IoT Engineering
1: Introduction to the Internet of Things

Slides: tmb.gr/iot-1

n|w

Overview

These slides introduce the *Internet of Things*.

Its definition and driving forces behind it.

The main use cases and how it is built.

Hands-on, 5': Defining IoT

What does *Internet of Things* mean to you?

Write down your definition on a post-it.

We will check out the result together.

Internet of Things (IoT)

"Internet-connected computers, with sensors and actuators." — @tamberg

"Physical objects with a Web API." — @hansamann

IoT: "Global network of computers, sensors and actuators, connected through Internet protocols."
Web of Things: "RESTful Web services that measure or manipulate physical properties." — @gsiot

Drivers of IoT

Small, inexpensive, low power computers.

Small, inexpensive, low power sensors.

Short and long range connectivity.

Cloud computing and storage.

Standard (IoT) protocols.

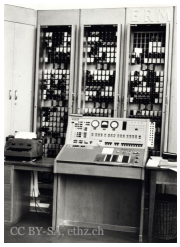
Moore's law

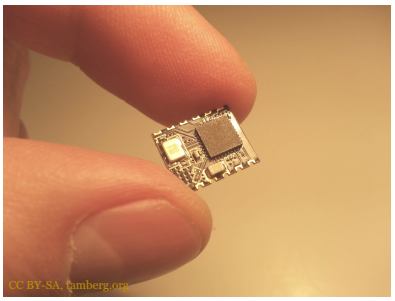
"Moore's law is the observation that the number of transistors in a dense integrated circuit doubles about every two years." — Wikipedia

Gordon Moore, a founder of Intel, noted this in 1965.

=> Computers become more powerful, less expensive.
=> The same power is available in a smaller package.
=> Small computers can be embedded into things.

Moore's law





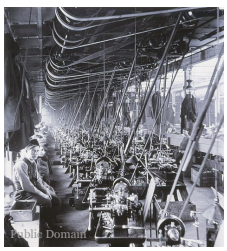
Ubiquitous computing

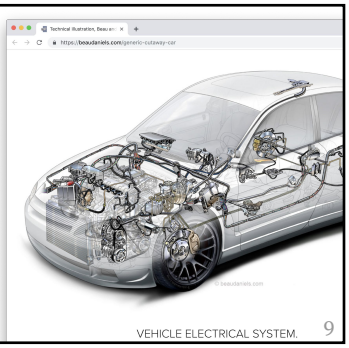
"The idea of integrating computers seamlessly into the world at large [...] *Ubiquitous computing*"

"How do technologies disappear into the background? The vanishing of electric motors may serve as an instructive example"

— Mark Weiser in The Computer for the 21st Century

Motors: 1 vs. n





Connectivity

Ability to communicate with another computer.

Personal area network (PAN, e.g. BLE, Zigbee).

Local area networks (LAN, e.g. Ethernet, Wi-Fi).

Wide area networks (WAN, e.g. 3/4G, LoRaWAN).

The range grows from "room" to "building" to "city" (e.g. BLE, 30m; Wi-Fi, 100m; LoRaWAN, 2-15km).

IoT high level use cases

IoT enables these core use cases, in different flavors.

Efficiency (e.g. trash bins let you know they are full).

Convenience (e.g. remotely preheat a holiday home).

New insights (e.g. a crowdsourced air quality map).

Sectors include connected consumer products, citizen sensing, industrial IoT and many more.

Connected products

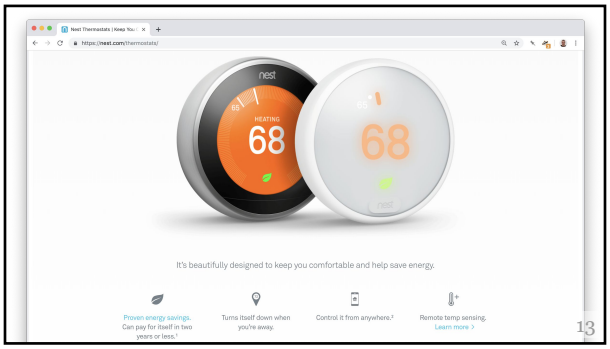
Internet-connected consumer products, e.g.

Nest, a connected, self-learning thermostat.

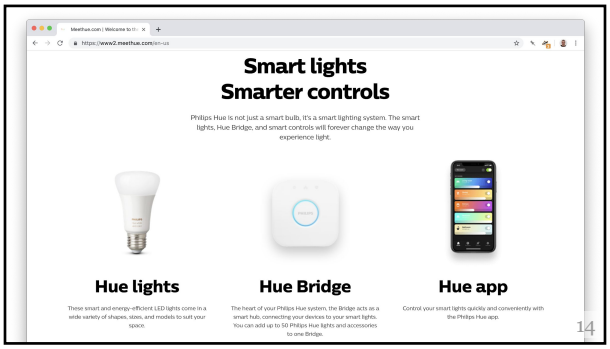
Philips Hue, connected lights with a Web API.

