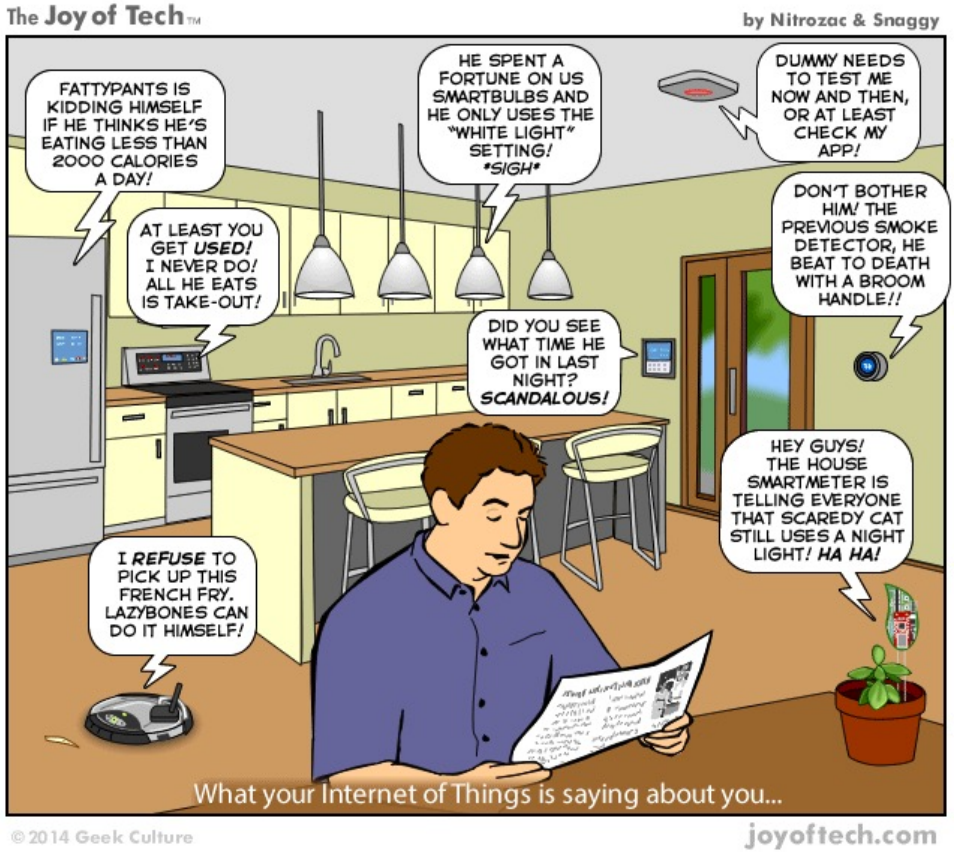
Internet of Things

- ✓ A hands-on, networking oriented perspective

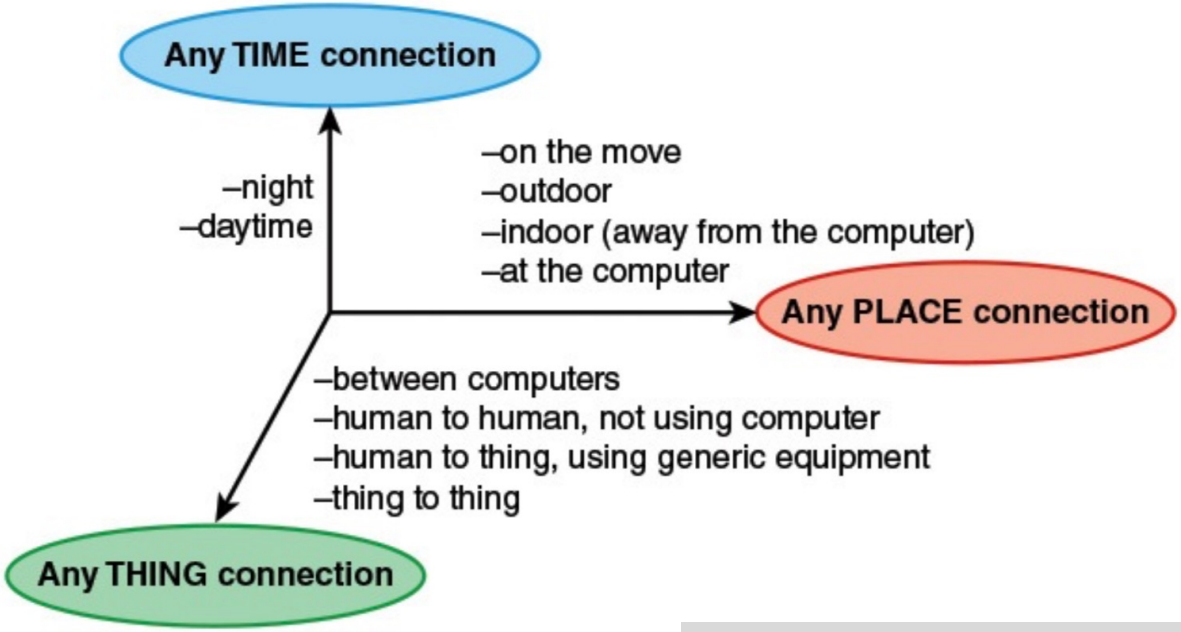
Internet of Things (IoT)

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

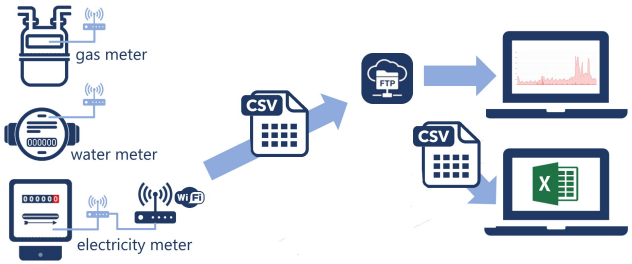
"The IoT can be viewed as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies (ICT)."



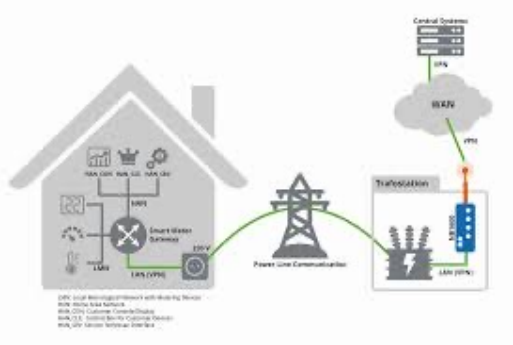
Source: Recommendation ITU-T Y.2060

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 simplifing integration across IoT platforms and application domains by using and extending existing, standardized Web technologies.

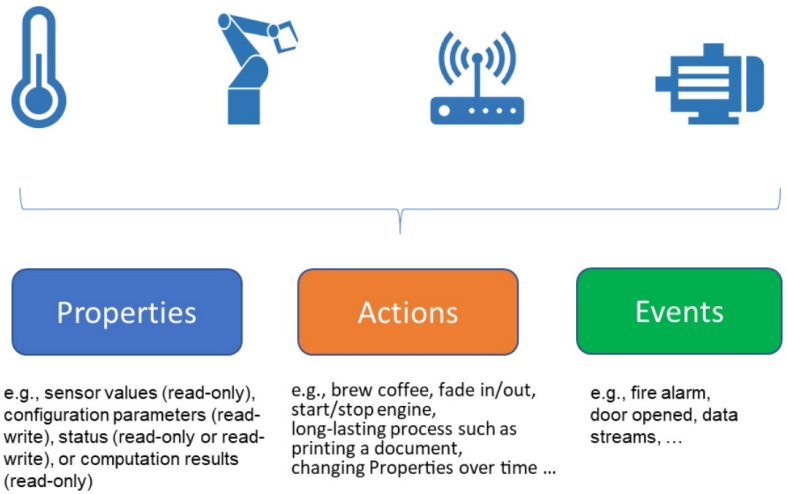
From the developer's perspective, the WoT enables access and control over IoT resources and applications using mainstream web technologies (such as **HTML 5.0, JavaScript, Ajax, PHP, Ruby n Rails**, etc)

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.

WoT Interactions

WoT introduces a simple interaction abstraction based on properties, events, and actions. Any IoT network interface can be described in terms of this abstraction. By using this abstraction, applications have a common anchor to retrieve an IoT service's metadata as well as way to understand what and how the data and an IoT services' functions can be accessed.

Please read here to get more background about the properties-action-event paradigm.

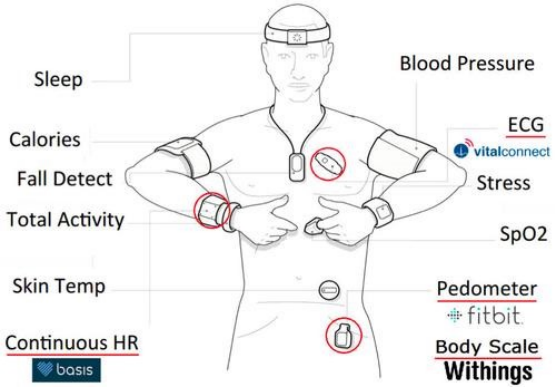


IoT Fundamental characteristics

- **Interconnectivity**: anything can be interconnected with the global information and communication infrastructure.
- **Heterogeneity**: The devices in the IoT are heterogeneous as based on different hardware platforms and networks.
- **Dynamic changes**: The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices can change dynamically.
- **Enormous scale**: The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet.

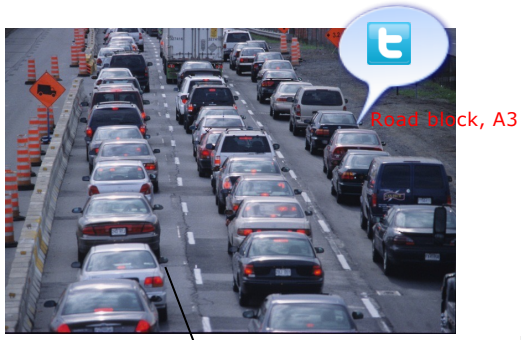
Source: Recommendation ITU-T Y.2060

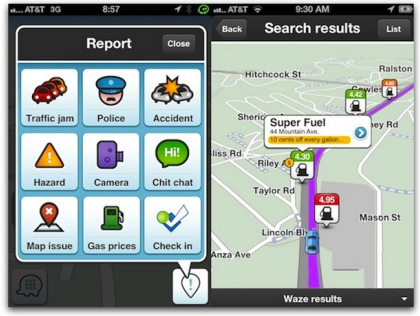
More examples of Things



Beyond conventional sensors

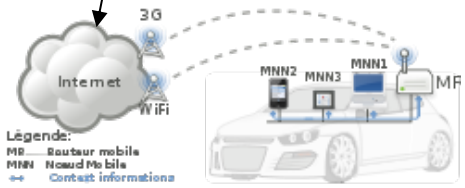
- Human as a sensor (citizen sensors)
  - e.g. tweeting real world data and/or events
- Virtual (software) sensors
  - e.g. Software agents/services generating/representing data





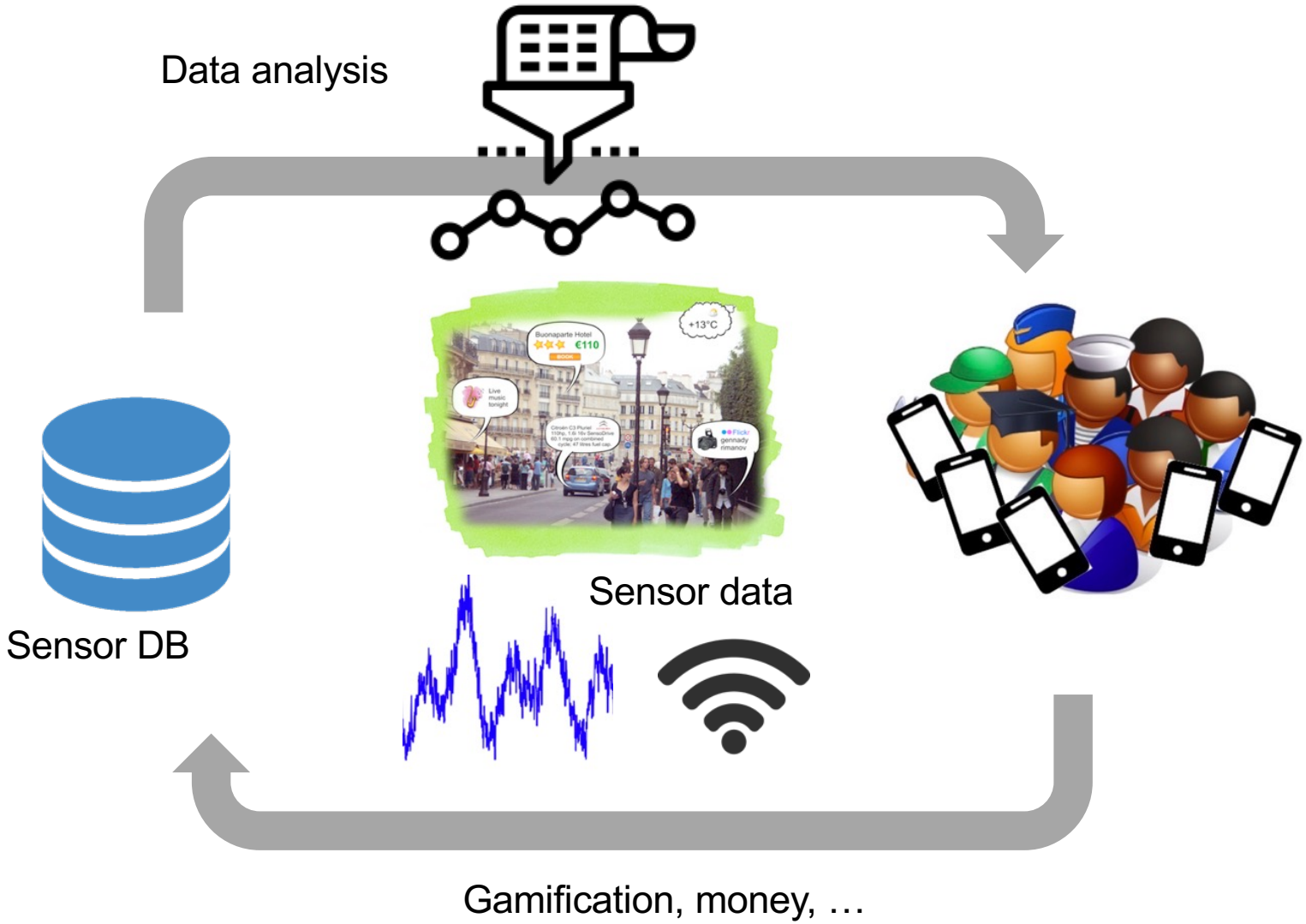
https://www.waze.com

Road block, A3

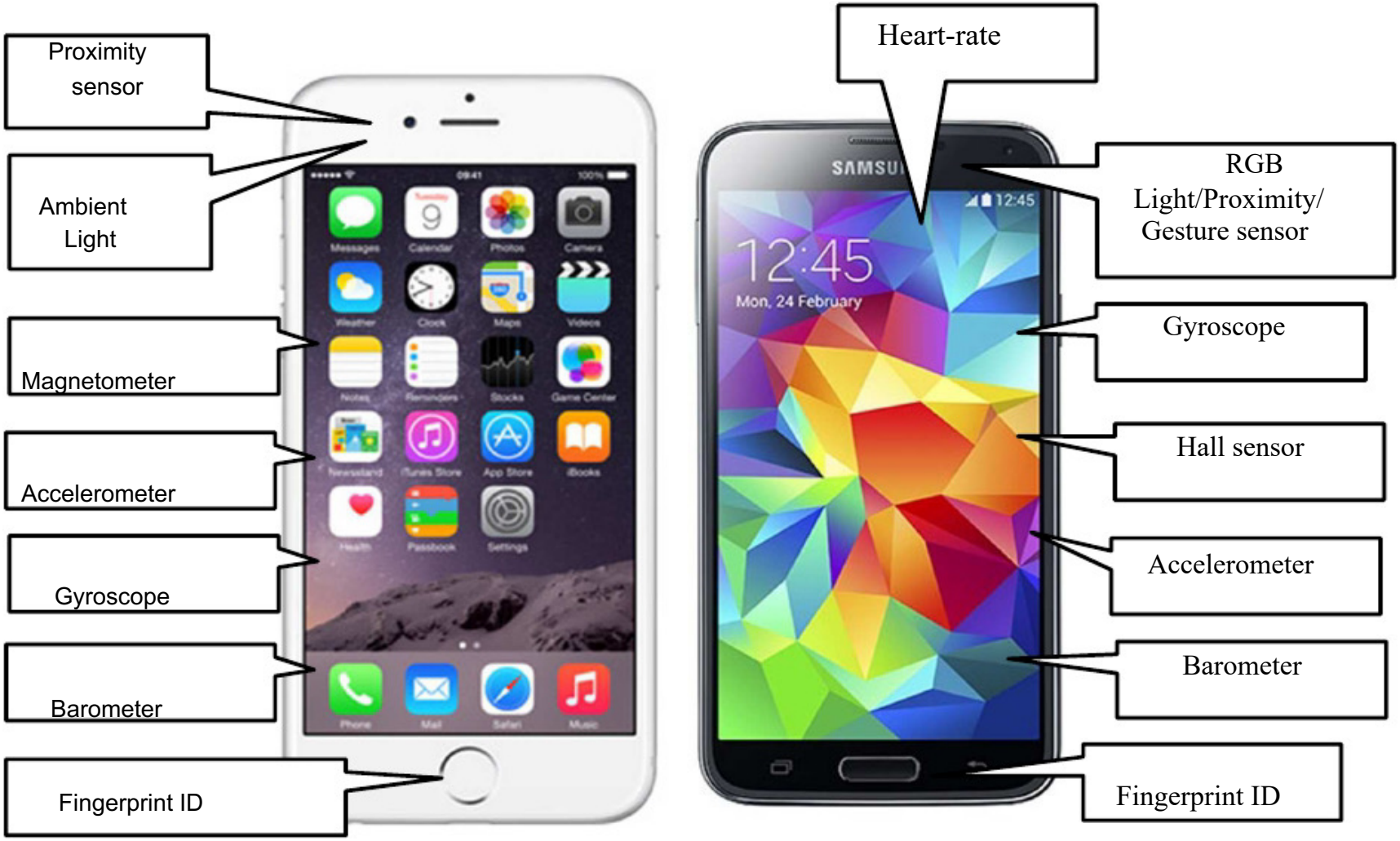


Suggest a different route

Crowdsensing



Sensors in Modern Smart Phones



Open Data

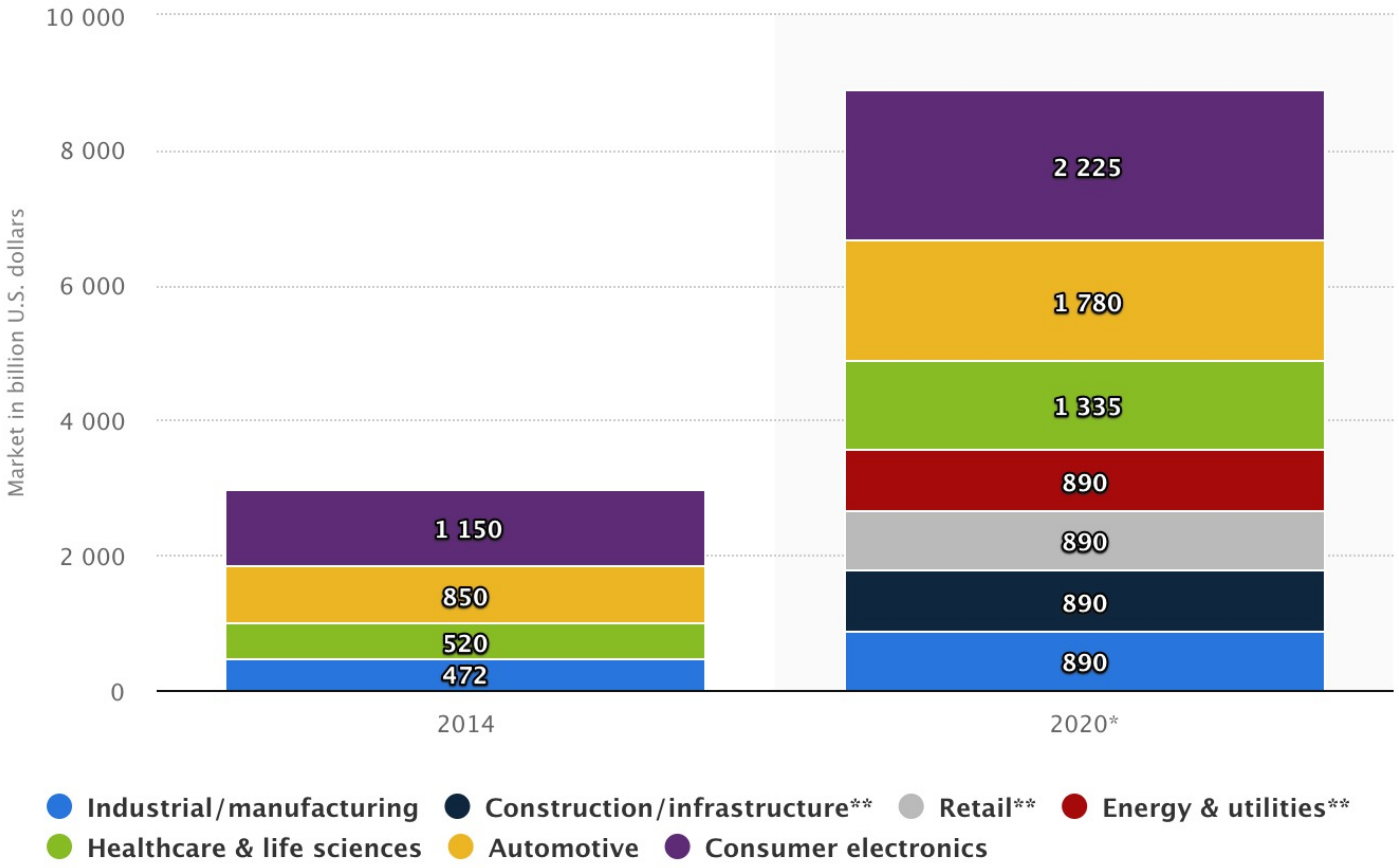
- "Cities today are vast repositories of information, endlessly collecting and archiving data. When semantically organized, the data can be exposed, shared, and interconnected. Giving people the right kind of access to this information can spark new applications and services, new ways of living, creating and being."
- Global movement to open up public data sets to make public data more accessible
  - Sparks innovation: Creation of apps and services
  - Greater transparency in government
- Big Data techniques!

