAR
More internet-connected devices than people!

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



The Four Corners of IoT Development

IoT Devices
- Embedded Software Development
- Various levels of software stacks: no OS, RTOS, language VM, mobile OS, full OS

IoT Apps
- Mobile Software Development (Android, iOS)
- Web Application Development (HTML5 + web frameworks)

IoT Gateways
- Embedded and/or Mobile Software Development
- Consumer IoT systems: Smartphone as the gateway
- Professional IoT systems: Dedicated gateways

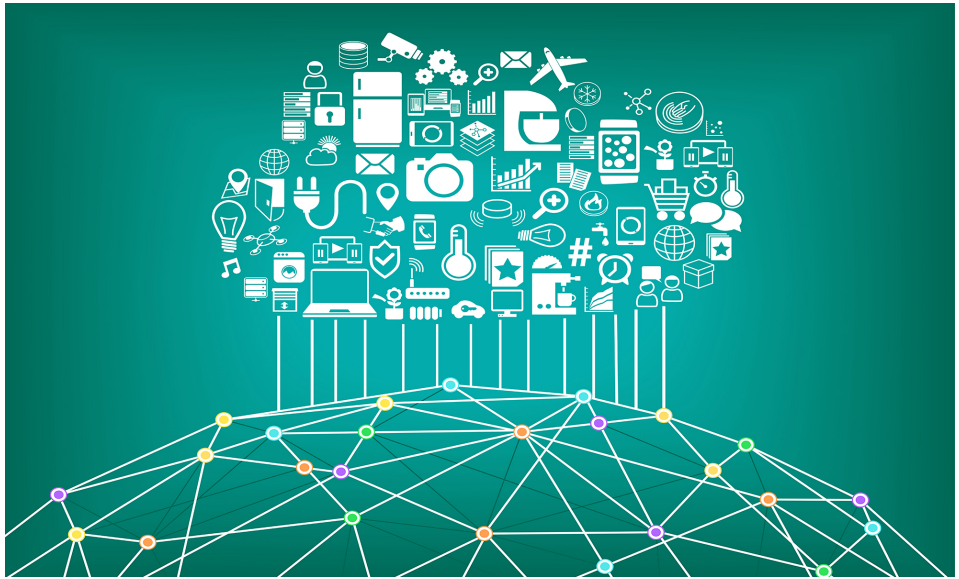
IoT Cloud and Analytics
- Cloud Backend Software Development
- Open source components available for all key areas, including data acquisition, storage & access, data analytics, device & identity management, and runtime & deployment support
- Various deployment options ranging from self-hosted to public clouds and services

Key observations from our industry projects:

The development of end-to-end IoT systems requires skills in nearly all areas of modern software development.

It is very difficult to find developers who can master all the necessary technologies.

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



The focus on these topics is not surprising, given the massive business potential arising from the ability to provide new insights into people's behavior and other real-world phenomena.

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.

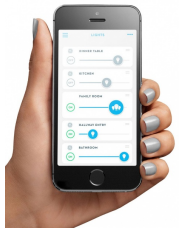
This is revolutionary from commercial perspective. Actual impact will be at least as significant as that of the emergence of virtual machines in mobile phones 20 years ago!

Where Will This Lead Us? Programmable World

Literally:

"Every *thing* in my realm programmable remotely"

Near-Term: A Lot of Incompatible Systems and APIs



THREAD GROUP

ipso Alliance

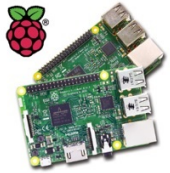
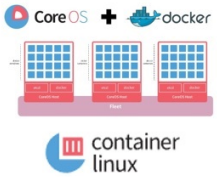
ALLSEEN ALLIANCE

