Seminar on "IoT basics using Docker Containers"

Pietro Manzoni

- Universitat Politecnica de Valencia (UPV)
- Valencia - SPAIN
- pmanzoni@disca.upv.es
- http://grc.webs.upv.es/

Universitat Politècnica de València

- The Universitat Politècnica de València (UPV) is a Spanish public educational institution founded in 1968.
- Its community is made up of around 34,000 students, 3,600 lecturers and researchers and 1,500 administration and services professionals distributed among its three campuses located in Alcoi, Gandia and València.
- The València Campus covers around 840.000 m2 and is almost 2 km long. It is a pedestrian campus with over 123.000 m2 of green areas.
- At present, the UPV comprises 13 university centers, of which nine are higher technical schools, two are faculties and another two, higher polytechnic schools. In addition, it has a Doctoral School and three affiliated centers (Florida University, Berklee College of Music and EDEM Business School).

Overall content

- This course focuses on developing and prototyping IoT applications using Docker containers.

- Containers are a lightweight approach to virtualization that developers can apply to rapidly develop, test, deploy, and update applications at scale.
  - **Docker** is an open platform for container-based virtualization. Docker makes it fast and easy to build containers and deploy them anywhere: in a private or public cloud, within a local VM, or on physical hardware, including IoT devices.

- The approach of the course is mainly practical, presenting the topic of IoT through examples based on Docker containers, integrating services and protocols like LoRAWAN, MQTT, REST, time-series database (e.g., InfluxDB), server agents (e.g., Telegraf), and so on.

- Finally, the course will offer basic concepts on TinyML. TinyML is a rapidly growing area where machine learning technologies and applications, including hardware, algorithms, and software, can perform sensor data analysis on devices with extremely low power consumption. These solutions enable a variety of broader use cases targeting battery-powered devices.

https://grc.webs.upv.es/teaching/corsounipd2022/

- 1. IoT: Basic concepts (slides)
- 2. An introduction to Docker (slides)
- 3. Docker: The basics
- 4. Docker: Building an image
- 5. Docker: Networking with standalone containers
- 6. LoRaWAN: a low power IoT networking technology (slides)
- 7. The TIG Stack for the processing and visualization of data
- 8. MQTT: a pub/sub approach to data sharing (slides)
- 9. Containerized MQTT clients
- 10. Bridging with MQTT brokers
- 11. Docker Swarm to manage multiple containers
- 12. IoT at the edge: TinyML (slides)
- 13. TinyML hands-on examples

How to follow the course

- Create your own Docker ID at https://hub.docker.com/signup
- And…

Option 1:

- You can download and install Docker on multiple platforms.
- Refer to the following link: https://docs.docker.com/get-docker/ and choose the best installation path for you.

Option 2:

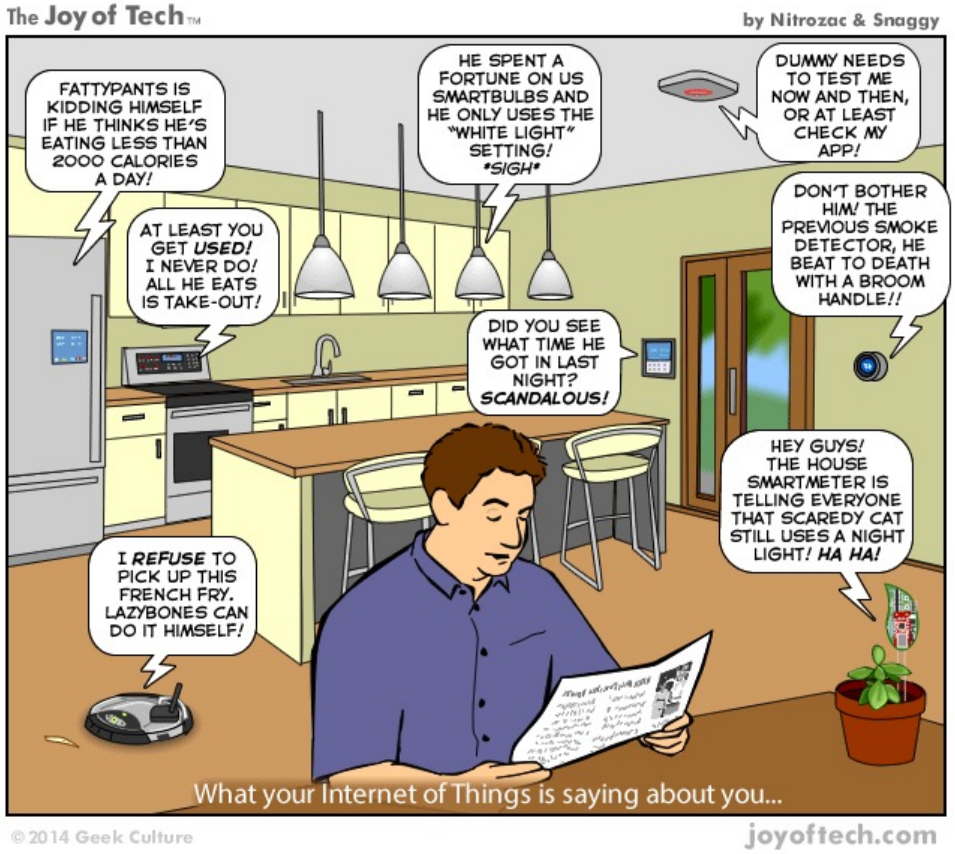
- You can use a virtual machine that you can download here:
  - https://bit.ly/dockervm_2022
- You have to use VirtualBox with the option: "Files / Import virtualized service".
- The VM has user "docker" with password "docker"

Option 3:

- You can execute it online: https://labs.play-with-docker.com/

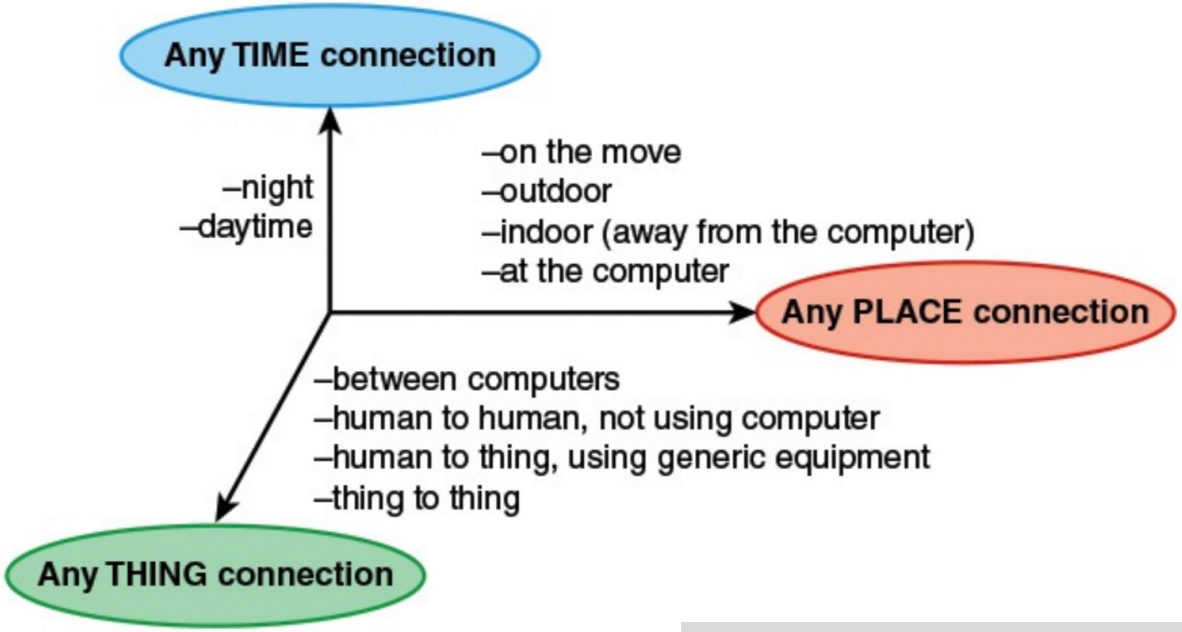
Internet of Things (IoT)

A quick and "physical" definition:
"A network of items—each embedded with sensors—which are connected to the Internet."



Overview of the IoT: a more formal definition

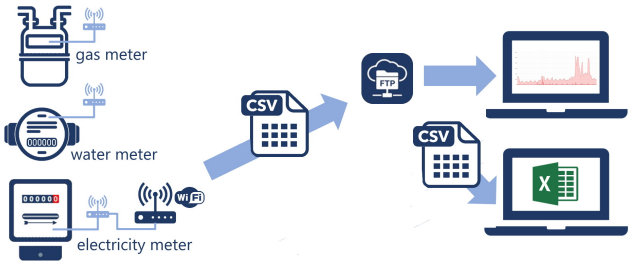
"The IoT can be viewed as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies (ICT)."



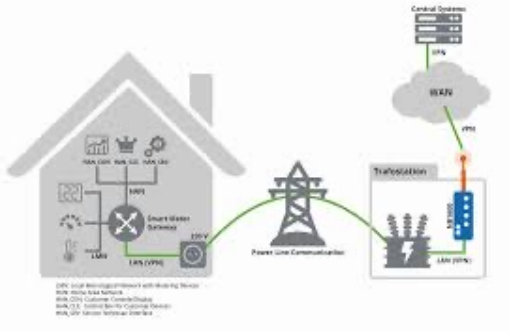
Source: Recommendation ITU-T Y.2060

Machine-to-Machine (M2M)

- The Machine-to-Machine Architecture model proposed by ETSI is considered a predecessor of IoT
- M2M was meant for automated interactions between devices



Smart metering

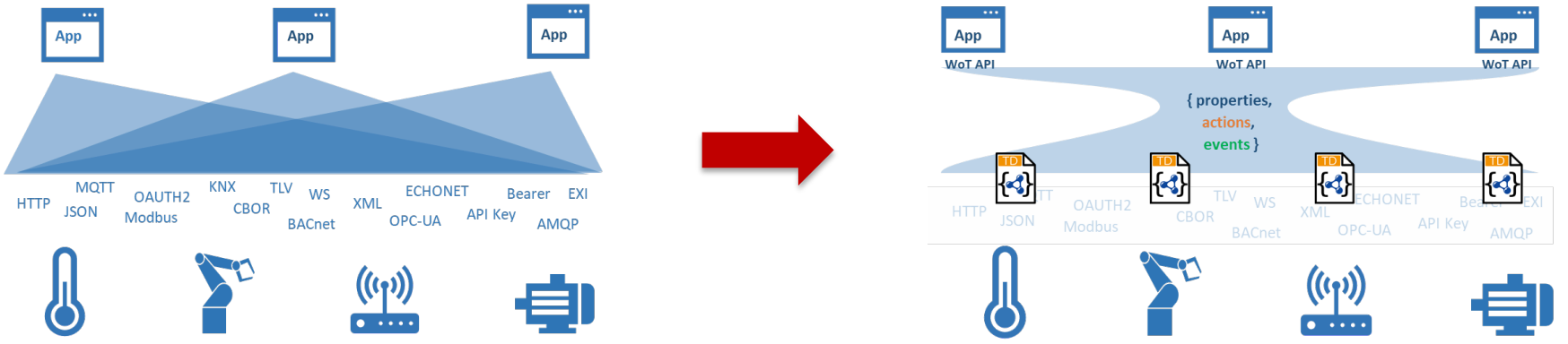


- Currently M2M and IoT are used interchangeably
  - https://www.etsi.org/technologies/internet-of-things
- Basic committees
  - https://www.etsi.org/committee/smartm2m
  - https://www.etsi.org/committee/onem2m

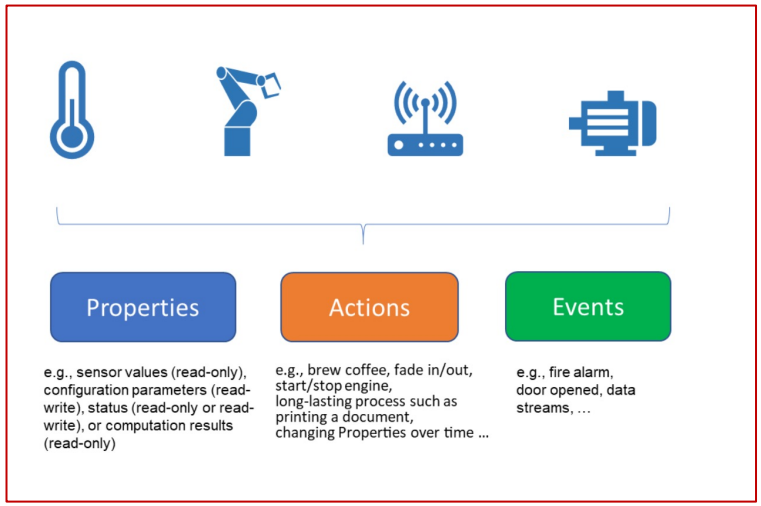
W3C Web of Things

https://www.w3.org/WoT/

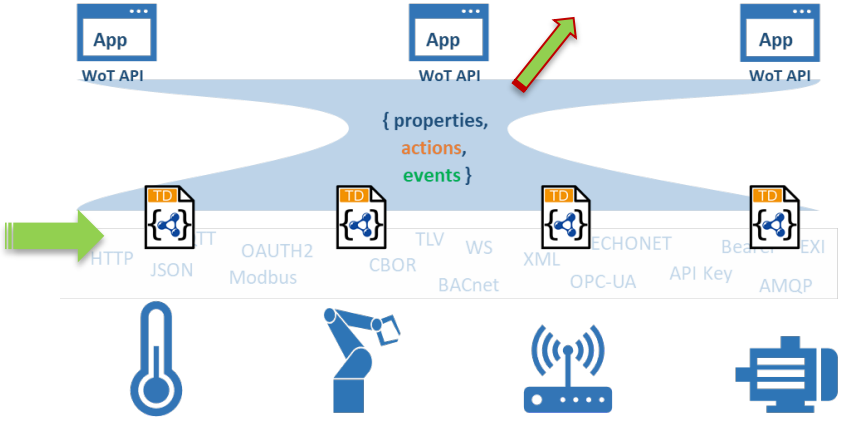
- **"The Web of Things (WoT) is a term used to describe approaches, software architectural styles and programming patterns that allow real-world objects to be part of the World Wide Web."**
- The Web of Things (WoT) tries to avoid the fragmentation of the IoT simplifying integration across IoT platforms and application domains by using and extending existing, standardized Web technologies.
- From the developer's perspective, the WoT enables access and control over IoT resources and applications using mainstream web technologies (such as **HTML 5.0, JavaScript, Ajax, PHP, Ruby n Rails**, etc)
- The approach to building WoT is therefore based on RESTful principles and REST API s, which enable s both developers and deployers to benefit from the popularity and maturity of web technologies.



