UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Internet of Things

a systemic perspective

Prof. Roberto Vezzani
University of Modena and Reggio Emilia

Milan, 10th May 2023

About me

- **Affiliation**:
  - Associate Professor at the University of Modena and Reggio Emilia
  - Member of the Artificial Intelligence Research and Innovation Center
  - Member of the AimageLab Group (head: Prof. Rita Cucchiara)
- **Teaching**: Computer Architecture and IoT
- **Research topics**: computer vision for people behavior understanding and human computer interaction; depth cameras; human robot interaction;
- **Email**: roberto.vezzani@unimore.it
- **Website**: https://aimagelab.ing.unimore.it

Objectives of this seminar

- Discussion on the meaning of «IoT»: an (often over)used term
- “digital twins” in IoT
- A generic IoT ecosystem
- A (very small) selection of useful tools and technologies for IoT
- Use cases and demos

IoT – First Definition

The Internet of Things (IoT) is a novel paradigm that is rapidly gaining ground in the scenario of modern wireless telecommunications. The basic idea of this concept is the pervasive presence around us of a variety of things or objects – such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, etc. – which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals

[D. Giusto, A. Iera, G. Morabito, L. Atzori (Eds.), The Internet of Things, Springer, 2010. ISBN: 978-1-4419-1673-0.]

Things started to have a "unique name in the digital world"

A more recent definition

"The Internet of Things is a network of physical **objects** – vehicles, machines, home appliances, and more – that uses **sensors** and APIs to **connect** and **exchange** data over the **Internet**. "

Physical objects

+

Controller, sensors, actuators

+

Internet and internet platforms

=

Internet of Things

Take home message: an application is not IoT if it is not based on a thing AND on internet

The Flavour of “The Internet Of Things”

The **alarm** rings. As you open your eyes blearily, you see that it’s five minutes later than your usual wake-up time. The clock has checked the train times online, and your train must be delayed, so it lets you sleep in a little longer. *(See http://makezine.com/magazine/make-11/my-trainschedule-alarm-clock/.)*

In your kitchen, a blinking light on the **medicine bottle cap** reminds you it’s time to take your tablets. If you forget, the medicine bottle cap goes online and emails your doctor to let her know. *(See www.vitality.net/glowcaps.html.)*

On your way out of the house, you catch a glow in the corner of your eye. Your **umbrella** handle is lit up, which means that it has checked the BBC weather reports and predicts rain. You sigh and pick it up. *(See www.materious.com/#/projects/forecast/.)*

As you pass the bus stop on the way to the station, you notice the large …

The Flavour of “The Internet Of Things”

The **alarm** rings. As you open your eyes blearily, you see that it’s five minutes later than your usual wake-up time. The clock has checked the train times online, and your train must be delayed, so it lets you sleep in a little longer. *(See http://makezine.com/magazine/make-11/my-trainschedule-alarm-clock/.)*

In your kitchen, a blinking light on the **medicine bottle cap** reminds you it’s time to take your tablets. If you forget, the medicine bottle cap goes online and emails your doctor to let her know. *(See www.vitality.net/glowcaps.html.)*

On your way out of the house, you catch a glow in the corner of your eye. Your **umbrella** handle is lit up, which means that it has checked the BBC weather reports and predicts rain. You sigh and pick it up. *(See www.materious.com/#/projects/forecast/.)*

