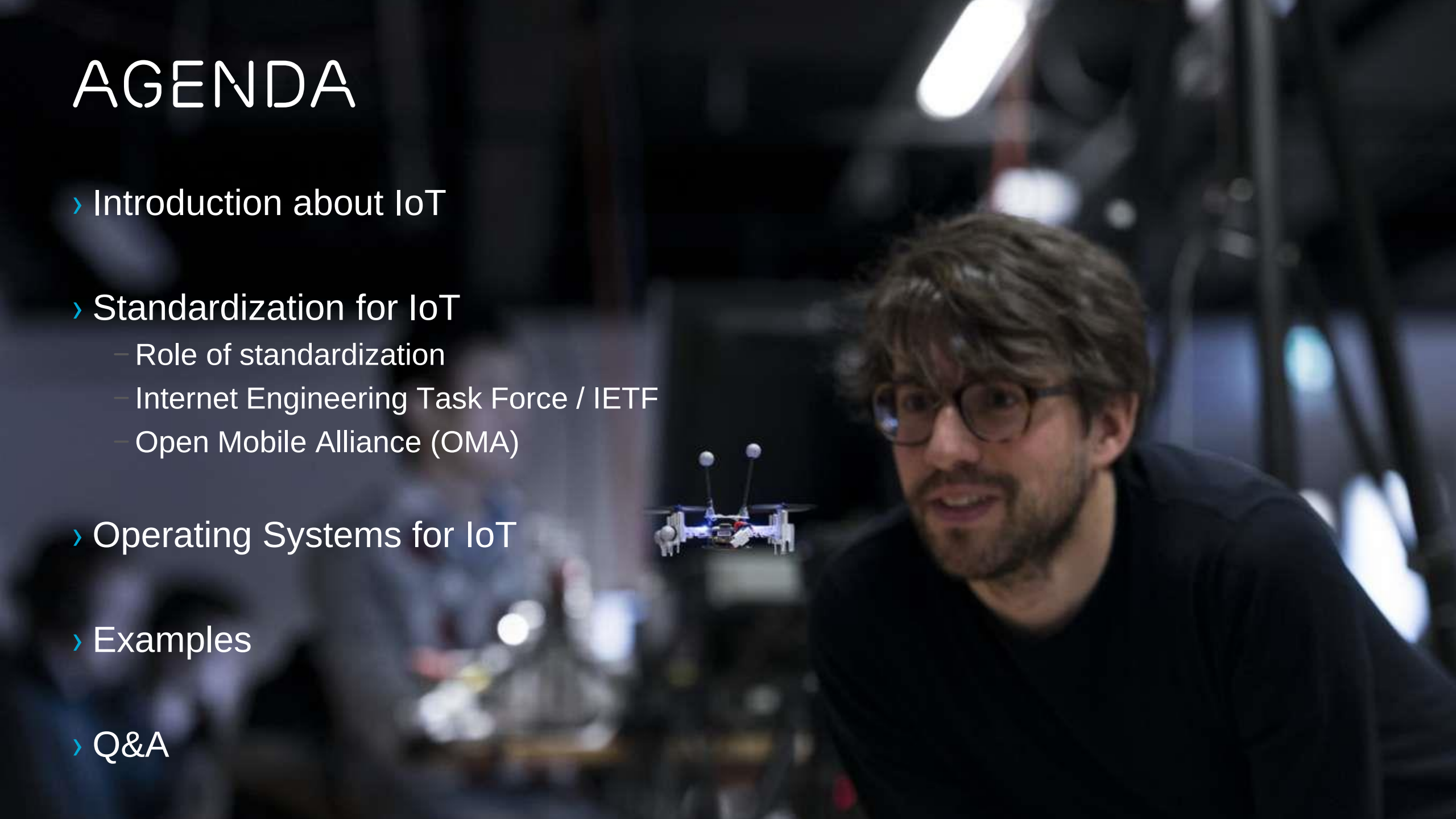
A VIEW ON CELLULAR AND WEB FOR IOT

December 2018

Jaime Jimenez – IoT Technologies Master Researcher, Finland
David Francino - IoT Sales Engagement Manager, Spain
Manuel Lorenzo – Technology & Innovation Manager, Spain

AGENDA

- Introduction about IoT
- Standardization for IoT
  - Role of standardization
  - Internet Engineering Task Force / IETF
  - Open Mobile Alliance (OMA)
- Operating Systems for IoT
- Examples
- Q&A



ONCE UPON A TIME …

"The social web of things"

DIGITAL REPRESENTATIONS OF THE REAL WORLD






temp
noise
air_quality
occupancy
energy
water

vibration
temperature
traffic_intensity
surface_condition
noise_level
route_to_work

energy
water
waste
CO2_emission
machine_tear
production

heart_rate
skin_conductance
gesture
mood
position
movement

location
occupancy
fuel
emissions
speed

irrigation
luminosity
nutrition
moisture
pesticides

DIVERSE IOT USE CASES

MASSIVE IOT

SMART BUILDING

LOGISTICS, TRACKING AND FLEET MANAGEMENT

SMART METER

SMART AGRICULTURE

CAPILLARY NETWORKS

CRITICAL IOT

AUTONOMOUS CAR

TRAFFIC SAFETY & CONTROL

INDUSTRIAL APPLICATION & CONTROL

REMOTE MANUFACTURING, TRAINING, SURGERY

LOW COST, LOW ENERGY
SMALL DATA VOLUMES
MASSIVE NUMBERS

ULTRA RELIABLE
VERY LOW LATENCY
VERY HIGH AVAILABILITY

CELLULAR FOR MASSIVE IOT

IoT on LTE

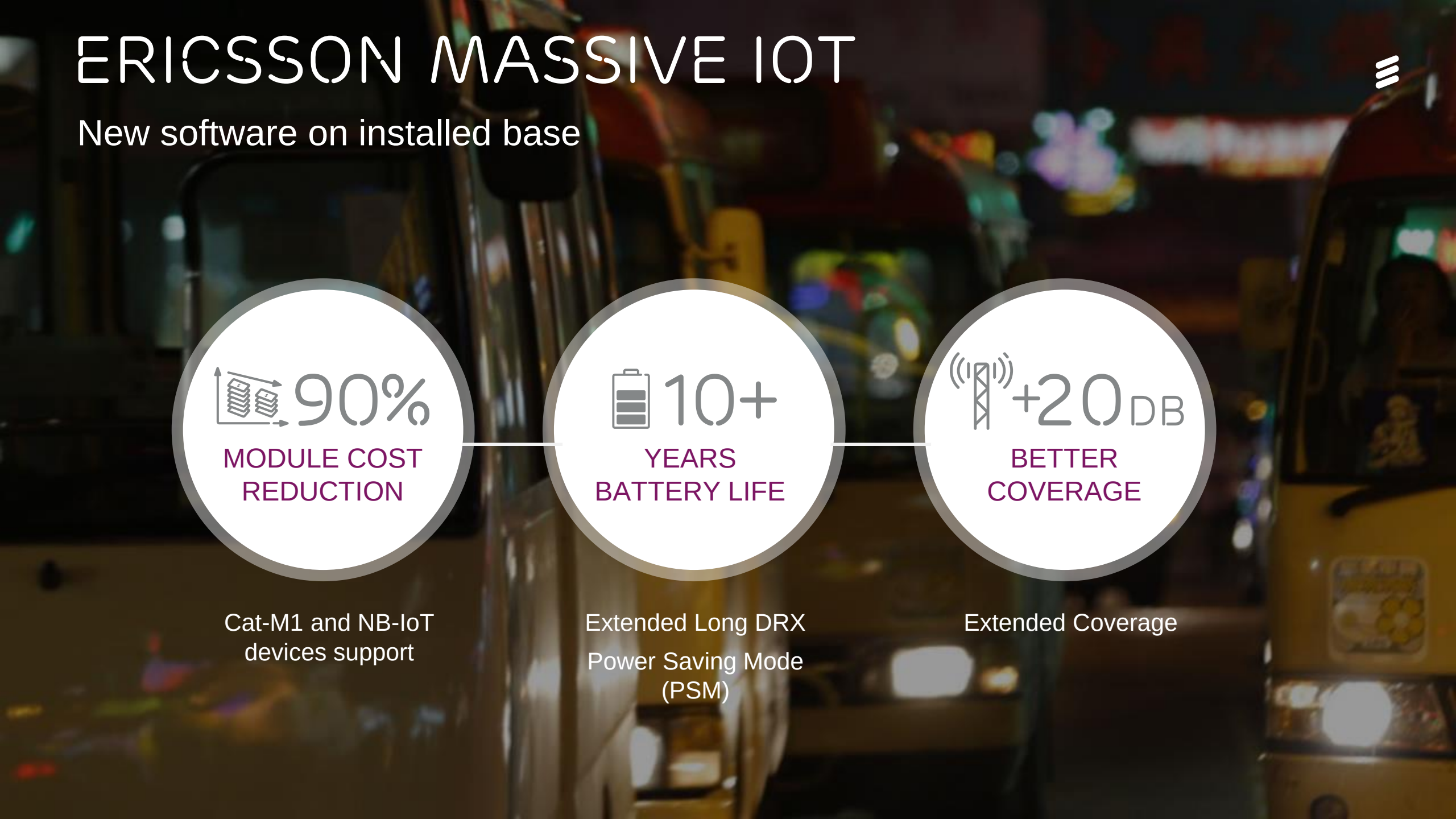
NB-IoT	Low-bitrate applications with extreme coverage and low cost devices
CAT-M1	Wider range of applications with low-medium bitrate, mobility and voice support

IoT on GSM

EC-GSM-IoT	GSM as fallback with improved battery life or extended coverage for the last mile

ERICSSON MASSIVE IOT

New software on installed base



Cat-M1 and NB-IoT devices support

Extended Long DRX

Power Saving Mode (PSM)