Berners Lee, "The year open data went worldwide", TED talks: http://www.ted.com/talks/tim_berners_lee_the_year_open_data_went_worldwide.html

Some example of open data projects

- http://datos.gob.es/
- https://www.valencia.es/dadesobertes/va/
- http://data.worldbank.org/
- https://data.baltimorecity.gov
- https://opendata.vancouver.ca/
- https://data.smartdublin.ie/
- https://data.rennesmetropole.fr/page/home/
- https://data.grandlyon.com/
- http://opendata.paris.fr
- http://opendata.comune.fi.it
- http://dati.comune.roma.it
- http://data.london.gov.uk
- http://data.gov.uk
- …

Size of the Internet of Things market worldwide

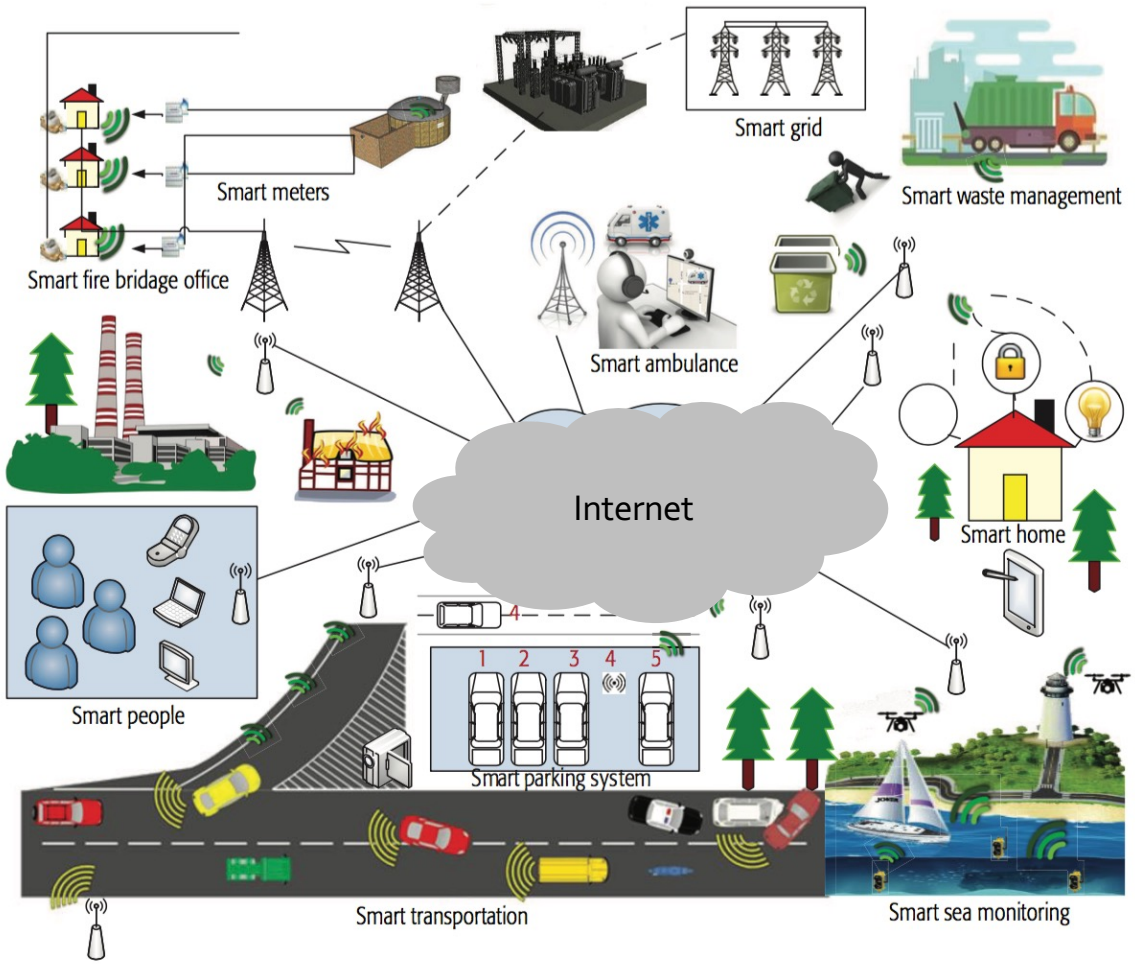


in 2014 and 2020, by industry (in billion U.S. dollars), © Statista 2019

All big companies are active in this area

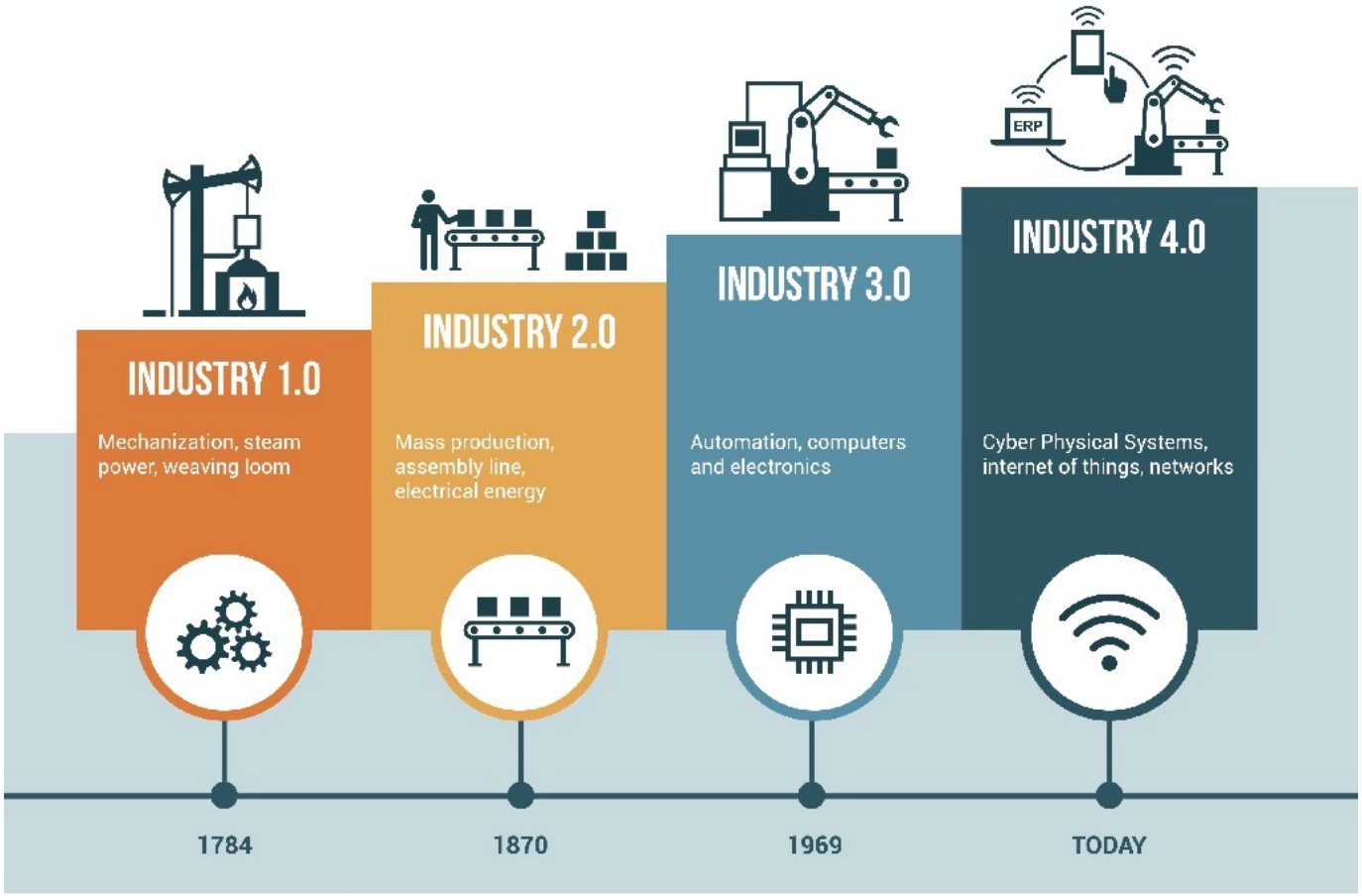
- Telefonica: https://iot.telefonica.com/en/about-us/the-thinx-iot-lab/
- CISCO: https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html
- Google: https://cloud.google.com/solutions/iot/
- HP: https://www.hpe.com/us/en/solutions/internet-of-things.html
- IBM: https://www.ibm.com/internet-of-things/
- Microsoft: https://azure.microsoft.com/es-es/overview/iot/
- Oracle: https://www.oracle.com/internet-of-things/
- Samsung: https://www.samsung.com/cl/mobile-iot/
- Apple: https://www.apple.com/shop/accessories/all-accessories/homekit
- …

Smarter everything



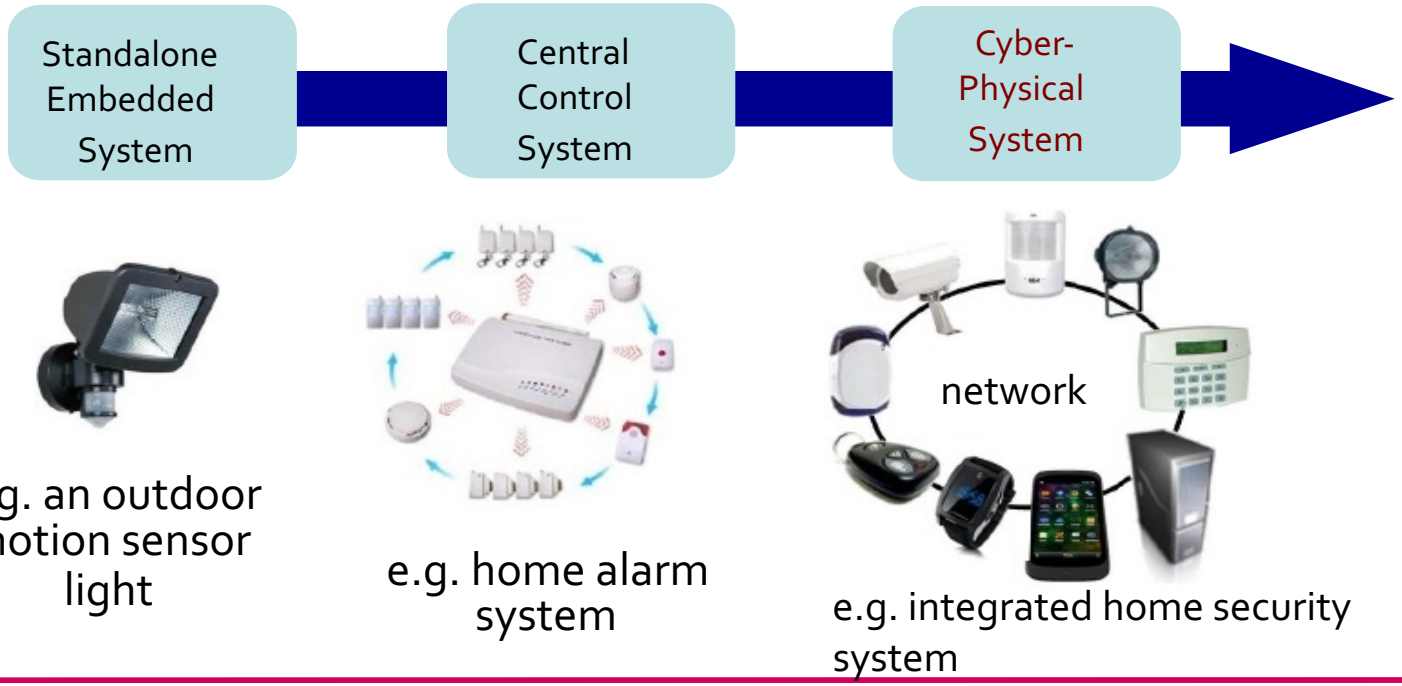
Enabling Communication Technologies for Smart Cities, IEEE Communications Magazine - January 2017

Industry 4.0 and Industrial IoT



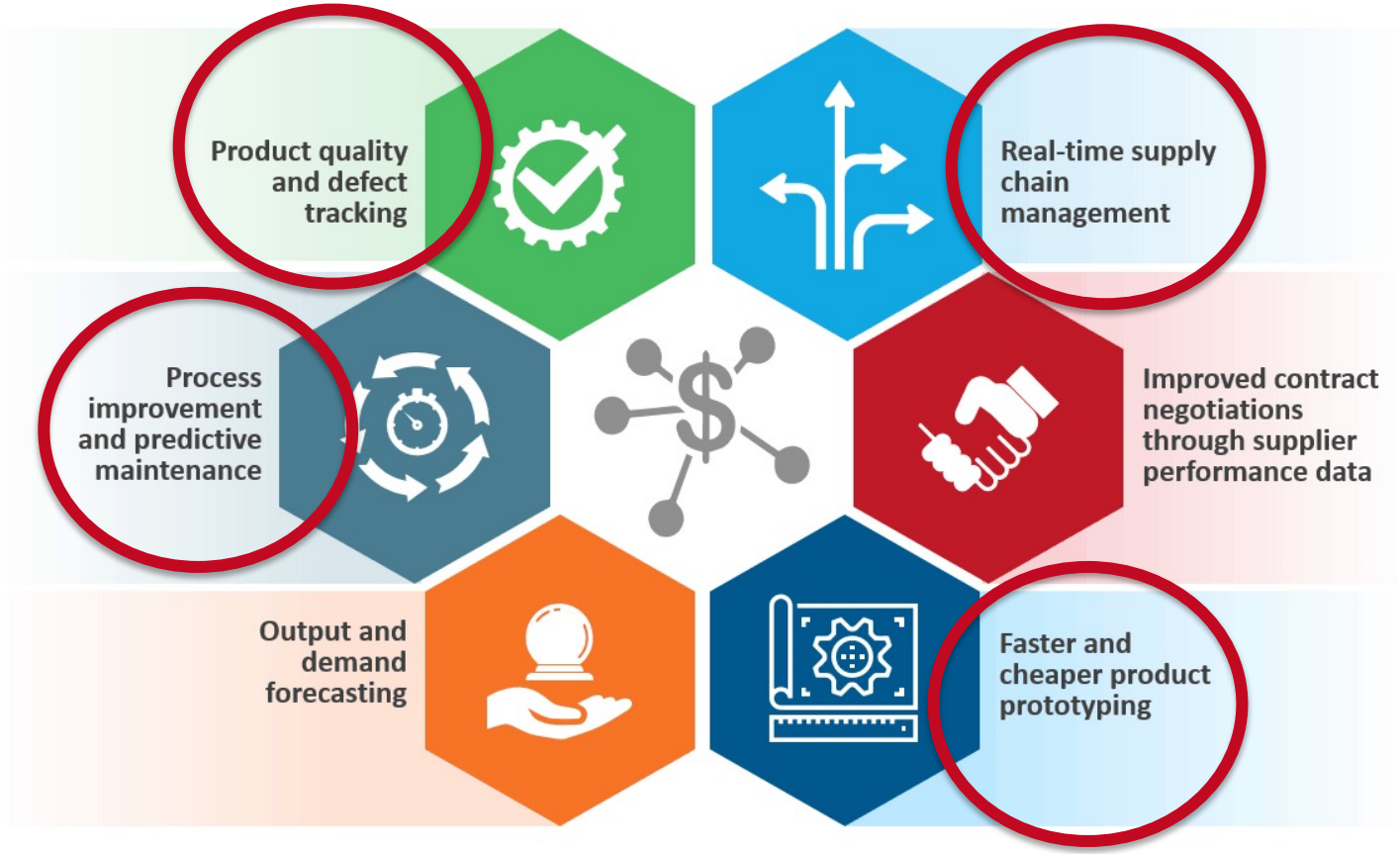
Cyber-Physical Systems

- CPS is commonly used (and preferred over IoT) by the engineering communities (e.g., mechanical engineering, aerospace engineering, aeronautics). It is also used extensively by computer scientists working on embedded systems
- CPS is typically used in the case of systems/problems that involve large scale real-time control (e.g., time critical problems), notably problems that combine control of combined organizational and physical processes.



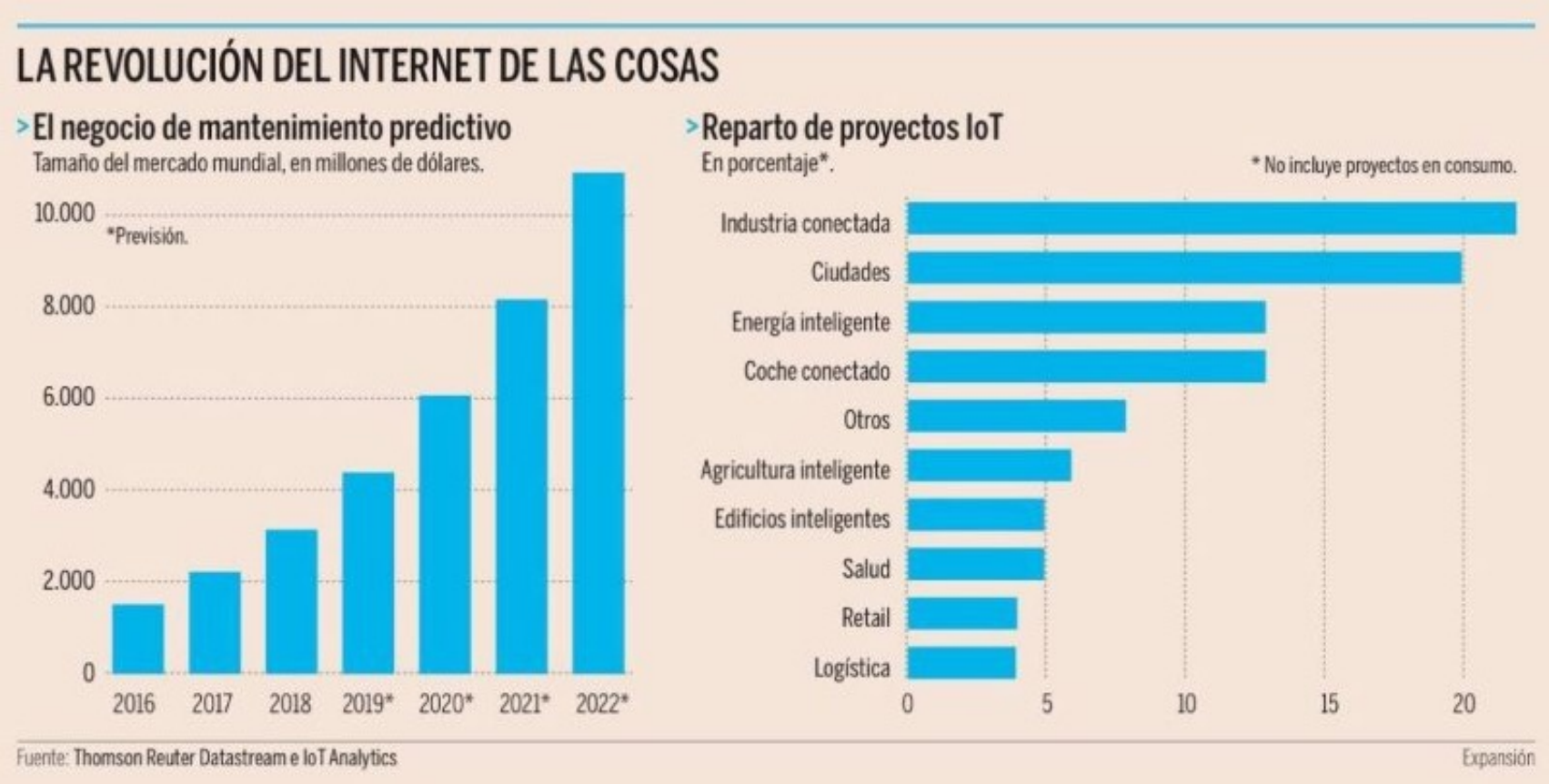
How Industry 4.0 is Helping Manufacturers

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

Market share



Industrial IoT example

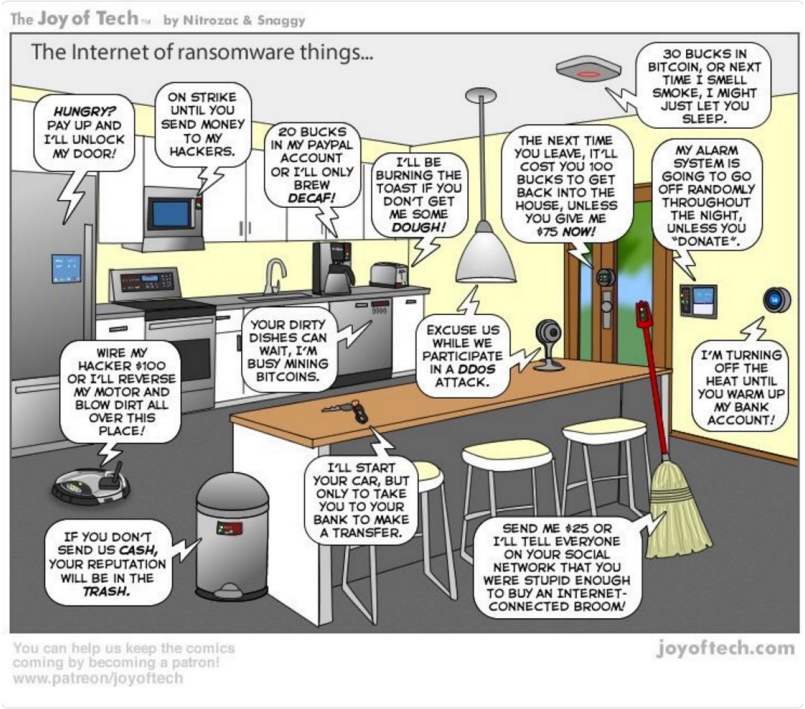
- **Rolls-Royce TotalCare**, "engine as a service".
  - Predictive maintenance (**know what will break**).
  - Anomaly detection (**find unknown issues**).
  - Live feedback (**from deployed engines**).

With the digital revolution blurring the boundaries between our physical products and the services we provide, we see a future where our engines are connected, contextually aware and even comprehending.

A future where we design and test engines digitally, service them remotely and manage them through their digital twin. A future where, once again, we are pioneering the power that matters, just like we have throughout our history.

Security considerations

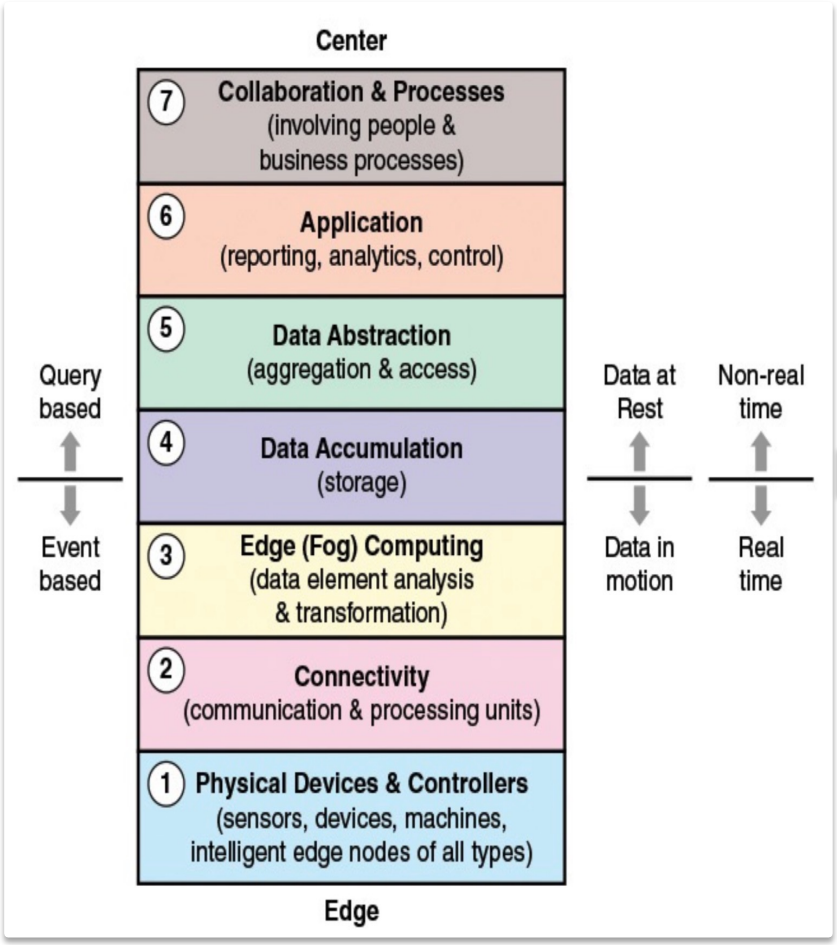
- There is no longer a solid perimeter
  - Wireless, mobile, web sites, computing everywhere
- Multiple vendors providing solutions
- Security is not a selling point - First to market
- Outsourcing

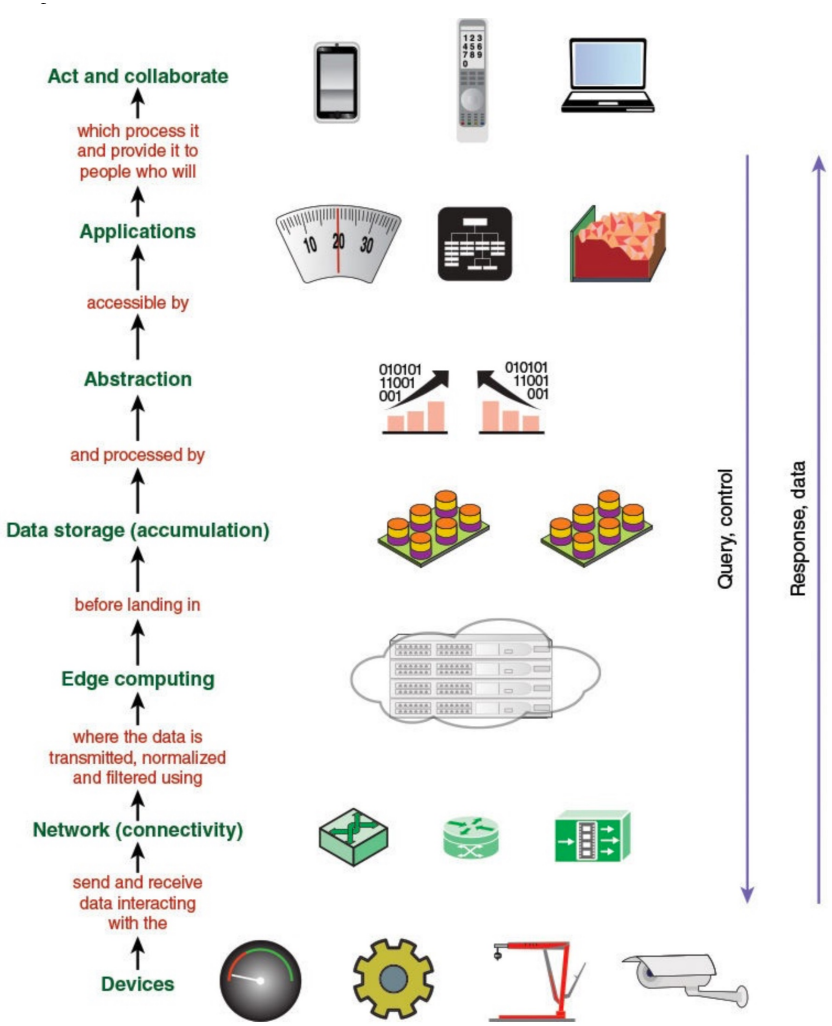


- Example: Dyn Attack
  - October 21, 2016
  - https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

https://github.com/jgamblin/Mirai-Source-Code

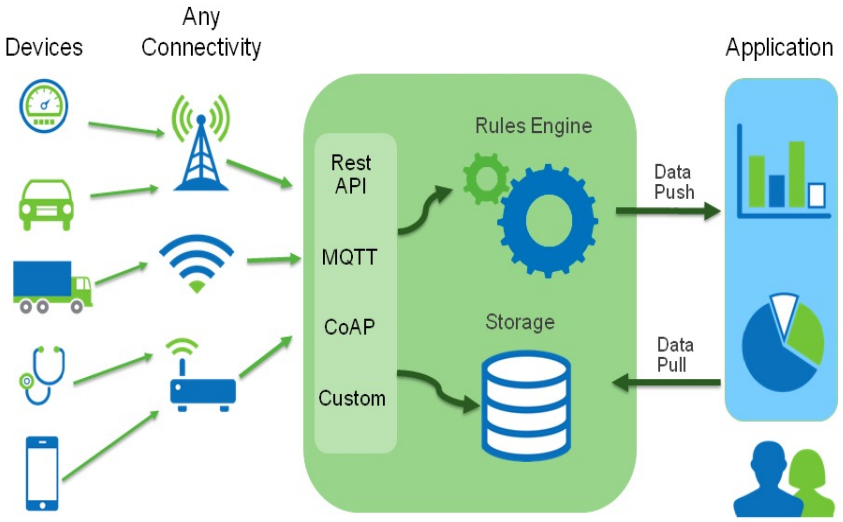
Internet of Things World Forum Reference Model