As you pass the bus stop on the way to the station, you notice the large …

Digital Twin

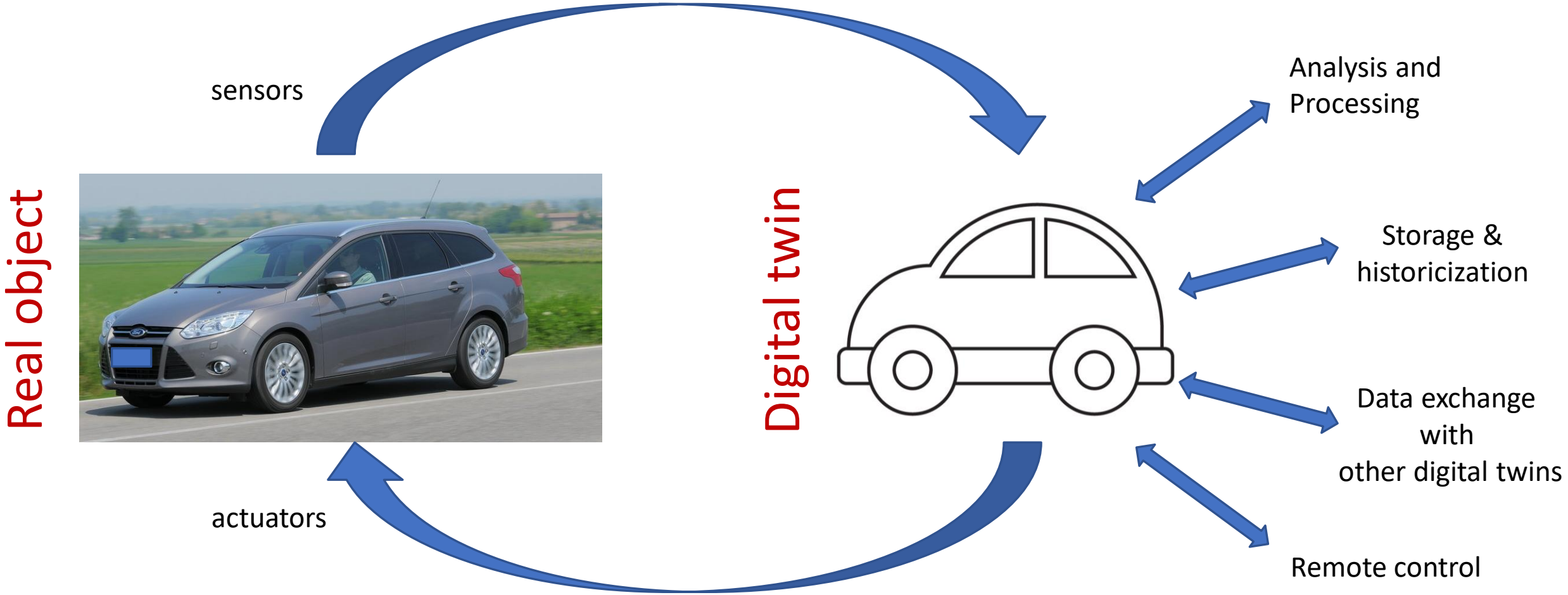
A digital twin is a live digital representation of a physical object.

Representing the essential physical content of the object, a digital twin provides real-time information on the configuration, condition, and state of the asset as well as historical data that can be queried through APIs.