Withings Scale, logs your weight to a dashboard.

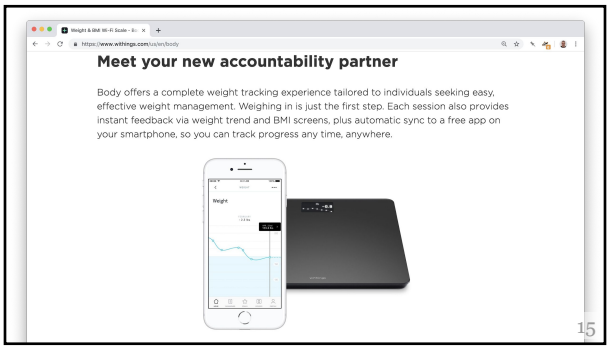
Good Night Lamp, linked lamps to share presence.



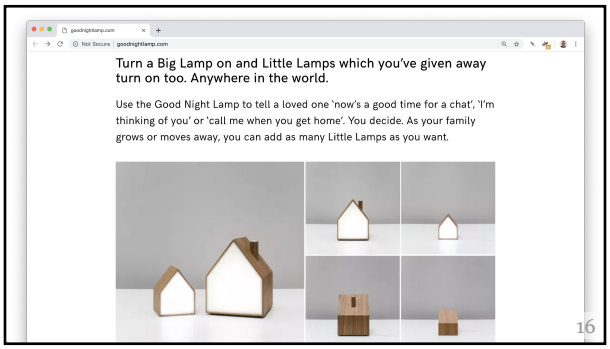
13



14



15



16

Citizen sensing

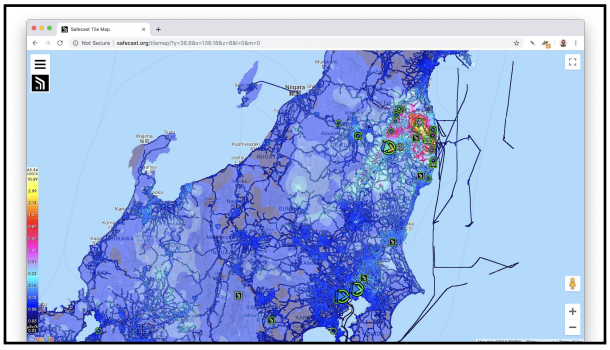
Self-built sensors, open data, nonprofit, e.g.

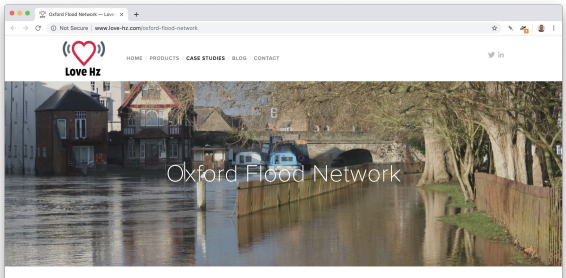
Safecast.org, a crowdsourced radiation map.

Oxford Flood Network, measuring water levels.

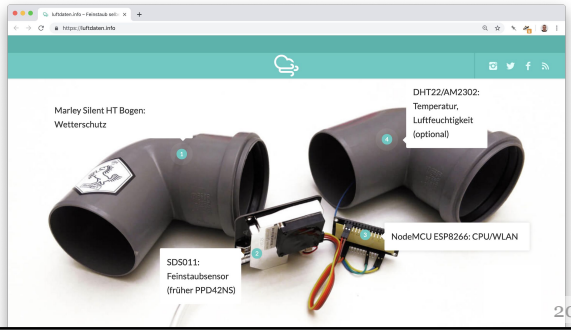
Luftdaten.info, particles and nitrogen oxides map.

17





Citizens, Building a Smart City



Industrial IoT

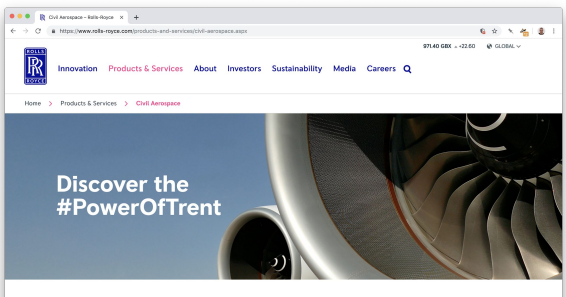
"Industrie 4.0" in German, cyber-physical systems.

E.g. Rolls-Royce TotalCare, "engine as a service".

Predictive maintenance (know what *will* break).

Anomaly detection (find *unknown* issues).

Live feedback (from *deployed* engines).