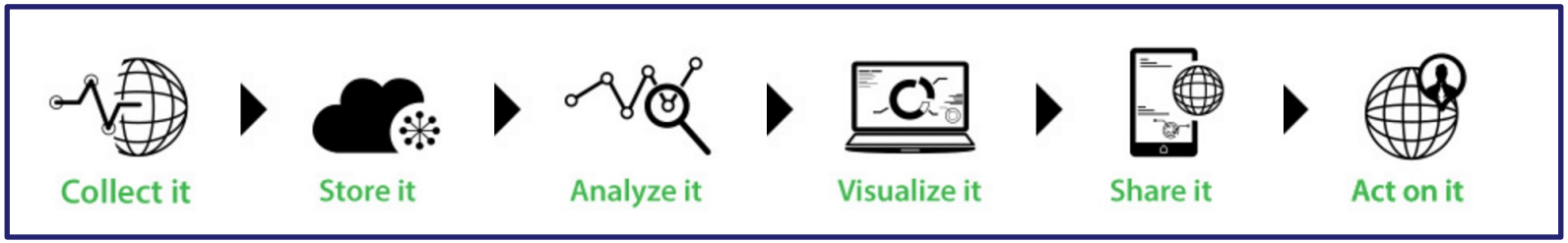
Simplified model

- 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



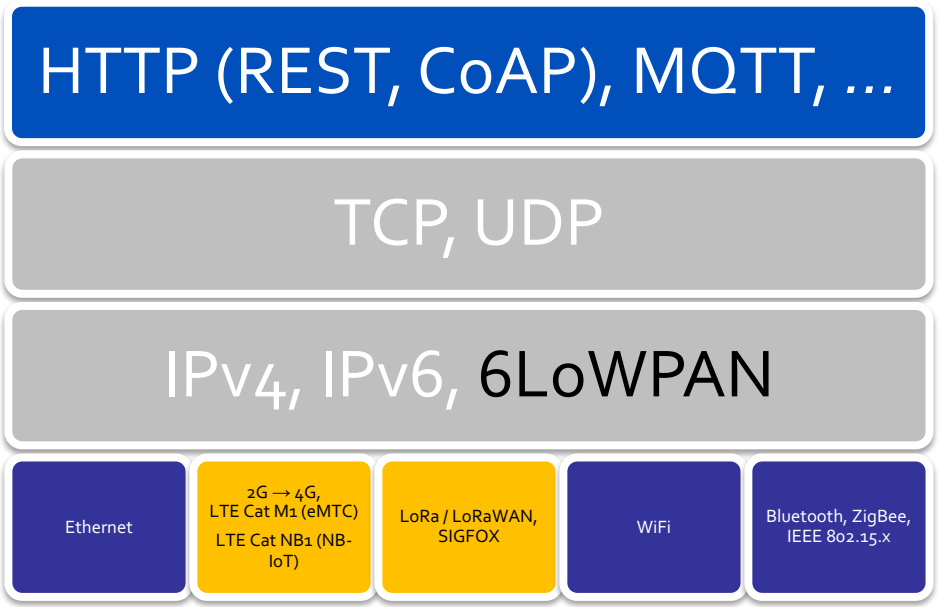
Platforms

- **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
- …



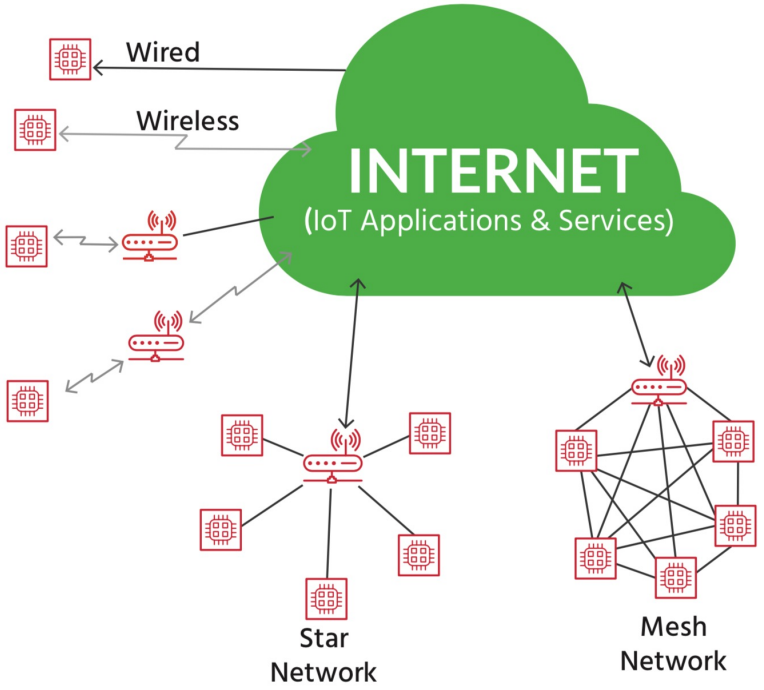
Connectivity

- In March 2015, the Internet Architecture Board (IAB) released a guiding architectural document for networking of smart objects (RFC 7452) which outlines a framework of **four common communication models used by IoT devices.**
- Most smart object deployments **can make use of the already-standardized Internet Protocol Suite.**



The 4 communication models

- communication patterns used in the smart object environment:
  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 example

Ellipse

See all reviews

A smart bike lock that connects to your phone to provide keyless entry, theft detection, bike sharing, crash alerts and more.

Quantity

- 1 +

$199

BUY NOW

Free shipping to the U.S.

International shipping starts at $20.

Device to cloud example

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

35,69 € (35,69 € / Unidad)

UNIDADES

\- 1 +

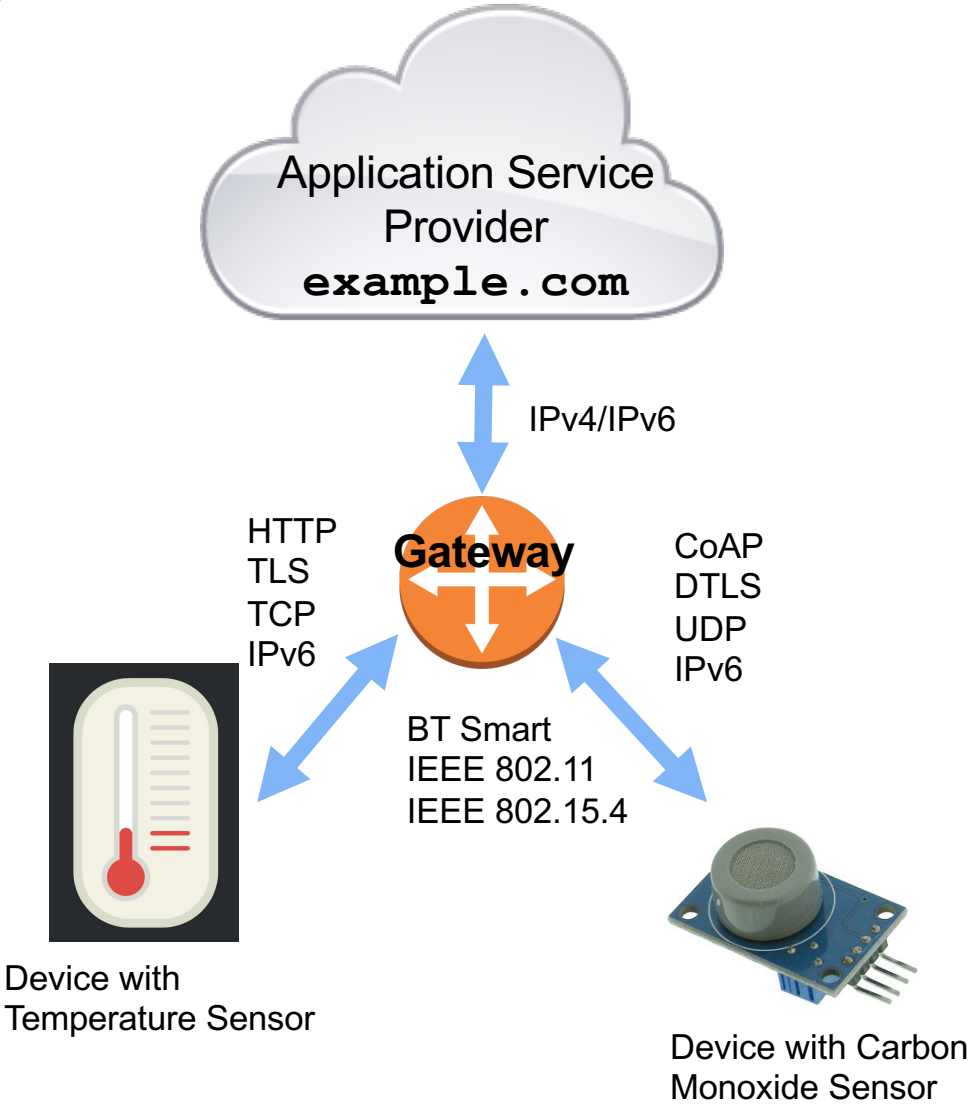
AÑADIR AL CARRO

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.

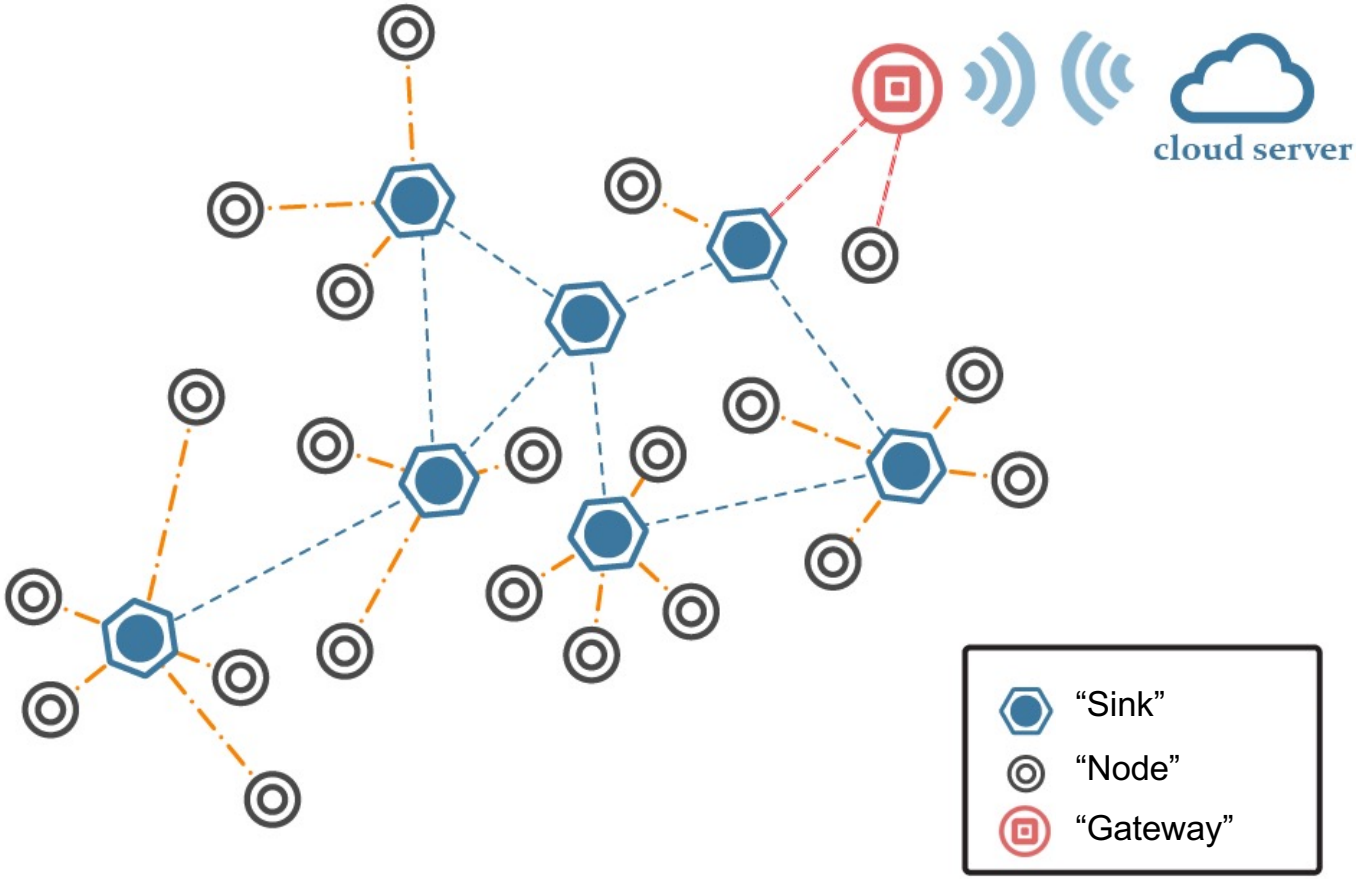
- Programable según tus necesidades.
- Se conecta y controla a distancia a través de tu Smartphone (aplicación disponible para Android & iOS). Cuando lo necesites y desde donde quieras
- Control de la intensidad: Normal y Max.
- Modo Inteligente: este modo ajustará la intensidad y duración de forma automática 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



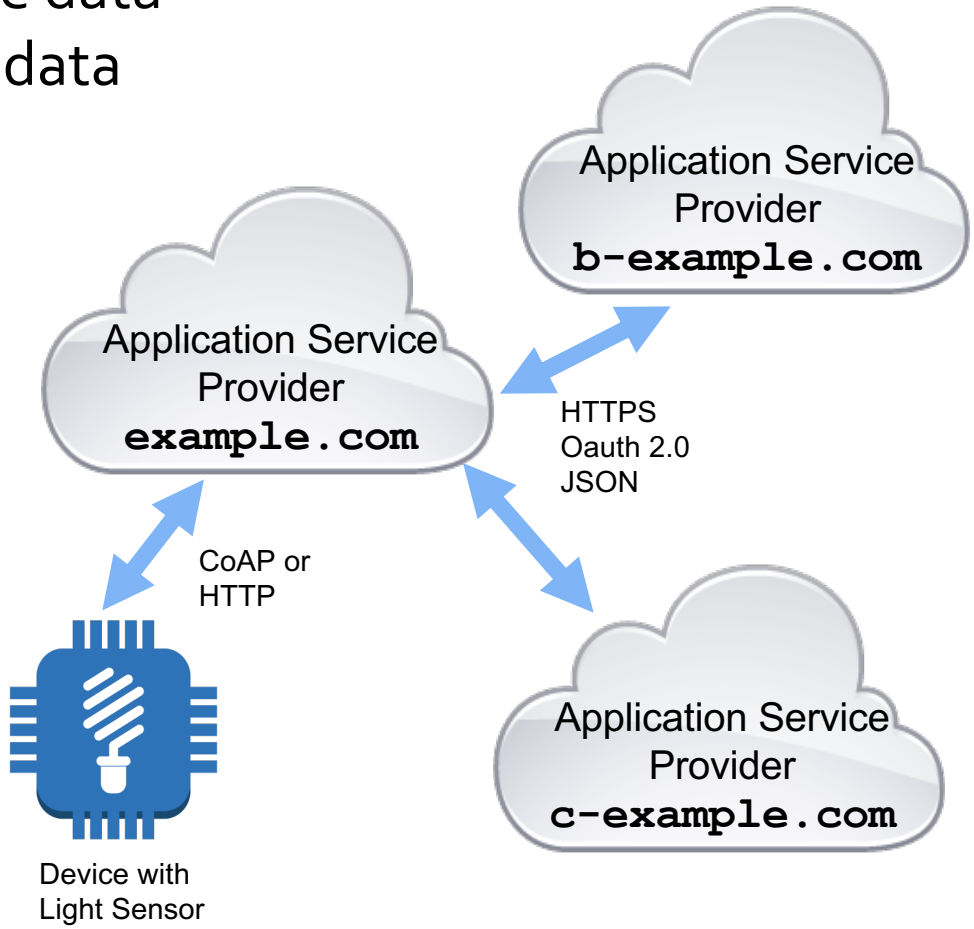
An example:

Device-to-Gateway: More complex configurations



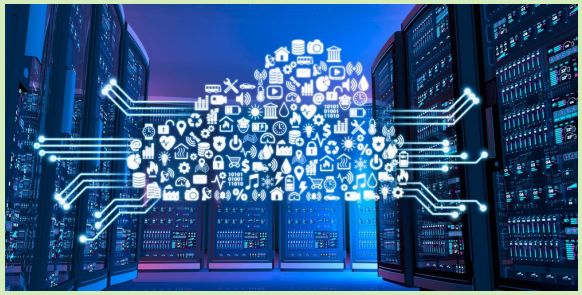
Back-End Data Sharing

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

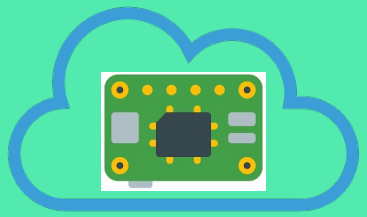


Edge computing

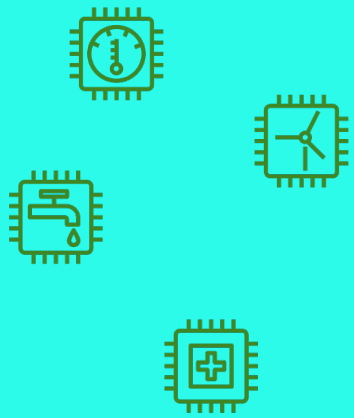
Core cloud



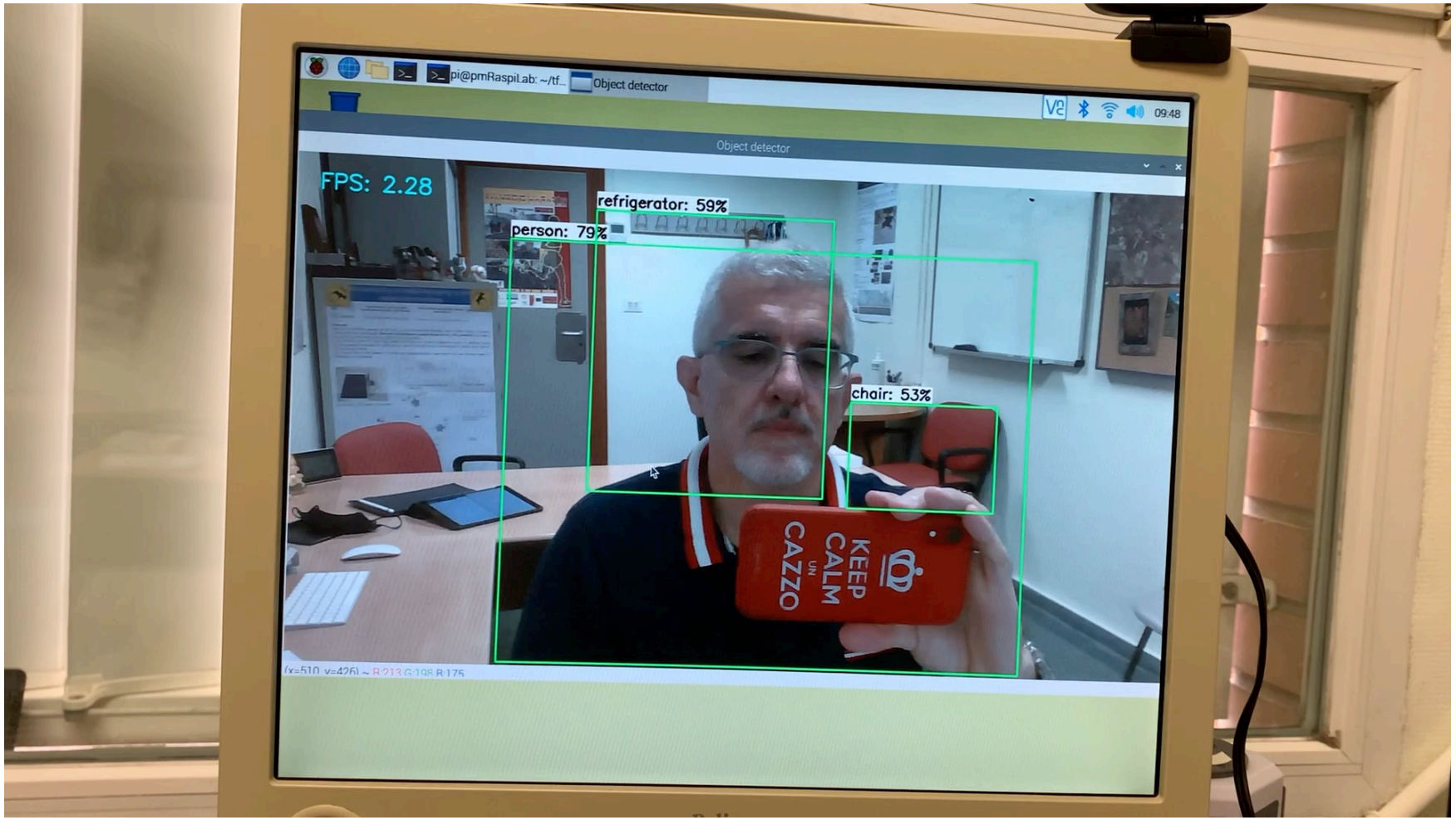
Edge Node



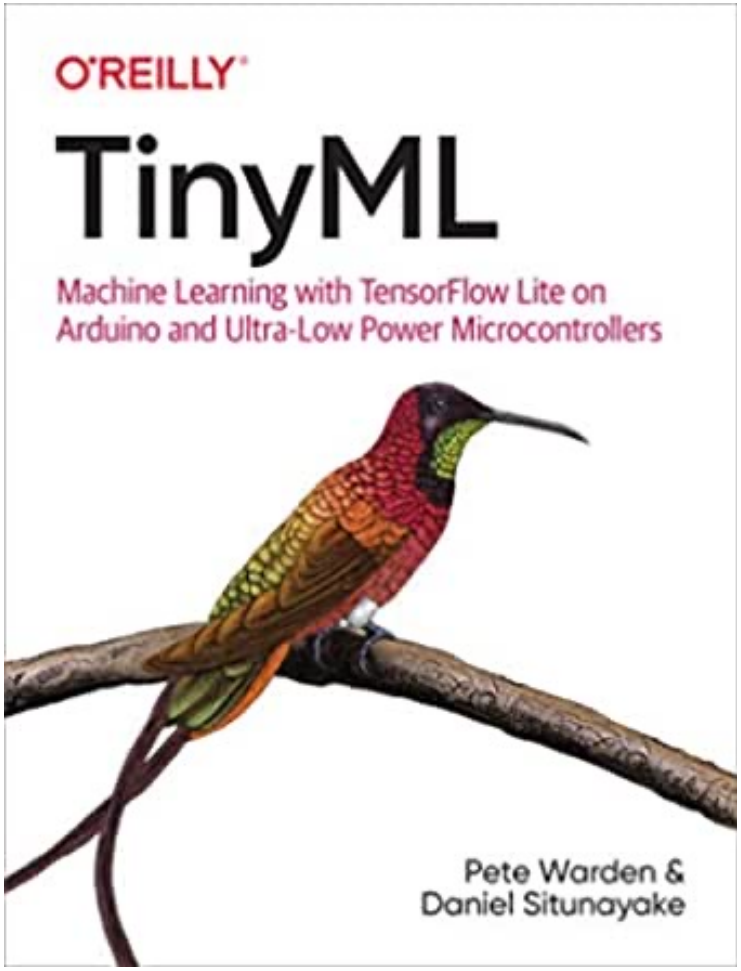
Edge devices

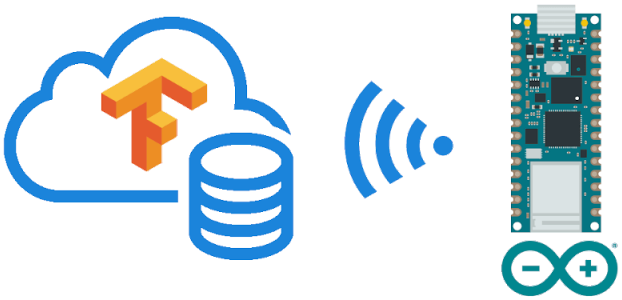


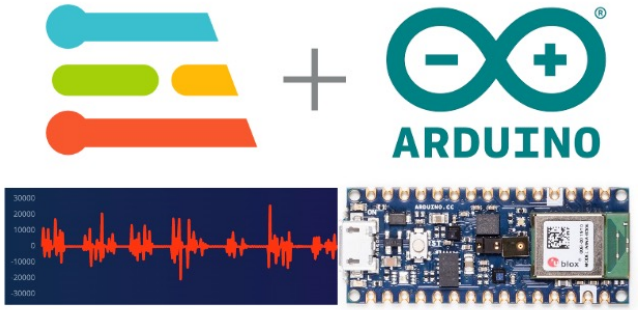
Edge Computing



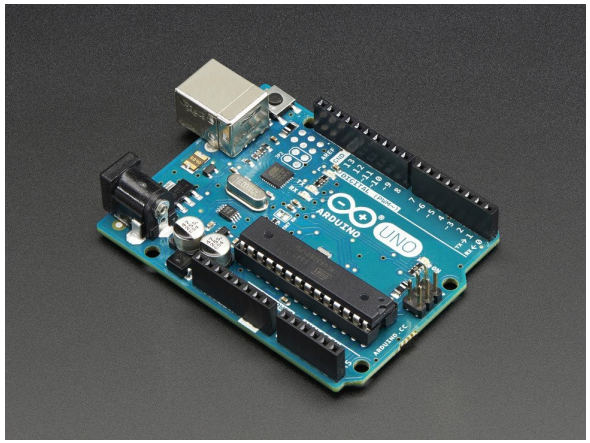
TinyML


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




ARDUINO

Prototyping IoT devices



https://www.postscapes.com/internet-of-things-hardware/

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

Proof of Value
(Engineering Samples)

Build an MVP to measure the ROI and to validate the value proposition interest

Pilot
(Pilot Run)

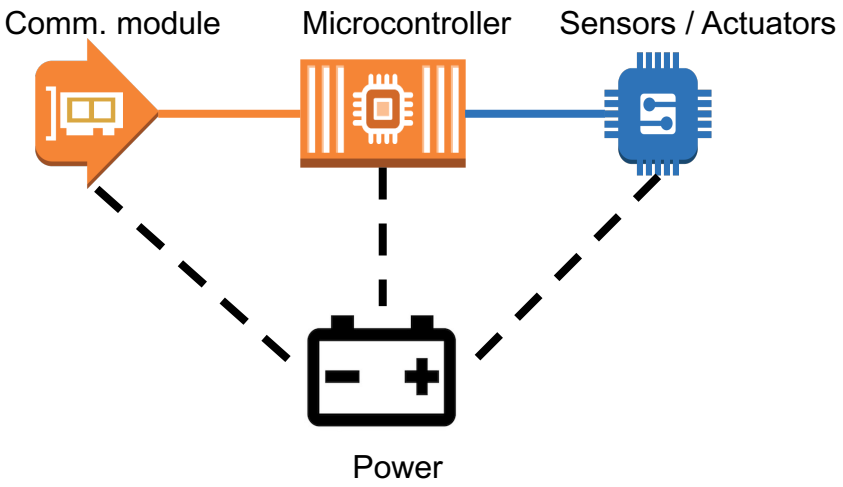
Build a first iteration of the platform to confirm the business case at scale and the functional requirements

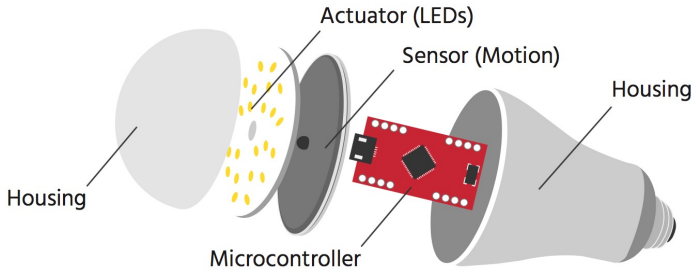
Production
(Mass Production)

Build a scalable and easily manageable platform that is loosely coupled to device manufacturers and protocols

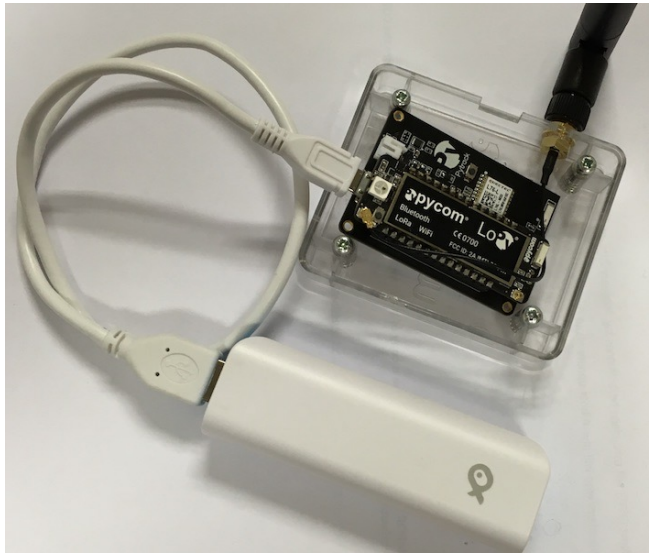
Things

- 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.