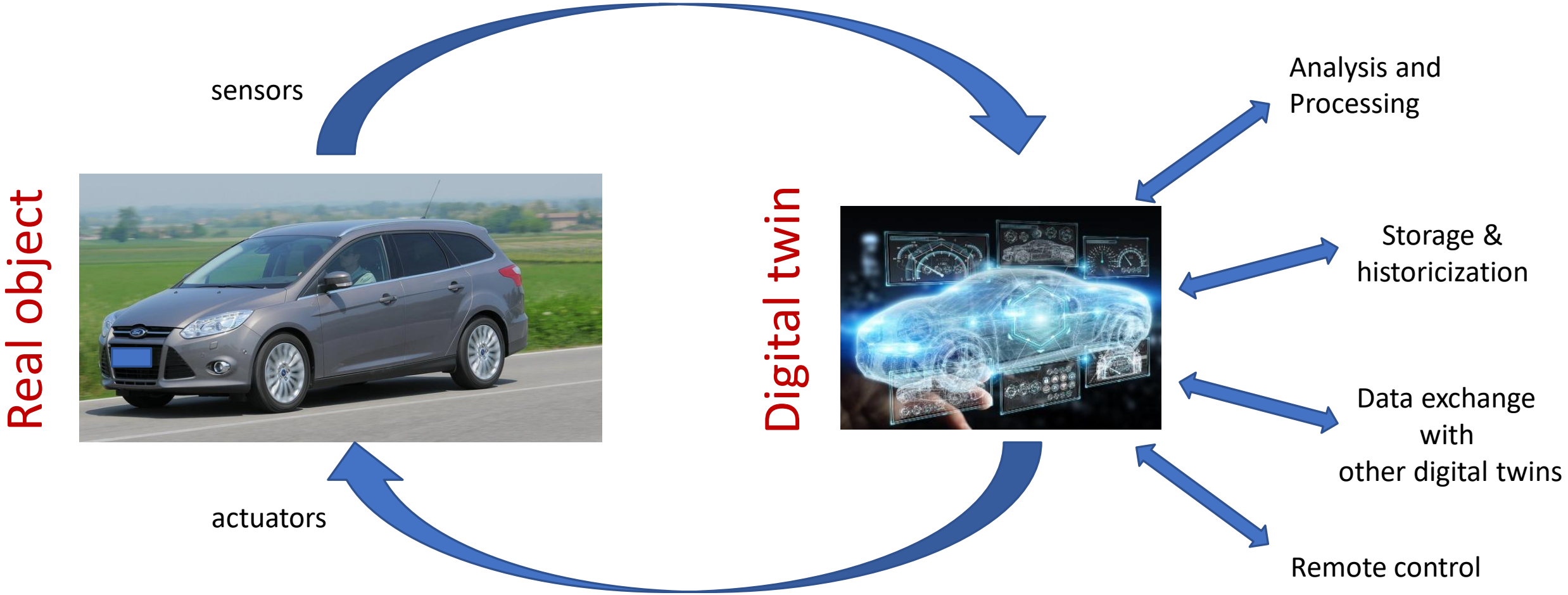
Digital twins act on behalf of physical objects by sending alerts and notifications while being able to initiate control flows to act in the physical world.

More complex digital twins feature hierarchies or relationships; contain derived data from statistical, machine learning, and physical simulations; and expose service capabilities.

Digital twin



Digital twin



Digital twin and IoT

Take home message:
it is not an IoT project if the digital twins are not in the cloud,
i.e., they are not accessible via the internet

Take home message:
it is not an IoT project if the digital twins are not accessible from other digital twins
(and not only from humans)