Extended Coverage

INDUSTRY IOT DRIVERS

Remote monitoring and control

Predictive maintenance

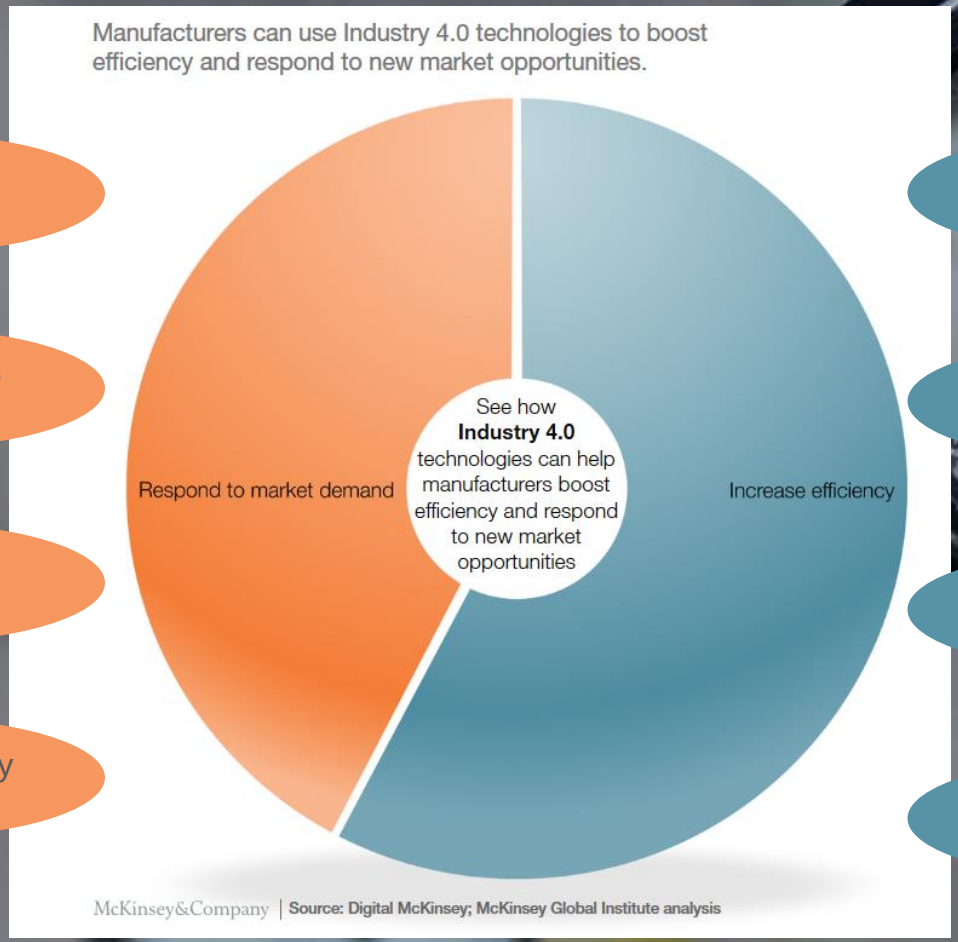
After-sales services

Faster time to market

Design

Advanced process control

Product Quality Improvement



Optimized processes

Greater asset utilization

Higher Labor productivity

Supply chain and inventories

Smart Energy Consumption Real-time optimization

Remote monitoring and control

Predictive maintenance

Source: McKinsey Global Institute - Making it in America: Revitalizing US manufacturing

IOT VALUE CHAIN QUANTIFIED

Estimated Technology Spending Split (2025)*

Services	~30%
Applications *inc Marketplaces*	~25%
Platforms *Enablement & Application*	~25%
Connectivity *Networks & Management*	~10%
Devices *inc Chipsets & Sensors*	~10%