W3C Web of Things



- An IoT device's metadata, including all information needed to enable this common abstraction, is documented in what is called **WoT Thing Description (TD)**.
- The TD is a central building block in the W3C Web of Things and can be considered as the entry point of an IoT instance (much like the index.html of a Web site).
- It provides information on which data and functions are provided, which protocol is used, how data is encoded and structured, and security mechanism is used to control access, and further machine-readable and human-readable metadata.
- A TD is expressed in JSON-LD and can be provided by an IoT device itself or hosted externally in a repository such as a TD Directory.



- In general, WoT is a protocol agnostic approach and provides a common mechanism to define how specific protocols such as MQTT, HTTP, CoAP or Modbus can be mapped to the WoT's interaction properties-action-event abstraction.
- This mapping and protocol specific metadata are provided by the **WoT Binding Templates**. A binding template for a specific protocol provides a guideline how a client can activate each WoT interaction abstraction through a corresponding network-facing interface for that protocol.

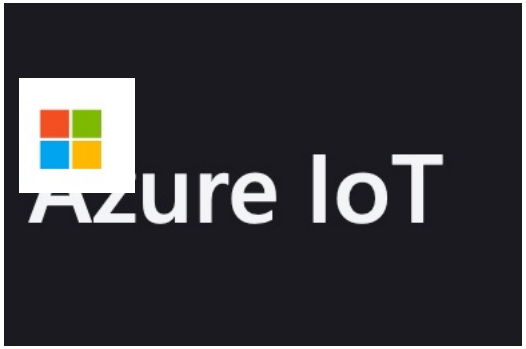
All big companies are active in this area

https://iot.telefonica.com/en/

https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html

https://cloud.google.com/solutions/iot/

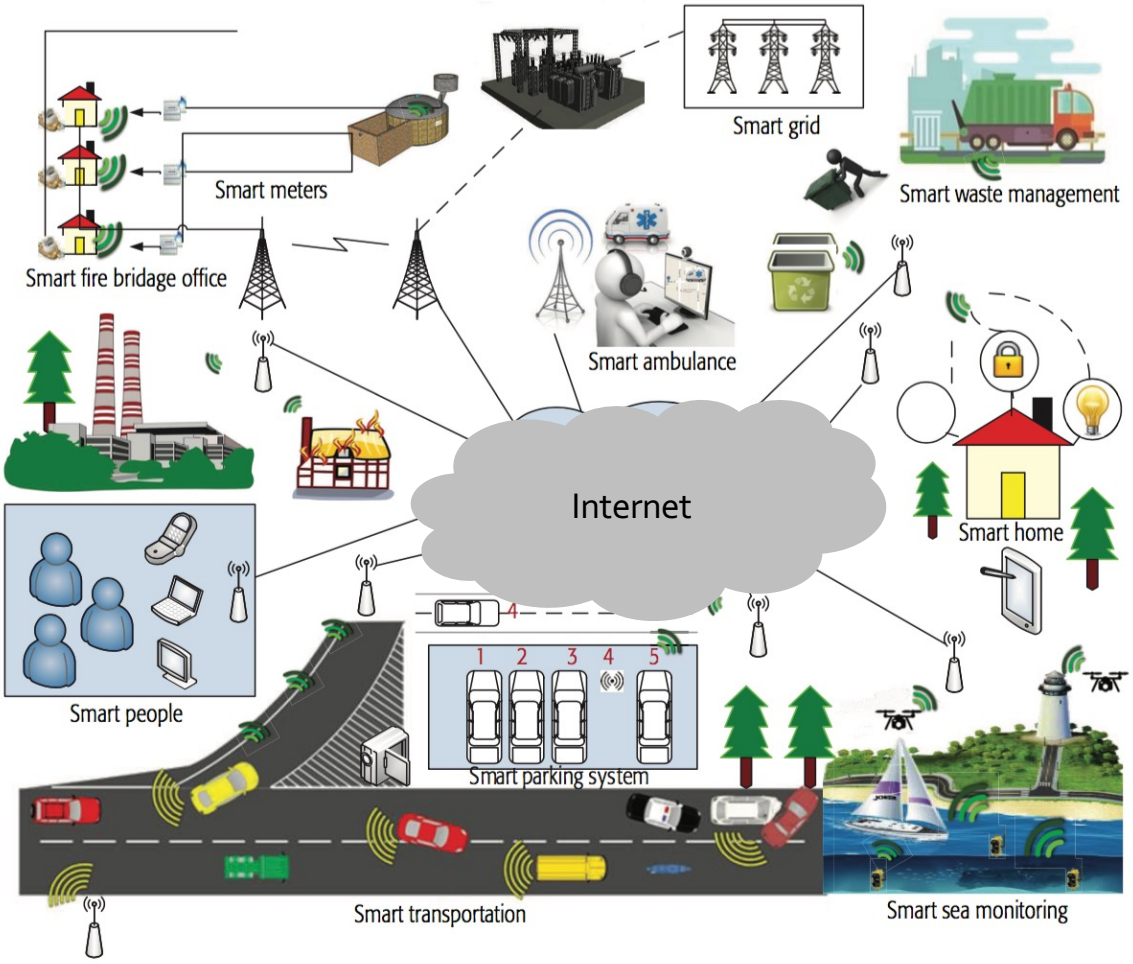
https://www.ibm.com/cloud/internet-of-things



https://azure.microsoft.com/es-es/overview/iot/#overview

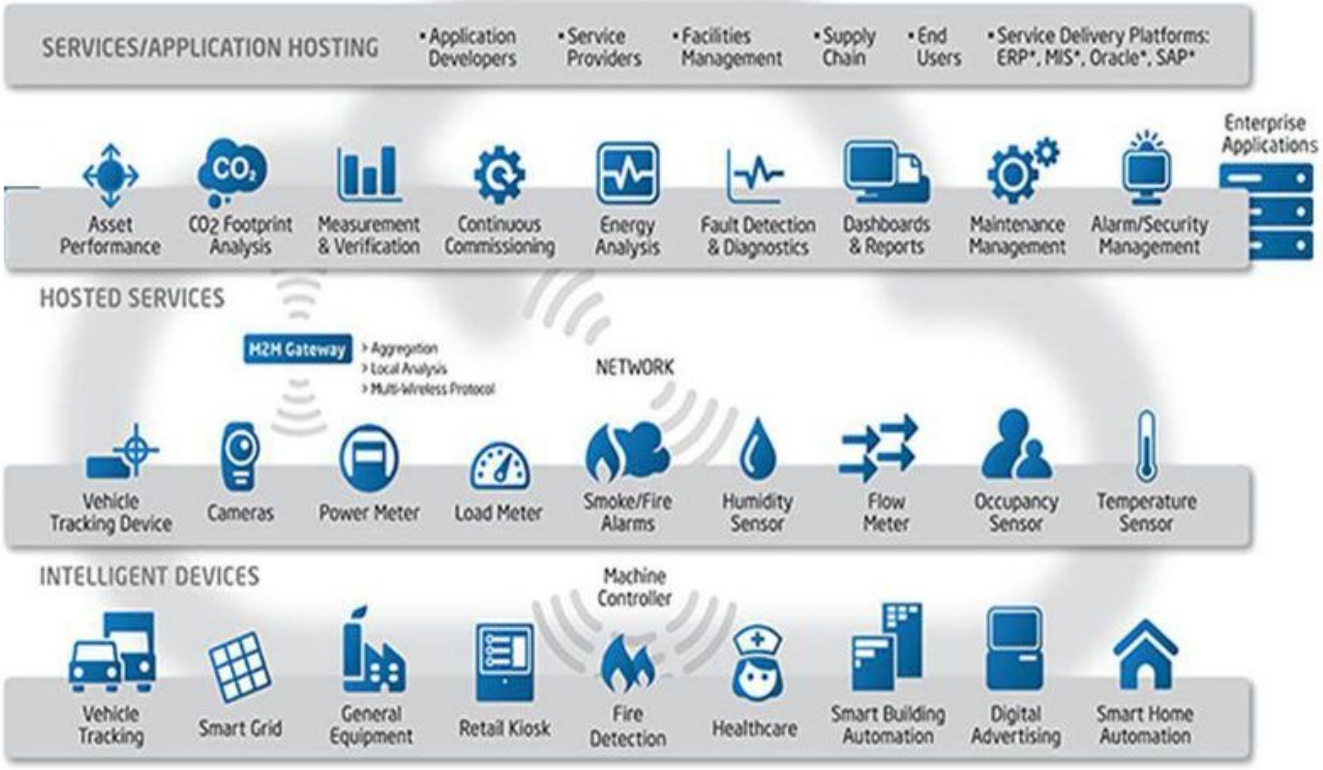
https://www.samsung.com/cl/smartthings/

Applications… smarter everything

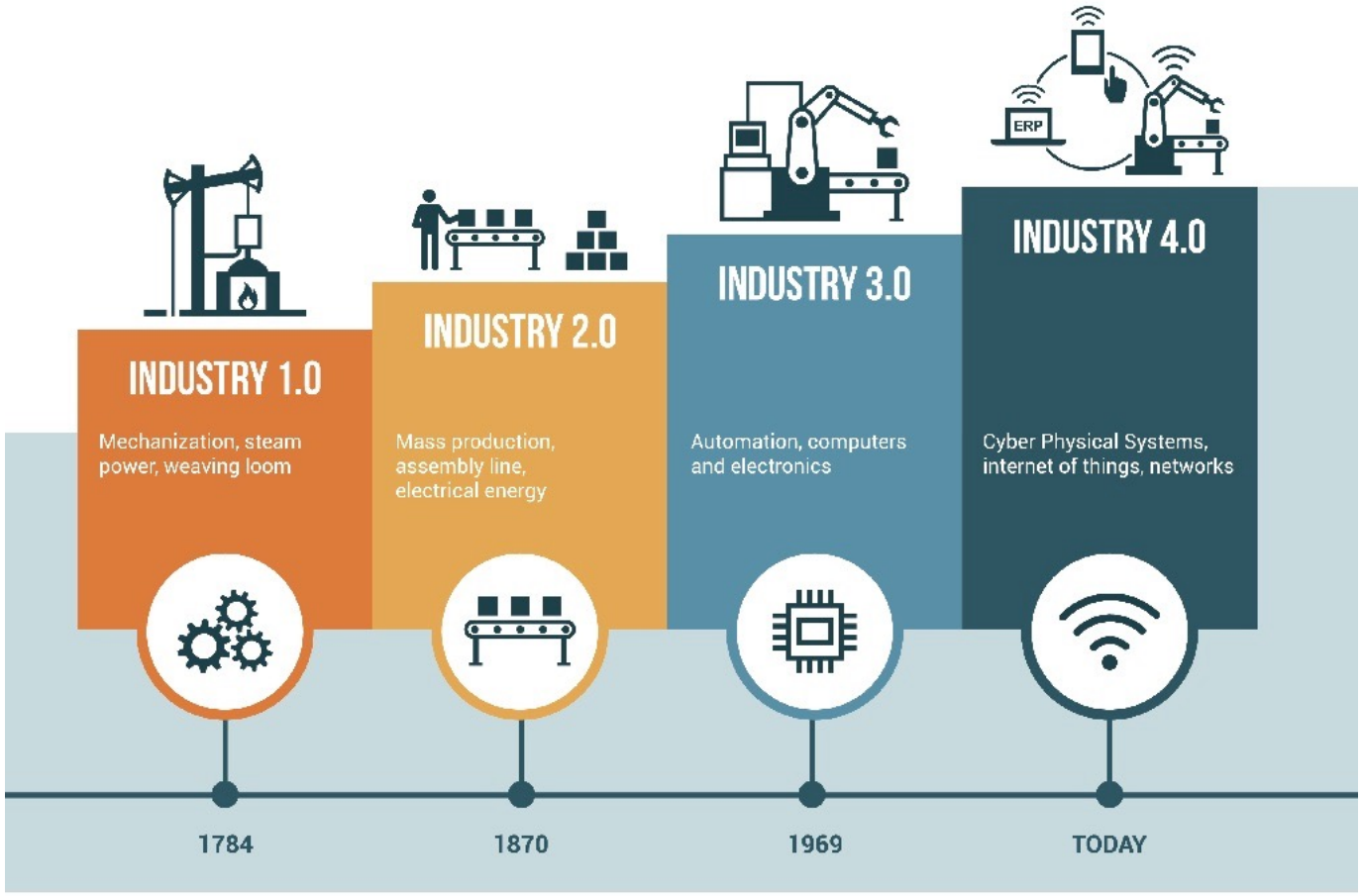


Enabling Communication Technologies for Smart Cities, IEEE Communications Magazine - January 2017