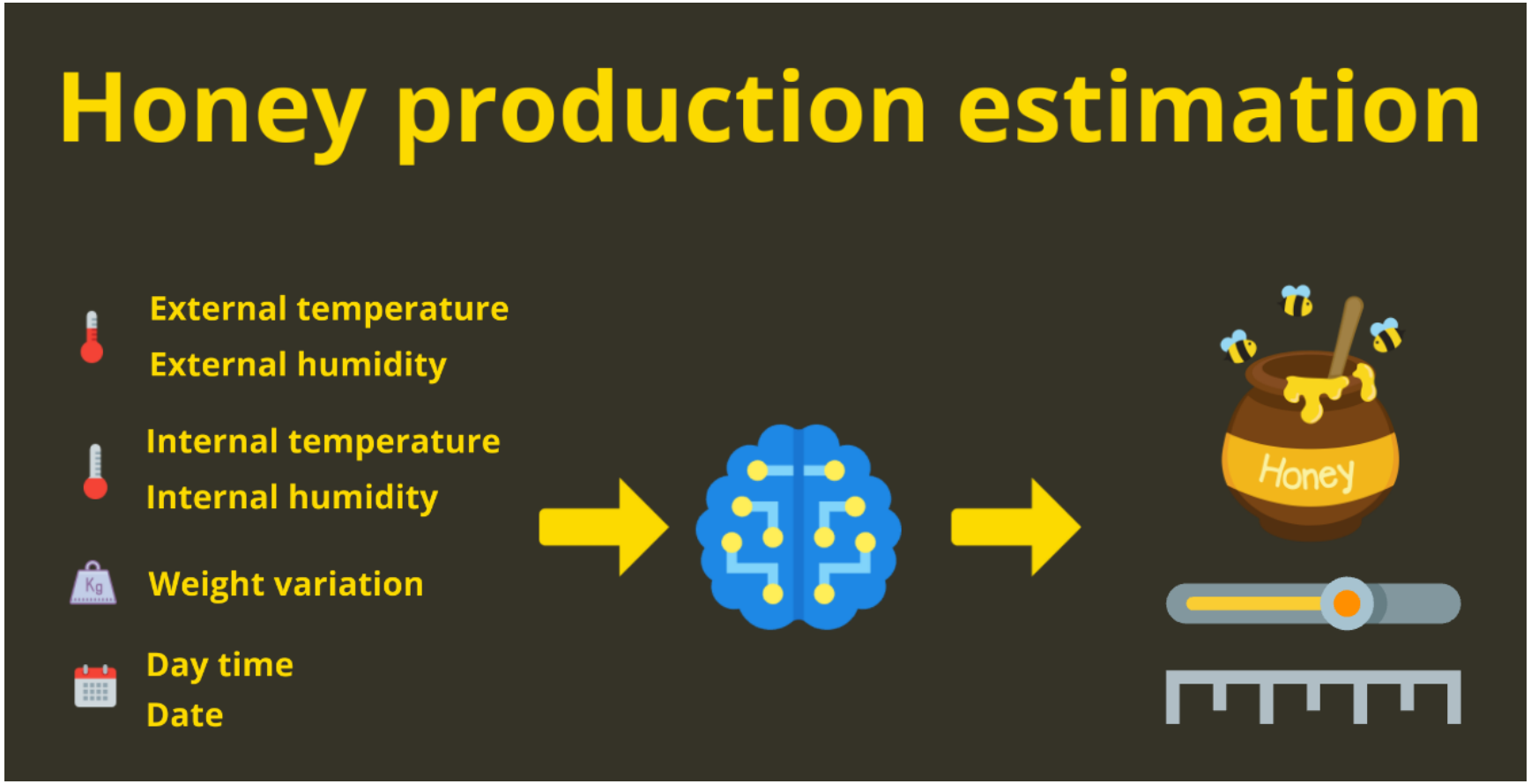
Some examples – Smart Hive

Final project of the IoT course at Unimore by Bertellini, Caputo, Doganieri

The real **object**: a hive inhabited by bees

https://github.com/paolobertellini/Smart-hive

Some examples – Smart Hive

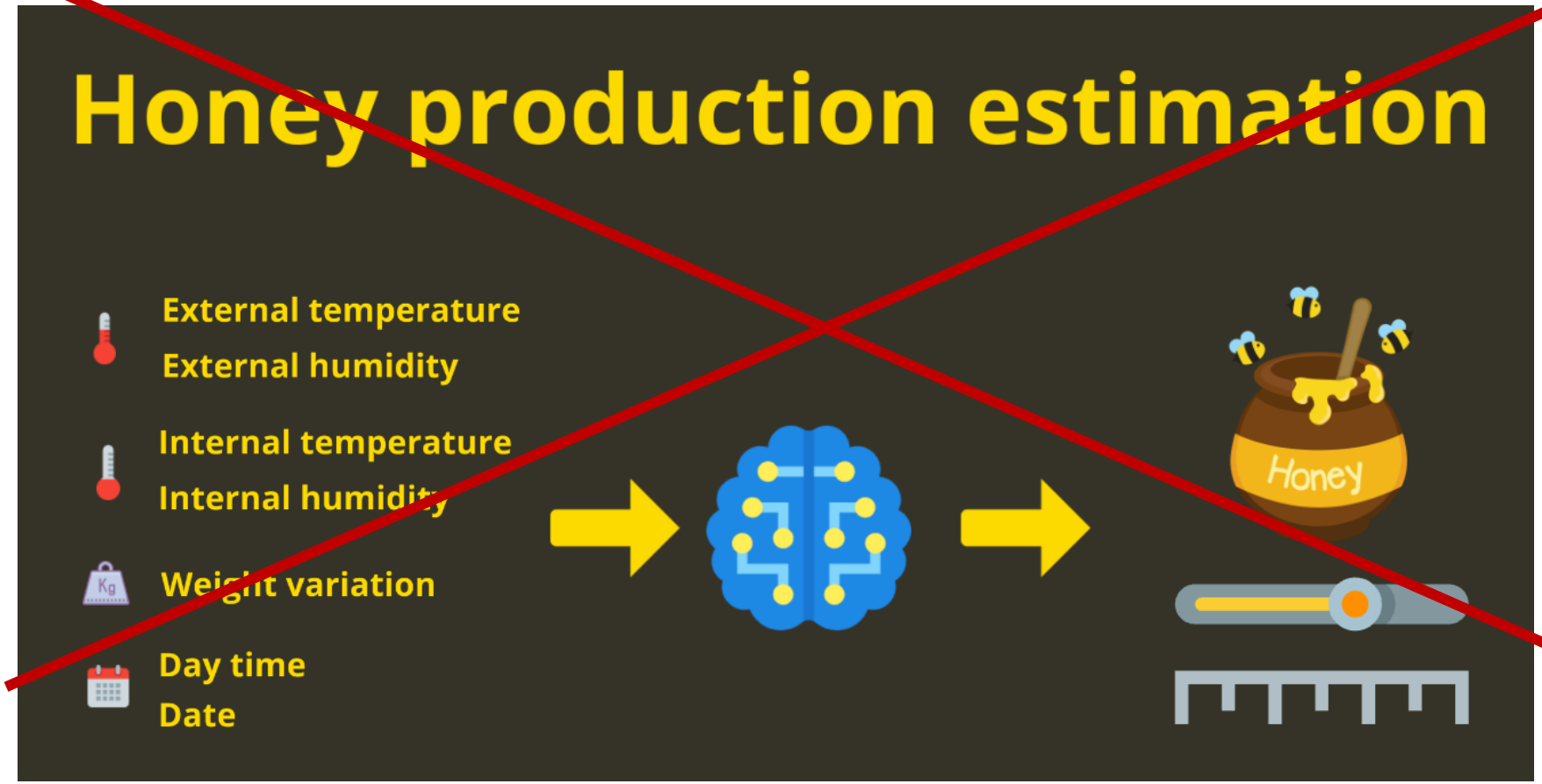


The first **Digital twin model**: temperatures, weight, day/time

Goals: honey estimation and remote control

https://github.com/paolobertellini/Smart-hive