* Source: Various inc McKinsey, AT Kearny, Heavy Reading, IDC, Machina

MARKET WITH EXPONENTIAL PROMISE

43%
ENTERPRISES ADOPTING IoT

$7.1T
GLOBAL IoT REVENUE

$11T
GLOBAL IoT ECONOMIC IMPACT

2016 2017 2020 2022 2025

4.1M
NEW IoT DEVICES CONNECTED DAILY

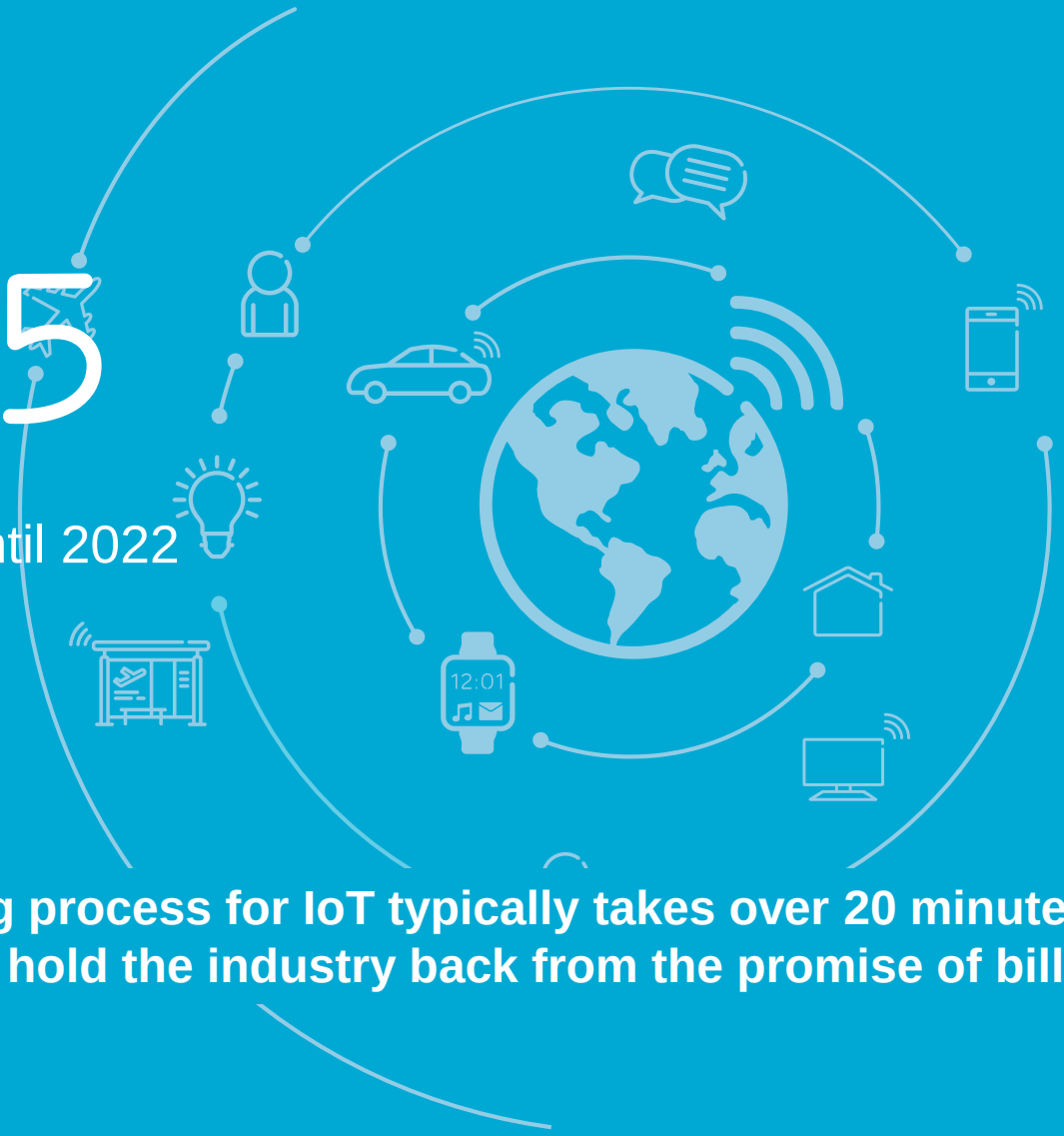
18B
GLOBAL IoT CONNECTIONS

€1T
EU28 IoT ECONOMIC VALUE

Source: Gartner, Ericsson, IDC, McKinsey, AT Kearney

12.5B NEW IOT DEVICES REQUIRE A NEW APPROACH TO ONBOARDING

12.5
Billion IoT devices until 2022

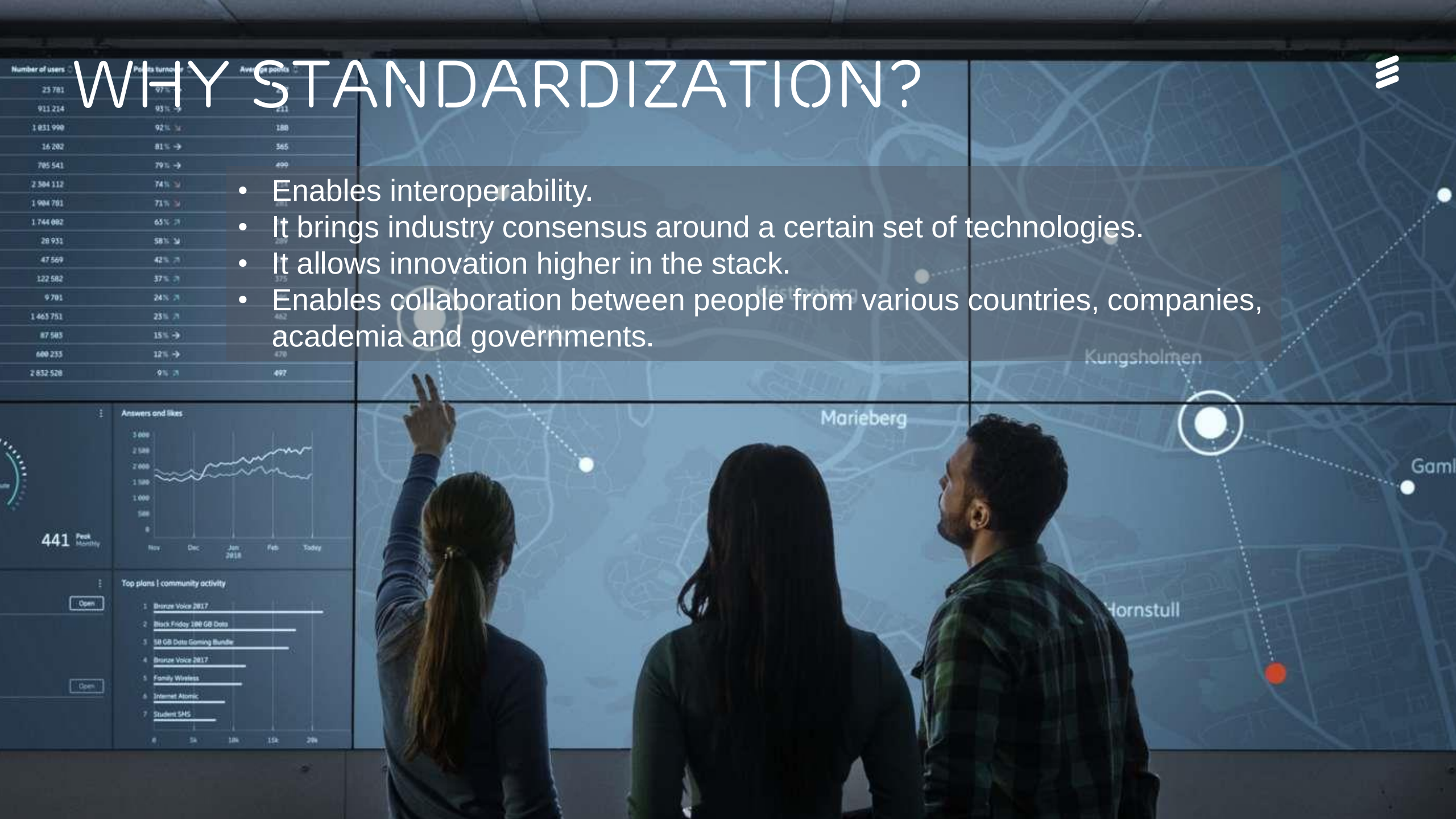


25
Million new applications

Today, the onboarding process for IoT typically takes over 20 minutes per device. This high cost, manual process will hold the industry back from the promise of billions of connected devices.

WHY STANDARDIZATION?

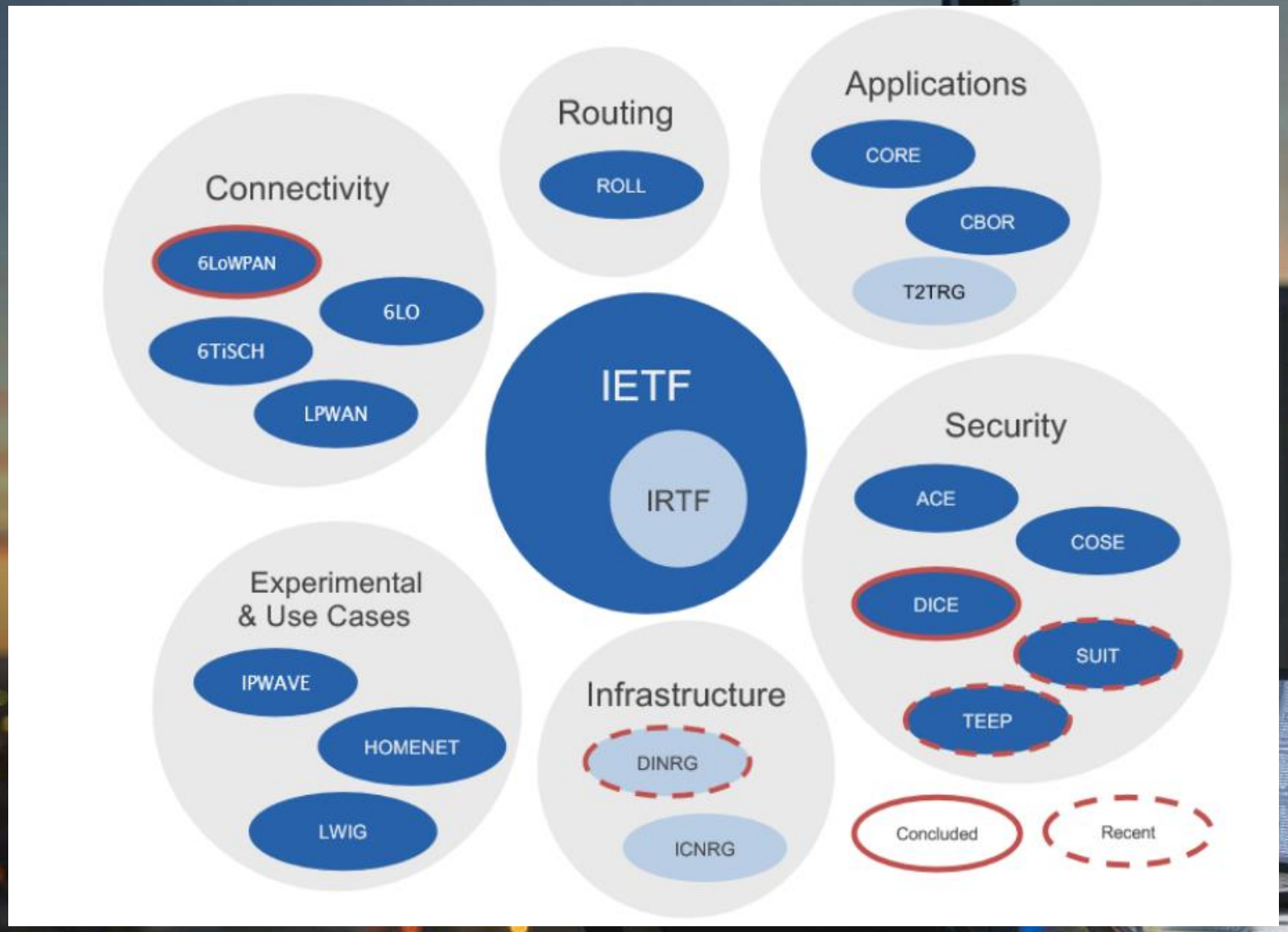
- Enables interoperability.
- It brings industry consensus around a certain set of technologies.
- It allows innovation higher in the stack.
- Enables collaboration between people from various countries, companies, academia and governments.



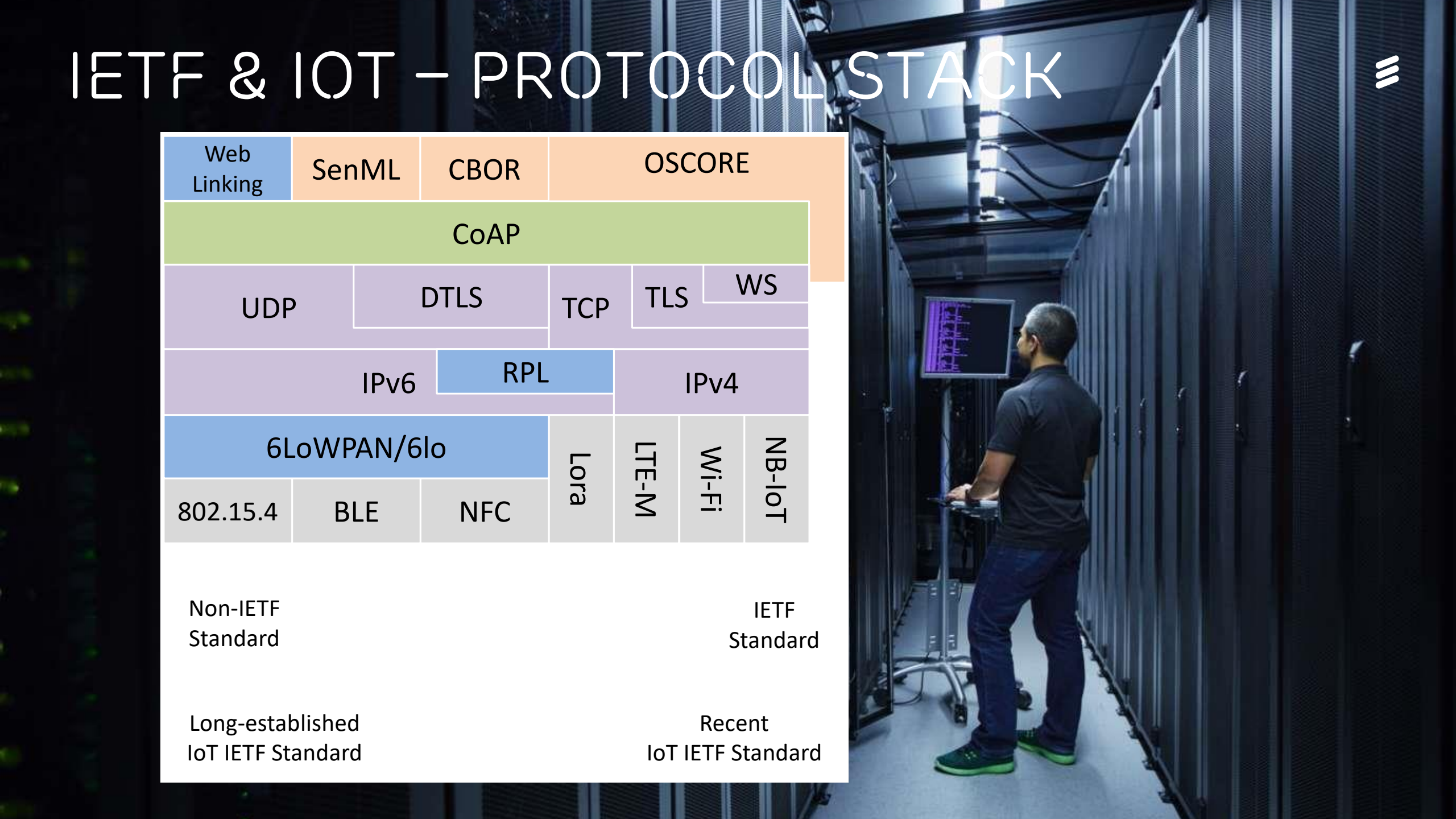
INTERNET ENGINEERING TASK FORCE (IETF)

- **Mission**: “The mission of the IETF is to make the Internet work better by producing high quality, relevant technical documents that influence the way people design, use, and manage the Internet.”
- **Role:** Specifying the underlying, fundamental Internet Technologies (e.g. IP, UDP, TCP, HTTP, TLS…)
- **Principles: “**We reject: kings, presidents and voting. We believe in: rough consensus and running code.”
- **Topics of interest:**
  - Automated network management
  - Internet of Things
  - New transport technology
  - Internet Security

IETF & IOT- WORKGROUPS



IETF & IOT – PROTOCOL STACK



IETF & IOT - CORE WG

Constrained Restfull Environments (CoRE) WG created the Constrained Application Protocol (CoAP).