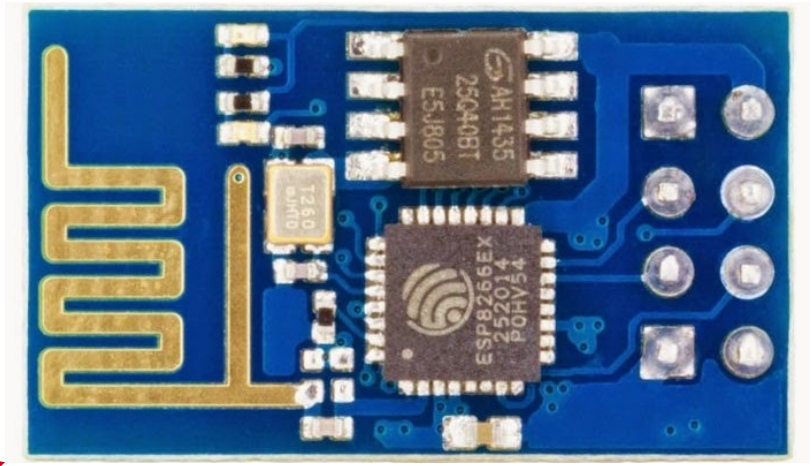


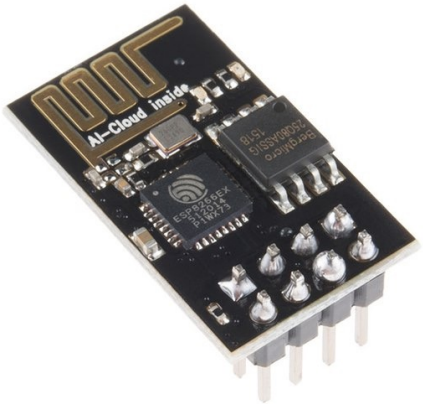
A Reference Guide to the Internet of Things Copyright © 2017 Bridgera LLC, RIoT



Microcontroller: ESP8266

- The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by Shanghai-based Chinese manufacturer, Espressif Systems.
  - http://www.esp8266.com
- There are multiple products out there labeled ESP8266, but there are at least a dozen variants and quality documentation is tough to find.
- The original chip: **the ESP-01 module:**
  - 32-bit microcontroller unit.
  - It uses TCP/IP protocol for transmission.
  - It works on 802.11 b/g/n WiFi with a frequency of 2.4 GHz and WPA (Wi-Fi Protected Access) or WPA2.
  - It also consists of 10 bit ADC 2.0, (H) SPI, UART, I2C, I2S, IR remort control, PWM and GPIO.
  - Its deep sleep power is less than 10uA and Power down leakage current is less than 5uA.
  - Its wake up and transmit packets is less than 2ms.
  - Its standby power consumption is < 1.0mW (DTIM3).
  - Its output power is +20 dBmin 802.11b mode.



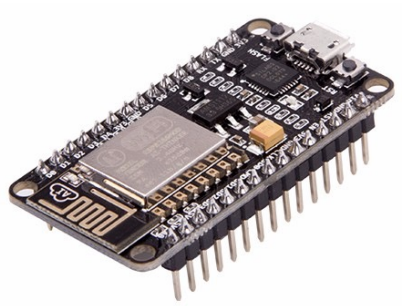


https://everythingesp.com

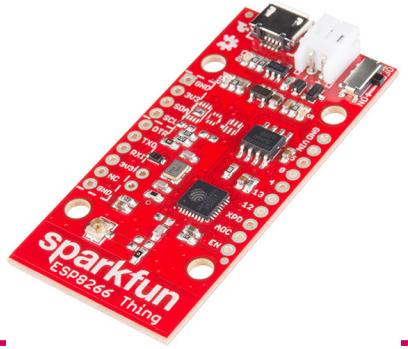
Microcontroller: ESP8266

- **ESP-12E** breaks out almost all of the ESP8266's pins, giving access to 11 GPIO pins and an analog-to-digital converter, and it includes an additional 6 pinouts along the bottom of the board, and a larger flash memory module (4MB).

- The NodeMCU includes a built-in USB to serial converter, making flashing the ESP8266 a simple endeavor
  - This board uses Lua as the programming language.
  - http://www.nodemcu.com/



- The SPARKFUN ESP8266 THING provides similar characteristics



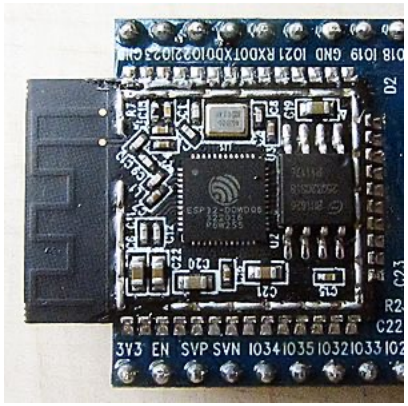
Microcontroller: ESP32

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

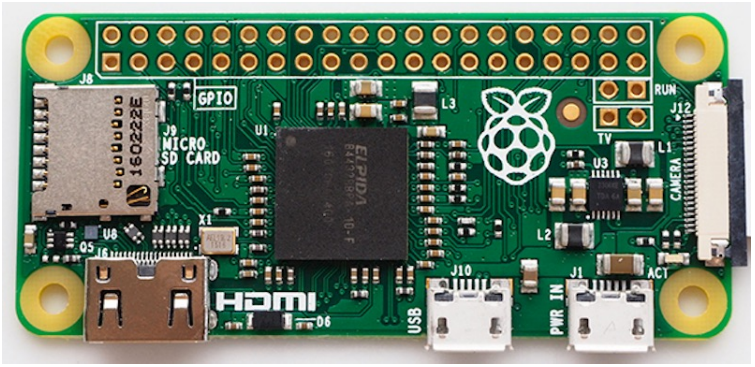
- CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS
- Memory: 520 KiB SRAM
- Wireless connectivity:
  - Wi-Fi: 802.11 b/g/n
  - Bluetooth: v4.2 BR/EDR and BLE
- Peripheral interfaces:
  - 12-bit SAR ADC up to 18 channels
 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

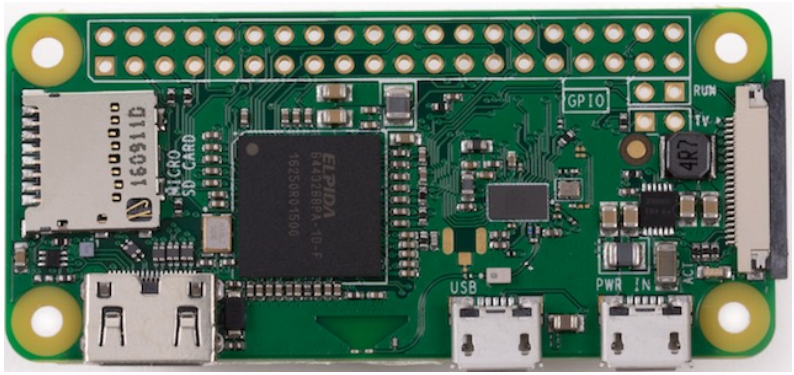
http://esp32.net



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

Raspberry Pi Zero & Zero W





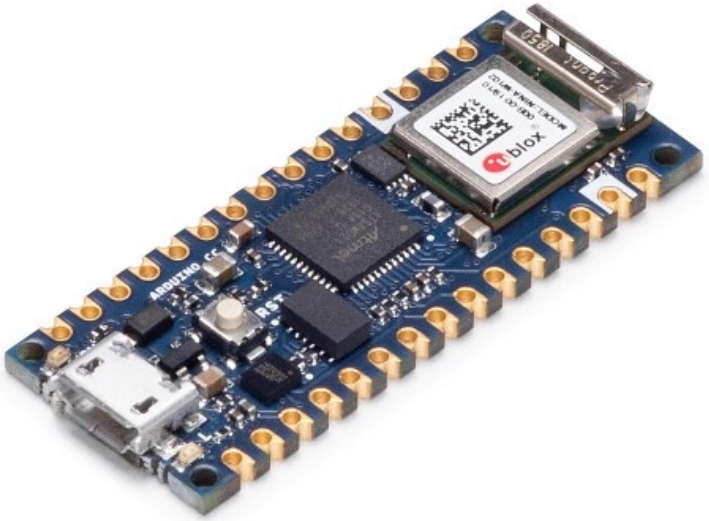
- 1GHz, single-core CPU
- 512MB RAM
- Mini HDMI and USB On-The-Go ports
- Micro USB power
- HAT-compatible 40-pin header
- Composite video and reset headers
- CSI camera connector

Plus:

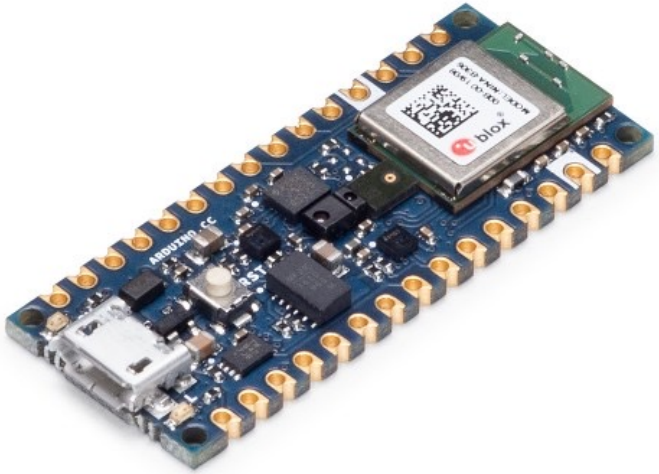
- 802.11 b/g/n wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)

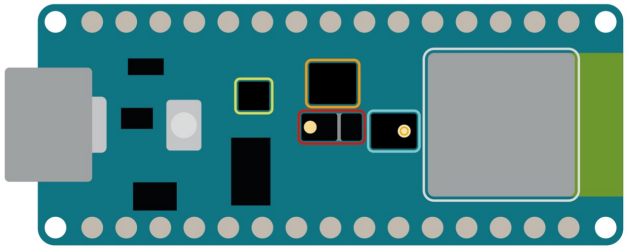
Arduino Nano

ARDUINO NANO 33 IOT



ARDUINO NANO 33 BLE SENSE



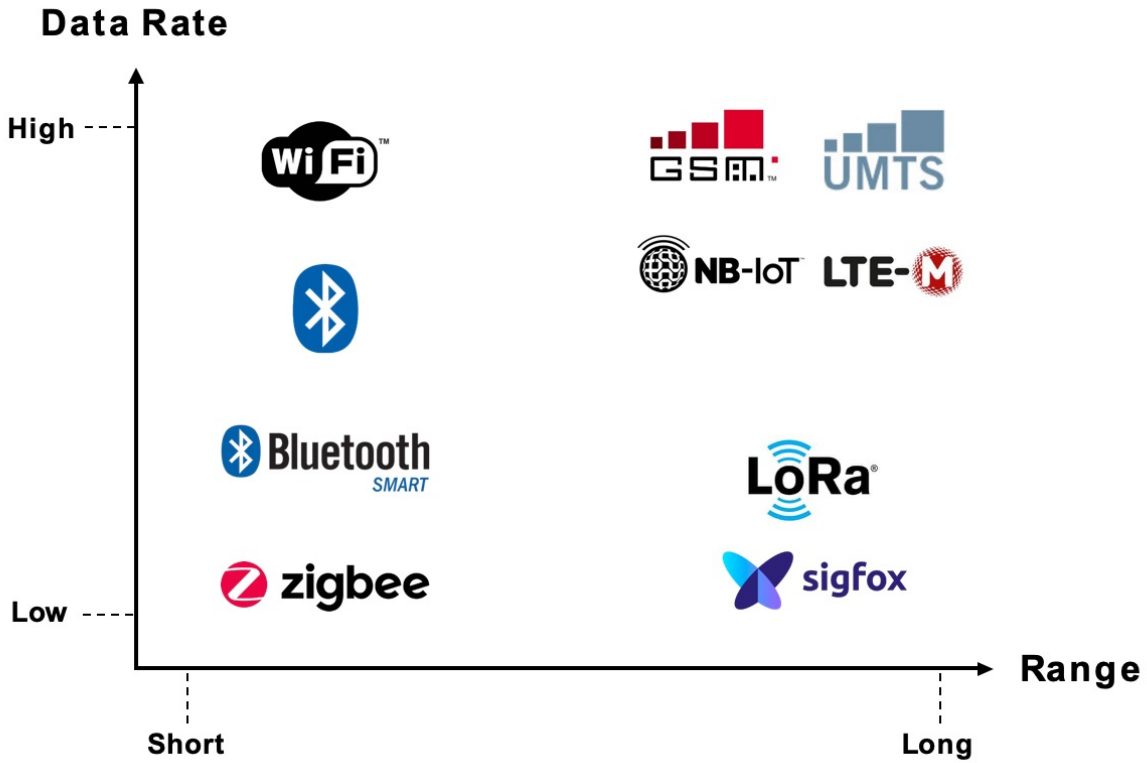


- Color, brightness, proximity and gesture sensor
- Digital microphone
- Motion, vibration and orientation sensor
- Temperature, humidity and pressure sensor
- Arm Cortex-M4 microcontroller and BLE module

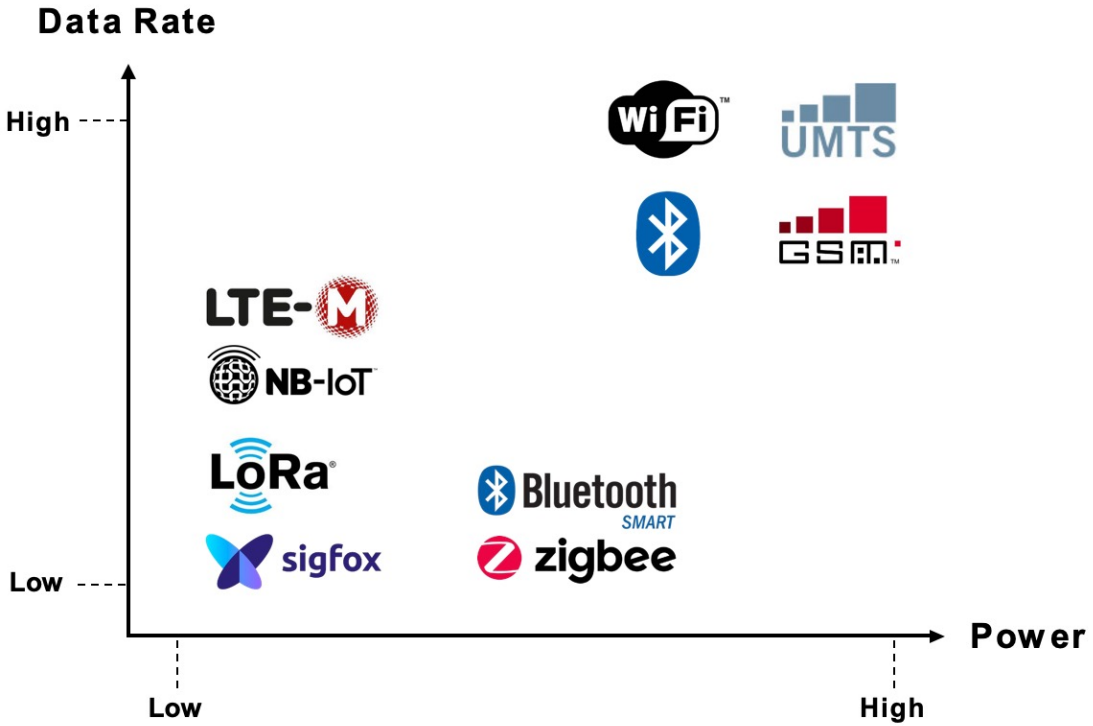
Two basic technologies for IoT

- ✓ LoRaWAN
- ✓ MQTT
- ✓ … and some experiments

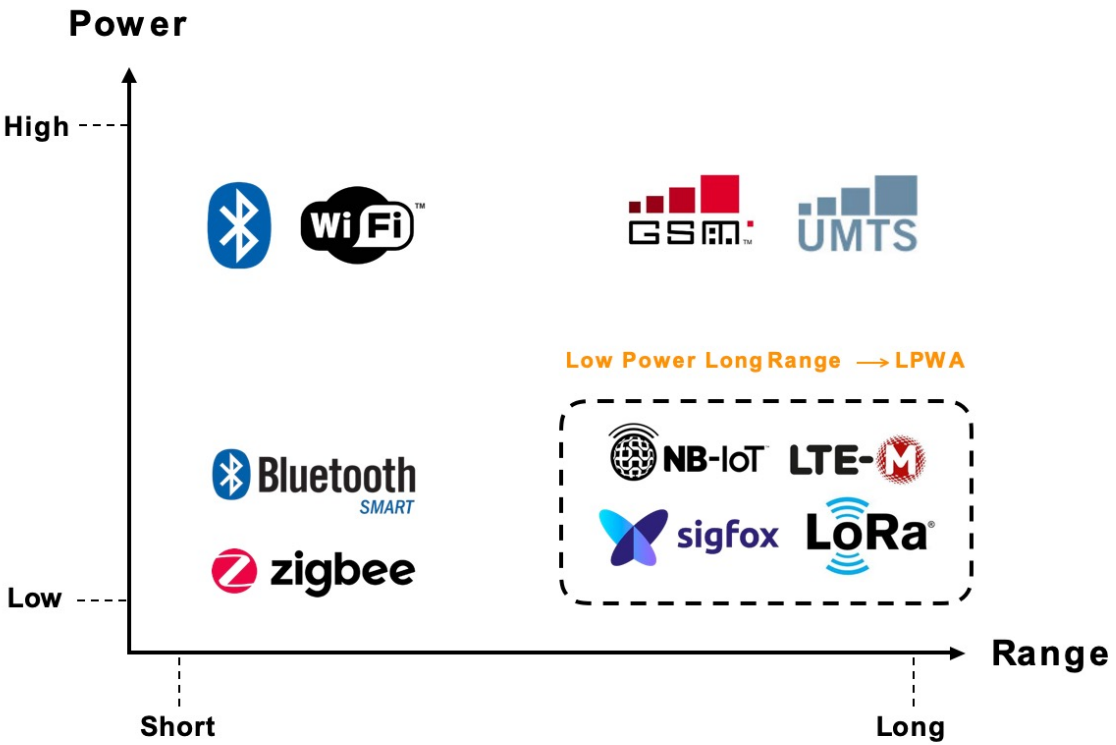
LPWAN: range vs data rate



LPWAN: power vs data rate

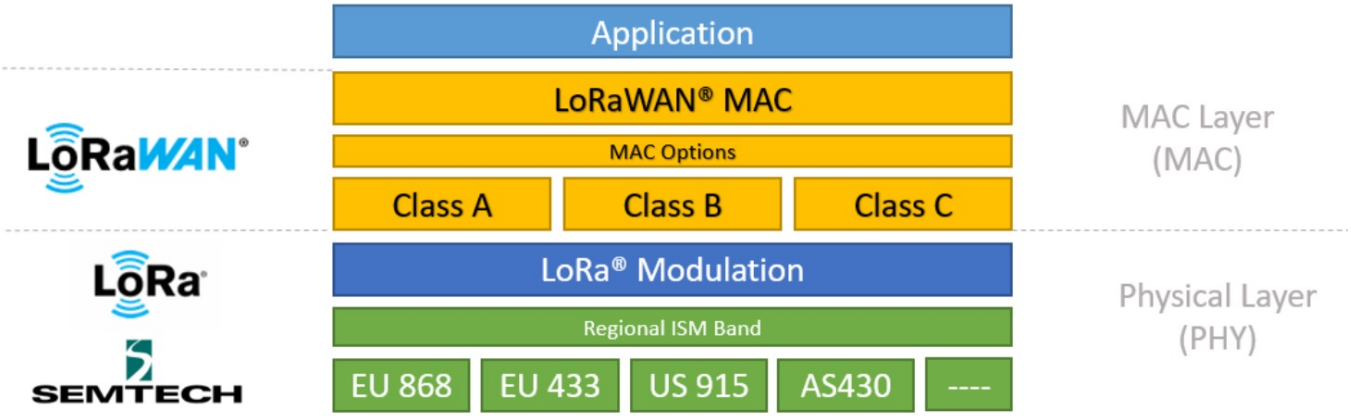


LPWAN: range vs power



LoRaWAN: a sub-gigahertz wireless technology

- LoRaWAN™ is a Low Power Wide Area Network (LPWAN) specification intended for wireless battery-operated Things in a regional, national or global network
  - by Semtech Corporation (http://www.semtech.com/)
- LoRaWAN™ defines the communication protocol and system architecture for the network, while the LoRa physical layer enables the long-range communication link.



https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/ ©

https://lora-alliance.org/

LoRa® Alliance

- The LoRa® Alliance is an open, non-profit association of members whose mission is:
  - "..promote and drive the success of the LoRaWAN® protocol as the leading open global standard for secure, carrier-grade IoT LPWAN connectivity…"
  - "To develop and promote LoRaWAN® technology and its ecosystem to deliver massive IoT"
- Specification is free to download:
  - https://lora-alliance.org/resource-hub/lorawan-104-specification-package

LoRaWAN® L2 1.0.4 Specification (TS001-1.0.4)

Authored by the LoRa Alliance Technical Committee

Technical Committee Chair and Vice-Chair:
A.YEGIN (Actility), O.SELLER (Semtech)

Editors:
T.KRAMP (Semtech), O.SELLER (Semtech)

Contributors (in alphabetical order):
A.BERTOLAUD (Gemalto), I.CALABRESE (A2A Smart City), J.CATALANO (Kerlink), J.DELCLEF (ST Microelectronics), V.DELPORT (Microchip Technology), P.DUFFY (Cisco), F.DYDUCH (Bouygues Telecom), T.EIRICH (Semtech), L.FERREIRA (Orange), Y.GAUDIN (Kerlink), S.GHAROUT (Orange), O.HERSENT (Actility), A.KASTTET (Birdz), D.KJENDAL (Senet), V.KLEBAN (Everynet), J.KNAPP (Semtech), T.KRAMP (Semtech), M.KUYPER (Semtech), P.KWOK (Objenious), M.LEGOURIEREC (Sagemcom), C.LEVASSEUR (Bouygues Telecom), M.LUIS (Semtech), M.PAULIAC (Gemalto), P.PIETRI (Orbiwise), O.SELLER (Semtech), D.SMITH (MultiTech), N.SORNIN (Semtech), R.SOSS (Actility), J.STOKKING (The Things Network), T.TASHIRO (M2B Communications), D.THOLL (Tektelic), P.THOMSEN (Orbiwise), A.YEGIN (Actility)