Some examples – Smart Hive



This was not IoT

The first **Digital twin model**: temperatures, weight, day/time

Goals: honey estimation and remote control

https://github.com/paolobertellini/Smart-hive

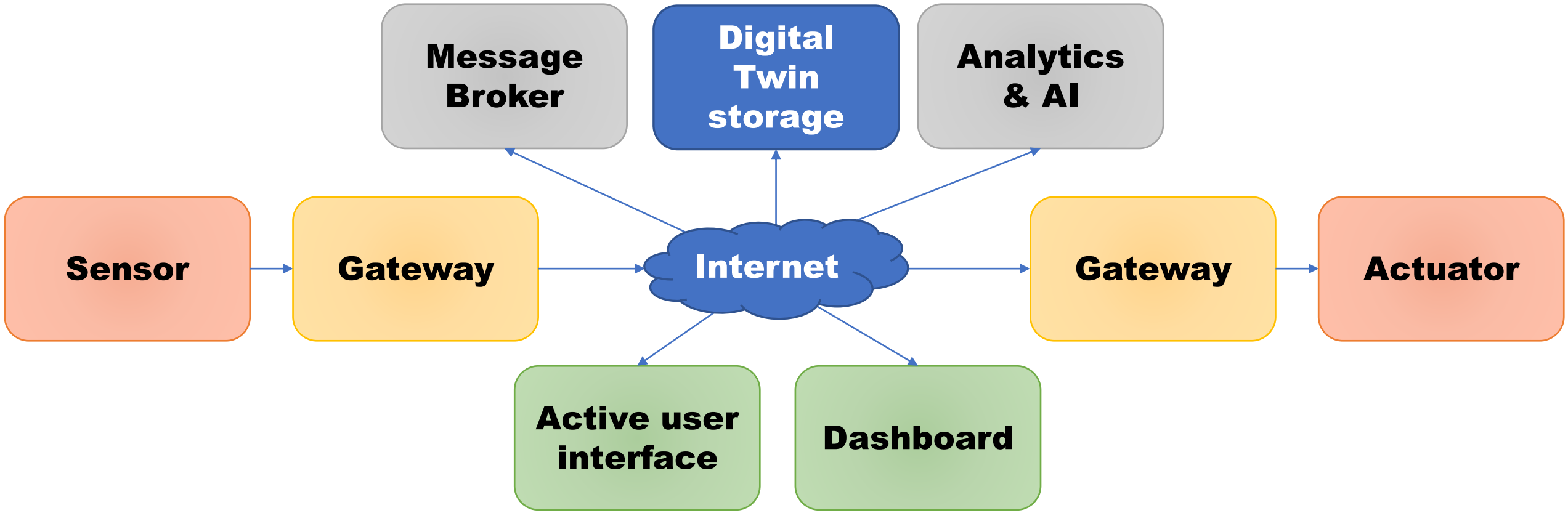
Some examples – Smart Hive

The ultimate **goal**: automatic swarm management. If a hive detects a swarm, nearby hives begin to attract bees (thanks to messages from internet)

This is IOT!

https://github.com/paolobertellini/Smart-hive