Applications: various levels

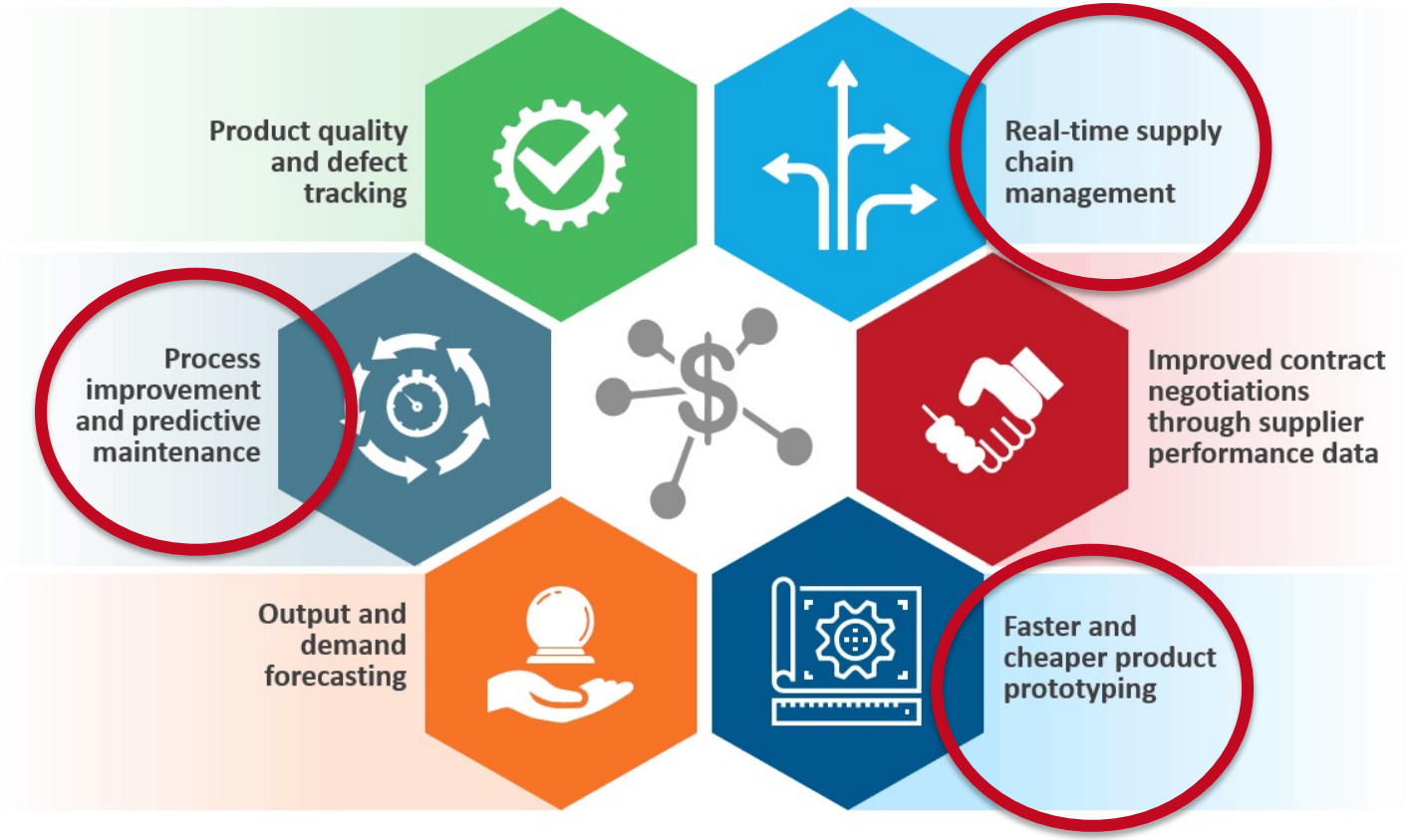


Industry 4.0 and Industrial IoT



How Industry 4.0 is Helping Manufacturers

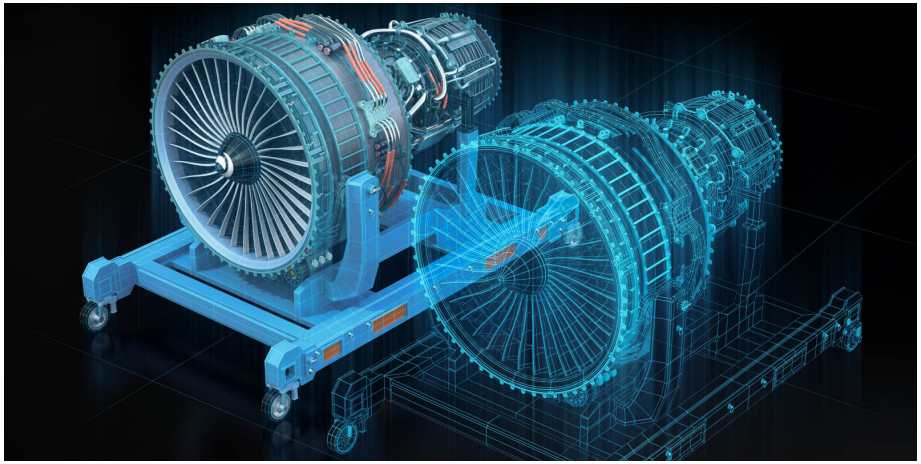
Industry 4.0 – automation and data exchange in manufacturing technologies – is helping manufacturers to achieve their goals of reducing cost and increasing profitability through improvements and optimization across the value chain



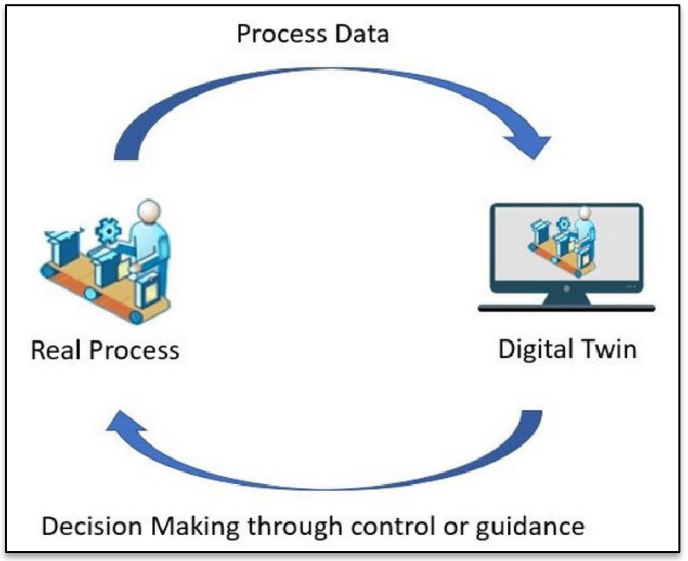
Everest Group® Enterprise Digital Adoption in Manufacturing | Pinnacle Model™ Assessment 2018

Industrial IoT example

- **Rolls-Royce TotalCare**, "engine as a service".
  - Predictive maintenance (**know what will break**).
  - Anomaly detection (**find unknown issues**).
  - Live feedback (**from deployed engines**).