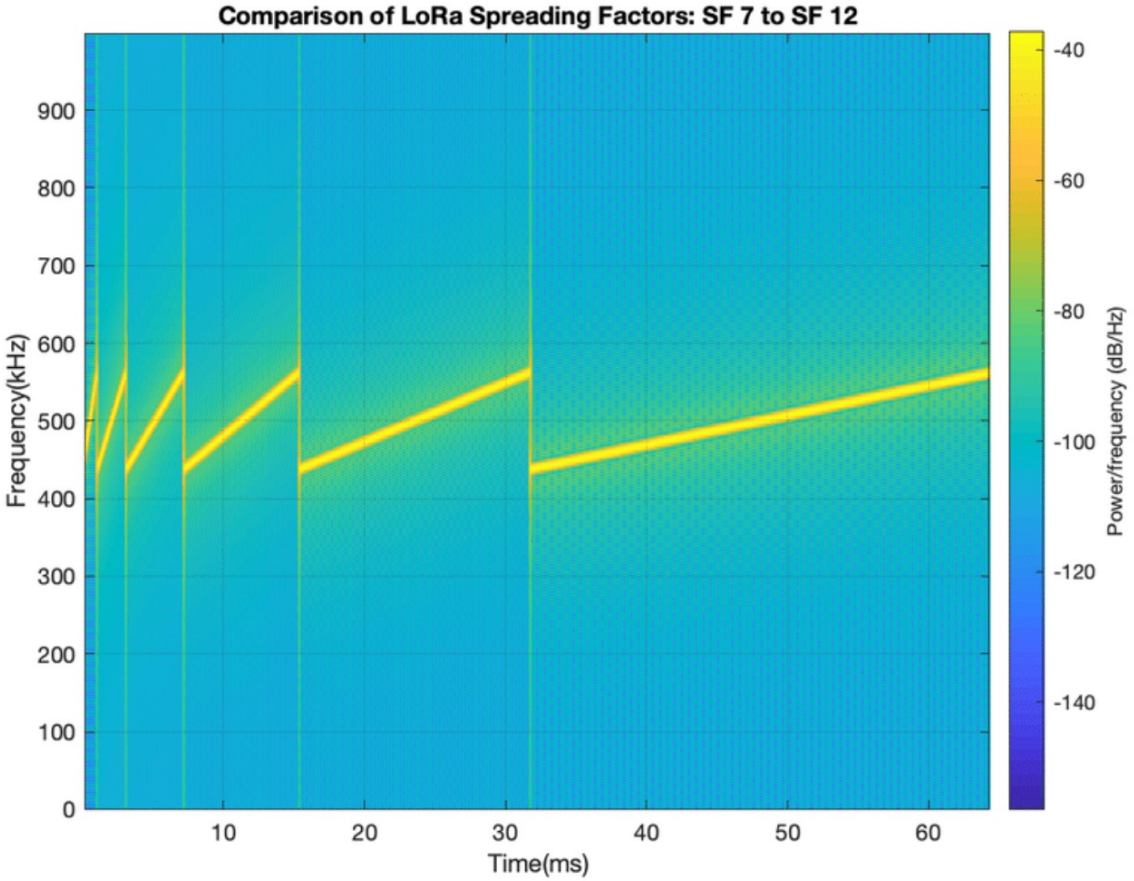
Version: 1.0.4
Date: October 2020
Status: Released

LoRaWAN specification depends on the region

	Europe	North America
Frequency band	867-869MHz	902-928MHz
Channels	10	64 + 8 +8
Channel BW Up	125/250kHz	125/500kHz
Channel BW Dn	125kHz	500kHz
TX Power Up	+14dBm	+20dBm typ (+30dBm allowed)
TX Power Dn	+14dBm	+27dBm
SF Up	7-12	7-10
Data rate	250bps- 50kbps	980bps-21.9kpbs
Link Budget Up	155dB	154dB
Link Budget Dn	155dB	157dB

https://lora-alliance.org/resource-hub/rp2-101-lorawanr-regional-parameters-0

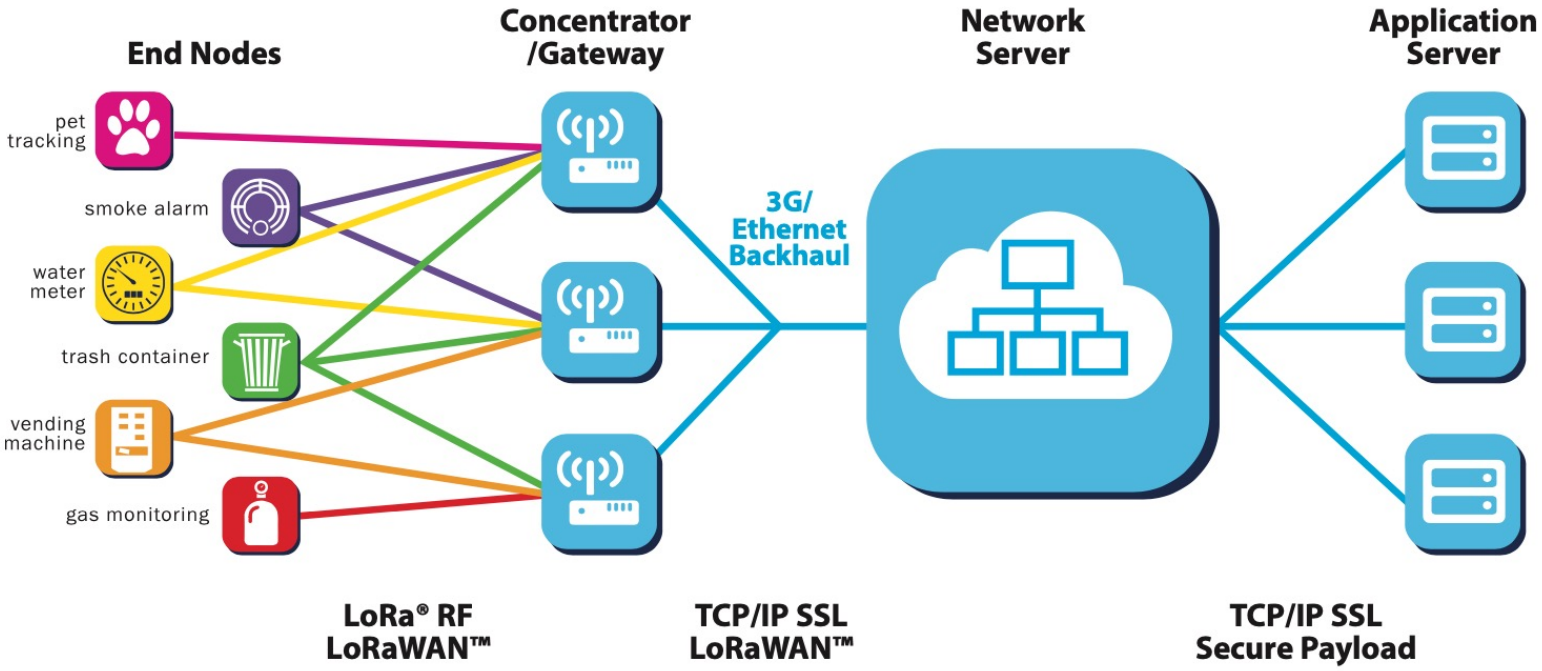
Chirp Modulation



https://www.youtube.com/watch?v=dxYY097QNs0

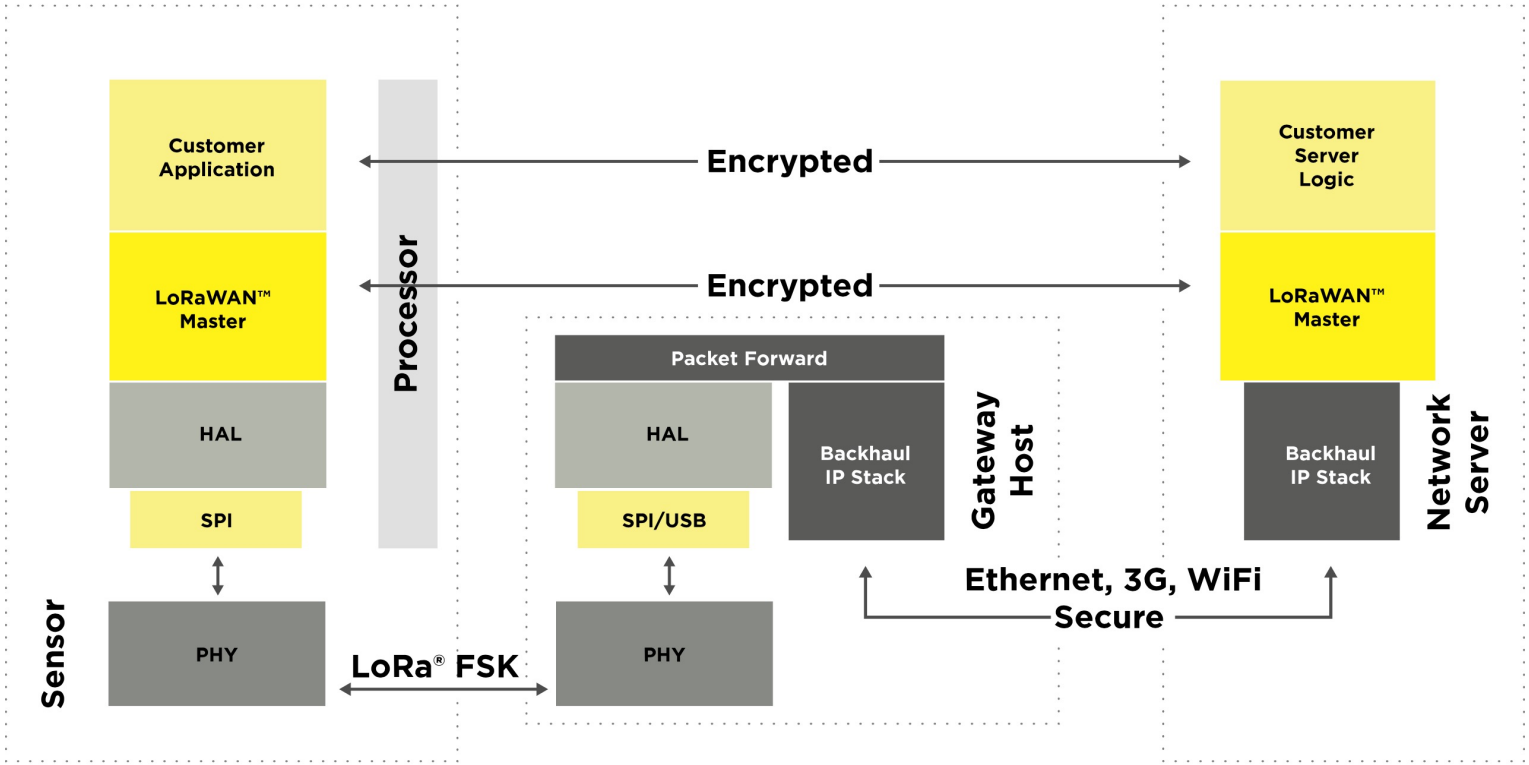
Kim, Dong-Hoon & Lee, Eun-Kyu & Kim, Jibum. (2019). Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed. Sustainability. 11. 1917. 10.3390/su11071917.

LoRaWAN network architecture



LoRa Alliance ©

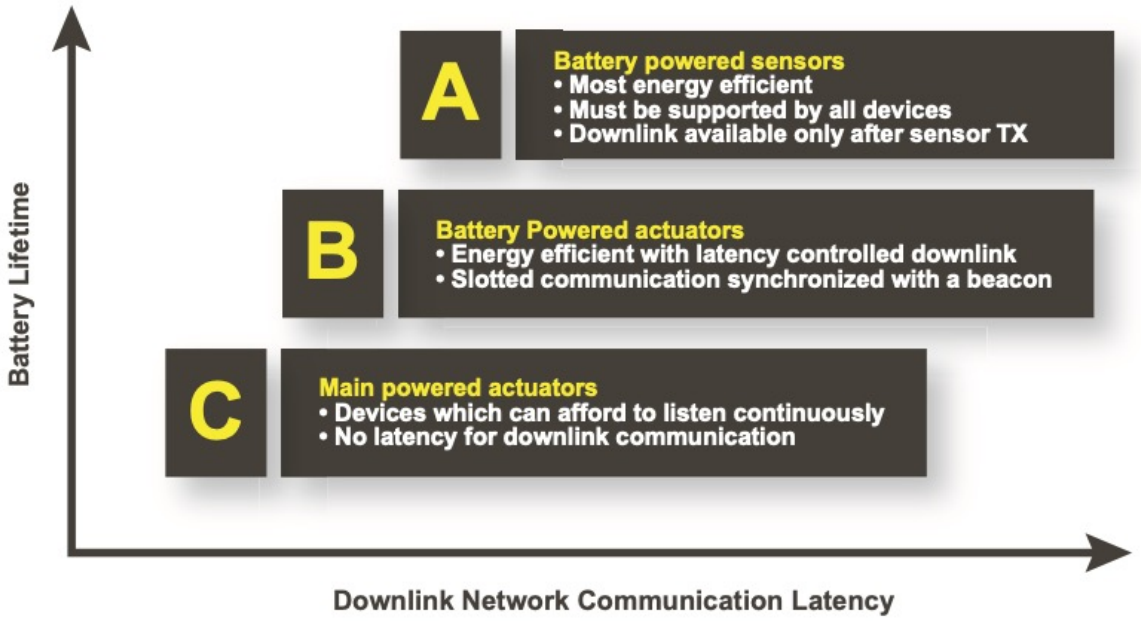
LoRaWAN data flow



HAL: Hardware Abstraction Layer

Three types of devices

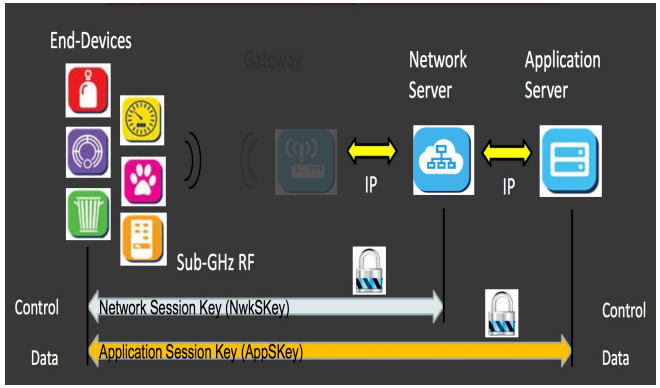
- LoRaWAN has three different classes of end-point devices to address the different needs reflected in the wide range of applications:



LoRa® Alliance Technical Marketing Workgroup

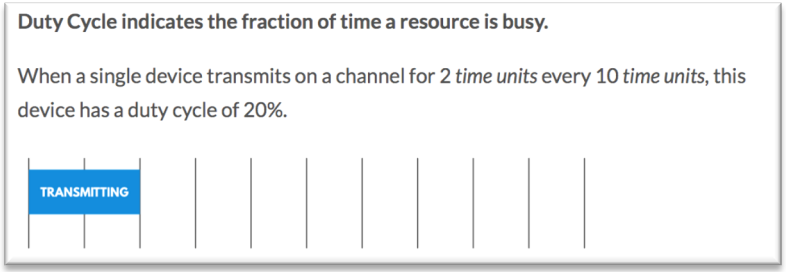
Devices activation

- LoRaWAN devices have a 64-bits unique identifier (`DevEUI`) that is assigned to the device by the chip manufacturer.
- However, all communication is done with a dynamic 32 bit device address (`DevAddr`) of which 7 bits are fixed (Network Server), leaving 25 bits that can be assigned to individual devices with a procedure called **Activation**.
- To exchange this information, two activation methods are available:
  - Over-the-Air Activation (OTAA)
    - Devices perform a join-procedure with the network, during which a dynamic DevAddr is assigned and security keys are negotiated with the device
  - Activation By Personalization (ABP)
    - Hardcode the DevAddr as well as the security keys in the device.



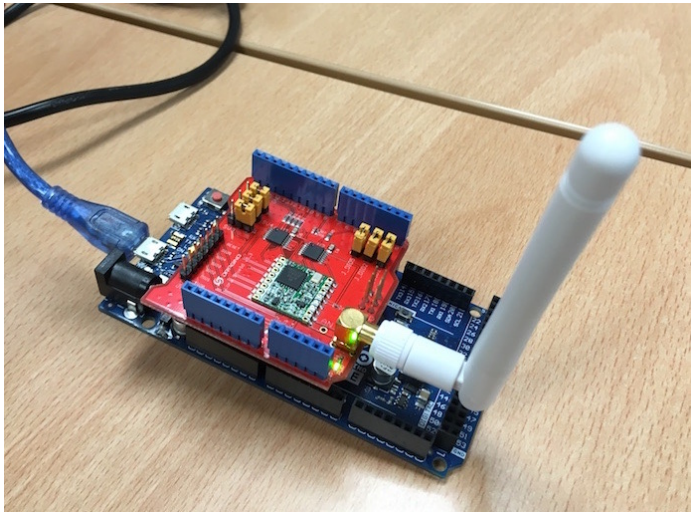
Maximum Duty Cycle

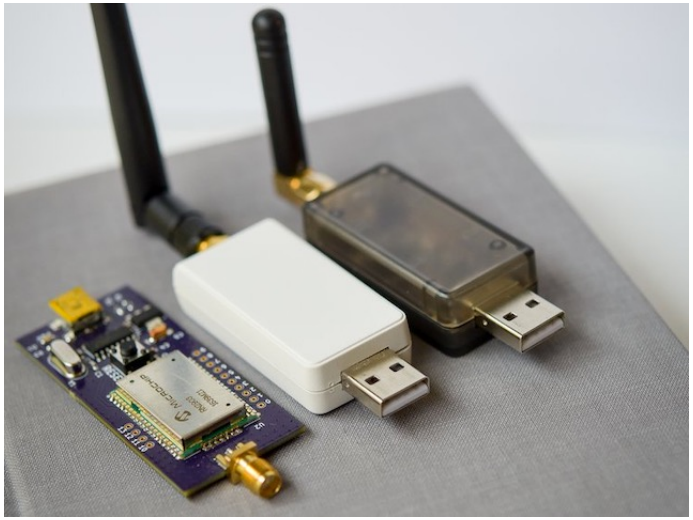
- The **duty cycle of radio devices is often regulated by government**.
  - In Europe, duty cycles are regulated by section 7.2.3 of the ETSI EN300.220 standard.



- Additionally, the **LoRaWAN specification dictates duty cycles for the join frequencies**, the frequencies devices of all LoRaWAN-compliant networks use for over-the-air activations (OTAA) of devices. In most regions this duty cycle is set to 1%.
- Finally, on "community network" like TTN there typically is a **Fair Access Policy** that limits the uplink airtime to 30 seconds per day (24 hours) per node and the downlink messages to 10 messages per day (24 hours) per node.

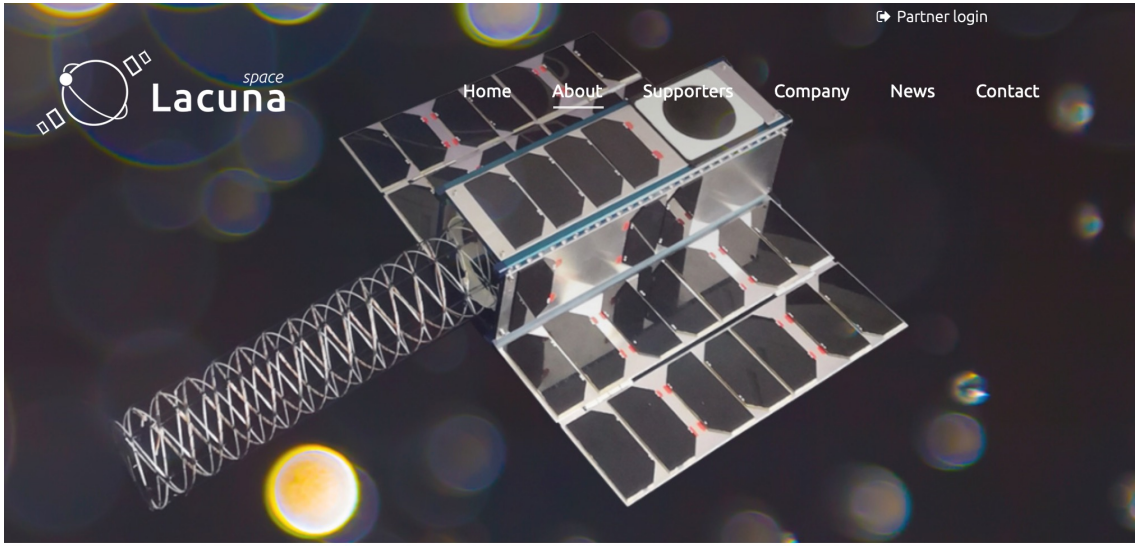
Some devices





https://lacuna.space

Nice project

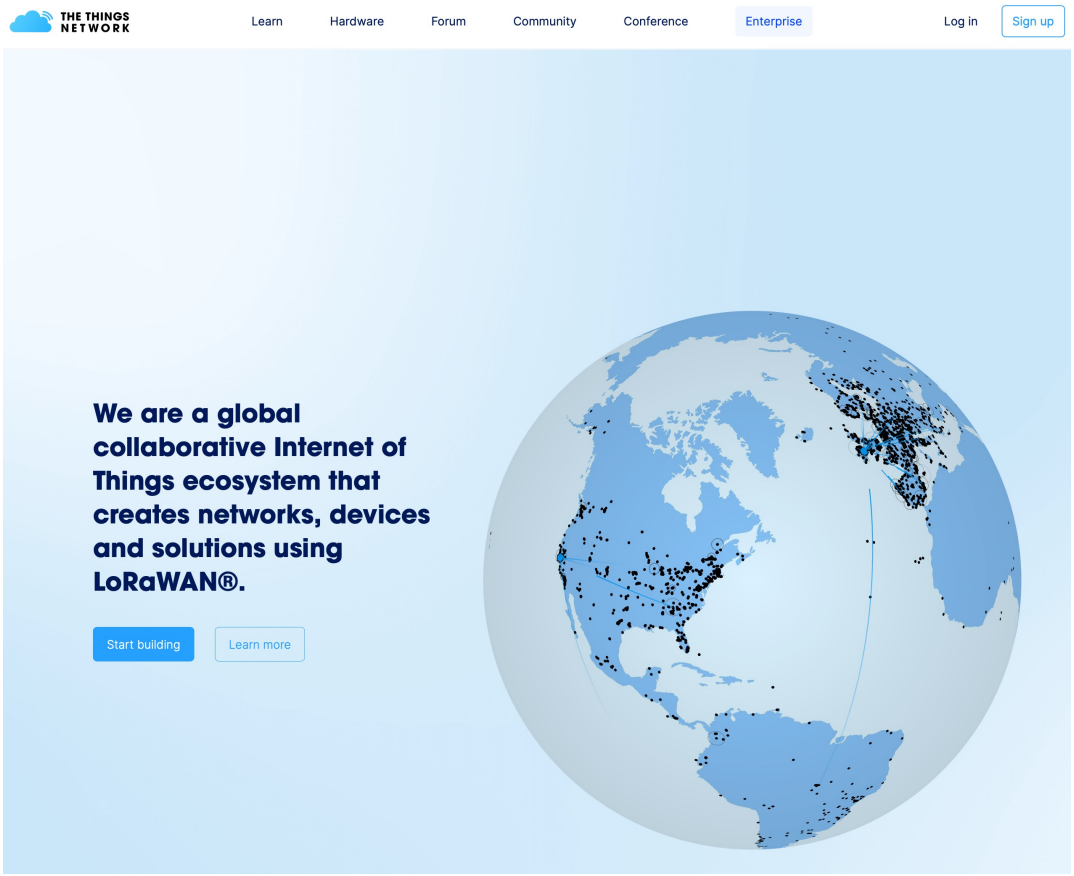


An ultra-low cost tracking and sensor detection service for small amounts of data

– think of it as 'things' rather than people tweeting short messages.

https://www.thethingsnetwork.org

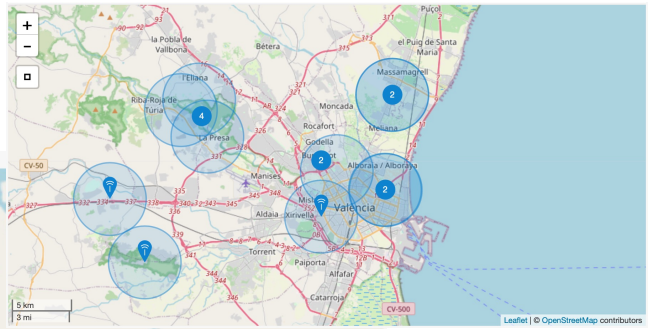
The Things Network (TTN)