A generic IoT Ecosystem

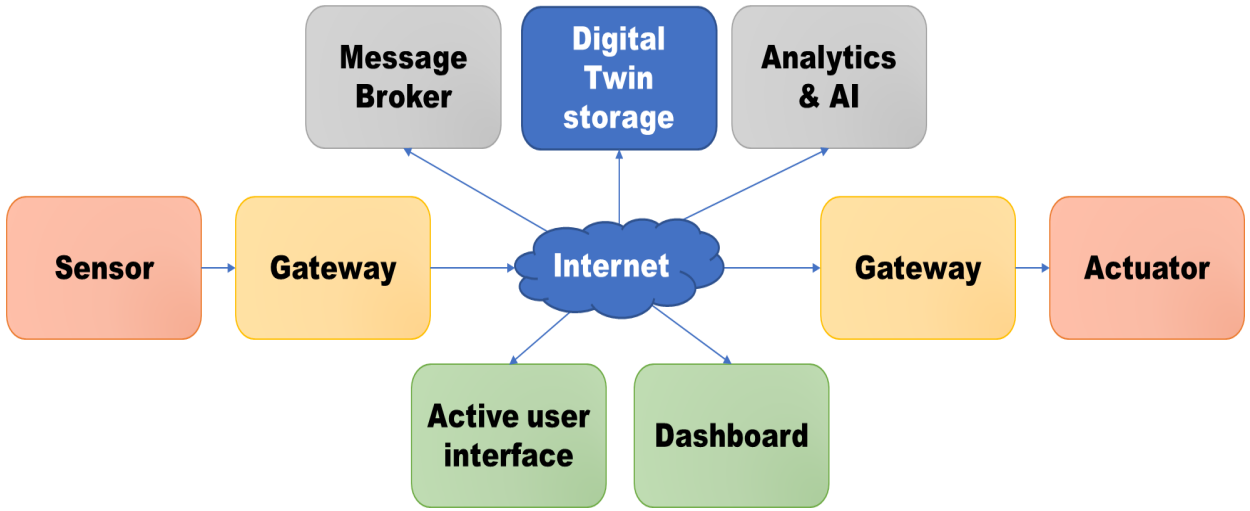


Key elements (1)

The Digital twin is the central element

The digital twin should be place in cloud, i.e., it should be reachable through a public address

The digital twin should have a unique URI



If you are the user, sensors are at your home

If you're the developer, you want to bring digital twins to your home. Cloud means “at the developer's home”. [remember, data is the new oil]