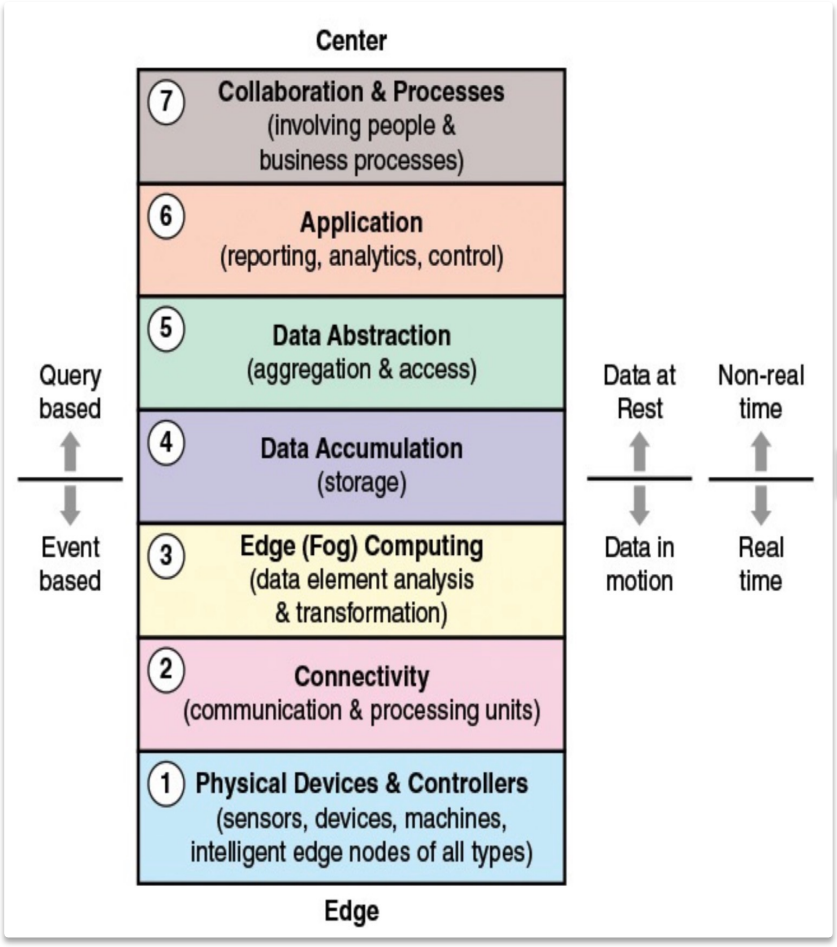


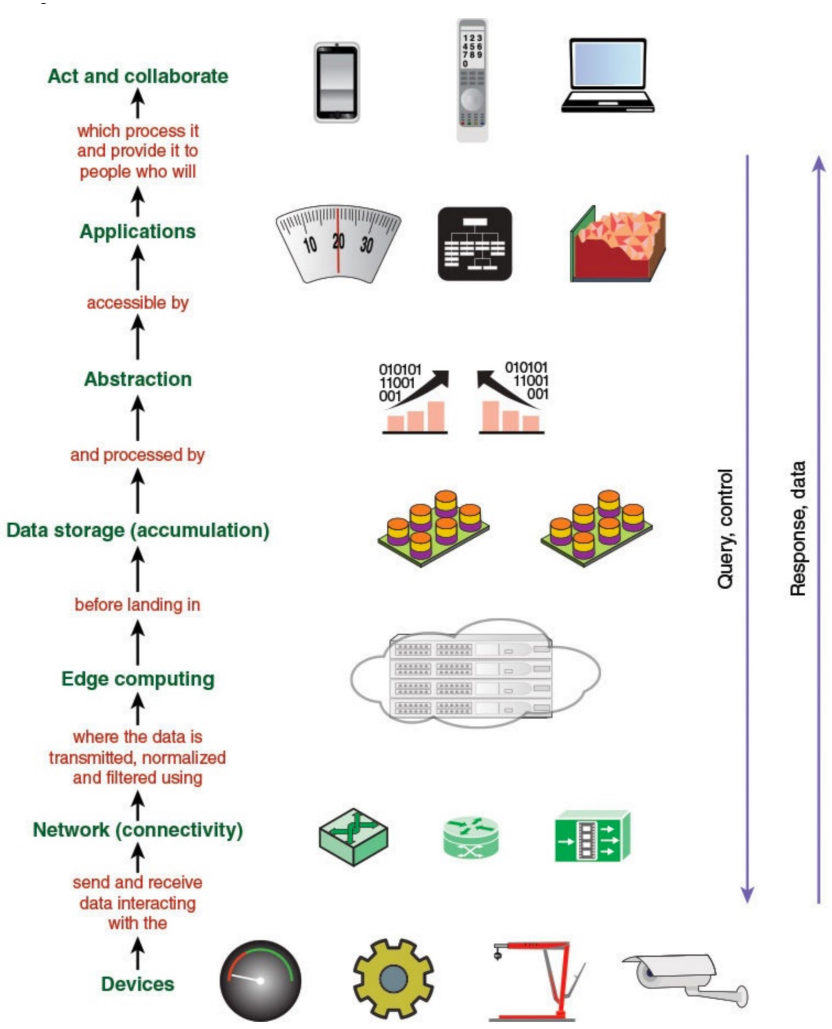
Digital Twins



Reference Model

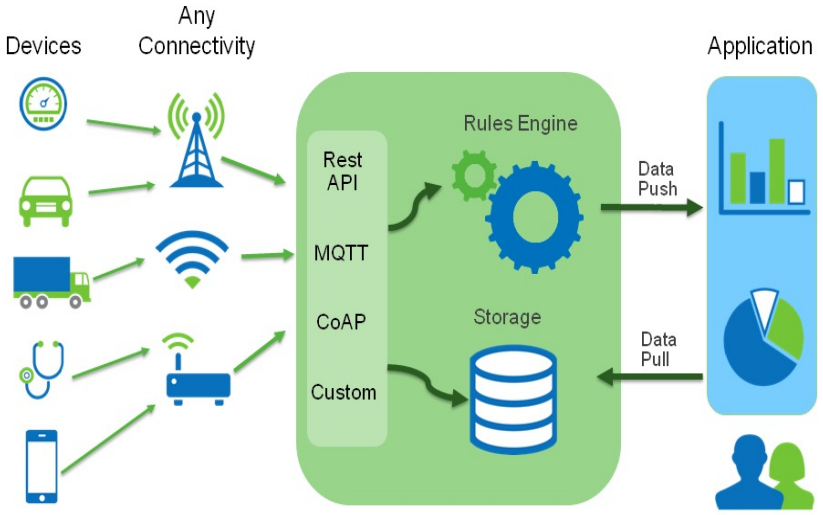
Internet of Things World Forum





Simplified model

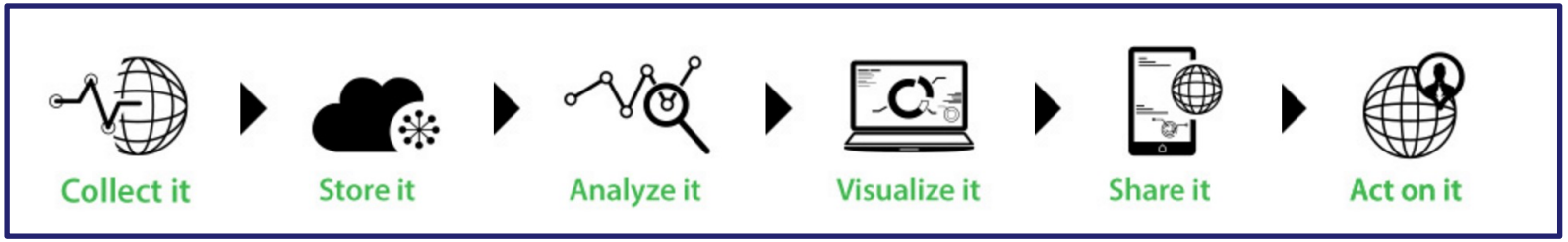
- Devices ("things")
  - These could be sensors, actuators, robots, cars, whatever can be connected.
  - A lot of inheritance from the world of "sensors networks"
- Connectivity
  - To connect things reliably to Internet.
  - Wireless connectivity is central to this task
- Platform
  - the collected data needs to be stored and processed somewhere. Typically cloud-based infrastructures



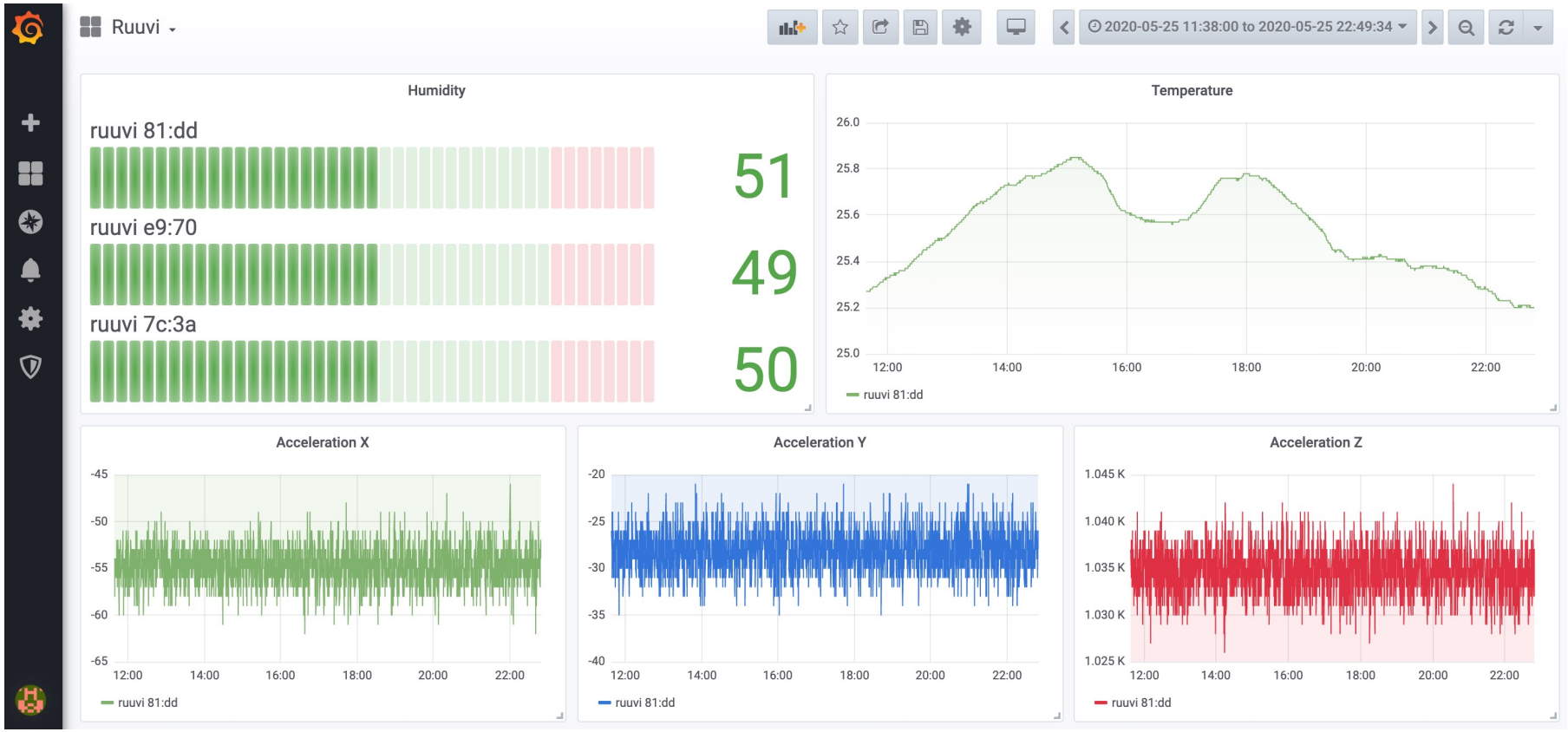
Platforms

Platforms

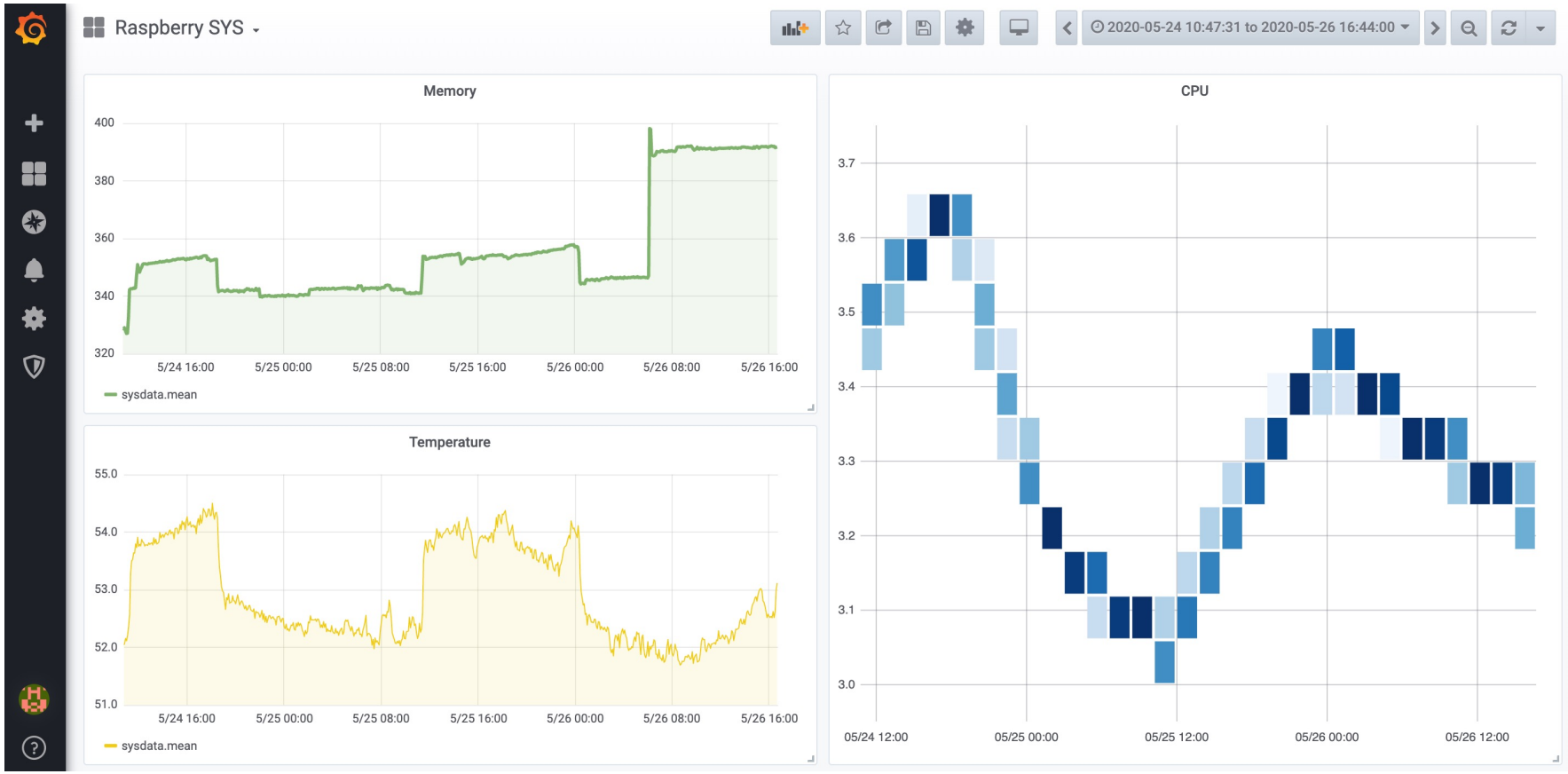
- Amazon Web Services
- Google Cloud IoT
  - Firebase
- Microsoft Azure IoT Suite
- Salesforce IoT
- Oracle Internet of Things
- Cisco IoT Cloud Connect
- IBM Watson Internet of Things
- FIWARE
- Ubidots
- ThingSpeak
  - Based on MATLAB
- ThingsBoard
- GroveStreams
- SensorCloud
- …