› Aims to make the REST paradigm available for devices and networks that might be too constrained to use the typical approaches around HTTP.

core	
RFC 6690 *Constrained RESTful Environments (CoRE) Link Format*	Defines a specific link format for being used by constrained web servers. The Constrained RESTful Environments (CoRE) Link Format is carried as a payload and allows describing hosted resources, their attributes, and other relationships between links.
RFC 7252 *The Constrained Application Protocol (CoAP)*	Defines the Constrained Application Protocol (CoAP), which is a RESTful application protocol mainly designed for being used in machine-to-machine (M2M) application, constrained nodes and constrained networks. Among the different CoAP features described in this document, we can find definition of the request/response interaction model between application endpoints, support for built-in discovery of services and resources, and definition of Web-related key concepts such as URIs and Internet media types.
RFC 7641 *Observing Resources in the Constrained Application Protocol (CoAP)*	Defines a CoAP extension for observing the resource state of a CoAP server. Specifically, it enables CoAP clients to be kept updated about the representation of a resource over a period of time.
RFC 8323 *CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets*	Delineates the changes required to use CoAP over TCP, TLS, and WebSockets transports.
RFC 8428 *Sensor Measurement Lists (SenML)*	Defines a format for representing simple sensor measurements and device parameters in Sensor Measurement Lists (SenML). It offers several representational notations, JSON, CBOR, EXI and XML.

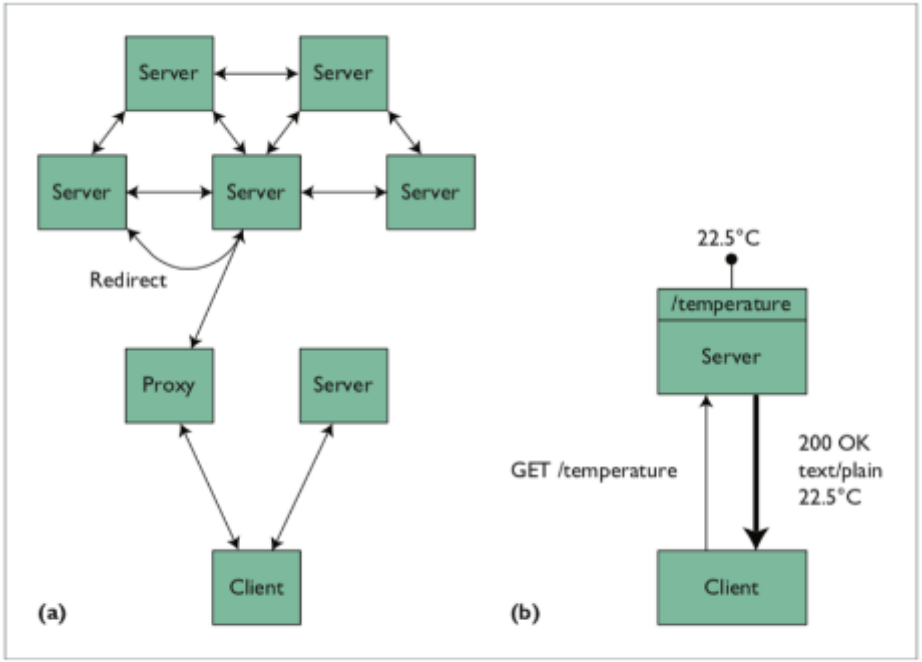


Figure 1. The Web architecture. (a) Clients access servers directly and via proxies; (b) a GET request elicits a 200 OK response.

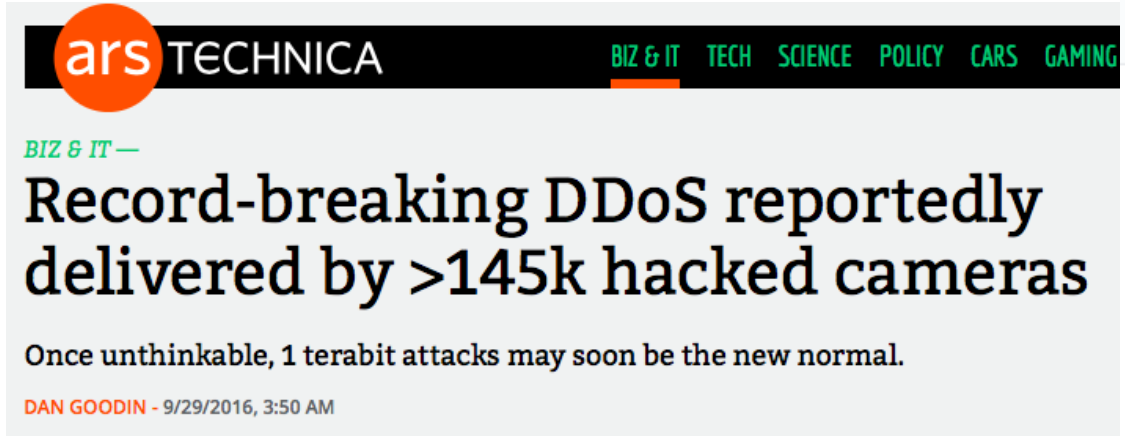
IETF®

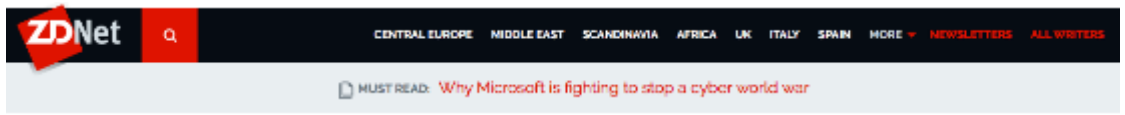
IETF & IOT - SUIT WG

[1]

Devices infected by Mirai continuously scan the internet for the IP address of Internet of things (IoT) devices. Mirai includes a table of IP Address ranges that it will not infect, including private networks and addresses allocated to the United States Postal Service and Department of Defense.[14]

Mirai then identifies vulnerable IoT devices using a table of more than 60 common factory default usernames and passwords, and logs into them to infect them with the Mirai malware.[6][15][16] Infected devices will continue to function normally, except for occasional sluggishness,[15] and an increased use of bandwidth. A device remains



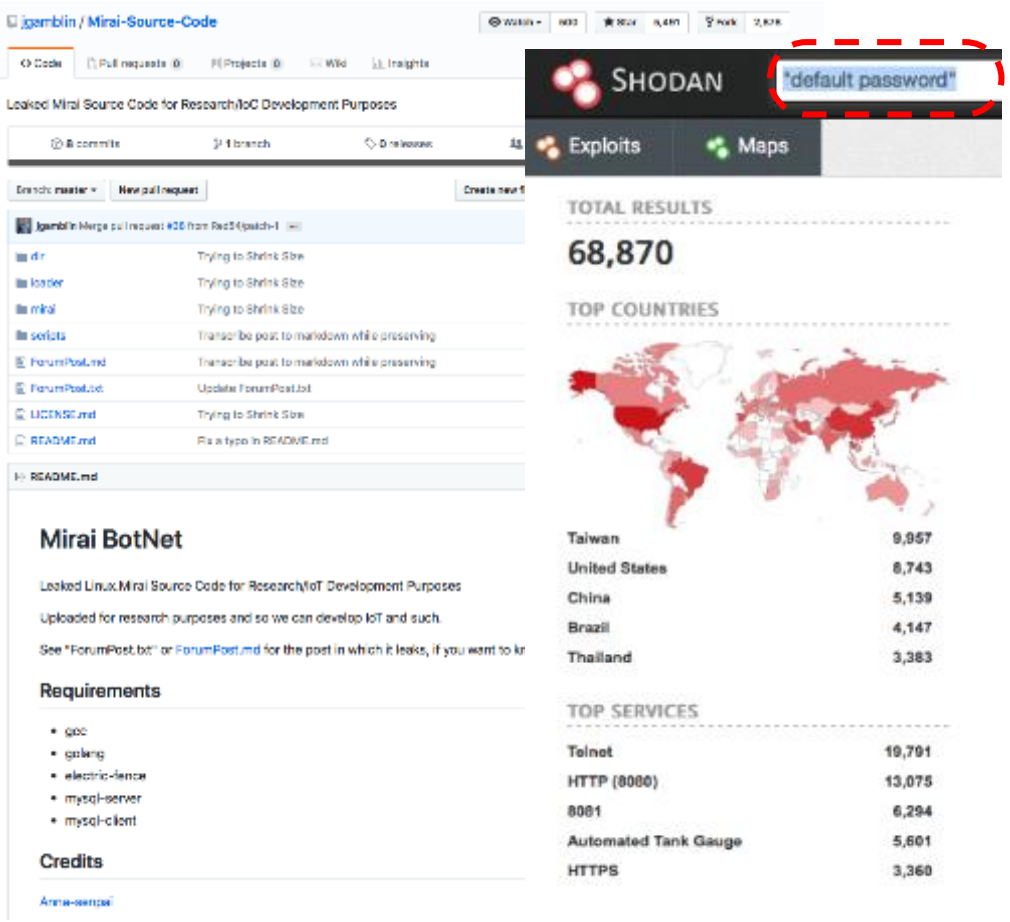


[2]

The CoAP protocol is the next big thing for DDoS attacks

CoAP DDoS attacks have already been detected in the wild, some clocking at 320Gbps.