Latest updates on our Trent engine family

IoT reference model

Device, ~~thing~~, ~~product~~ (with sensors & actuators).

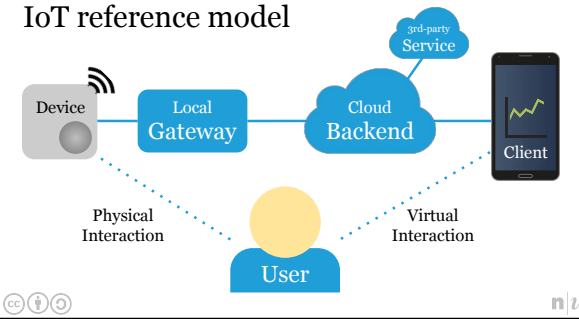
Gateway, ~~hub~~, ~~bridge~~ (in the local network).

Backend (IoT platform, "in the cloud").

Client (app or 3rd-party service).

User (local or remote).

IoT reference model



Device

Embedded computer with sensors and actuators.

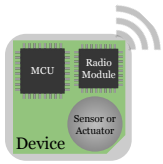
Connectivity on the chip or as an external module.

Microcontroller (MCU) with constrained resources.

Small, slow processor, limited memory, low power.

Often battery powered or harvesting energy.

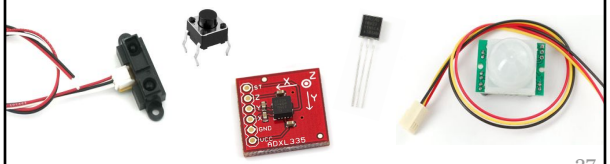
Device



Sensors

Convert physical properties to electrical signals.

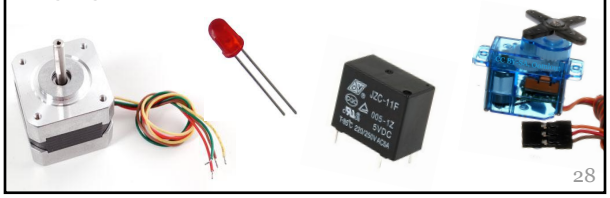
E.g. temperature, sound, light, distance, flow.



Actuators

Convert electrical signals to physical properties.

E.g. light, movement, sound, heat, current.



Gateway

Computer in the local network, with more resources.

Connects local devices/network to the Internet, e.g.

LoRaWAN to Wi-Fi gateway (TTN indoor gateway).

Zigbee to Ethernet gateway (Philips Hue bridge).

Or the Wi-Fi router itself (for Wi-Fi devices).

Transparent, depending on the perspective.

Backend

Backend server(s), service endpoint "in the cloud".

Provides data to clients, receives commands.

High availability, scalability, bandwidth.

Can provide storage or data analysis.

Can call 3rd-party (Web) services.

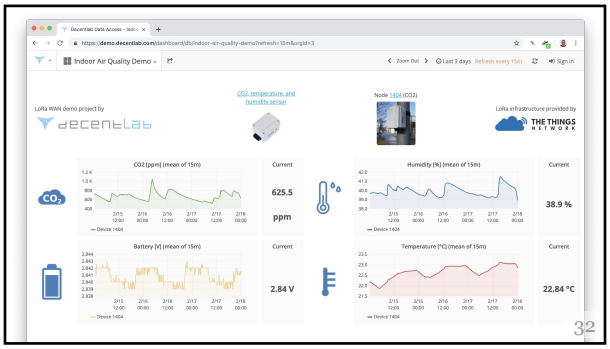
Client

Client app, e.g. dashboard or 3rd-party service client.

Reads measurement data from devices via backend.

Writes control data to the device via backend.

Multiple client apps can share a backend.



Hands-on, 15': Deconstructing IoT

Click here to pick one of the above products/projects.

Nest[1], Hue[2], Withings[3], Good Night Lamp[4], Safecast[5], Luftdaten[6], Oxford Flood Network[7], Rolls-Royce[8].

Draw a reference model of how it works.

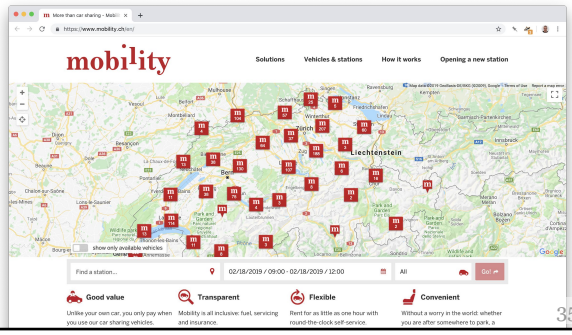
Here is a connected door example.

Be ready to present your result.

Connected product

The entirety of one or more devices, gateways, backends, apps and the services they represent.