Data visualization & analytics: an example

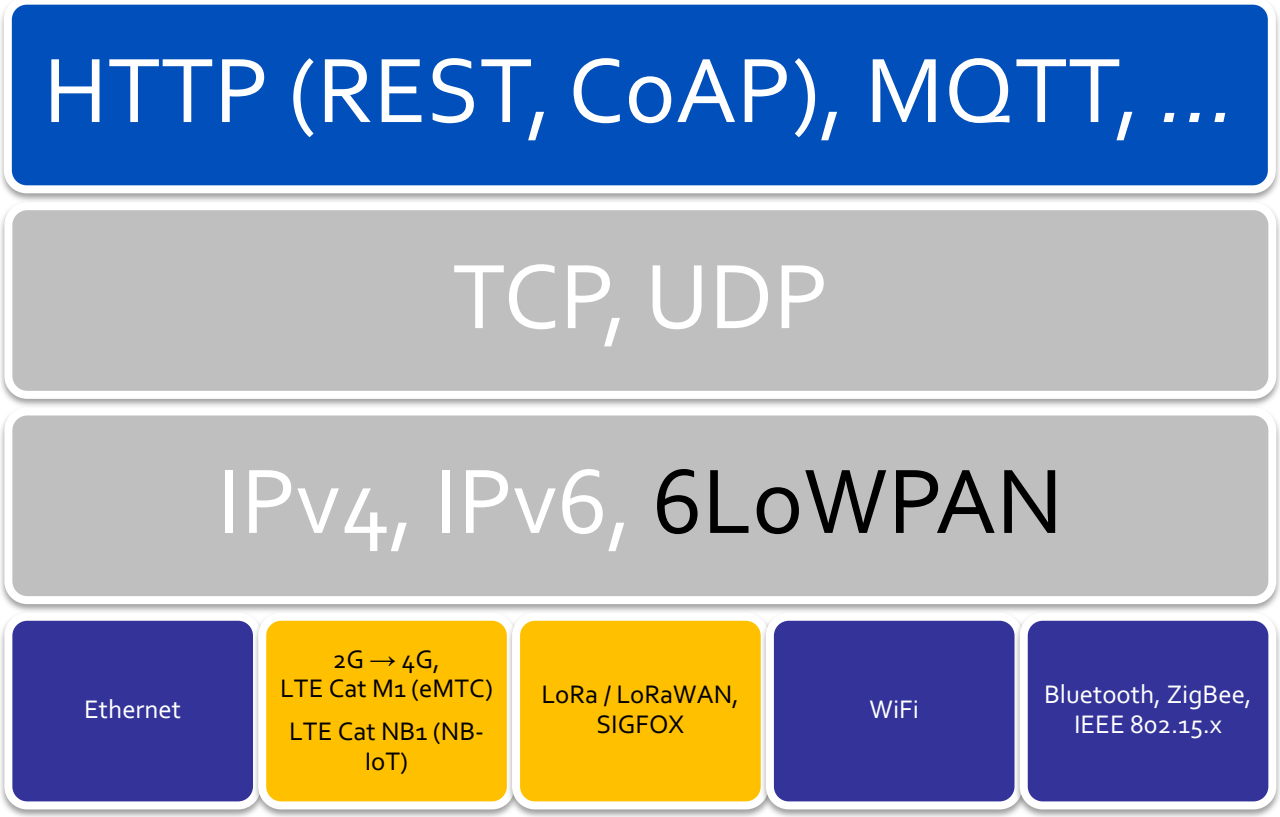


Data visualization & analytics: an example



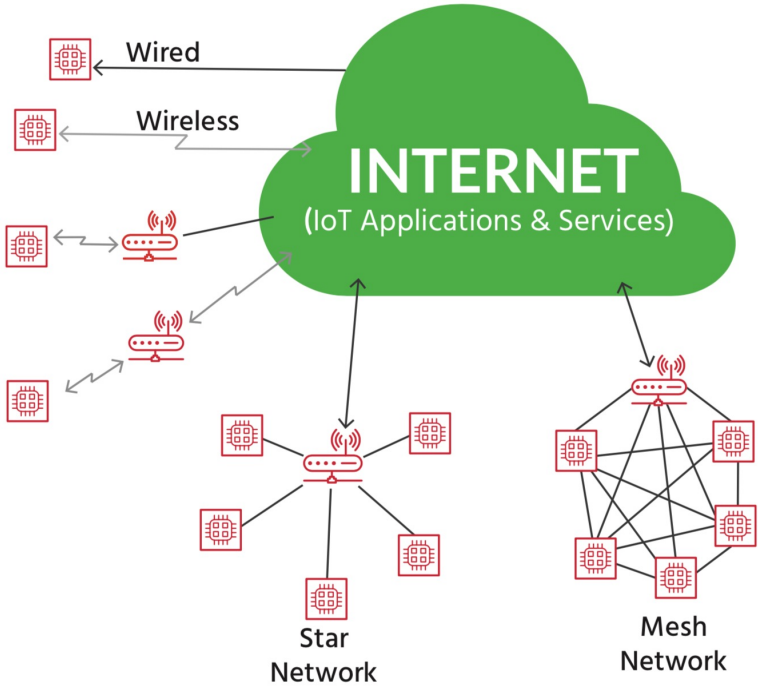
Connectivity

Connectivity

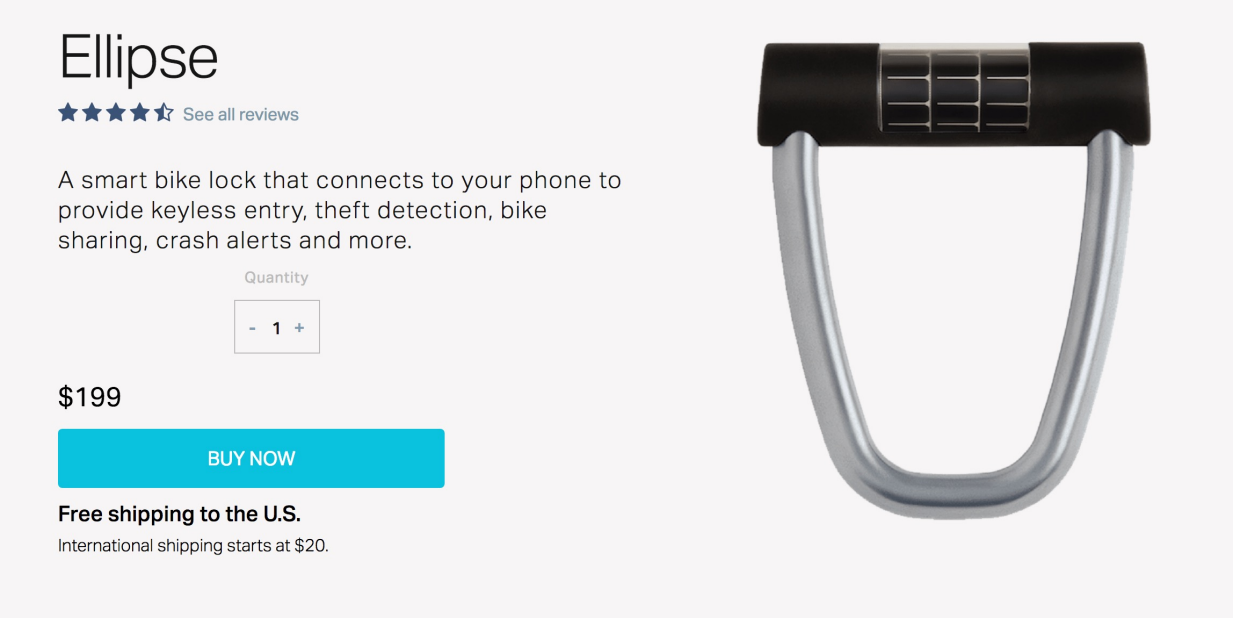


The 4 communication models

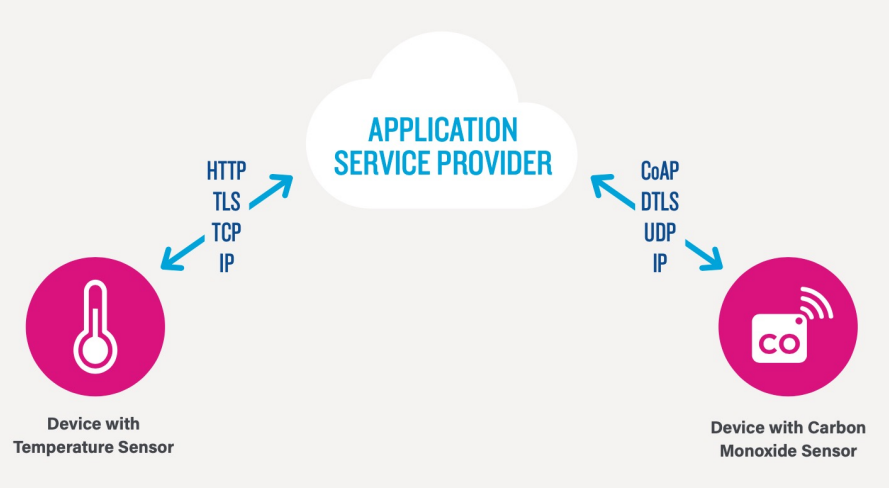
- Internet Architecture Board (IAB) released (2015) a guiding architectural document for networking of smart objects (RFC 7452) which outlines a framework of four common communication models used by IoT devices.
  1. Device-to-Device
  2. Device-to-Cloud
  3. Device-to-Gateway
  4. Back-End Data Sharing
  - It is possible that more than one pattern can be applied at the same time in a product.



Device-to-Device Communication Pattern



Device-to-Cloud Communication Pattern



BLOOM Home Control insecticida volador eléctrico líquido controlable desde el móvil aparato + 1 recambio

35,69 € (35,69 € / Unidad)

UNIDADES

\- 1 +

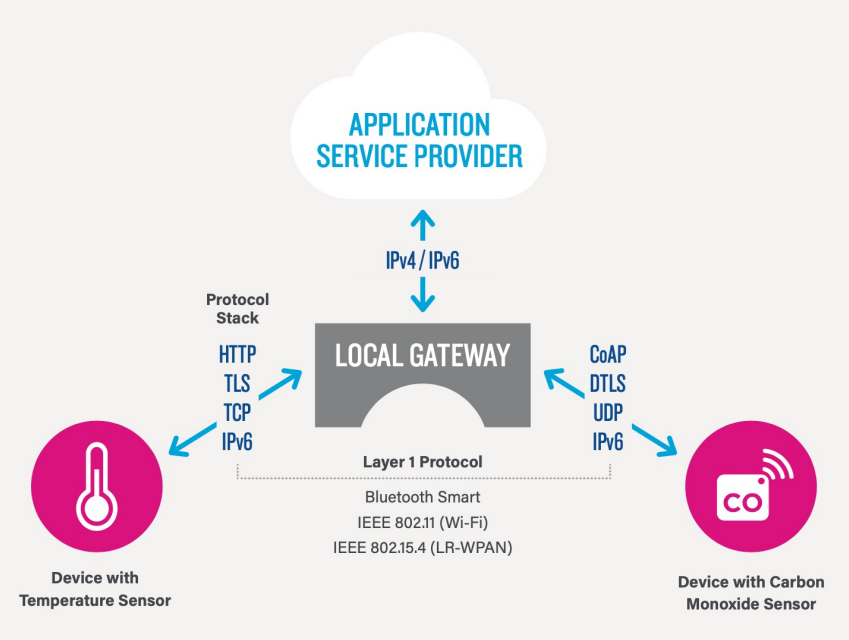
AÑADIR AL CARRO

Información general

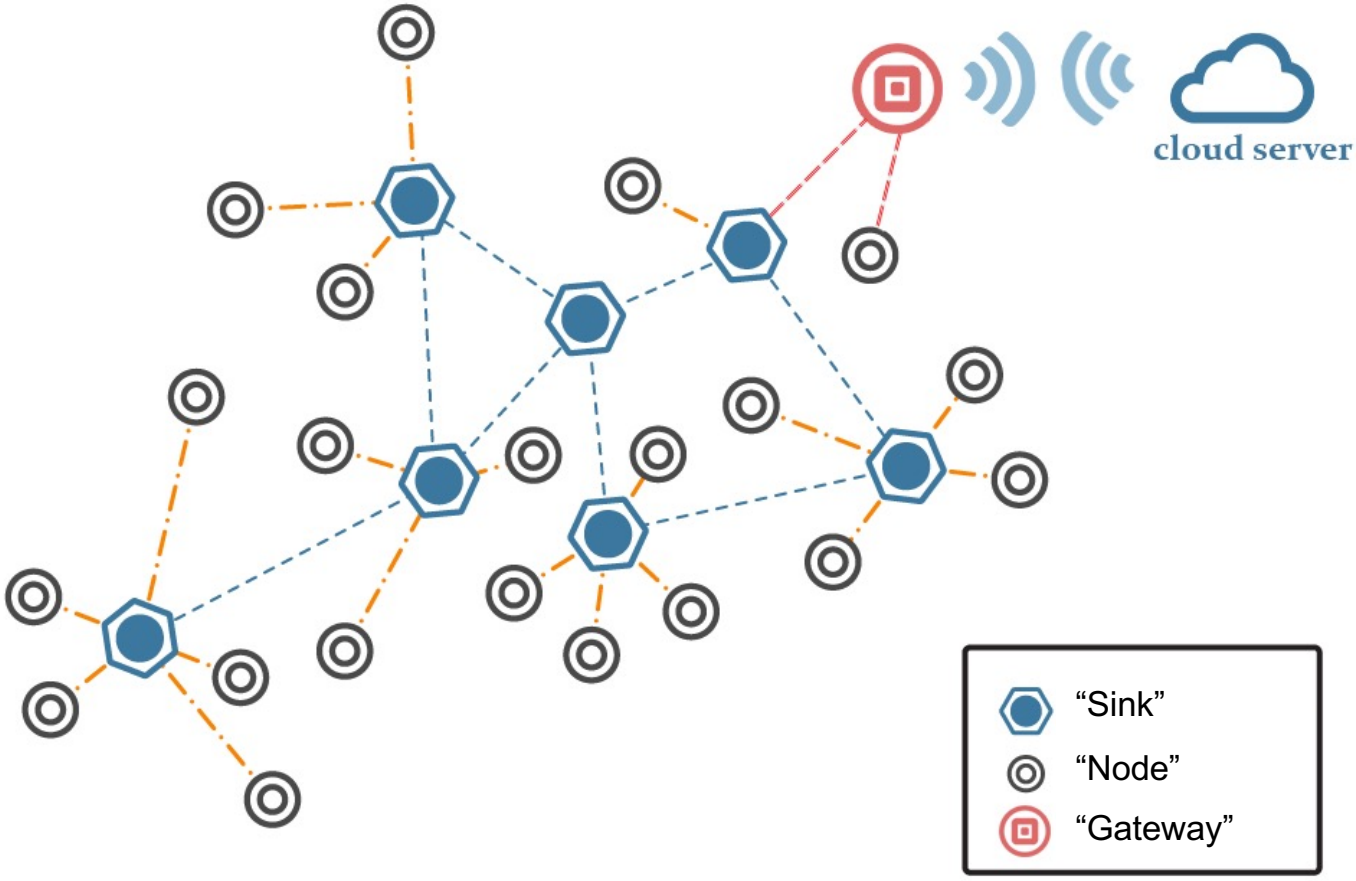
Bloom Home Control es un insecticida eléctrico líquido con la fórmula más avanzada para protegeros a ti y a tu familia de los mosquitos común y tigre.