industrial internet CONSORTIUM

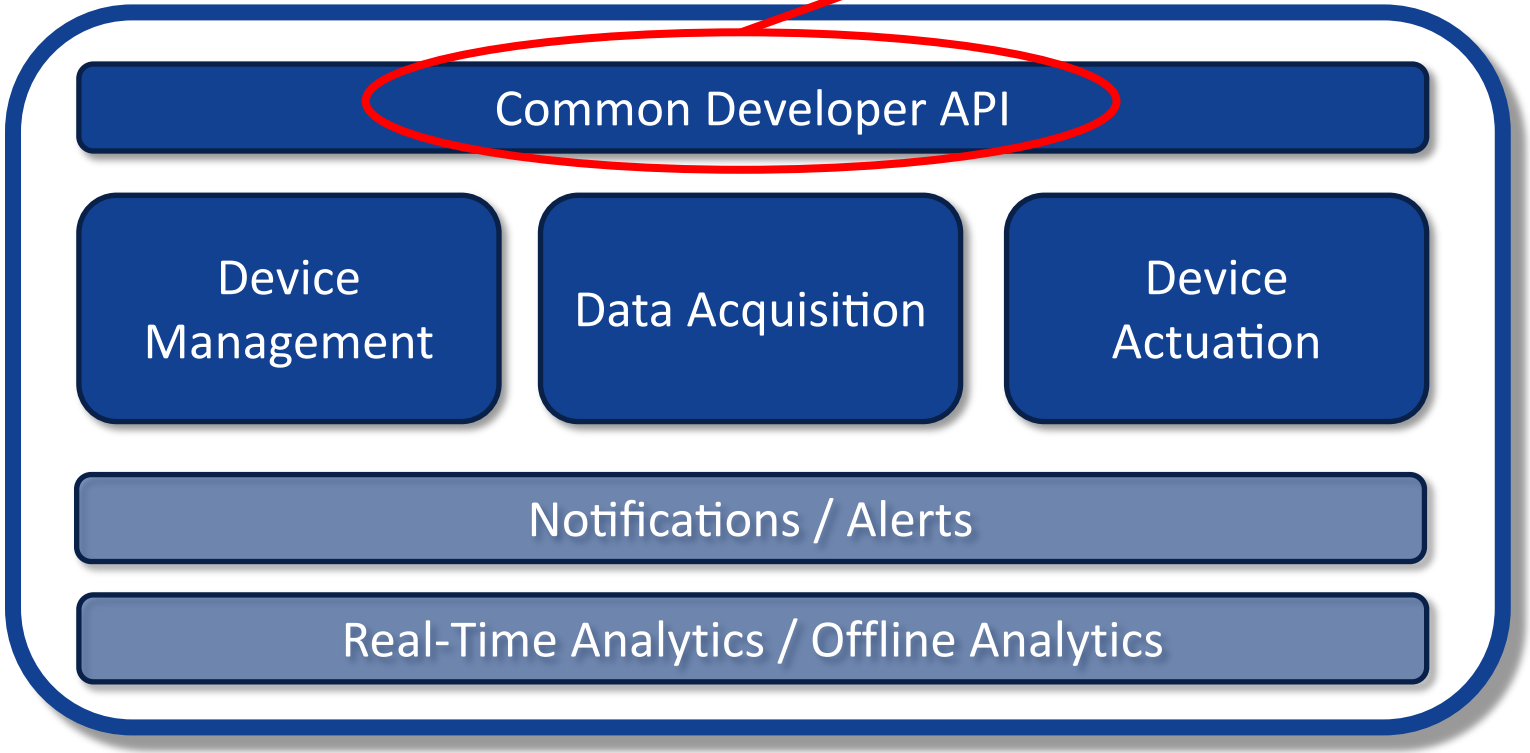
OPEN CONNECTIVITY FOUNDATION™

Broad Range of Software Stacks for IoT Devices

Over time, IoT devices will become more and more capable, harmonizing the E2E development needs

No OS	RTOS	Language Runtime	Full OS	App OS	Server OS	Container OS
						
Typical devices: - simple sensor devices, - heartbeat sensors, etc. Hardware architecture based on low-end microcontrollers No OS: Basic drivers only Minimum RAM: tens of kB No support for 3rd party SW development SW updates by reflashing Battery duration: weeks to months	Typical devices: - feature watches, - more advanced sensors Real-time OS (e.g., FreeRTOS, Nucleus, QNX) Minimum RAM: tens or hundreds of kilobytes No support for 3rd party SW development SW updates by reflashing Remote firmware updates possible Battery duration: days to weeks	Typical devices: - "maker" devices, - generic sensing solutions Off-the-shelf hardware (several choices available) RTOS + virtual machine (VM) supporting specific programming language Minimum RAM: hundreds of kilobytes 3rd party SW development supported Applications can be installed / updated dynamically Battery duration: days	Typical devices: - "maker" devices, - generic sensing solutions Off-the-shelf hardware (several choices available) Linux OS (typically) Minimum RAM: ½ to a few megabytes 3rd party SW development supported Software and applications can be installed / updated dynamically Battery duration: N/A or days	Typical devices: - high-end smartwatches Android Wear or Apple Watch OS Minimum RAM: 512 megabytes Rich APIs and tools for 3rd party SW development App stores Software and applications can be installed / updated dynamically Battery duration: hours to one day	Typical devices: - solutions benefiting from portable web server Off-the-shelf hardware (several choices available) Linux OS + Node.js Minimum RAM: tens of megabytes 3rd party SW development supported Software and applications can be installed / updated dynamically Battery duration: hours to one day	Typical devices: - solutions benefiting from fully "isomorphic" apps, i.e., code that can be migrated between cloud and edge Linux OS + Docker (or something equivalent) Minimum RAM: gigabytes Fully virtualized, isomorphic apps possible Applications isolated fully from the underlying HW architecture Battery duration: N/A or hours