E.g. ATM, "money, now".
Philips Hue, "smart lighting".
Kindle, "never be without a book".
Good Night Lamp, "share your presence".
Echo, "control your home, using just your voice".



Architectural Patterns

Systems with various degrees of connectedness.

Physical computing, on device.

App + accessory, local/personal network.

Remote sensing, device via gateway to cloud.

Remote control, cloud via gateway to device.

Edge computing, on edge device/gateway.

Physical computing

On device sensing/control, no connectivity.

Sensor → Device, e.g. logging temperature.

Device → Actuator, e.g. time-triggered buzzer.

Sensor → Device → Actuator, e.g. RFID door lock.

A → B: measurement or control data flow.

App + accessory

Local sensing/control, local connectivity.

Sensor → Device → Client app
E.g. blood sugar measurements.

Actuator ← Device ← Client app
E.g. insulin pump control data.

A → B: measurement or control data flow.

Remote sensing

Sensor → Device → Gateway → Backend → Client

E.g. air quality data via LoRaWAN to shared map.
(LoRaWAN would introduce an additional backend.)

Or machine telemetry via 3/4G to analysis tool client.
(3/4G gateway would be transparent, it's TCP/IP.)

A → B: measurement or control data flow.

Remote control

Client → Backend → Gateway → Device → Actuator

E.g. app sends command via backend to dim a light.

Or a stormy weather service triggers a blind to go up.

Remote sensing and control can be combined.

This is sometimes called "physical mashup".

A → B: control data flow.

Edge computing

Sensor → Device → Edge GW → Device → Actuator.

Variant: Sensor → Edge Device → Actuator.

Use cases: Low latency or big amounts of data.

E.g. cloudless voice recognition unlocks a door.

Or FFT over local machine data, trigger alerts.

A → B: measurement or control data flow.

Important IoT System Qualities

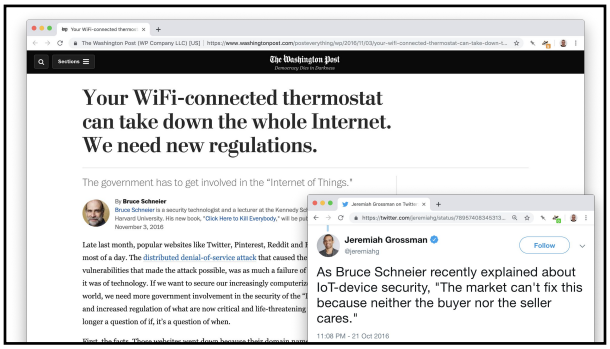
Security, to keep devices, network & backend secure.

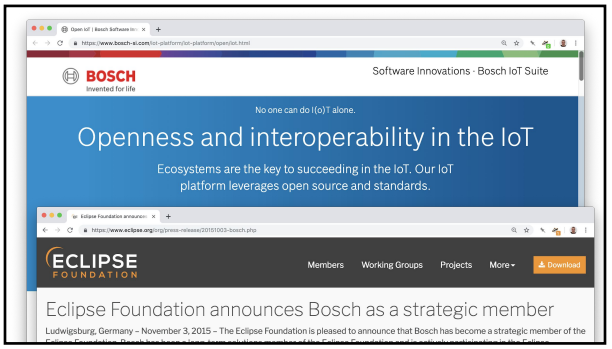
Privacy, to keep people in control of their own data.

Interoperability, to become part of an ecosystem.

Openness, standards & open source build trust.

See, e.g. betteriot.org principles for guidance.





Summary

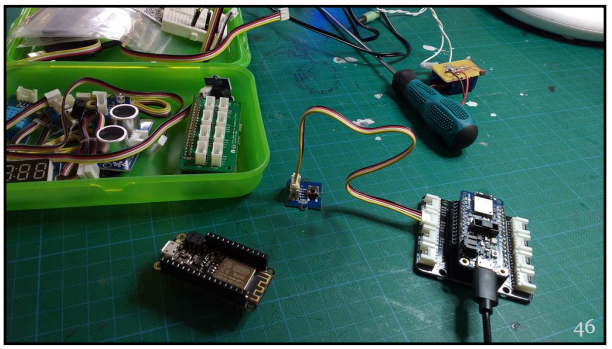
We defined IoT, understood the drivers behind it.

We looked at connected products in three sectors.

We know a simple reference model for IoT systems.

We've seen some patterns with varying connectivity.

Next: Microcontrollers, Sensors & Actuators.



Homework, max. 3h

Install the Arduino IDE and set up microcontrollers:

Check the Wiki entry on Installing the Arduino IDE.

Set up the Feather nRF52840 Express for Arduino.

Set up the Feather Huzzah ESP8266 for Arduino.

And take a first look at the IoT Engineering Wiki.

Feedback or questions?

Write me on https://fhnw-iot.slack.com/

Or email thomas.amberg@fhnw.ch

Thanks for your time.