- Programable según tus necesidades.
- Se conecta y controla a distancia a través de tu Smartphone (aplicación disponible para Android & iOS). Cuando lo necesites y desde donde quieras
- Control de la intensidad: Normal y Max.
- Modo Inteligente: este modo ajustará la intensidad y duración de forma automática para ti dependiendo del tamaño de la habitación, hábitos de uso y la intensidad de mosquitos según AccuWeather.
- Recordatorio para comprar un recambio. Recibe una notificación automática cuando el contenido líquido esté por debajo del 20%.

Device-to-Gateway Communication Pattern

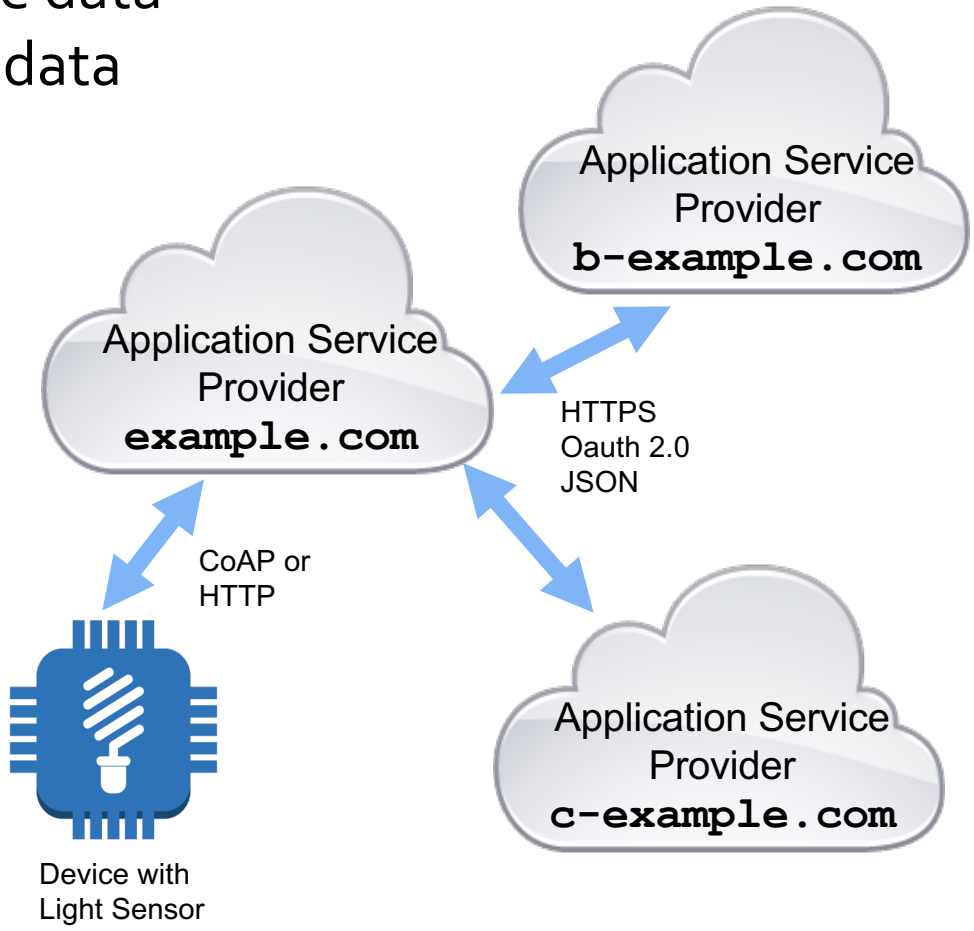


Device-to-Gateway: More complex configurations



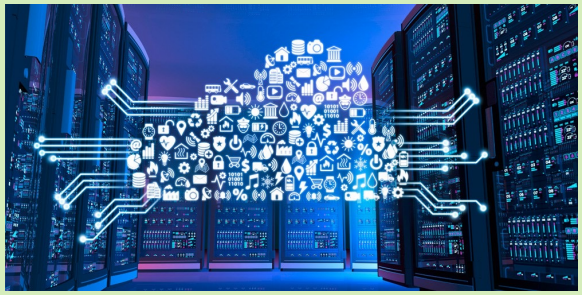
Back-End Data Sharing

- Provides the possibility to export and to analyse data in combination with data from other sources.

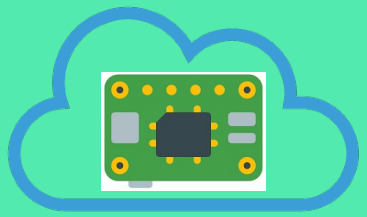


Edge computing

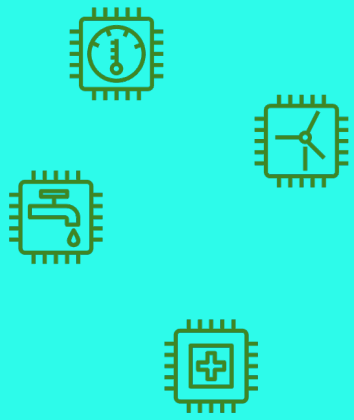
Core cloud



Edge Node

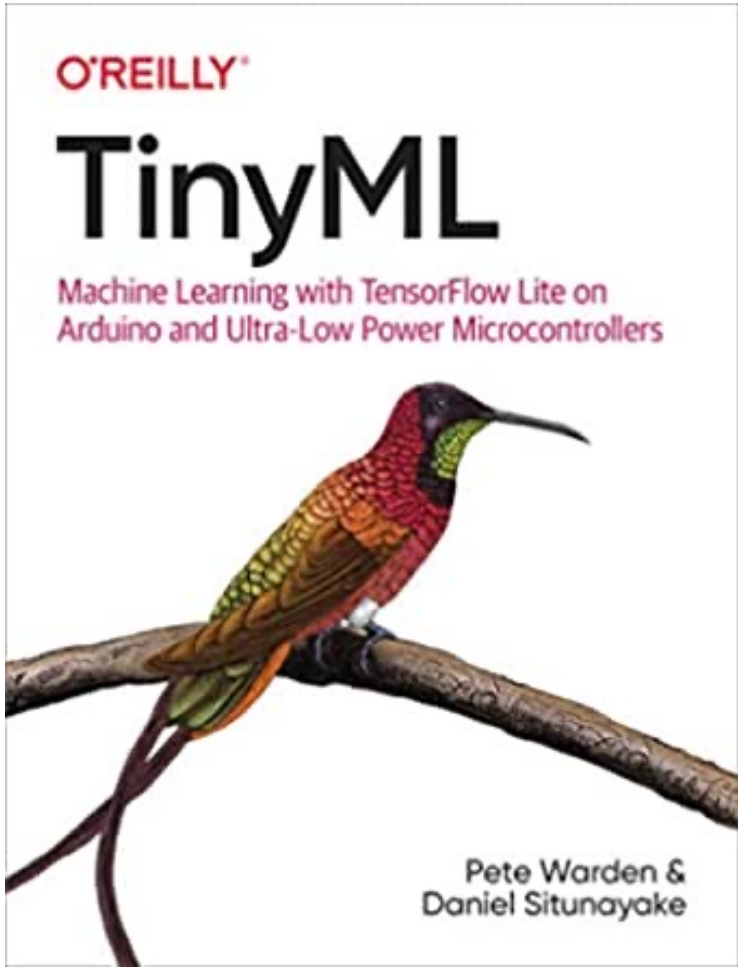


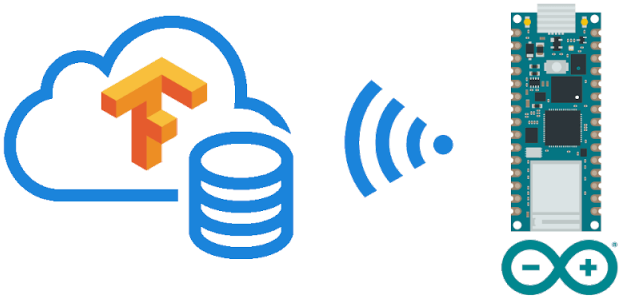
Edge devices

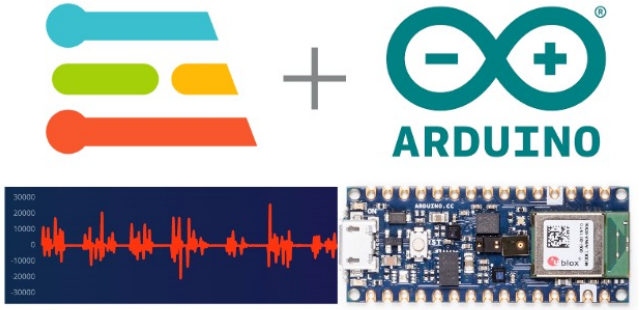


Edge Computing → TinyML

TinyML

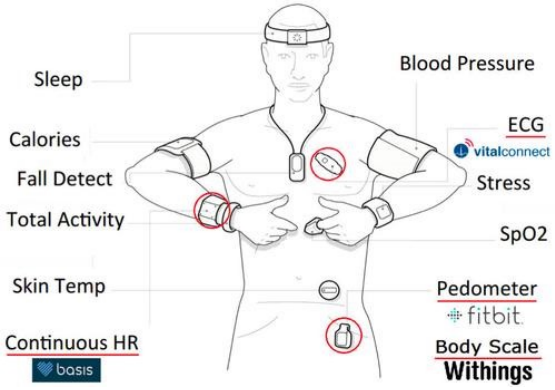

O'REILLY
TinyML
Machine Learning with TensorFlow Lite on Arduino and Ultra-Low Power Microcontrollers
Pete Warden &
Daniel Situnayake




ARDUINO

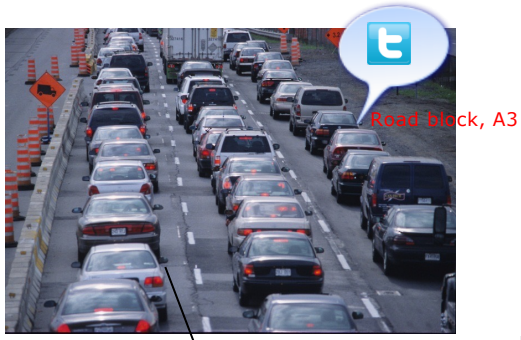
Things

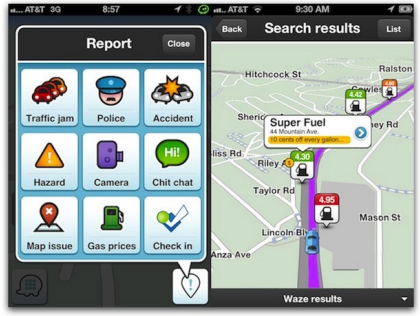
Examples of Things



Beyond conventional things/sensors

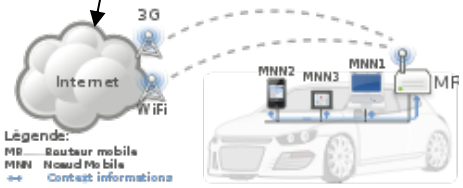
- Human as a sensor (citizen sensors)
  - e.g. tweeting real world data and/or events
- Virtual (software) sensors
  - e.g. Software agents/services generating/representing data