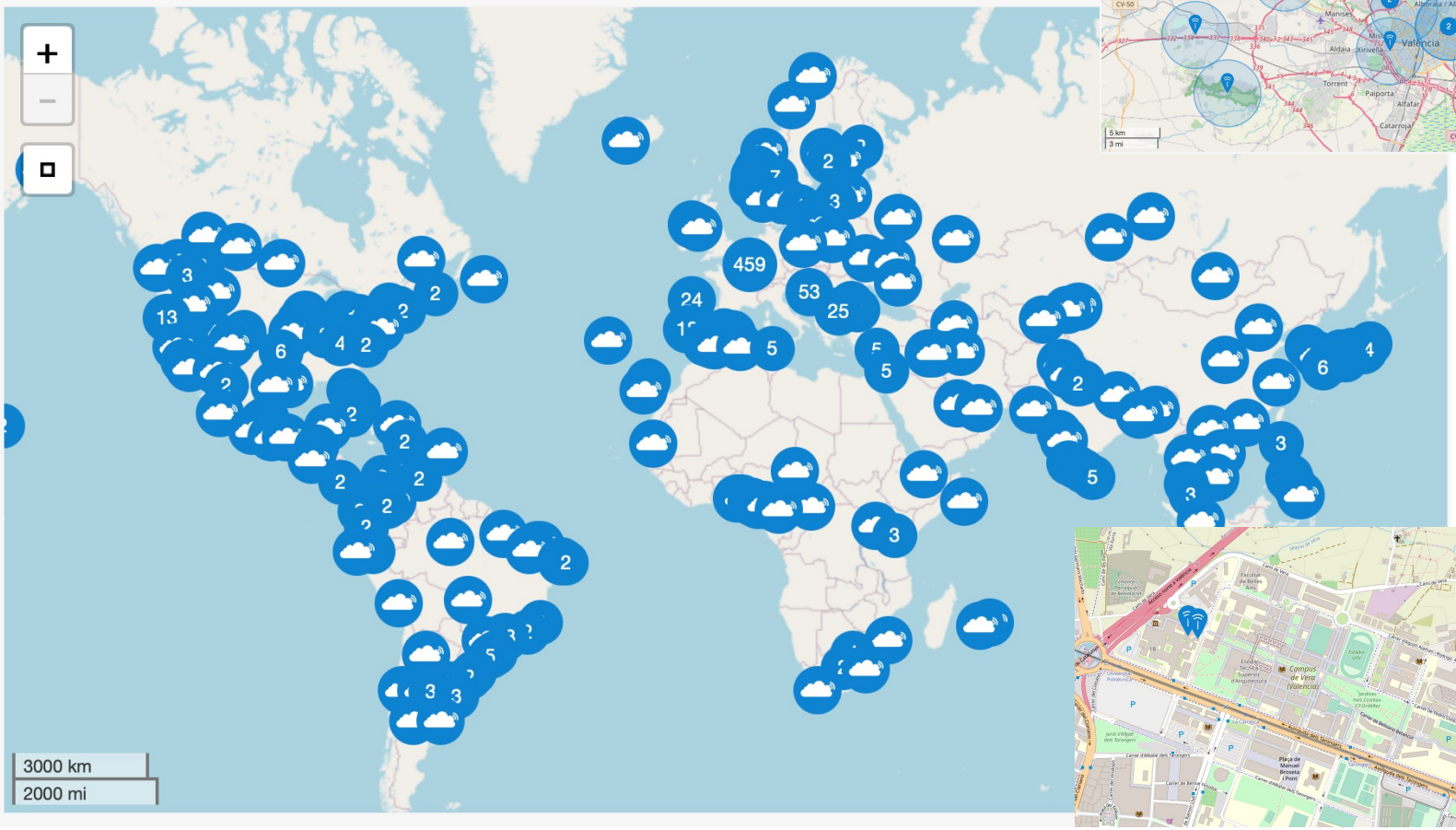


27.2M	151	970	154.6K	21.3K
Messages today	Countries	Certified developers	Members	Gateways

The Things Network (TTN)

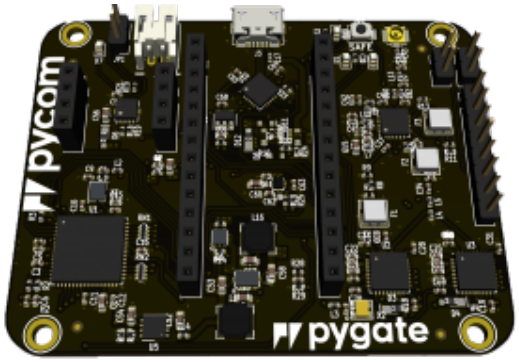
Currently (1oct2021) 21300 gateways active worldwide

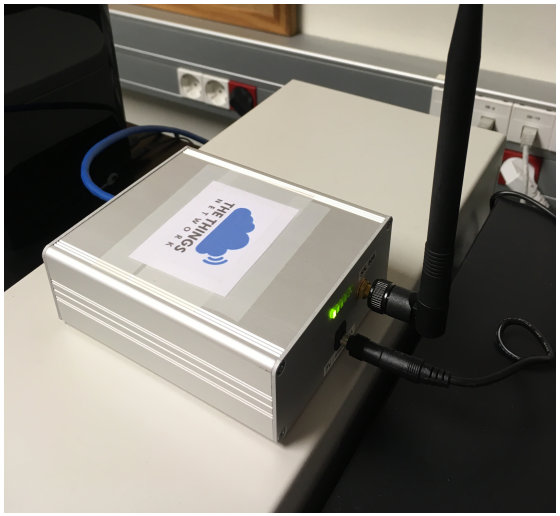


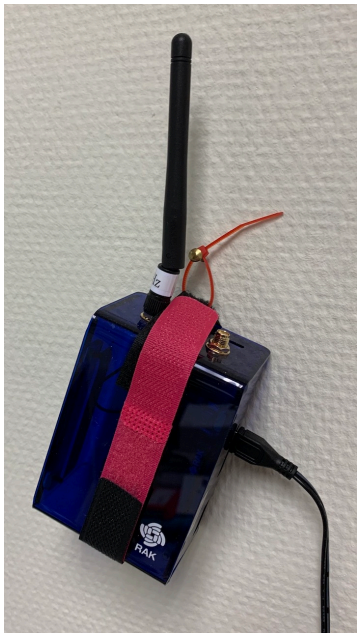


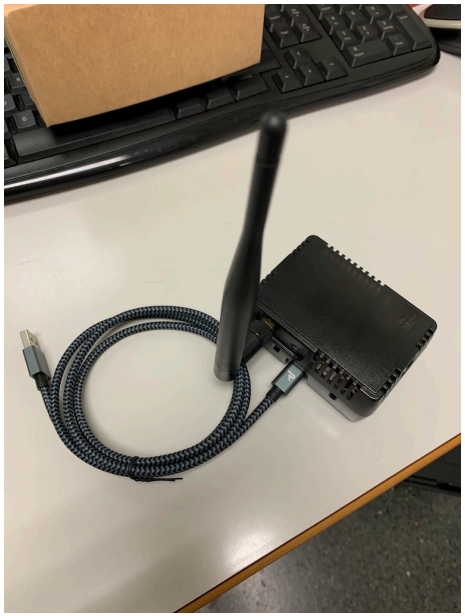
https://www.thethingsnetwork.org/community

GRC Gateways

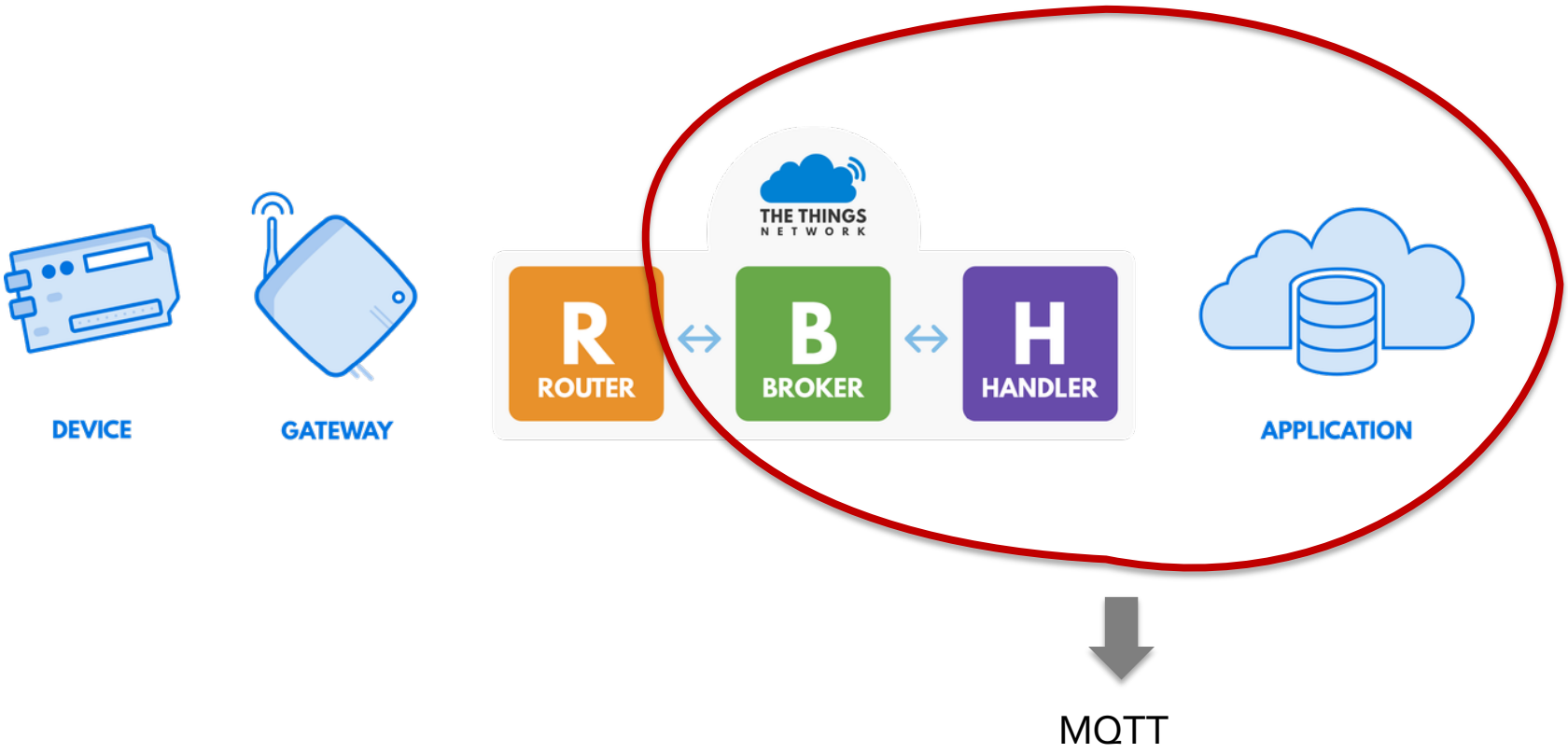




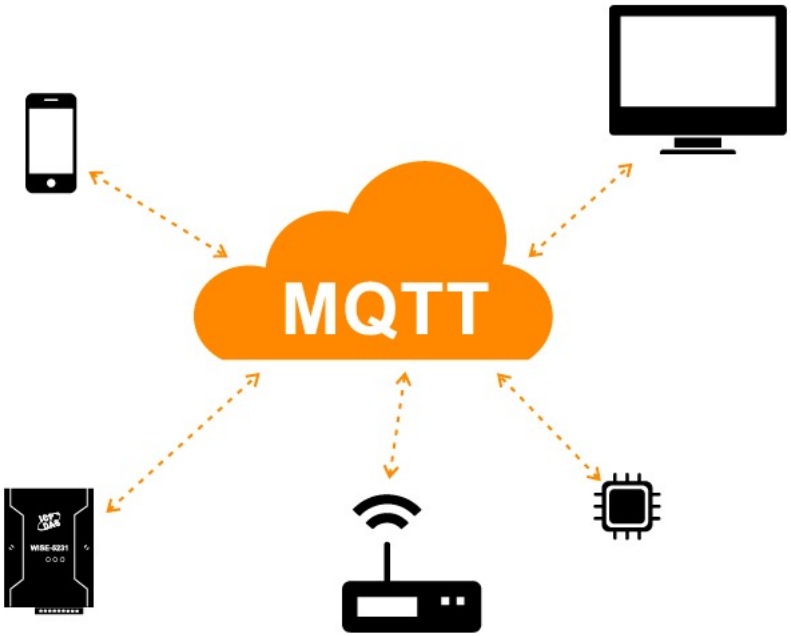




The Things Network (TTN)

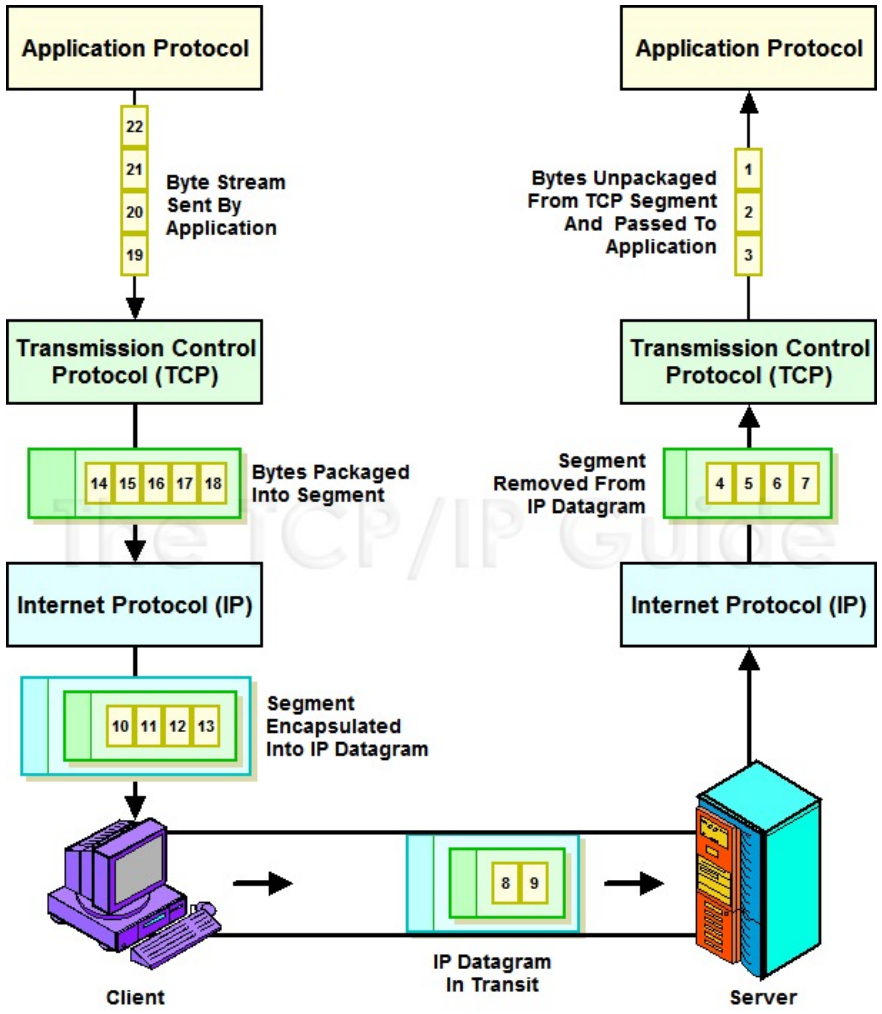


MQTT basics

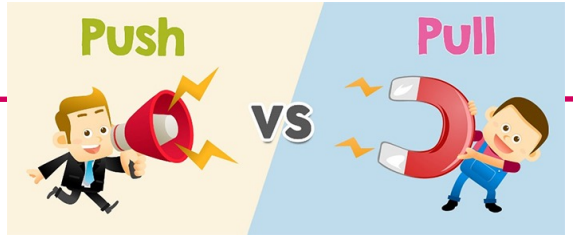


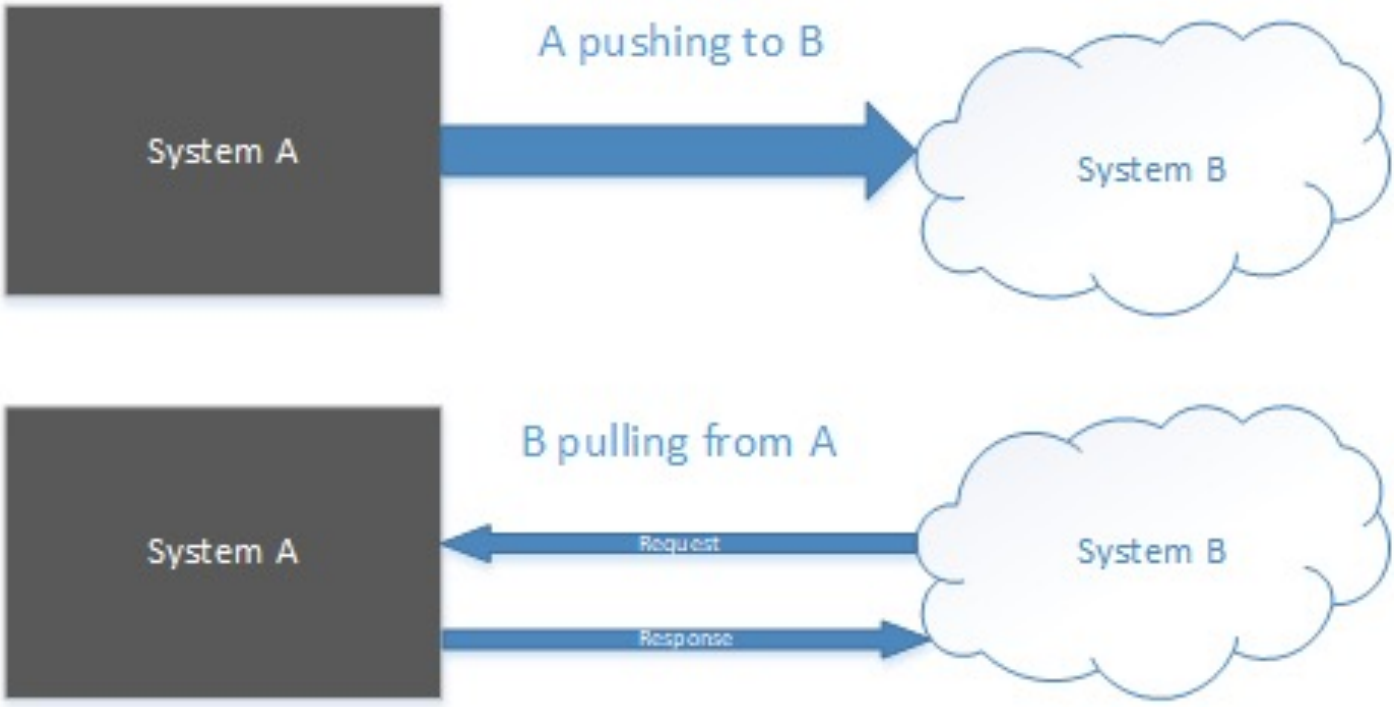
From "byte streams" to "messages"

- The "old" vision of data communication was based on **reliable byte streams**, i.e., TCP
- Nowadays **messages interchange** is becoming more common
  - E.g., Twitter, Whatsapp, Instagram, Snapchat, Facebook,…
- Actually is not that new…
  - emails: SMTP+MIME,
  - FTP,



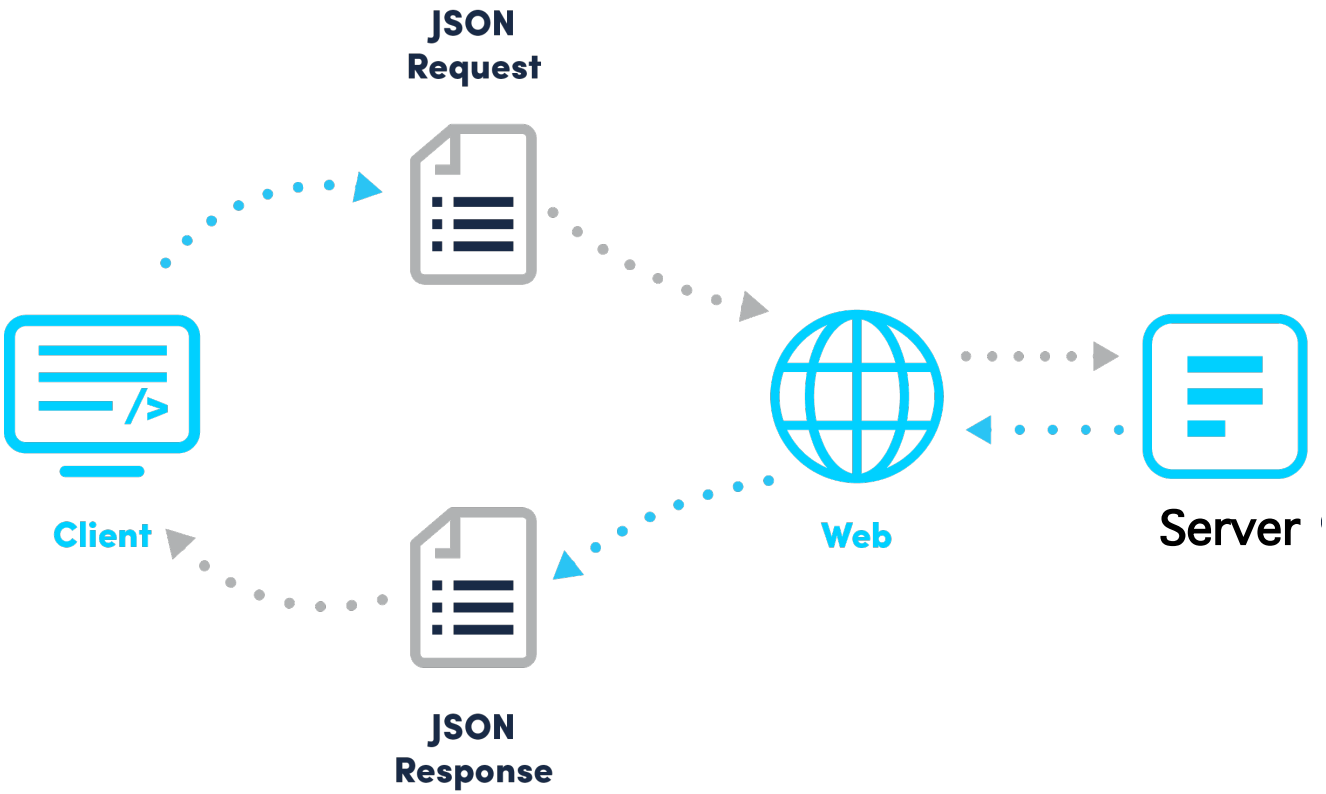
Ways to interchange "messages"





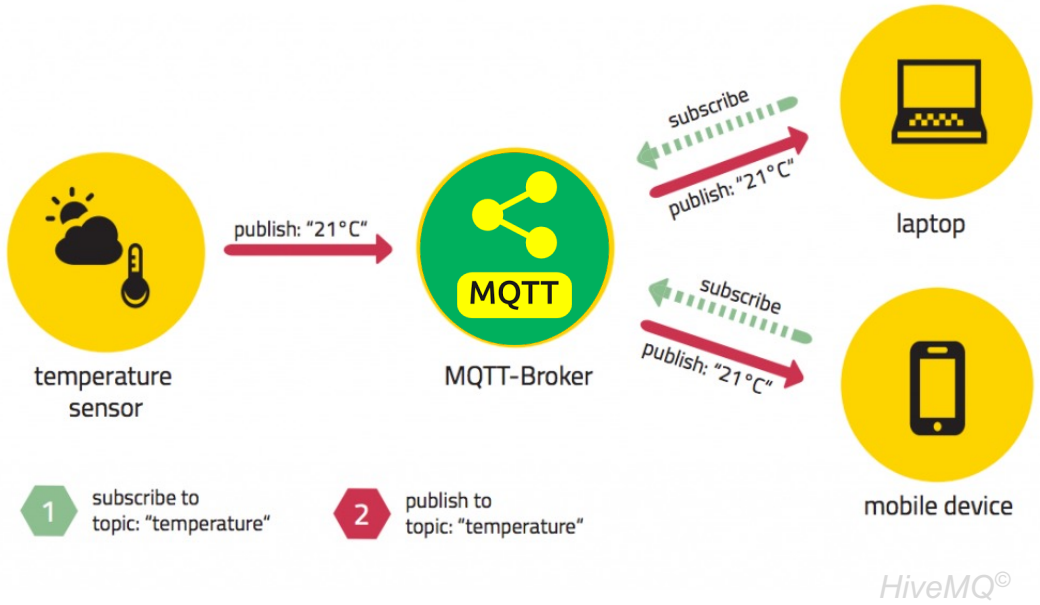
Request/response approach

- **REST**: Representational State Transfer
- Widely used; based on HTTP
- *Lighter version: **CoAP** (Constrained Application Protocol)*



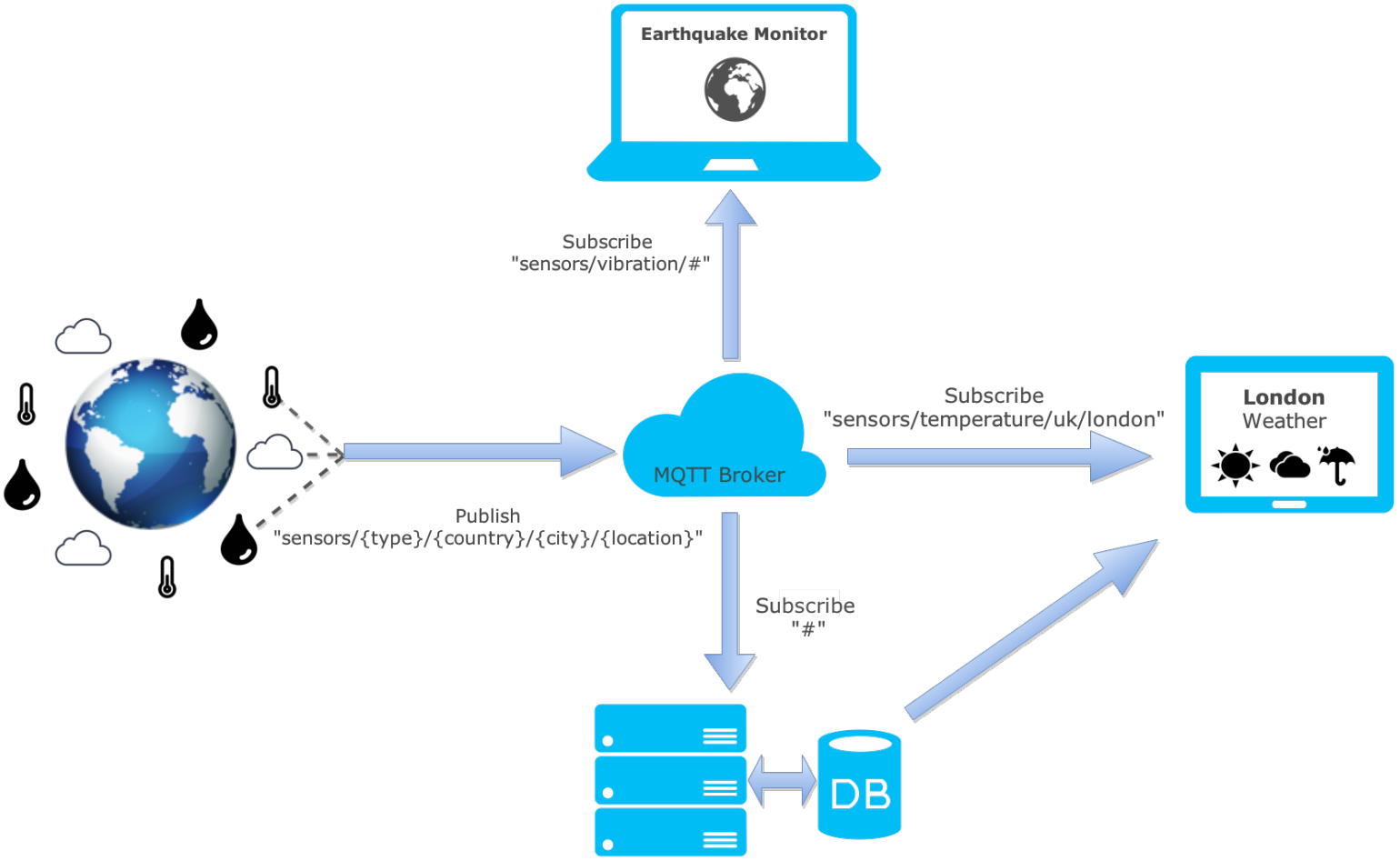
Pub/sub approach

- Pub/Sub separate a client, who is sending a message about a specific **topic**, called publisher, from another client (or more clients), who is receiving the message, called subscriber.
- There is a third component, called broker, which is known by both the publisher and subscriber, which filters all incoming messages and distributes them accordingly.