By Catalin Cimpanu for Zero Day | December 5, 2018 -- 04:13 GMT (04:13 GMT) | Topic: Security

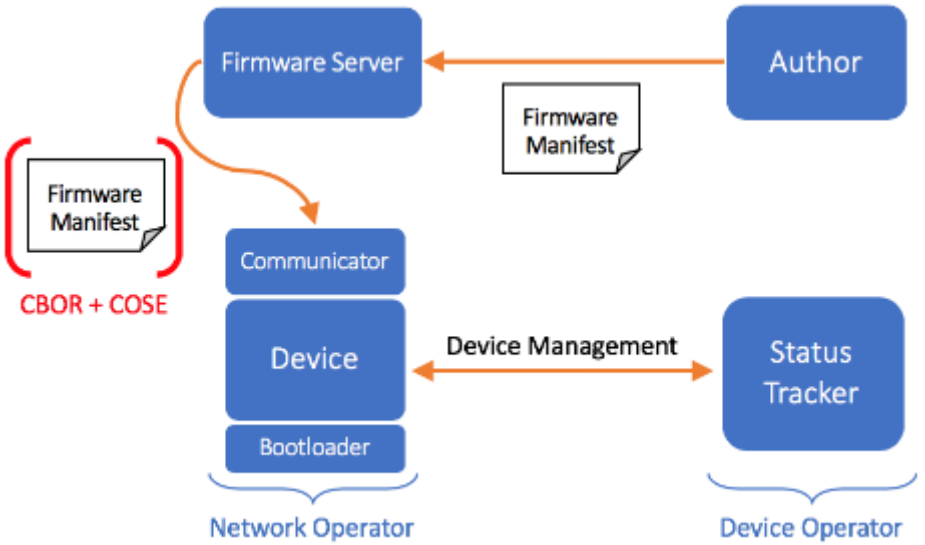


[1] https://en.wikipedia.org/wiki/Mirai_(malware)
[2] https://www.zdnet.com/article/the-coap-protocol-is-the-next-big-thing-for-ddos-attacks/

IETF & IOT - SUIT WG

Software Updates for Internet of Things (SUIT) WG was chartered to mitigate IoT attacks by improving software update mechanisms.

› The group focuses on defining a firmware update solution that can be used by devices with as little as 10 kB RAM and 100 kB flash.



suit	
A Firmware Update Architecture for Internet of Things Devices	Defines requirements and architecture for a firmware update mechanism suitable for constrained IoT devices. The architecture is defined in such a way to decouple it from the underlying transport of the firmware images and associated meta-data.
Firmware Updates for Internet of Things Devices - An Information Model for Manifests	Defines all the information that must be present in the meta-data ("manifest") used for ensuring a secure firmware update mechanism. The manifest describes the firmware image(s) and ensures appropriate protection.
A CBOR-based Firmware Manifest Serialisation Format	Describes the format of a manifest about the firmware for an IoT device (e.g where to find the firmware, the devices to which it applies, and cryptographic information protecting the manifest).

OMA LWM2M

IETF delivers technology to other SDOs.

› Open Mobile Alliance has built a device management protocol (LWM2M) for constrained devices around open standards.

LWM2M version 1.0 architecture

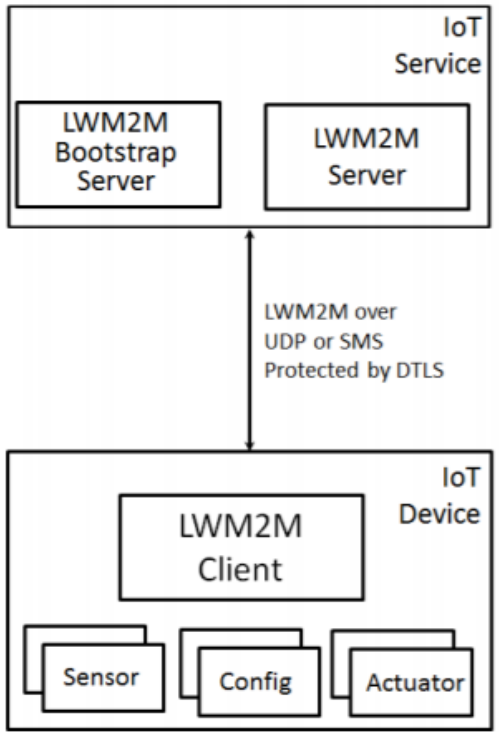


Figure 1: Entities in the LWM2M Architecture.

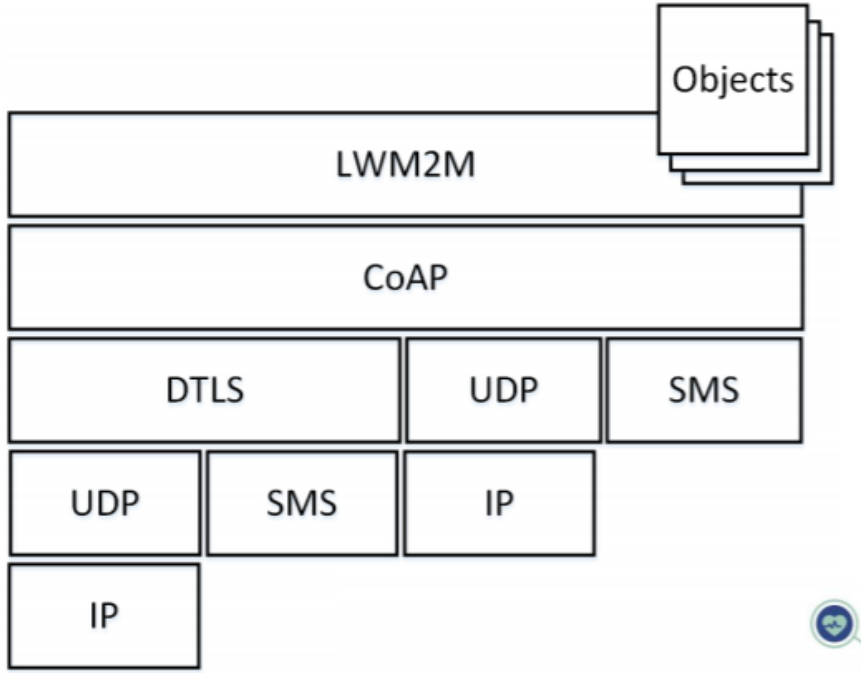
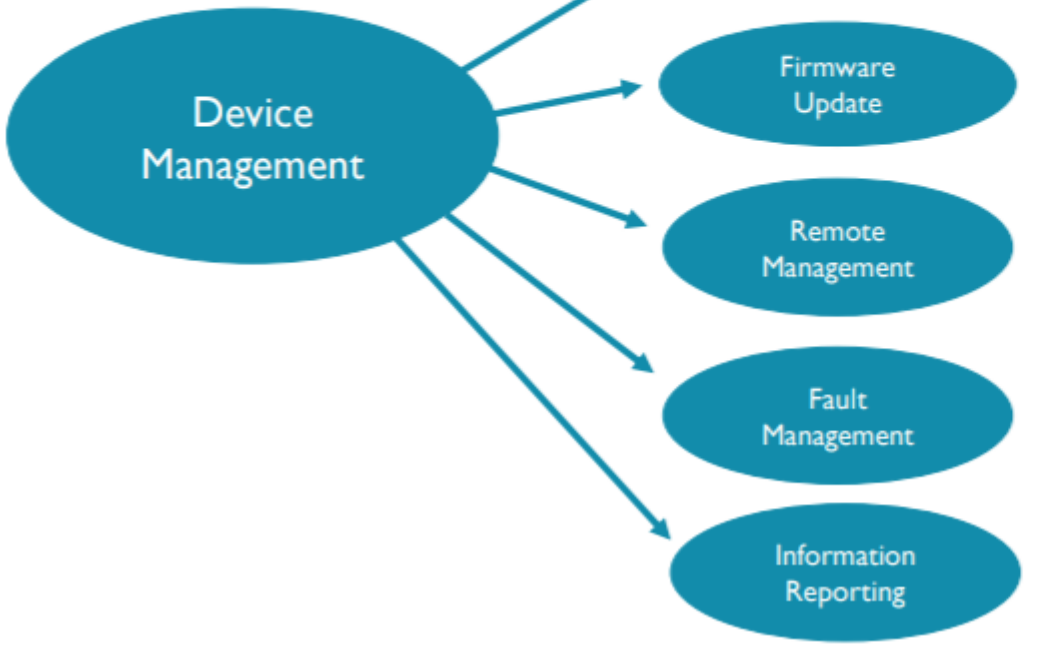


Figure 2: Protocol Stack

LightweightM2M
specified at oma

OMA LWM2M

Diverse IoT deployments with common needs



- Service provisioning
- Key management
- Provisioning of access control lists

- Update application and system software
- Apply bug fixes and add new features

- Changes to settings
- Trigger actuators

- Report errors from devices
- Query status of devices

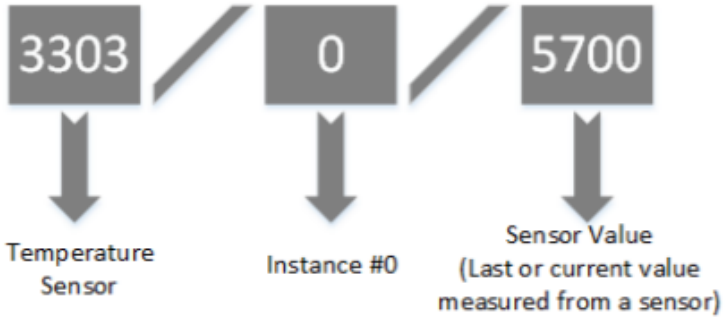
- Notify changes in sensor values
- Retrieve configuration settings and device status