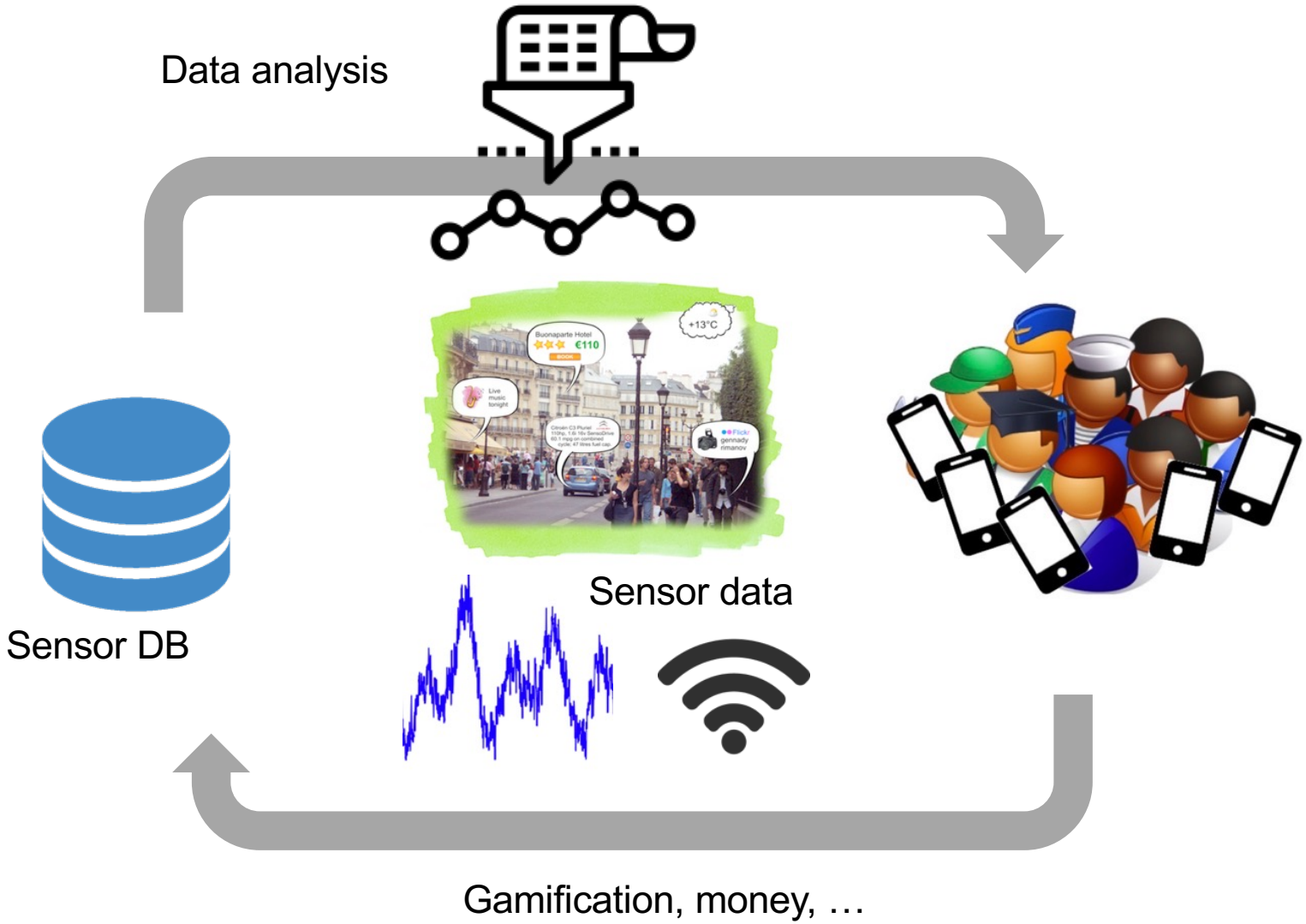
https://www.waze.com

Road block, A3

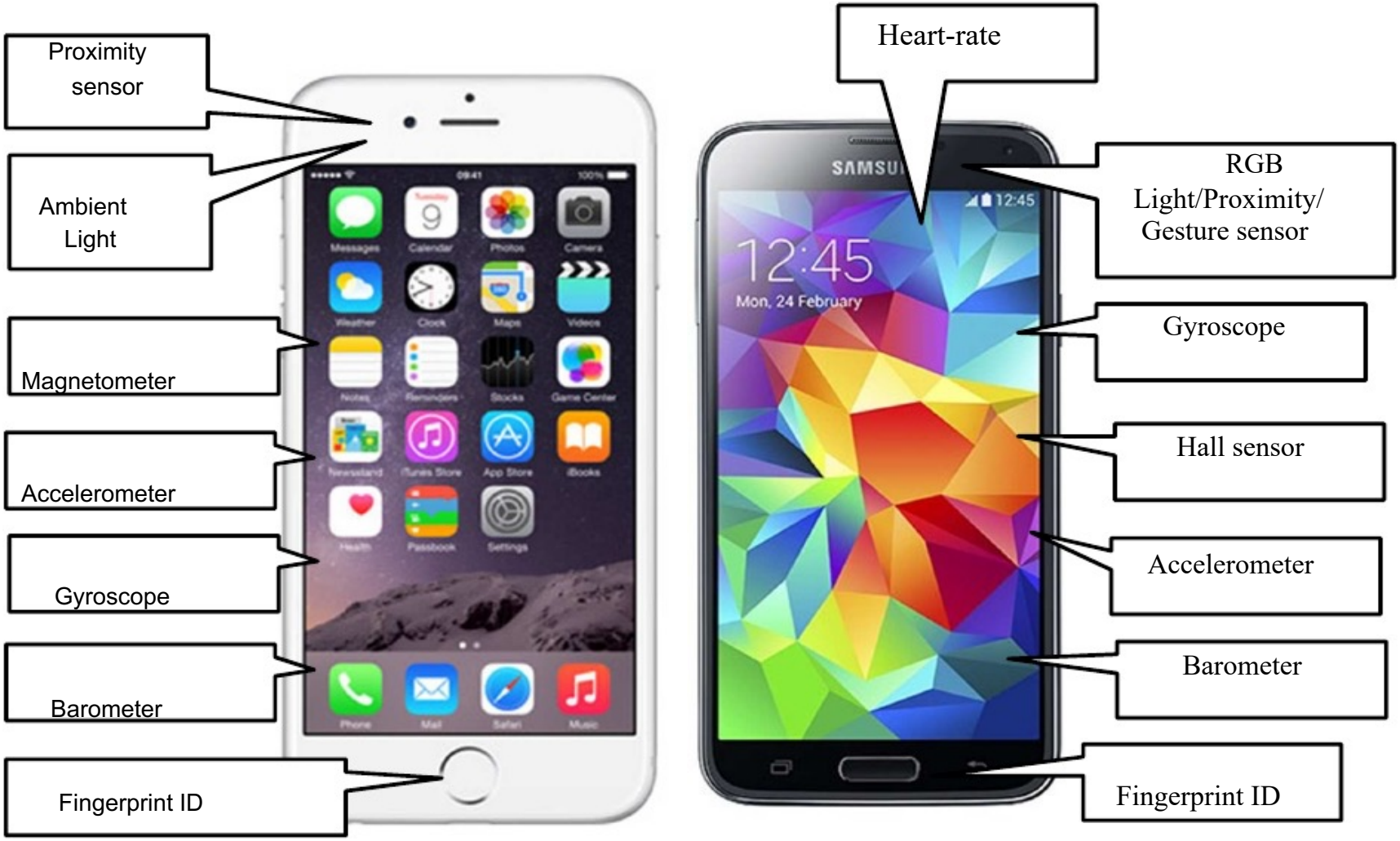


Suggest a different route

Crowdsensing

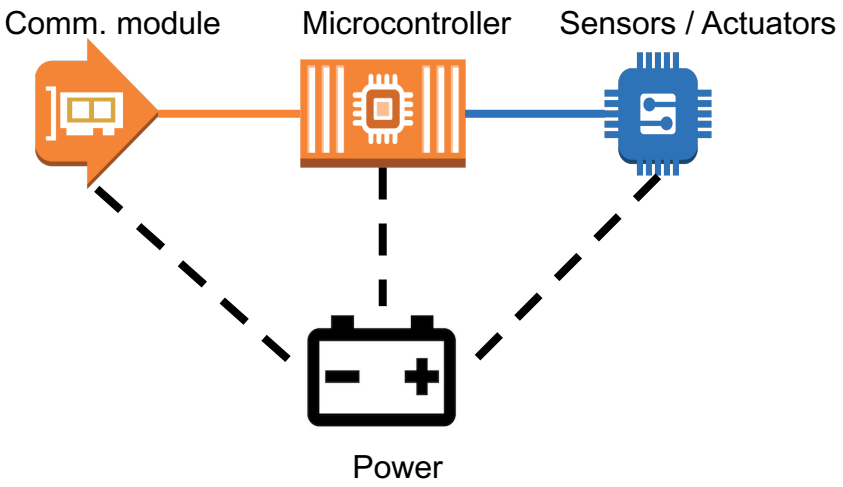


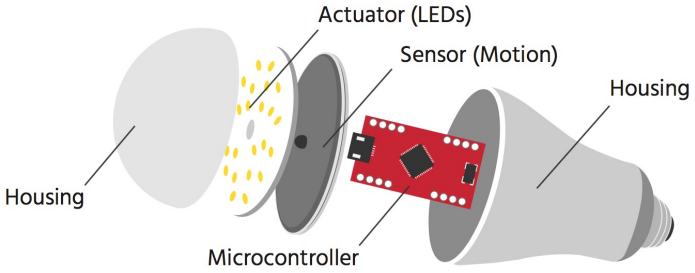
Sensors in Modern Smart Phones



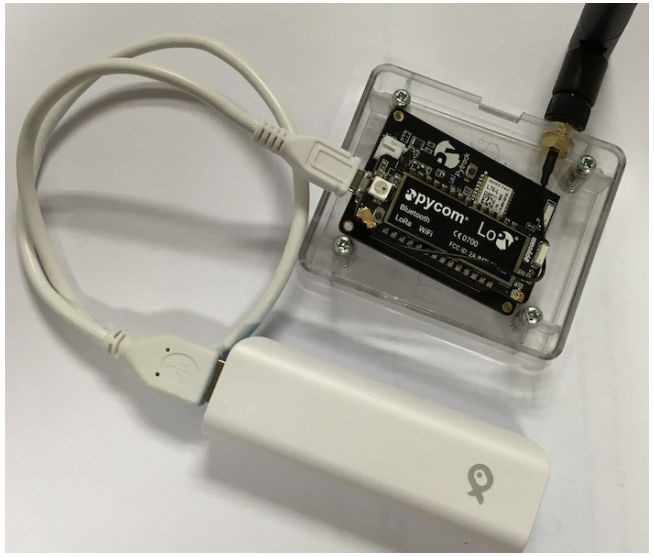
A basic model of a thing

- A "thing" generally consists of **four main parts:**
  - Sensors & actuators
  - Microcontroller
  - Communication unit
  - Power supply
- A "thing" has the **following properties**:
  - It's usually powered by battery. This implies limited source of energy.
  - It's generally small in size and low in cost. This limits their computing capability.





A Reference Guide to the Internet of Things Copyright © 2017 Bridgera LLC, RIoT



Things: sensors

Sensors

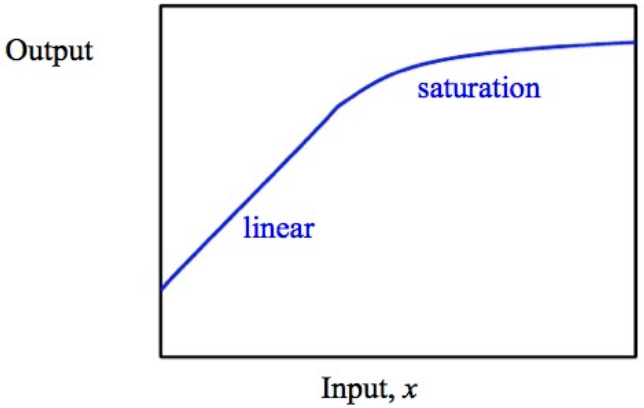
- They are mainly input components in IoT; are devices that receive a stimulus and responds with an electrical signal
- A transfer function for a sensor represent the input-output relation.
  - Input: a physical measured parameter
  - Output: usually an electrical output signal.
- The simplest form of transfer function is a linear function which can be described as follows.

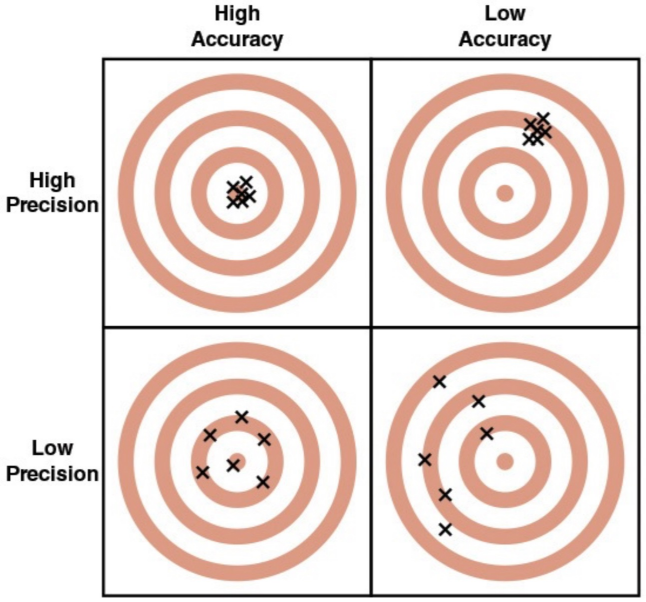
$$S = a + bx$$

  - where x is the input, b is the slope (and sometimes called sensitivity), and a is the offset (or the output when the input is zero).

Sensors transfer function

- **Saturation**:
  - The output becomes flat at some levels of input
- **Resolution**:
  - Describe the smallest increment of stir
  - Outputs change in small steps
- **Accuracy**:
  - Describes how close the measurement is to the true value
- **Precision or Repeatability:**
  - Describes the difference in results when measurements are taken repeatedly under the same
  - conditions





Things: power sources

Power sources

- Today, the **most common power source is a battery**, but there are several other possibilities for power, such as **solar cells, piezoelectricity, radio-transmitted energy, and other forms of power scavenging.**
- Rechargeable batteries are not particularly well-suited to smart objects.
  - Instead of using rechargeable batteries, battery-equipped smart objects are typically designed so a single battery should last the entire lifetime of the smart object.