- Various protocols:
  MQTT, AMQP, XMPP (was Jabber)
- Growing technique
  E.g., https://cloud.google.com/iot/docs/how-tos/mqtt-bridge

An example



Source: https://zoetrope.io/tech-blog/brief-practical-introduction-mqtt-protocol-and-its-application-iot

Message Queuing Telemetry Transport

- A lightweight publish-subscribe protocol that can run on embedded devices and mobile platforms ➔ http://mqtt.org/
  - Low power usage.
  - Binary compressed headers
  - Maximum message size of 256MB
    - not really designed for sending large amounts of data
    - better at a high volume of low size messages.
- Documentation sources:
  - The MQTT community wiki:
    - `https://github.com/mqtt/mqtt.github.io/wiki`
  - A very good tutorial:
    - `http://www.hivemq.com/mqtt-essentials/`

Some details about versions

- **MQTT 3.1.1 is the current version of the protocol.**
  - Standard document here:
    - http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html
  - October 29th 2014: MQTT was officially approved as OASIS Standard.
    - https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=mqtt

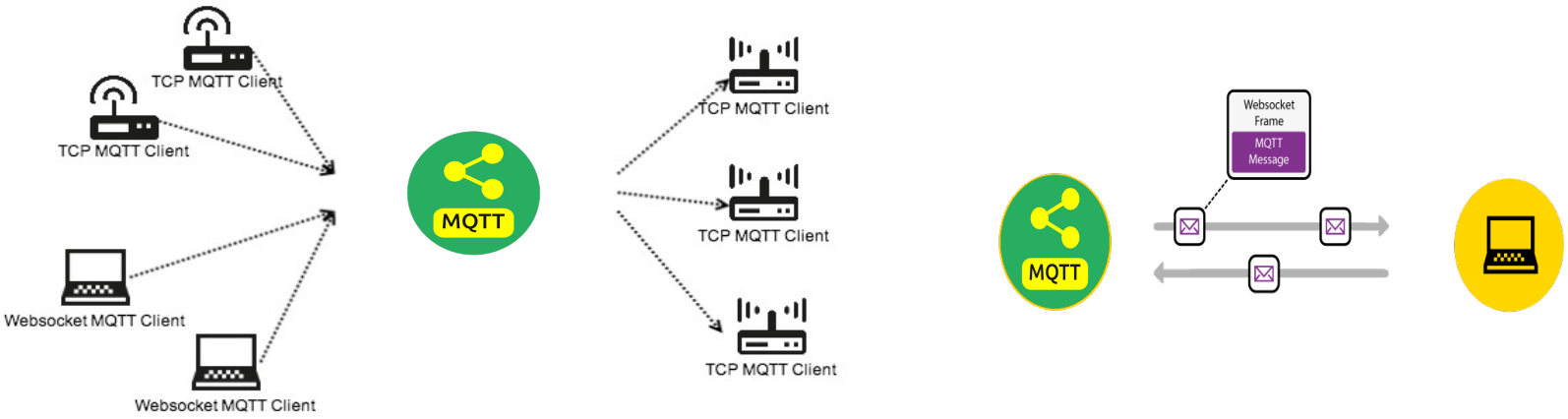
- MQTT v5.0 is the successor of MQTT 3.1.1
  - Current status: Committee Specification 02 (7 March 2019)
    - http://docs.oasis-open.org/mqtt/mqtt/v5.0/cs02/mqtt-v5.0-cs02.html
    - **Not backward compatible**; too many new things are introduced so existing implementations have to be revisited, for example:

1. More extensibility → user properties
2. Improved error reporting (Reason Code & Reason String)
3. Performance improvements and improved support for small clients
   - shared subscriptions
   - topic alias
4. Formalized common patterns → payload format description
5. Improved authentication

MQTT works on top of…

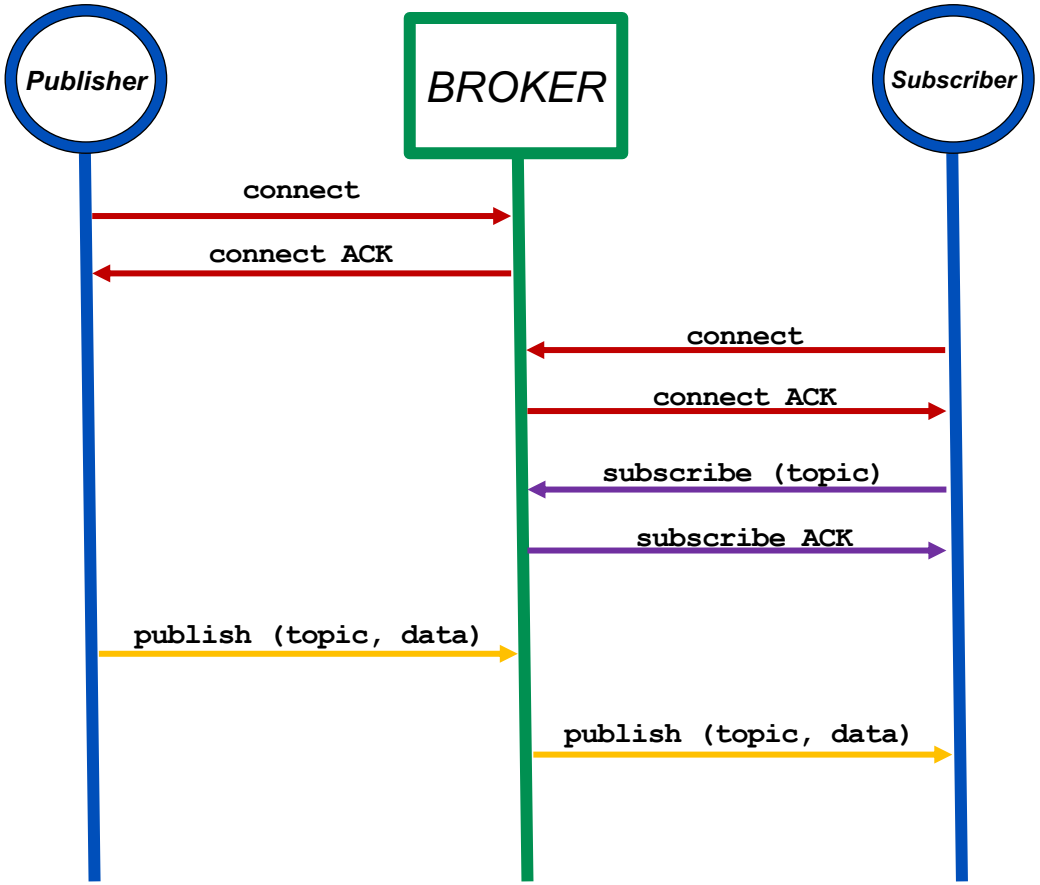
- mainly of TCP
  - There is also the closely related MQTT for Sensor Networks (MQTT-SN) where TCP is replaced by UDP → TCP stack is too complex for WSN

- websockets can be used, too!
  - Websockets allows you to receive MQTT data directly into a web browser.



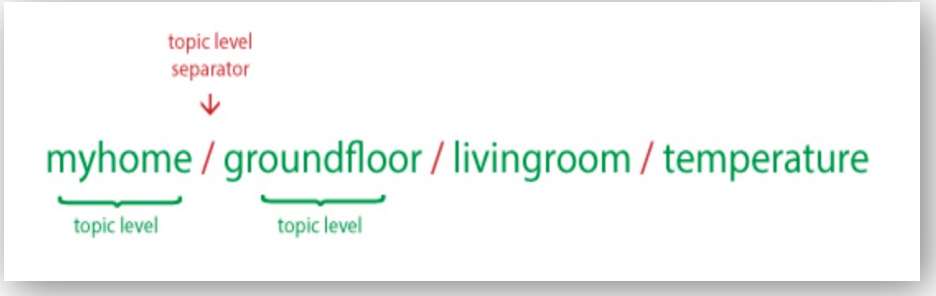
- Both, TCP & websockets can work on top of "Transport Layer Security (TLS)" (and its predecessor, Secure Sockets Layer (SSL))

Publish/subscribe interactions sequence



Topics

- MQTT Topics are structured in a hierarchy similar to folders and files in a file system using the forward slash (/) as a delimiter.
- Allow to create a user friendly and self descriptive **naming structures**



- Topic names are:
  - Case sensitive
  - use UTF-8 strings.
  - Must consist of at least one character to be valid.
- Except for the $SYS topic **there is no default or standard topic structure.**

Special $SYS/ topics

- $SYS/broker/clients/connected
- $SYS/broker/clients/disconnected
- $SYS/broker/clients/total
- $SYS/broker/messages/sent
- $SYS/broker/uptime

Topics wildcards

- Topic subscriptions can have wildcards. These enable nodes to subscribe to groups of topics that don't exist yet, allowing greater flexibility in the network's messaging structure.
  - '+' matches anything at a given tree level
  - '#' matches a whole sub-tree
- Examples:
  - Subscribing to topic `house/#` covers:
    - house/room1/main-light
    - house/room1/alarm
    - house/garage/main-light
    - house/main-door
  - Subscribing to topic `house/+/main-light` covers:
    - house/room1/main-light
    - house/room2/main-light
    - house/garage/main-light
  - but doesn't cover
    - house/room1/side-light
    - house/room2/side-light

A few words on security

- MQTT has the option for Transport Layer Security (TLS) encryption.
- MQTT also provides username/password authentication with the broker.
  - Note that the password is transmitted in clear text. Thus, be sure to use TLS encryption if you are using authentication.

"It's not just you. We're all insecure in one way or another."

A few words on security

Smart homes can be easily hacked via unsecured MQTT servers

https://www.helpnetsecurity.com/2018/08/20/unsecured-mqtt-servers/

In fact, by using the Shodan IoT search engine, Avast researchers found over 49,000 MQTT servers exposed on the Internet and, of these, nearly 33,000 servers have no password protection, allowing attackers to access them and all the messages flowing through it.

TOTAL RESULTS

49,197

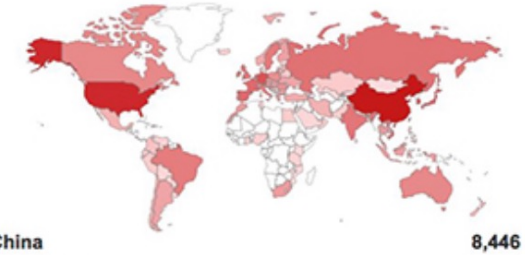
TOP COUNTRIES

China	12,151
United States	8,257
Germany	3,092
Korea, Republic of	2,003
Hong Kong	2,002

TOTAL RESULTS

32,888

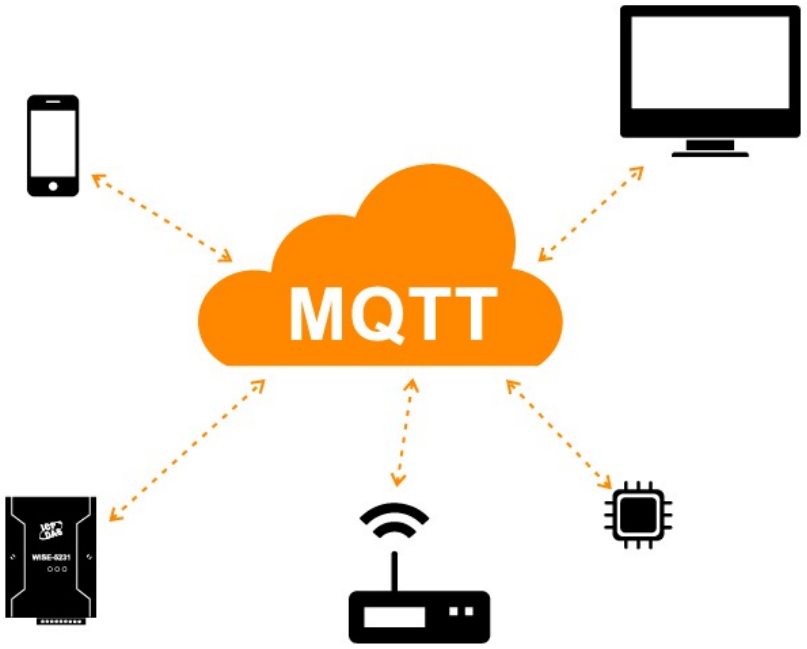
TOP COUNTRIES



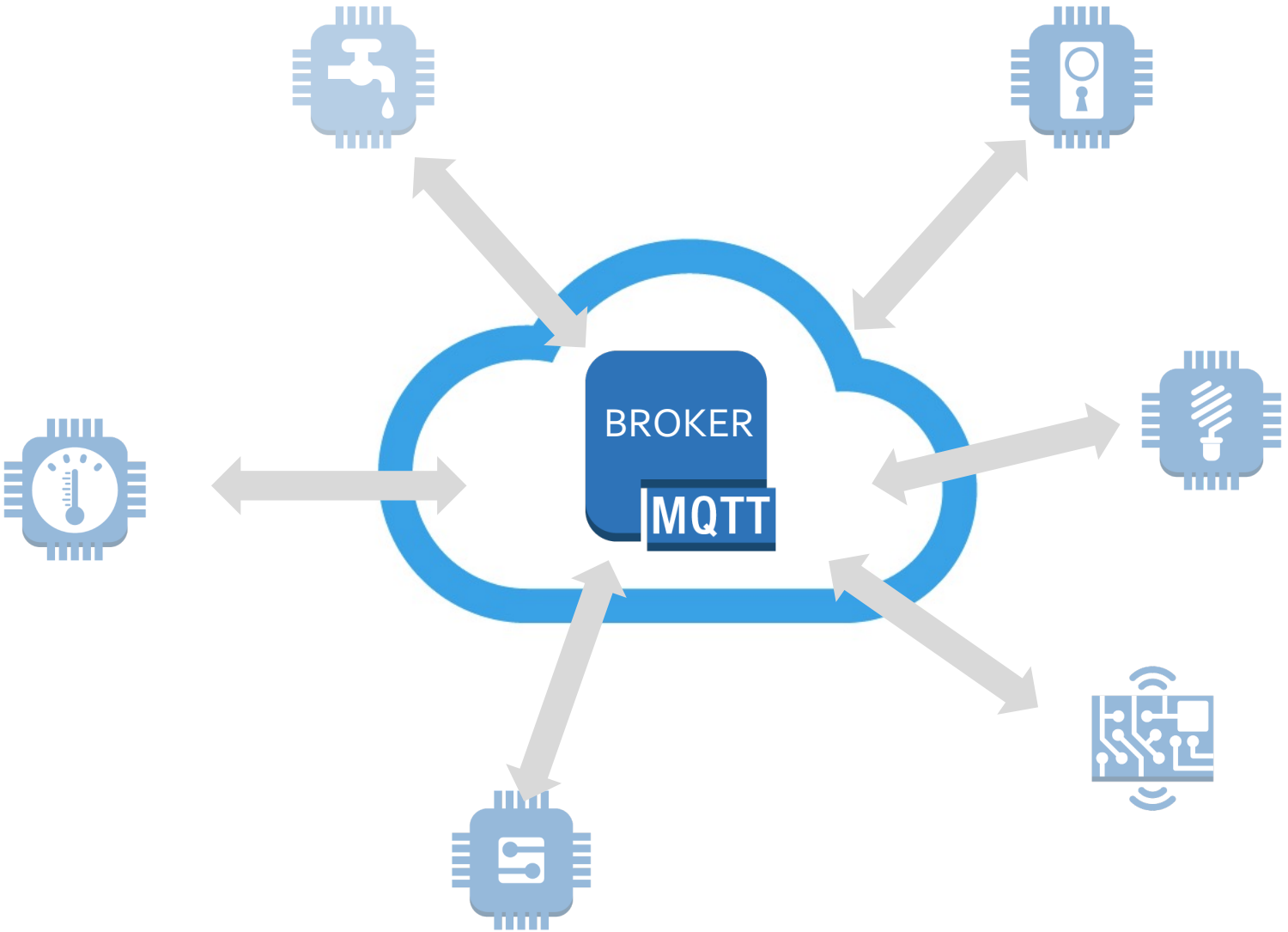
China	8,446
United States	4,733
Germany	1,719
Hong Kong	1,614
Taiwan	1,565

Intro to MQTT

- ✓ Brokers and clients



Creating a broker



Available MQTT brokers

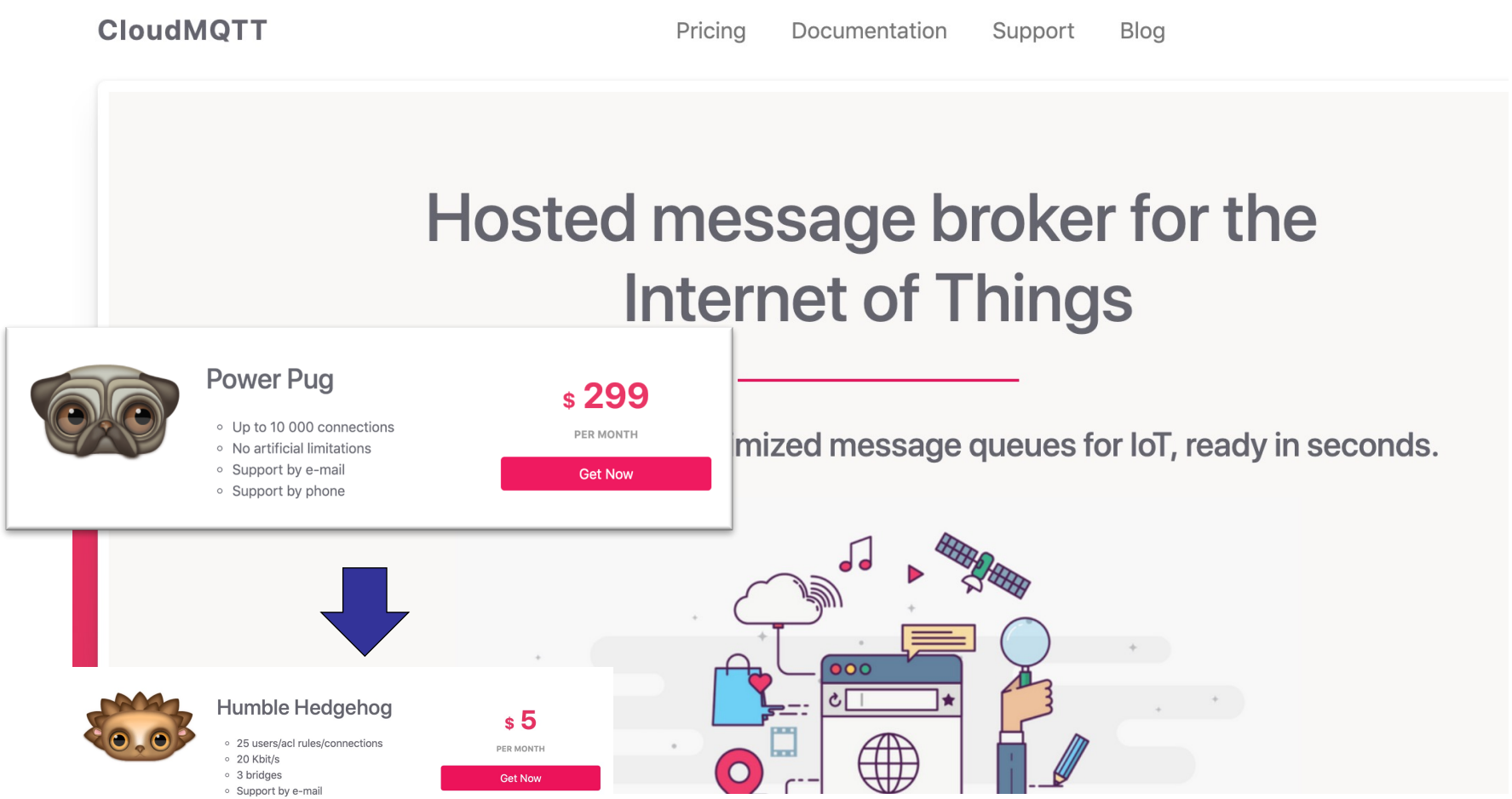
- The most widely used are:
  - http://mosquitto.org/
    - man page: https://mosquitto.org/man/mosquitto-8.html
  - http://www.hivemq.com/
    - The standard trial version only supports 25 connections.