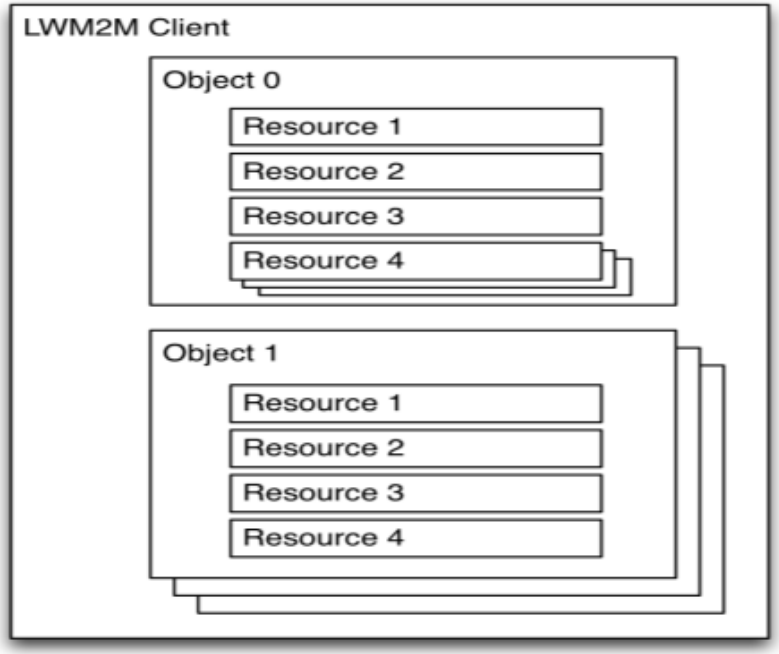
OMA LWM2M

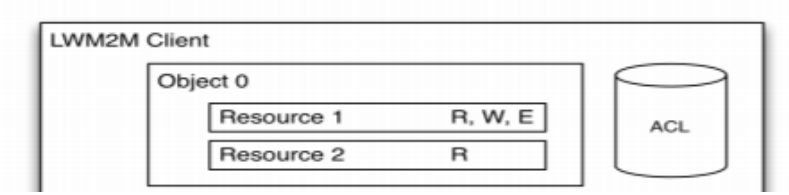
Object model

- Objects/Resources are accessed with simple URIs:
  /{Object ID}/{Object Instance}/{Resource ID}

- Example:







OMA LWM2M

Objects

The LWM2M technical specification itself defines eight objects; the repository contains many more contributed by IPSO alliance, oneM2M, and from vendors.

Object Name	ID	Description
LWM2M Security	0	Keying material of a LWM2M Client to access a LWM2M server.
LWM2M Server	1	Data related to a LWM2M server.
Access Control	2	Information used to check whether a LWM2M Server has access to object.
Device	3	Device related information, including device reboot and factory reset function.
Connectivity Monitoring	4	Parameters related to network connectivity.
Firmware	5	Capability to update firmware
Location	6	Device location information
Connectivity Statistics	7	Information like transmit and receive counters

SEE MORE IN ERICSSON BLOG

Automated Discovery of CoAP Devices

Concept

Discovering CoAP endpoints is a well-established procedure and much necessary on IoT deployments. We analyzed all mechanism available to date and built a crawler based on them.

Automated Discovery of CoAP Devices is the outcome of that work.

Background

The Constrained Application Protocol (CoAP) has been designed by the IETF as a universal transfer protocol with uniform identifiers for resources, methods for enabling resources interactions, resource discovery mechanisms, and security extensions. CoAP endpoints use a *typed link* specified in "Web Linking" (RFC5988). All endpoints have an URI called "`/.well-known/core`" that exposes their resources.

A mechanism for discovery is the Resource Directory, which provides links for the resources hosted by CoAP servers, as well as other metadata such as attributes and possible link relations.

Making the discovery automatic needs the application to be structured meeting the REST design principles. The principles from HATEOAS allow to create an underlying continuous process of functionalities discovery. Similar to that of a crawler, a simple peer would then:

1. Receive requests on `/.well-known/core` from devices with pre-configured connections;
2. Answer providing the list of resources hosted;
3. Extract the IP address from the request just received and start a "Mutual Discovery" process;
4. Receive list of resources hosted by another device, process them and create new links via the "`hostedby`" relation type.

Evaluation

We emulate IoT devices that run CoAP C/S and host few resources. Each of these devices engages one or more discovery mechanisms and builds virtual connections with other nodes. This process is periodically triggered, new devices that are deployed at run-time can discover nodes already present on the network and build relations with them.

End-points perform the discovery using the well-known URI as entry point. The discovery can be performed either via Unicast or Multicast.

The crawler tries every discovery technique, shows the devices discovered, their resources and the link paths that things have built over time.

The testbed is built on docker, with each device having its own IP network interface. When using docker-compose we will have multicast communication too. Each node will expose port 5683 for UDP connections over CoAP. Resources are simulated. The workload characterization is that of 150 simple peers and a multicast peer.

Bytes exchanged (left) - Distribution of discovery times mean (right). The Multicast Peer takes about 74 seconds to discover 150 devices.

We run 12 experiments, starting from 300 requests/minute and ending with 180000 requests/minute.

Knee Capacity happens between 12000 req/min and 24000 requests/minute.

Conclusion

This work makes use of common standards and protocols, presenting an autonomous way for devices to discover other endpoints as well as autonomously update their own state based on that. Future work will be on adding new features to the crawling algorithm.

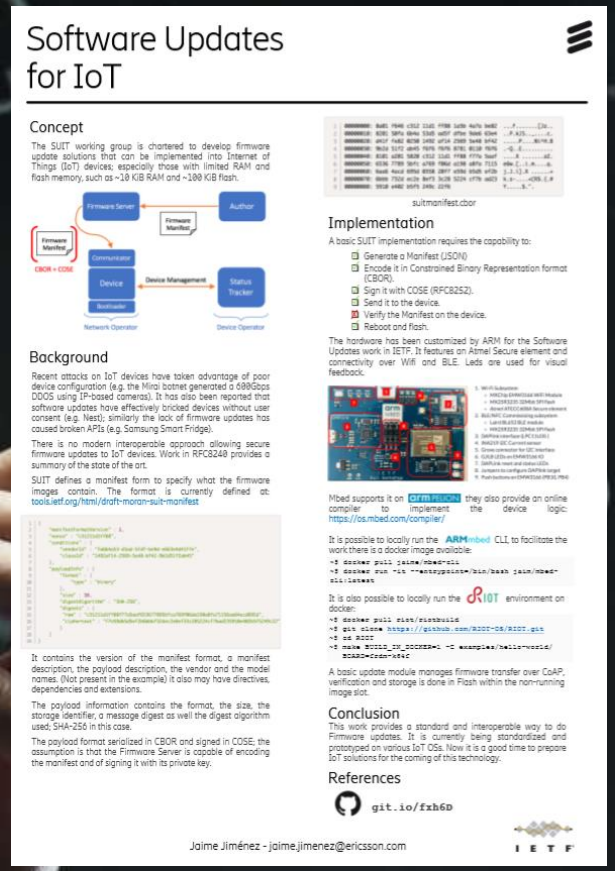
References

`git clone https://git.overleaf.com/17326933pxvtmfztcfrv`

Jaime Jiménez, Francesco Caturano, Simon Pietro Romano

IEEE COMMUNICATIONS MAGAZINE

http://jaimejim.github.io/docs/automatic_discovery_poster.pdf



Software Updates for IoT

Concept

The SUIT working group is chartered to develop firmware update solutions that can be implemented into Internet of Things (IoT) devices; especially those with limited RAM and flash memory, such as ~10 KiB RAM and ~100 KiB flash.

Background

Recent attacks on IoT devices have taken advantage of poor device configuration (e.g. the Mirai botnet generated a 600Gbps DDOS using IP-based cameras). It has also been reported that software updates have effectively bricked devices without user consent (e.g. Nest); similarly the lack of firmware updates has caused broken APIs (e.g. Samsung Smart Fridge).

There is no modern interoperable approach allowing secure firmware updates to IoT devices. Work in RFC8240 provides a summary of the state of the art.

SUIT defines a manifest form to specify what the firmware images contain. The format is currently defined at: tools.ietf.org/html/draft-moran-suit-manifest

It contains the version of the manifest format, a manifest description, the payload description, the vendor and the model names. (Not present in the example) it also may have directives, dependencies and extensions.

The payload information contains the format, the size, the storage identifier, a message digest as well the digest algorithm used; SHA-256 in this case.

The payload format serialized in CBOR and signed in COSE; the assumption is that the Firmware Server is capable of encoding the manifest and of signing it with its private key.

suitmanifest.cbor

Implementation

A basic SUIT implementation requires the capability to:

- Generate a Manifest (JSON)
- Encode it in Constrained Binary Representation format (CBOR).
- Sign it with COSE (RFC8252).
- Send it to the device.
- Verify the Manifest on the device.
- Reboot and flash.

The hardware has been customized by ARM for the Software Updates work in IETF. It features an Atmel Secure element and connectivity over Wifi and BLE. Leds are used for visual feedback.

Mbed supports it on GITHUB.COM/ARMMBED they also provide an online compiler to implement the device logic: https://os.mbed.com/compiler/

It is possible to locally run the ARMmbed CLI, to facilitate the work there is a docker image available:

```
~$ docker pull jaime/mbed-cli
~$ docker run -it --entrypoint=/bin/bash jaim/mbed-cli:latest
```

It is also possible to locally run the RIOT environment on docker:

```
~$ docker pull riot/riotbuild
~$ git clone https://github.com/RIOT-OS/RIOT.git
~$ cd RIOT
~$ make BUILD_IN_DOCKER=1 -C examples/hello-world/ BOARD=cc2538dk
```

A basic update module manages firmware transfer over CoAP, verification and storage is done in Flash within the non-running image slot.

Conclusion

This work provides a standard and interoperable way to do Firmware updates. It is currently being standardized and prototyped on various IoT OSs. Now it is a good time to prepare IoT solutions for the coming of this technology.

References

`git.io/fxh6D`

Jaime Jiménez - jaime.jimenez@ericsson.com

IETF

http://jaimejim.github.io/docs/automatic_discovery_poster.pdf

https://www.ericsson.com/en/blog

https://jaimejim.github.io/

OPERATING SYSTEMS FOR IOT

2014 – YANZI NETWORKS

The challenge
Creating smart buildings and tracking. Lack of interoperability at network, application and semantics.

The solution
Using common standards (802.15.4, IPv6, RPL, CoAP, IPSO). Use of Open Source (Contiki).