LiPo

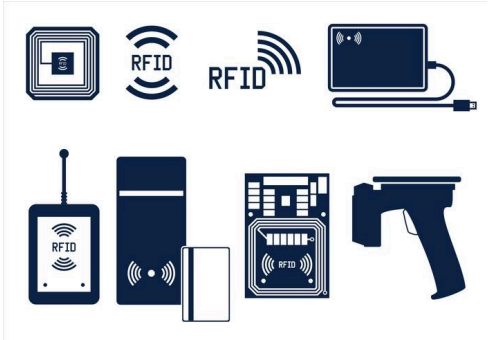

LiPo


Li-ion

Lithium Ion (Li-ion)
Lithium-Ion Polymer (Li-Po)

Power sources

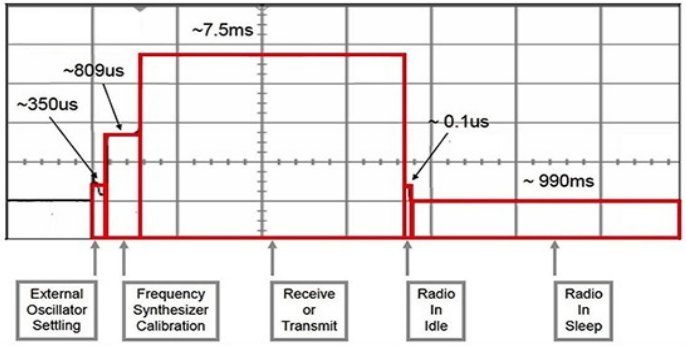
- Power scavenging is a technique that harvests power from the physical environment.
  - **Solar cells** represent the most common form of power scavenging..
  - **Piezoelectricity** is another source for power scavenging. For this source, physical movement is converted into energy used to power the smart object.
    - For example, EnOcean's smart light switches are completely driven by the energy harvested from the act of pressing the light switch.
- **The energy in radio waves can also be used as a power source.**
  - A well-known example of this are Radio Frequency Identification (RFID) tags that use radio energy to power a radio transceiver for a short while.



Things: communication module

Communication module

- Of the hardware components of a "thing", the radio is usually the most power-consuming component. Compared to the power consumption of the microcontroller or the sensors, the radio transceiver often uses ten times as much power.
- This is due to the processing required for modulating and demodulating the radio signal.
  - For low-power radios, only a small portion of the power consumption is used to send the radio signal into the air.
  - ➔ listening is as power consuming as sending.



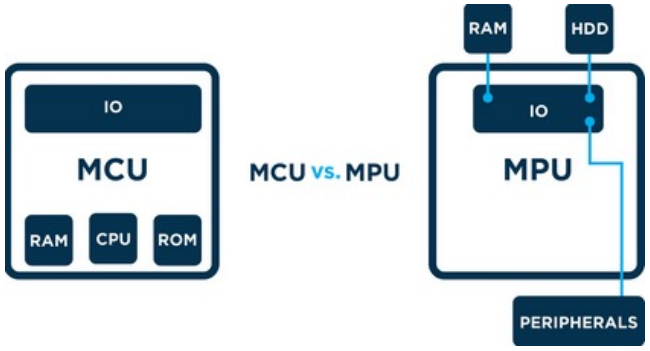
Communication module

- Because the radio is the most power-consuming component, and because idle listening is as expensive as sending data, the radio must be switched off to conserve power.
  - …. But, when the radio is switched off, it is clearly not able to receive any data.
- The radios of the devices in the network must somehow be "synchronized" to find the best compromise between saving energy and sending/receiving

Things: microcontrollers

Microcontroller

- The microcontroller gives "things" their intelligence. It runs the software of the devices and is also responsible for connecting the radio with the sensors and actuators.
- A microcontroller is a microprocessor with built-in memory, timers, and hardware for connecting external devices such as sensors, actuators, and radio transceivers.
- Typically, a smart object microcontroller has a few kilobytes of on-chip memory and is run at a clock speed of a few megahertz.



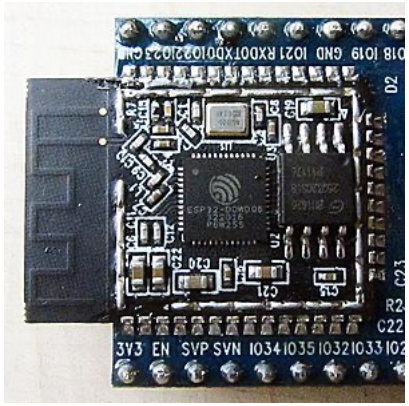
Microcontroller: ESP32

https://www.espressif.com/en/products/socs/esp32

- The ESP32 series employs a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations. ESP32 is created and developed by Espressif System

http://esp32.net

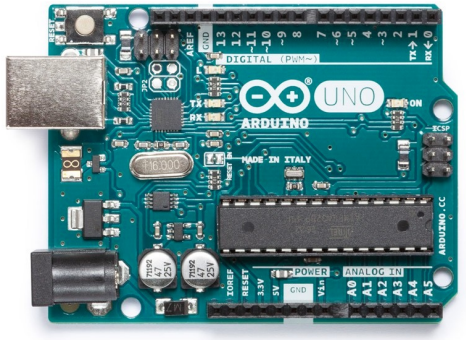
- CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS
- Memory: 520 KiB SRAM
- Wireless connectivity:
  - Wi-Fi: 802.11 b/g/n
  - Bluetooth: v4.2 BR/EDR and BLE
- Peripheral interfaces:
  - 12-bit SAR ADC up to 18 channels
  - 2 × 8-bit DACs
  - 10 × touch sensors (capacitive sensing GPIOs)
  - …

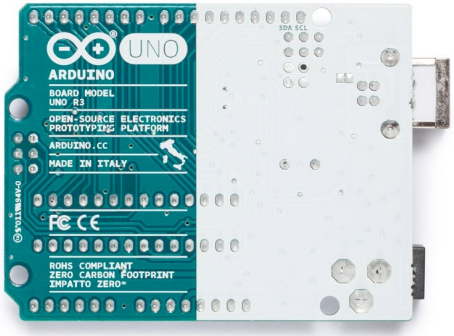


https://www.espressif.com/en/products/socs/esp32

Arduino Uno

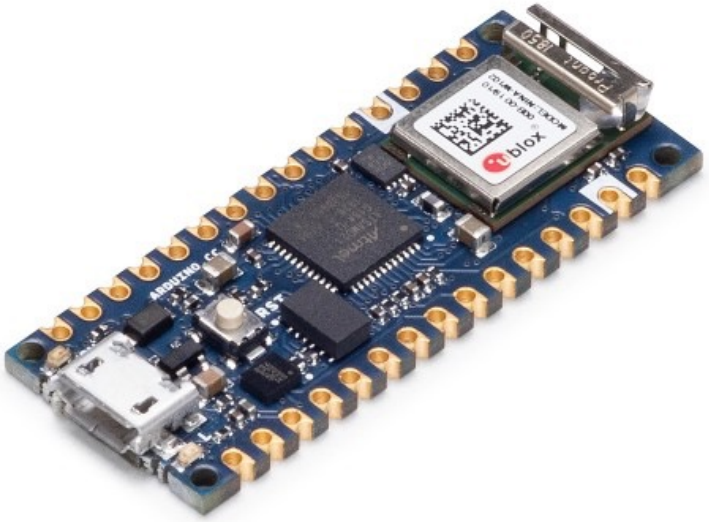
- The Arduino Uno model is widely used in Arduino development.
- It's built on top of a MCU ATmega328P microcontroller
- The board provides several digital and analog I/O pins, to which we can attach our sensor and actuator devices.
- SPI and I2C protocols are also provided by the Arduino Uno.
- More infos: https://store.arduino.cc/arduino-uno-rev3



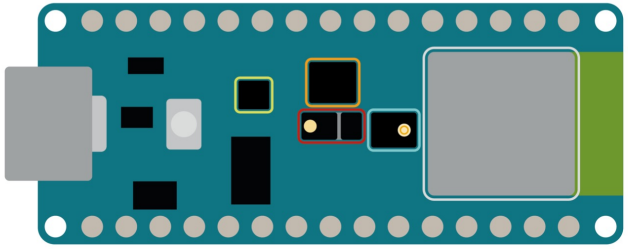


Arduino Nano

ARDUINO NANO 33 BLE SENSE



ARDUINO NANO 33 IOT



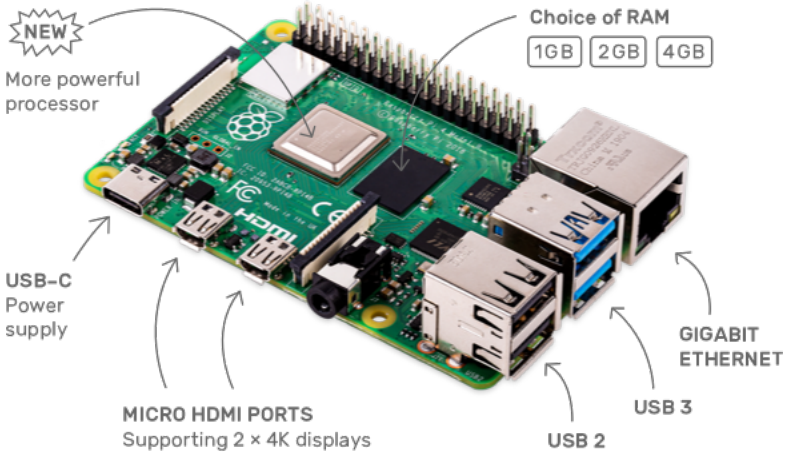
- Color, brightness, proximity and gesture sensor
- Digital microphone
- Motion, vibration and orientation sensor
- Temperature, humidity and pressure sensor
- Arm Cortex-M4 microcontroller and BLE module

Operating Systems

- Do we need an OS?
  - Arduino basically do not have an OS, there's some bootstrap code, to handle loading new code via USB and some other housekeeping
- Operating systems for smart objects are tailored to the specific requirements of smart objects and to the specific constraints imposed by the hardware.
  - E.g., Smart object operating systems do not have a user interface; no user directly interacts with the smart object operating system.
- The memory constraints make the programming model different from general purpose operating systems.

Raspberry Pi 4

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)

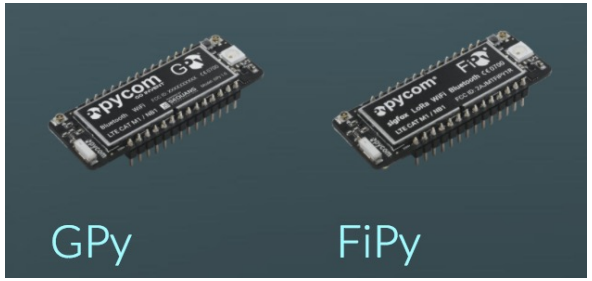


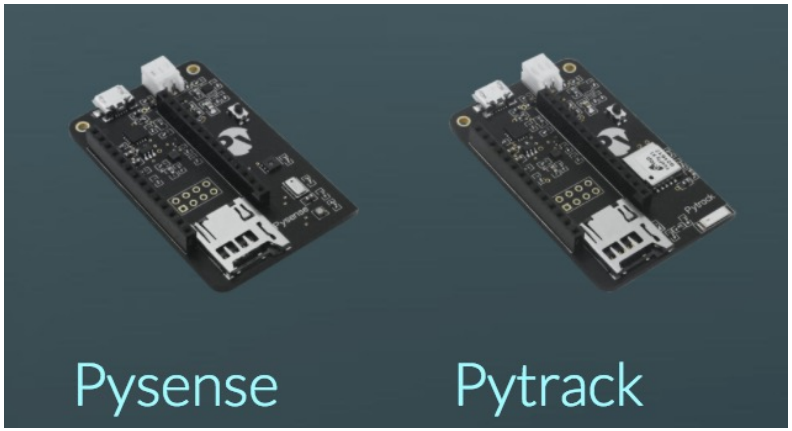
MicroPython

- MicroPython is a lean and efficient implementation of the Python 3 programming language optimised to run on microcontrollers and in constrained environments.
  - It is compact enough to fit and run within just 256k of code space and 16k of RAM.
  - It is packed full of advanced features such as an interactive prompt, arbitrary precision integers, closures, list comprehension, generators, exception handling and more.
  - **In addition to implementing a selection of core Python libraries, MicroPython includes modules such as "machine" for accessing low-level hardware.**
- You get an interactive prompt (the REPL) to execute commands immediately, along with the ability to run and import scripts from the built-in filesystem.
- MicroPython strives to be as compatible as possible with normal Python (known as CPython) so that if you know Python you already know MicroPython.
  - https://micropython.org/
  - Use MicroPython online: https://micropython.org/unicorn/

Nodes: HW alternatives for prototypes: Pycom

- Multi-network hardware.
  - Wifi, Bluetooth, Sigfox, LoRa and LTE-M
- Based on MicroPython
- The portfolio includes a comprehensive range of development boards, expansion boards, sensor shields and OEM hardware





https://pycom.io/hardware/

To conclude: the 3 stages of an IoT journey

"The minimum viable product (MVP) is that version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort." by Eric Ries