- And also:
  - https://www.rabbitmq.com/mqtt.html
  - http://activemq.apache.org/mqtt.html

- A quite complete list can be found here:
  - https://github.com/mqtt/mqtt.github.io/wiki/servers

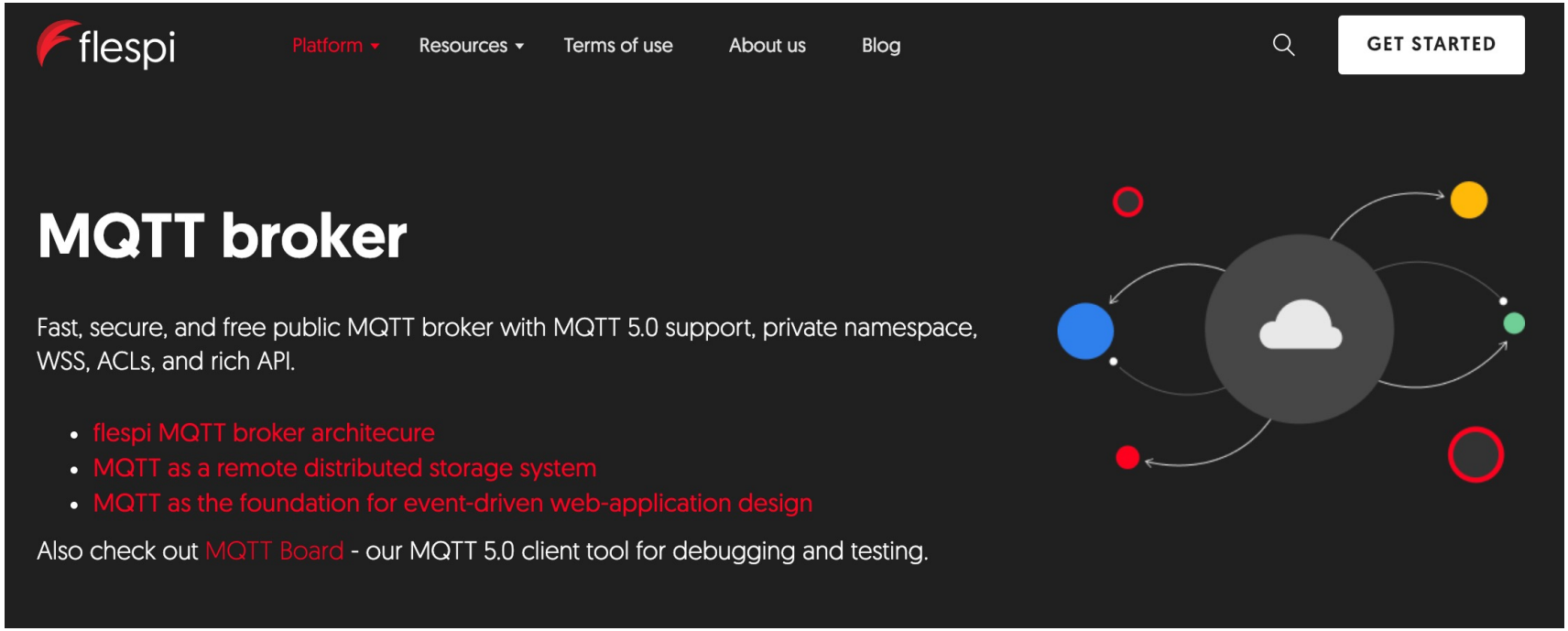
Cloud based MQTT brokers: CloudMQTT

https://www.cloudmqtt.com/ ➔ based on Mosquitto



Cloud based brokers: flespi

https://flespi.com/mqtt-broker



Cloud based brokers: flespi

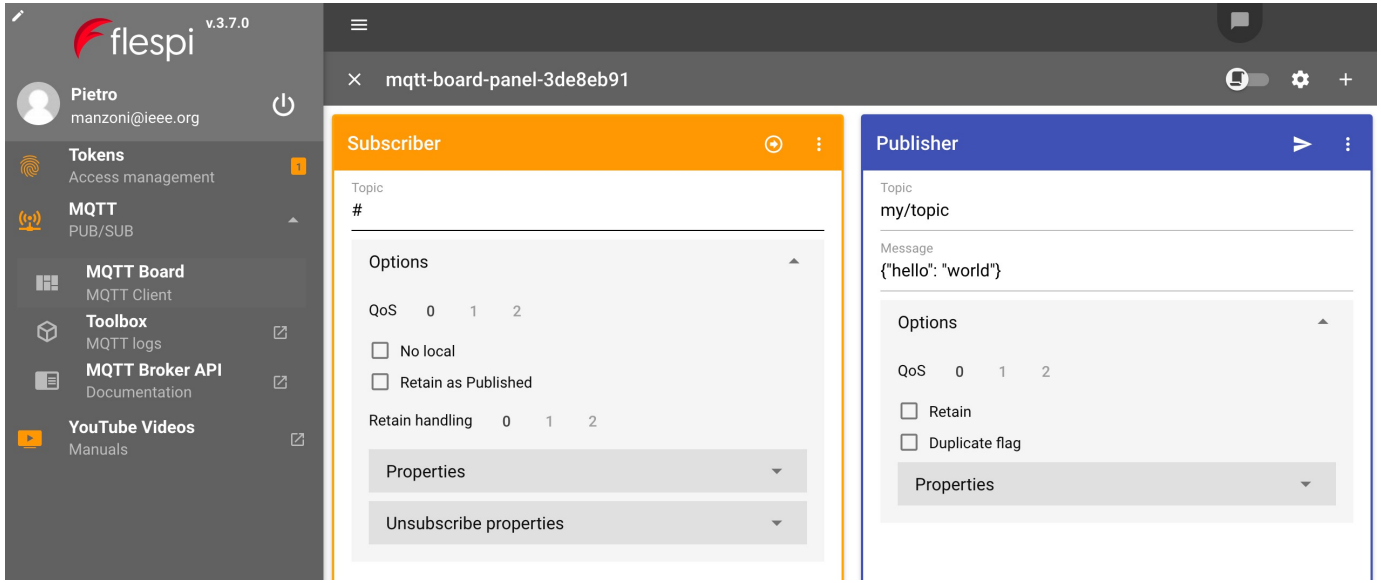
Terms of use

Free $0/mo

MQTT

100 active MQTT sessions

https://flespi.io/#/panel/mqttboard



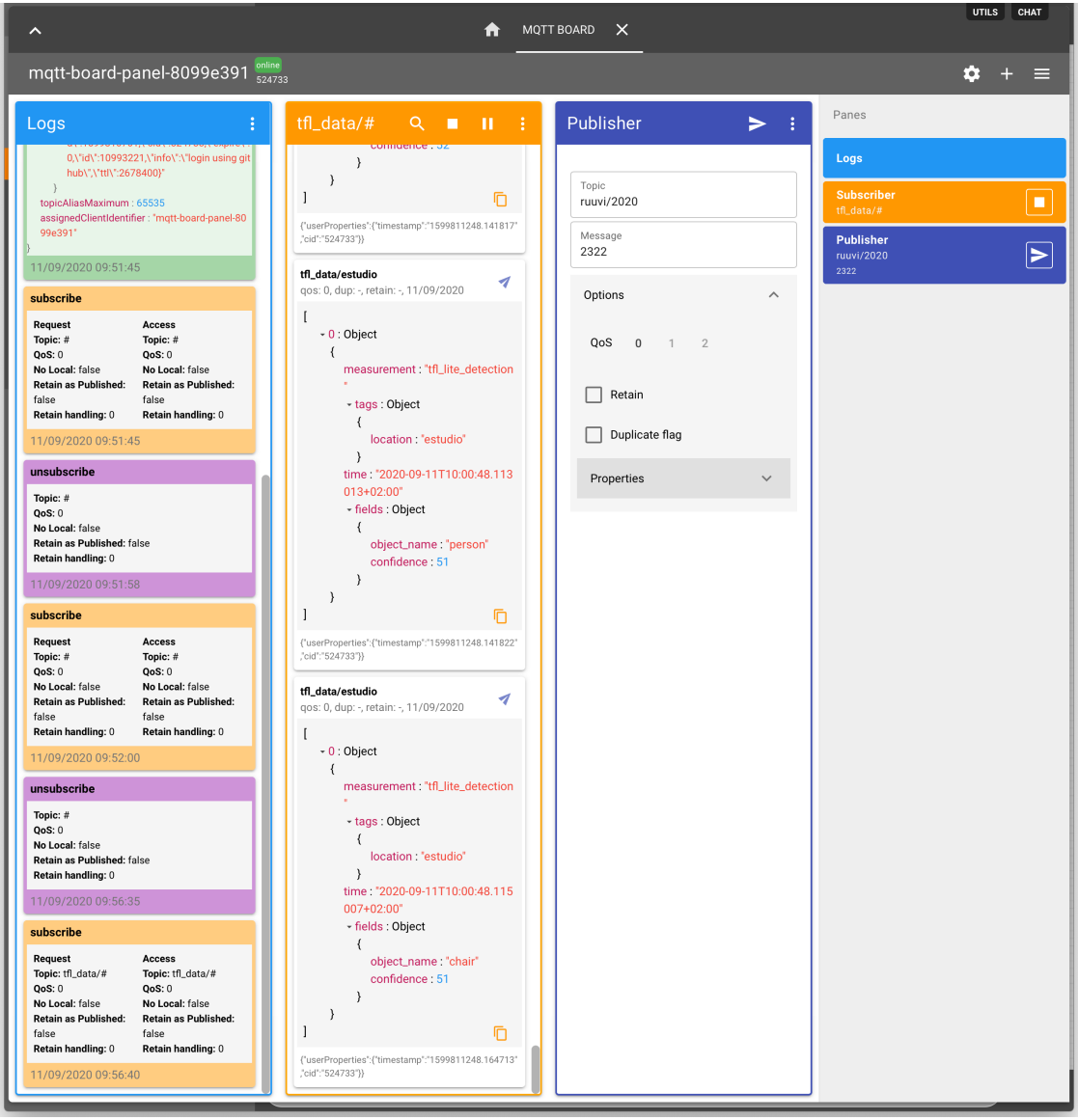
https://flespi.com/mqtt-api

flespi MQTT broker connection details

- **Host** — **mqtt.flespi.io**.
  Port — **8883 (SSL)** or **1883 (non-SSL)**; for MQTT over WebSockets: **443 (SSL)** or **80 (non-SSL)**.
- **Authorization** — use a flespi platform token as MQTT session username; no password.
- **Client ID** — use any unique identifier within your flespi user session.
- **Topic** — you can publish messages to any topic except **flespi/**.
- **ACL** — both **flespi/** and **MQTT pub/sub** restrictions determined by the token.

```
I1RKMMIUJpplQoSgAQ8MvDUJWNNJ9R2HIJgijO1S1gt5rajaeIOaiaKWwlHt2z1z
```

Flespi dashboard



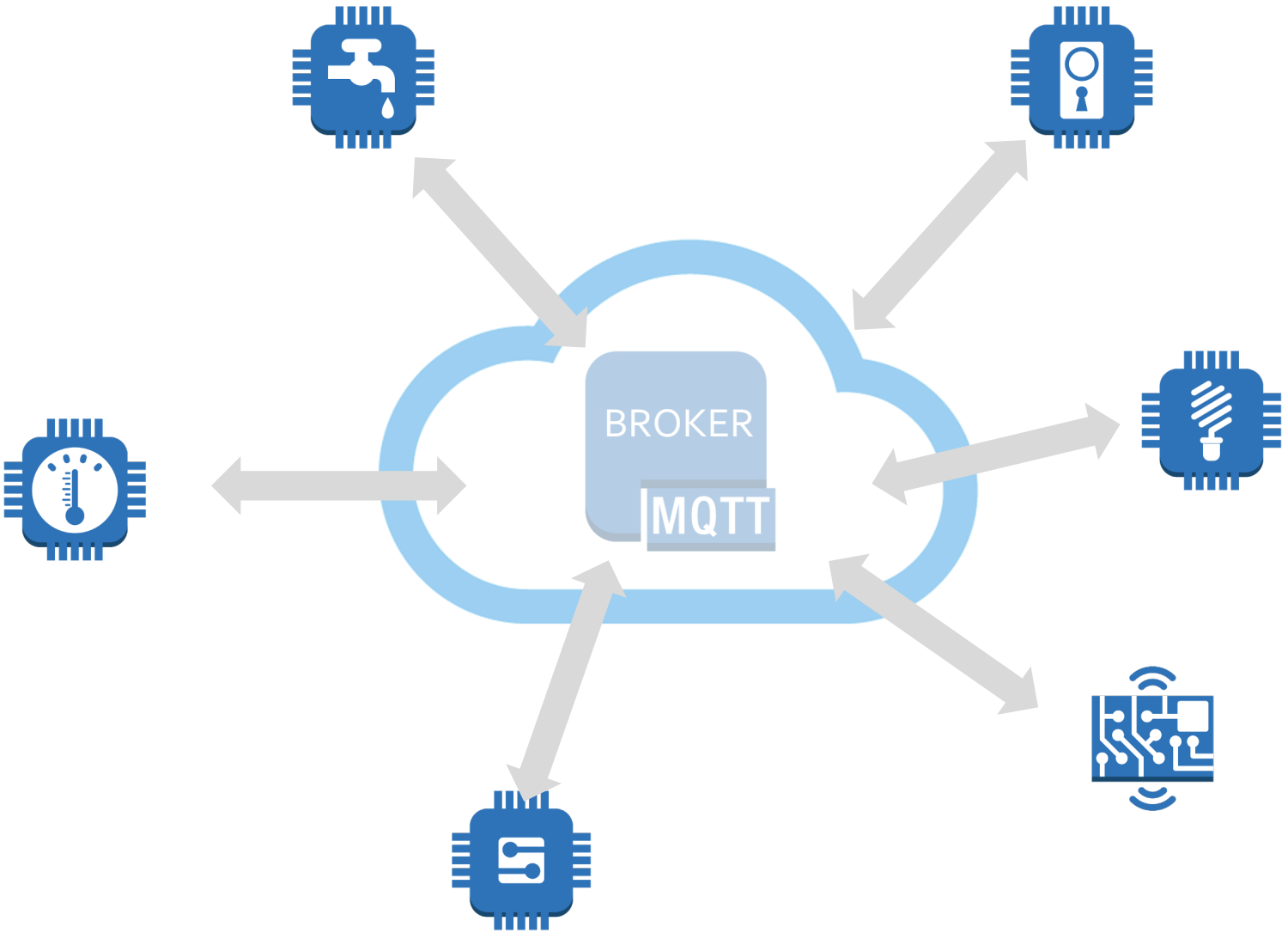
Open brokers ("Sandboxes")

- TCP based:
  - https://iot.eclipse.org/getting-started/#sandboxes
    - Hostname: **iot.eclipse.org**
  - http://test.mosquitto.org/
    - Hostname: **test.mosquitto.org**
  - https://www.hivemq.com/mqtt-demo/
    - Hostname: **broker.hivemq.com**
    - http://www.mqtt-dashboard.com/
  - Ports:
    - standard: 1883
    - encrypted: 8883 *(TLS v1.2, v1.1 or v1.0 with x509 certificates)*

- Websockets based:
  - broker.mqttdashboard.com port: 8000
  - test.mosquitto.org port: 8080
  - broker.hivemq.com port: 8000

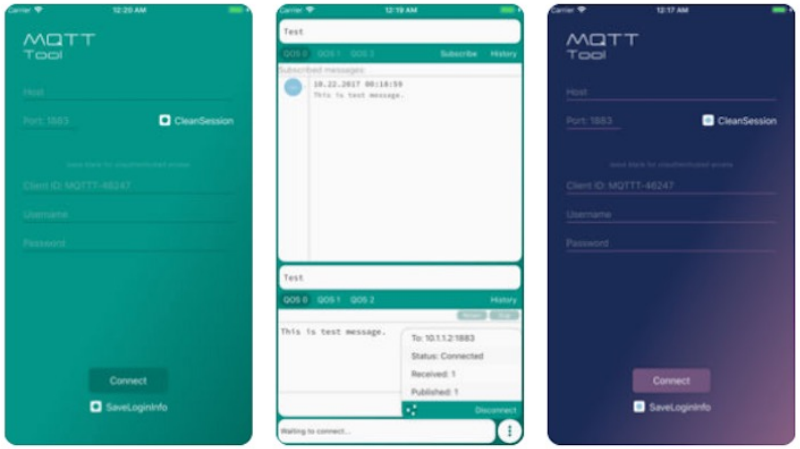
- https://github.com/mqtt/mqtt.github.io/wiki/public_brokers

Creating clients

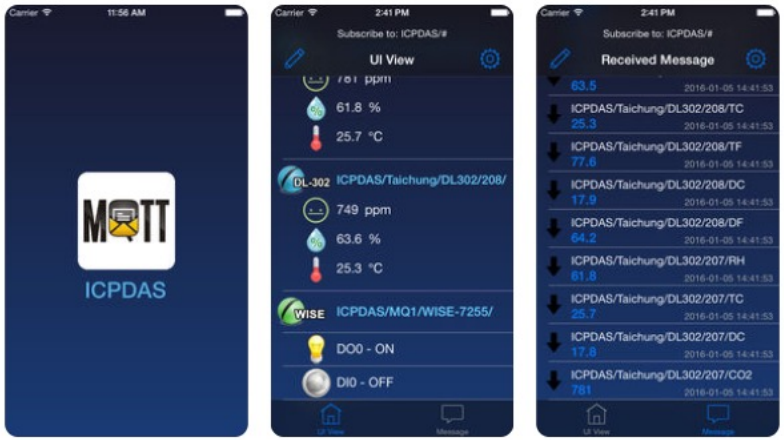


MQTT clients: iOS

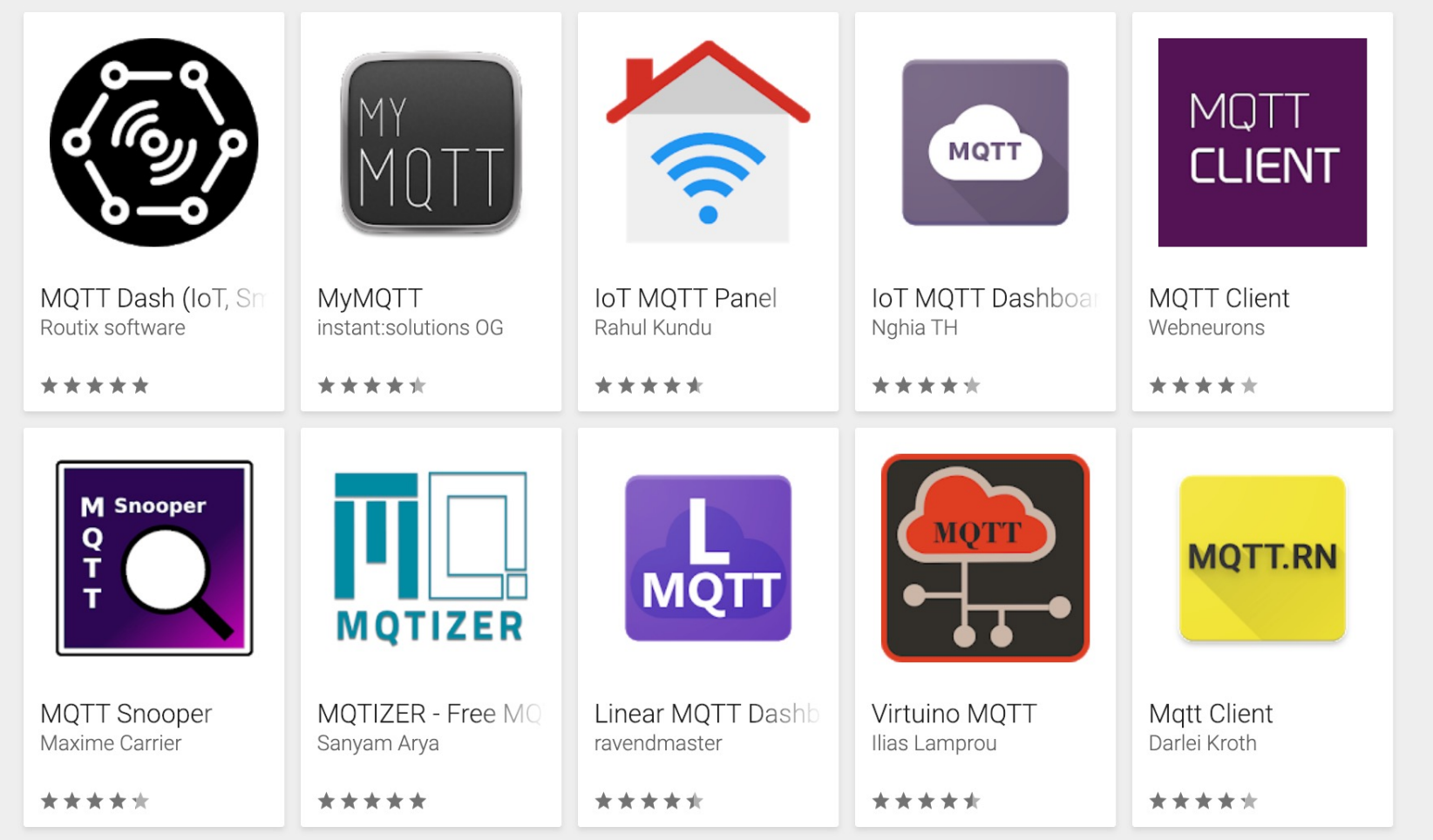
Mqttt
Utilidades



ICPDAS MQ...
Utilidades

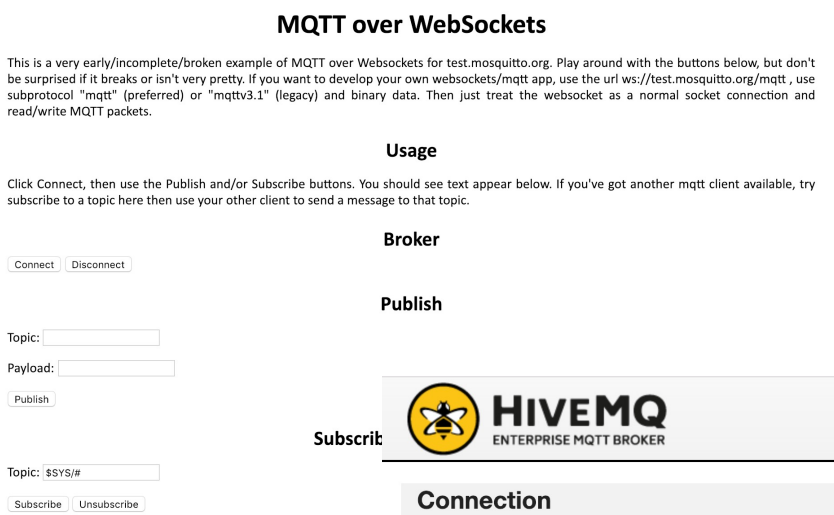


MQTT clients: Android

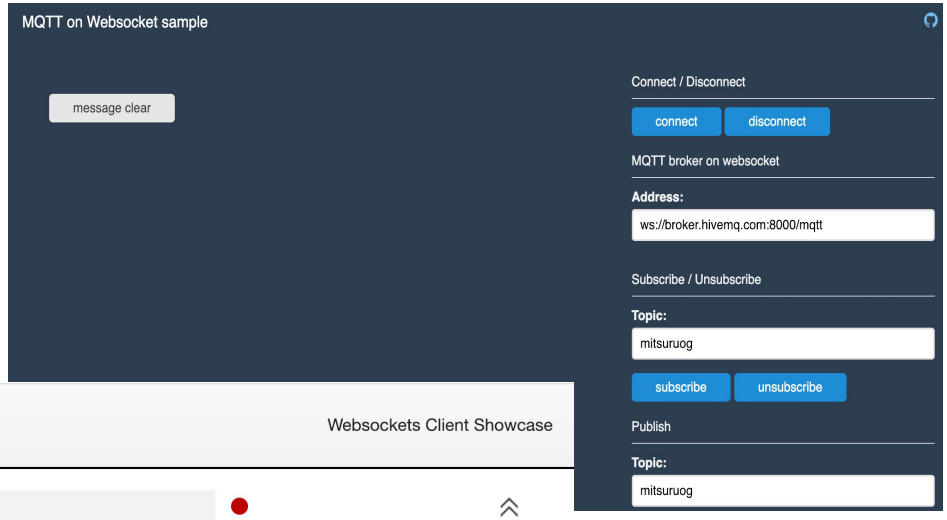


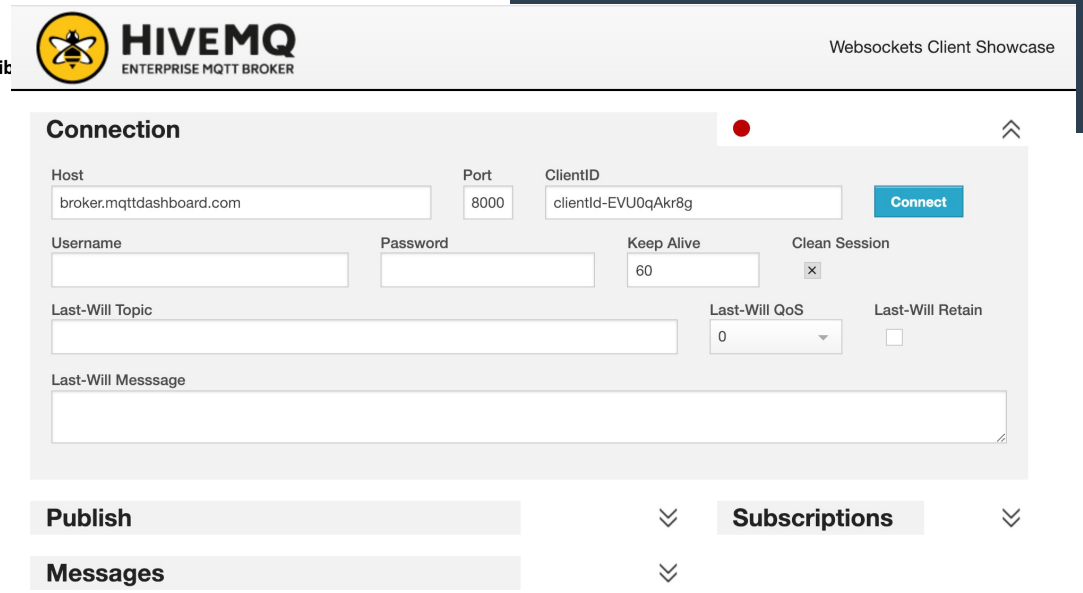
MQTT websocket clients

http://test.mosquitto.org/ws.html



http://mitsuruog.github.io/what-mqtt/





http://www.hivemq.com/demos/websocket-client/

http://mqtt-explorer.com

Python example 1: the simplest subscriber

```
# File: sisub.py

import paho.mqtt.client as mqtt

THE_BROKER = "test.mosquitto.org"
THE_TOPIC = "$SYS/#”
C_ID = ""

# The callback for when the client receives a CONNACK response from the server.
def on_connect(client, userdata, flags, rc):
    print("Connected to ", client._host, "port: ", client._port)
    print("Flags: ", flags, "return code: ", rc)

    # Subscribing in on_connect() means that if we lose the connection and
    # reconnect then subscriptions will be renewed.
    client.subscribe(THE_TOPIC, qos=0)

# The callback for when a PUBLISH message is received from the server.
def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload))

client = mqtt.Client(client_id=C_ID, clean_session=True, userdata=None, protocol=mqtt.MQTTv311, transport="tcp")

client.on_connect = on_connect
client.on_message = on_message

client.username_pw_set(None, password=None)
client.connect(THE_BROKER, port=1883, keepalive=60)

client.loop_forever()
```

Python example 2: very basic periodic producer

```
# File: sipub.py

import random
import time

import paho.mqtt.client as mqtt

THE_BROKER = "test.mosquitto.org"
THE_TOPIC = "PMtest/rndvalue"
CLIENT_ID = ""

# The callback for when the client receives a CONNACK response from the server.
def on_connect(client, userdata, flags, rc):
    print("Connected to ", client._host, "port: ", client._port)
    print("Flags: ", flags, "returned code: ", rc)

# The callback for when a message is published.
def on_publish(client, userdata, mid):
    print("sipub: msg published (mid={})".format(mid))


client = mqtt.Client(client_id=CLIENT_ID,
                     clean_session=True,
                     userdata=None,
                     protocol=mqtt.MQTTv311,
                     transport="tcp")

client.on_connect = on_connect
client.on_publish = on_publish

client.username_pw_set(None, password=None)
client.connect(THE_BROKER, port=1883, keepalive=60)
```

```
client.loop_start()

while True:

    msg_to_be_sent = random.randint(0, 100)
    client.publish(THE_TOPIC,
                   payload=msg_to_be_sent,
                   qos=0,
                   retain=False)

    time.sleep(5)

client.loop_stop()
```

Generates a new data every 5 secs