The result

- Very connected to research with RISE (Research Institutes of Sweden) and the Contiki developers.
- Focus on simple and quick device deployment as well as security.
- Acquired (control stock) for roughly 50M€ this July.

2015 – SMART ROCKBOLT

The challenge
Structural damages in mines. Lack of adequate alert systems and of measuring capabilities. No interest in reinventing the wheel.

The solution
Collaboration btw companies, Luleå University (PIMM project). Use of well-known standards.

The result

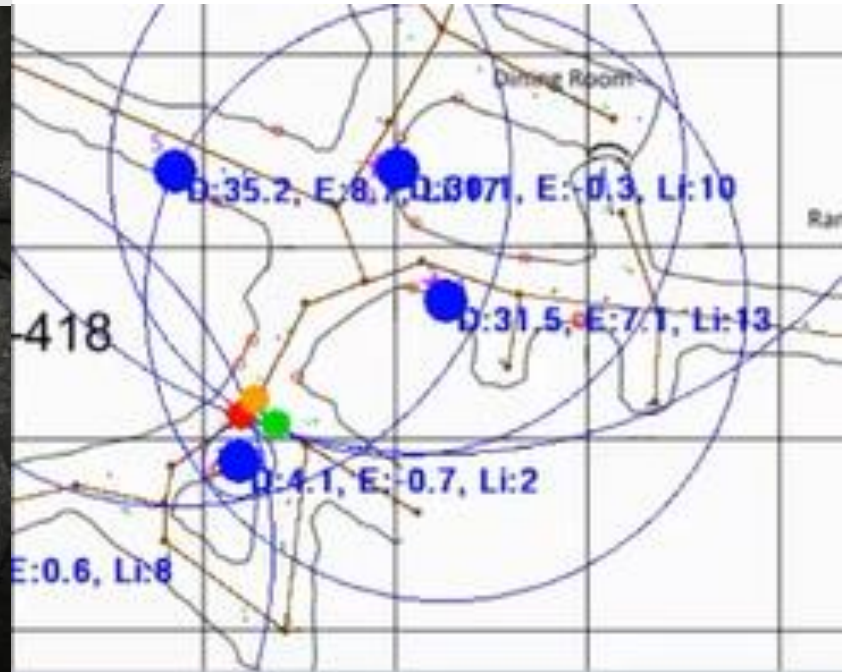
- Smart Rockbolt providing visual cues about the cave status. Gathering of telemetry from the bolt creating a “Digitalized mining Area”.
- Thingwave company as spinoff (8 people team).

IOT FOR SECURITY & SAFETY

“As a mine operator I want to secure good air quality for improved safety by more advanced control of ventilation”

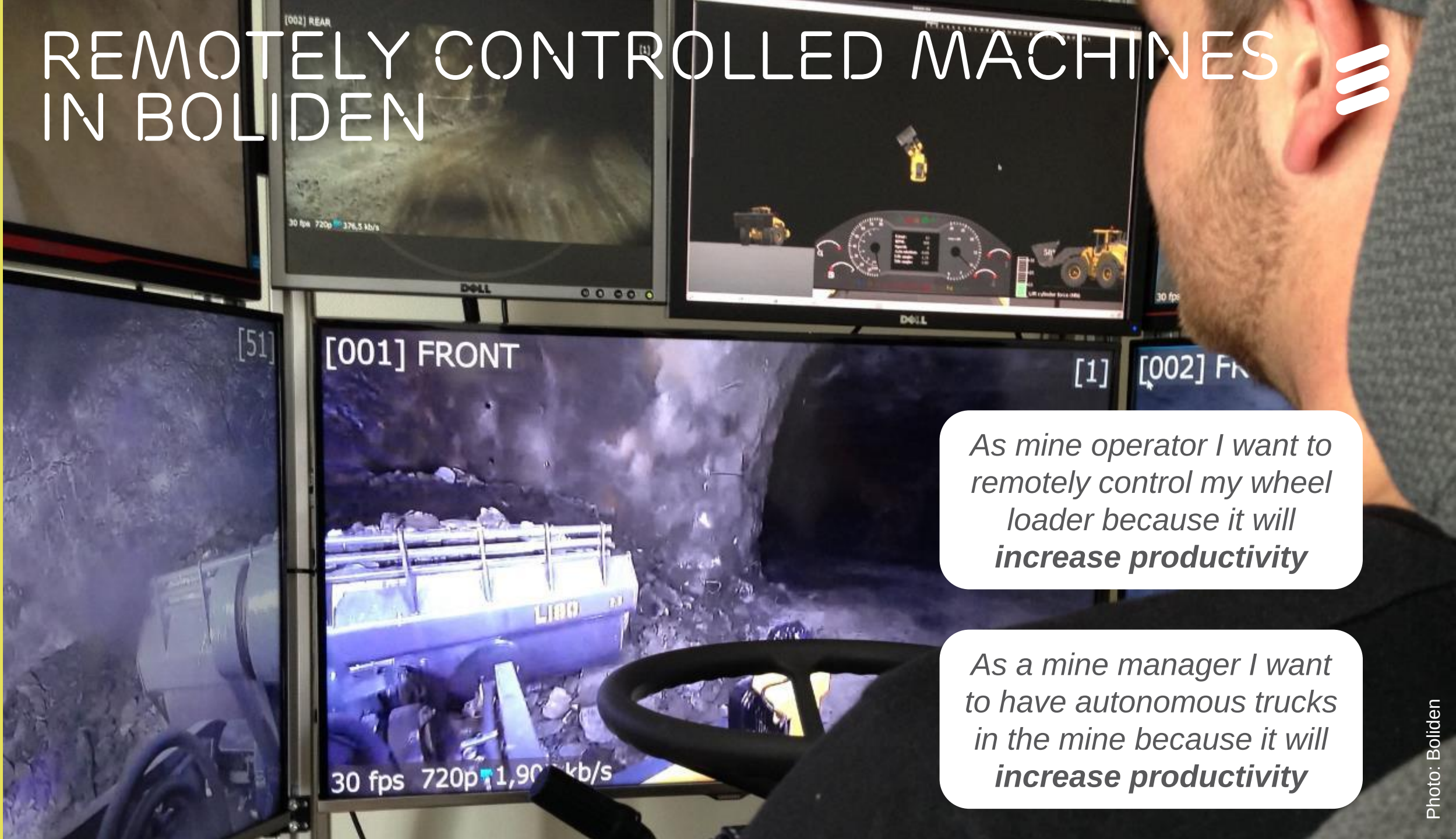
“As responsible for mine safety I want to secure improved safety by understanding rock movements”

“As responsible for mine safety I want accurate positioning because it will improve safety”



Photos: Boliden

REMOTELY CONTROLLED MACHINES IN BOLIDEN



As mine operator I want to remotely control my wheel loader because it will ***increase productivity***

As a mine manager I want to have autonomous trucks in the mine because it will ***increase productivity***

Photo: Boliden

MIXED REALITY USE CASES TRANSPARENT ROCK

- Visual augmentations to make rock see-through, e.g. allowing to see co-workers in other tunnels

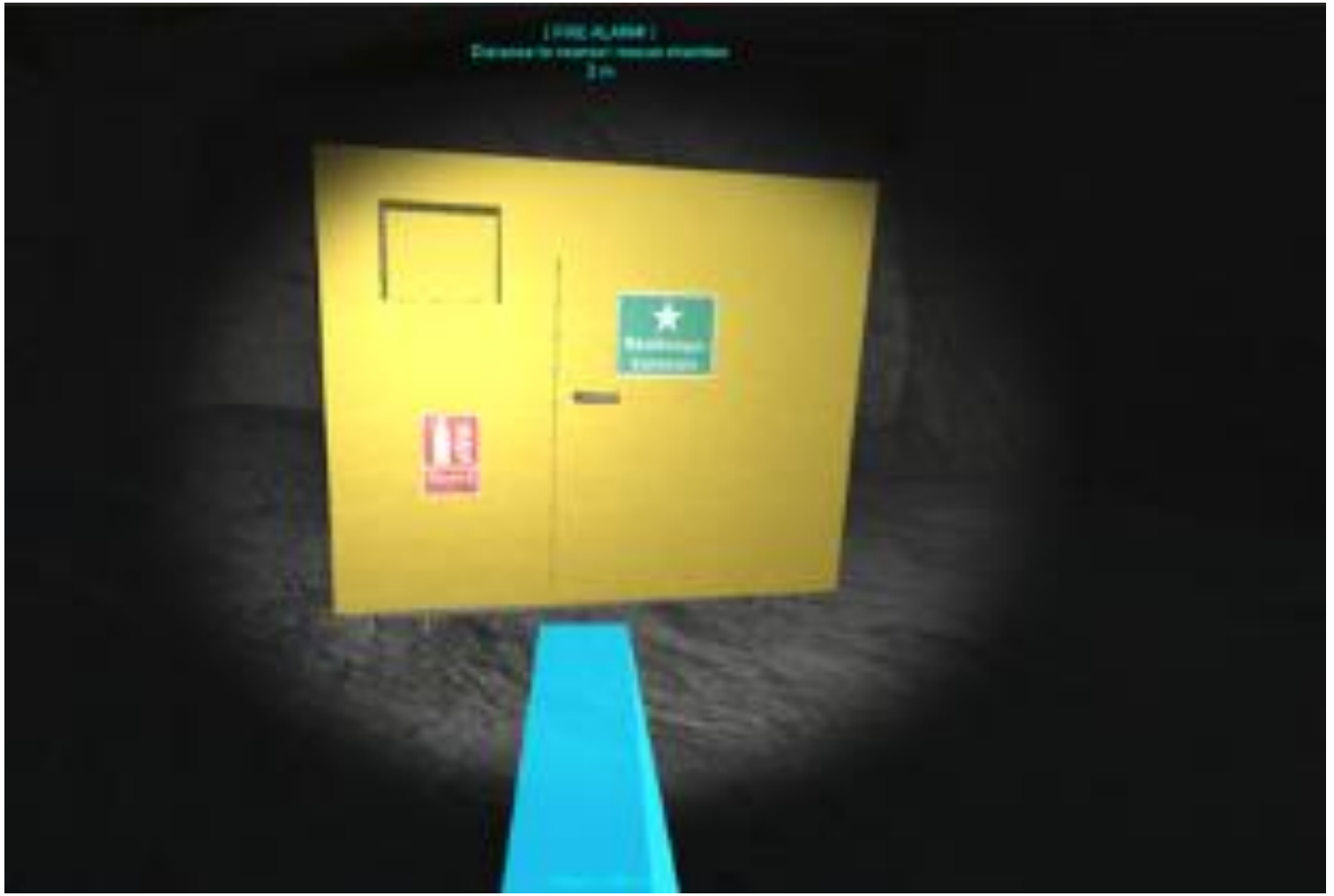
MIXED REALITY USE CASES
SAFETY GUIDES

› Lane guide and alerts of hazardous materials.