Long-Term: Common Programmable World Solution



Programmable World – Software Engineering Challenges

- How to establish a common programmable world API set that would work across devices and systems from various different manufacturers?
- 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?

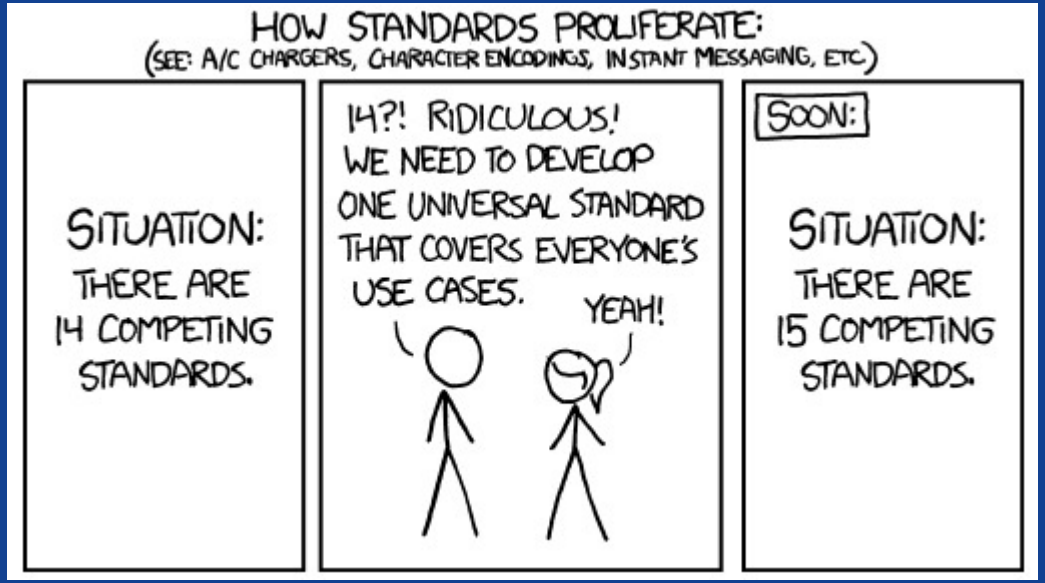
Summary and Key Takeaways

- 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 typical IoT program is *continuous* and *reactive* in nature. Programs are essentially *asynchronous*, *parallel* and *distributed*.
  - Devices are *weakly connected*, with potentially unreliable, intermittent connectivity.
  - Device configurations and topologies can be heterogeneous and vary considerably over time.
  - The number of devices/CPUs can be considerably larger than in traditional applications.
- 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.
- Most likely, it will still take 5-10 years until we get there – meanwhile, there are very interesting and relevant research challenges!

For Further Reading

- Taivalsaari, A. and Mikkonen, T. A Taxonomy of IoT Systems. IEEE Software, May/June 2018, in print.
  Mäkitalo, N., Ometov, A., Kannisto, J., Koucheryavy, Y., Andreev, S. and Mikkonen, T. Safe and Secure Execution at the Network Edge: A Framework for Coordinating Cloud, Fog, and Edge. IEEE Software, pp. 30-37, Jan/Feb 2018.
- Taivalsaari, A. and Mikkonen, T. Roadmap to the Programmable World: Software Challenges in the IoT Era. IEEE Software, Jan/Feb 2017.
- Gallidabino, A., Pautasso, C., Mikkonen, T., Systä, K., Voutilainen J.-P., and Taivalsaari, A. Architecting Liquid Applications. Journal of Web Engineering, Vol. 16, No. 5-6, pages 433-470, 2017.
- Taivalsaari, A. and Mikkonen, T. Beyond the Next 700 IoT Platforms. 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, Canada, October 5-8, 2017.

Thank you!


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.

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"