Virtual path to continue operation even without light avoiding accidents.

MIXED REALITY USE CASES
SAFETY GUIDES

› Guide to rescue chamber, exit, or to the one in distress in case of emergency



2016 – HUSQVARNA GARDENA

The challenge
New generation of “smart mowers“ and “smart gardening” solutions. Management of devices.

The solution
Common R&D project between few companies. Looking into state-of-the-art protocols (LWM2M) and Open Source (Contiki) technologies.

The result

- Part of the Gardena portfolio uses these standards and technologies.
- New IoT product market segment validated last month via stock crash (19%) of other unit.
  - “**exit from low price point product segments** and brands, particularly in petrol powered lawnmowers and garden tractors.”
  - “To **focus on future premium product** and service offerings under the core brands of Husqvarna and **Gardena**.”

Swedish Chainsaw Massacre Hits Husqvarna

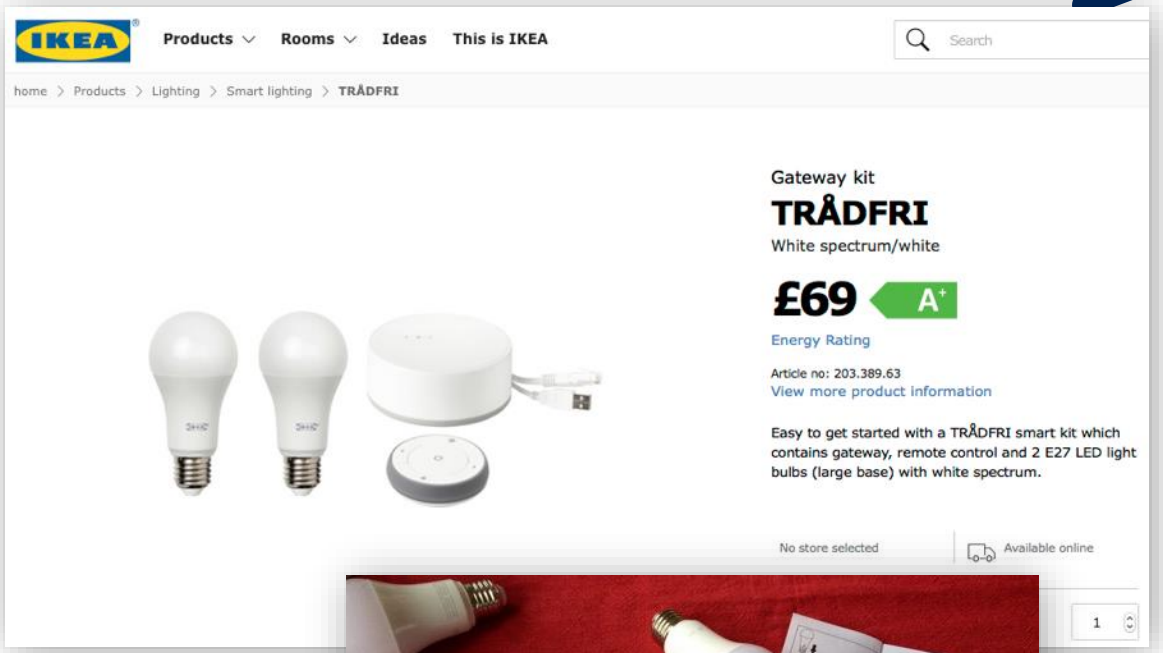
By Niklas Magnusson
July 17, 2018, 11:08 AM GMT+3
Updated on July 17, 2018, 11:31 AM GMT+3

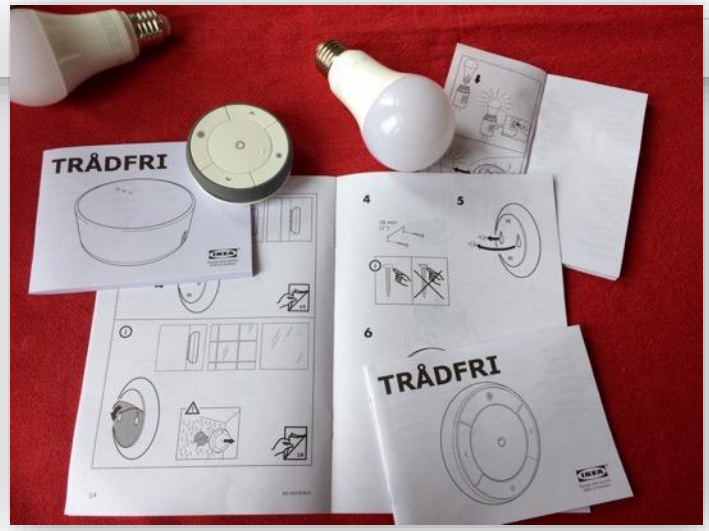
- Shares fall the most since Husqvarna was listed in June 2006
- Company to restructure Consumer Brands Division, exit segments

2017 - IKEA TRÅDFRI

The challenge
Smart lighting control is expensive and complex.

The solution
Hire experienced consultants.
Used Open Standards (802.15.4, CoAP, IPSO) and LwM2M.





The result

- Build IKEA Trådfri on state-of-the-art standards.
- Proves that there is no need to ask for permission to the community at large…. nobody knew about this. Found out because of IPSO TLV format:

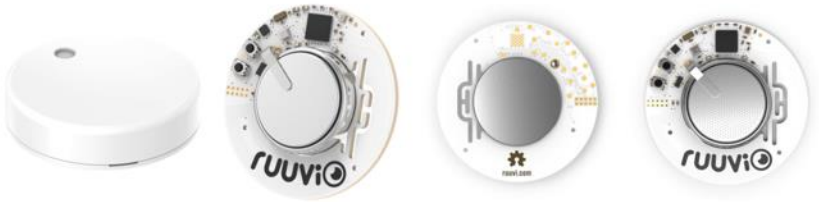
```
{"5850":1,"5851":127,"5707":0,"5708":0,"5709":24930
,"5710":24694,"9003":0,"5711":250,"5706":"f5faf6"}
```

- Runs RIOT too!

2018 – RUUVI TAG

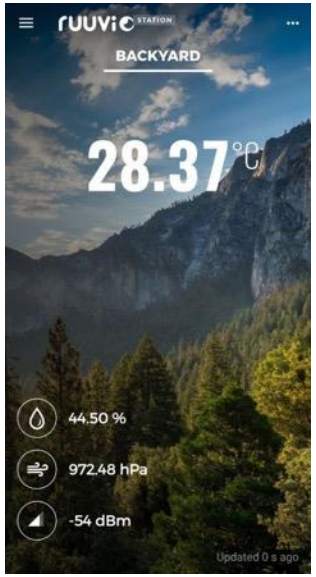
The challenge
Complexity and cost of environmental monitoring. Lack of open source hardware development.

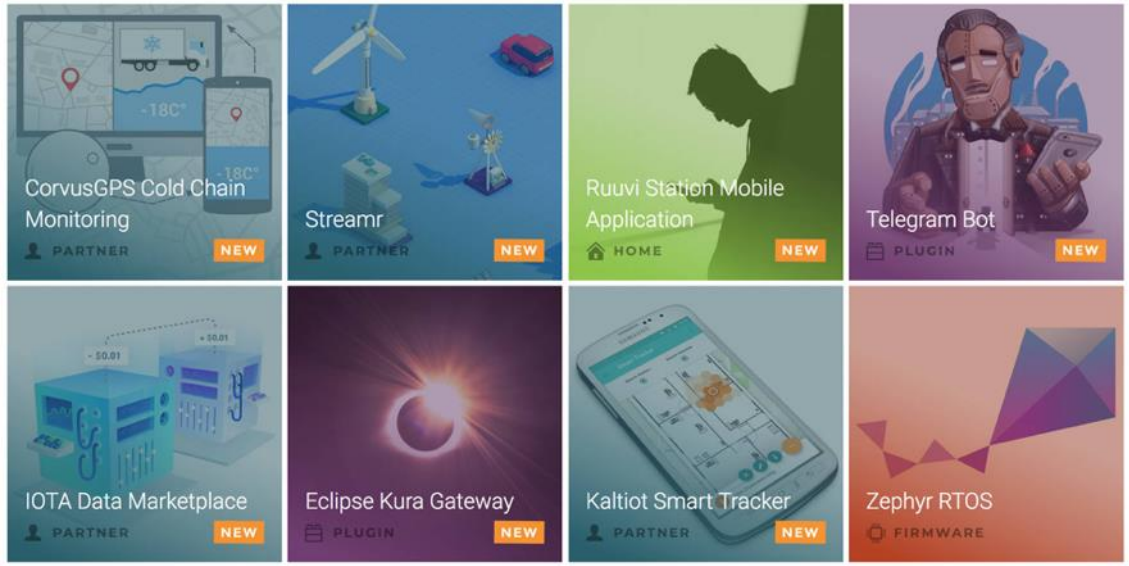
The solution
Create an open-source to avoid future vendor lock-ins, save money and simplify prototyping. Benefit from larger community.



The result

- RuuviTag open-source sensor beacon platform.
- Becomes an generic IoT Platform for telemetry.
- Benefits from its community to get ideas and be used on other applications: Kaltiot tracker, ColdChain monitoring, RIOT, Zephyr…





Q&A

THANK YOU