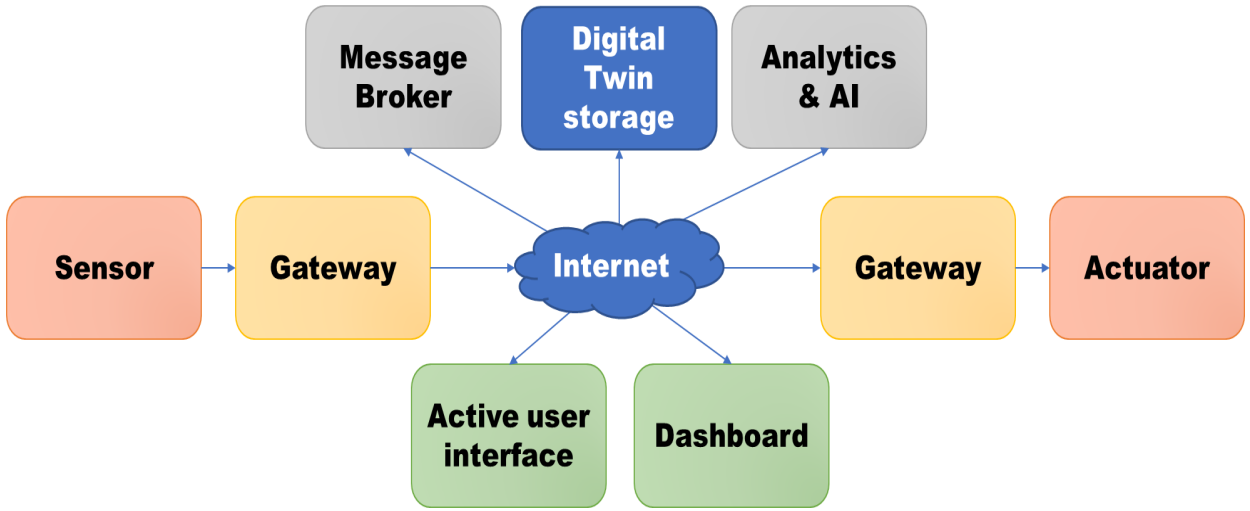
Key elements (2)

Sensors and actuators are usually connected to microcontrollers

Microcontrollers (usually) do not have internet connection. Gateways are used instead

Sensors, actuators and gateway are placed "at your home“

Gateways have two sides. One for the sensors/actuators and one toward internet



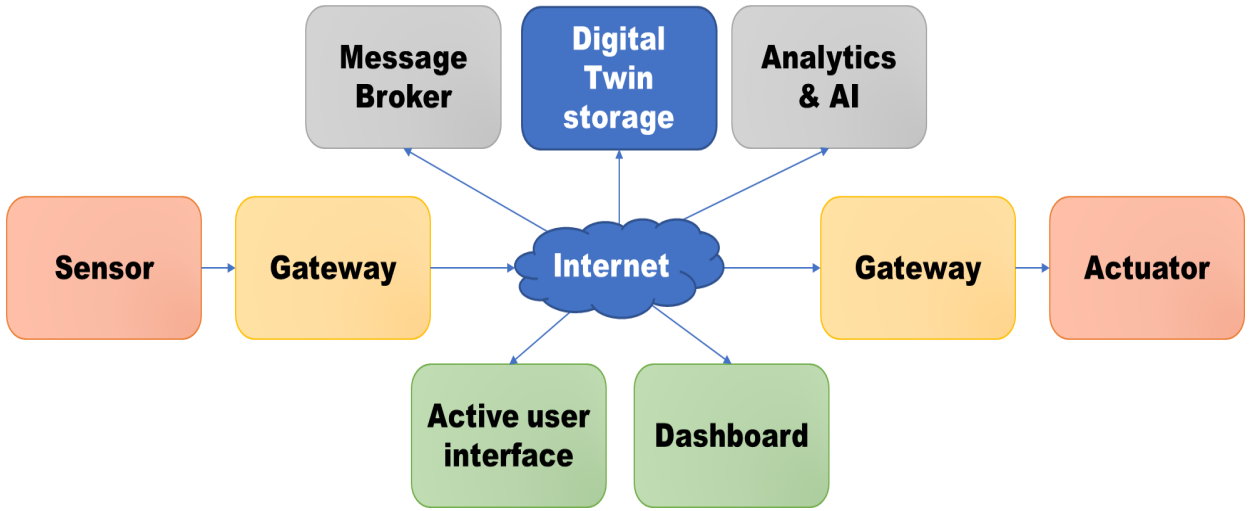
Key elements (3)

The communication protocol between sensor <-> bridge and actuator <-> bridge is usually custom.

Serial protocols are a good choice since they have a low communication overhead

Communication via the Internet mainly exploits standard protocols and common practices

Two main models can be used: request/response (e.g., HTTP REST) and publish/subscribe (e.g., MQTT)

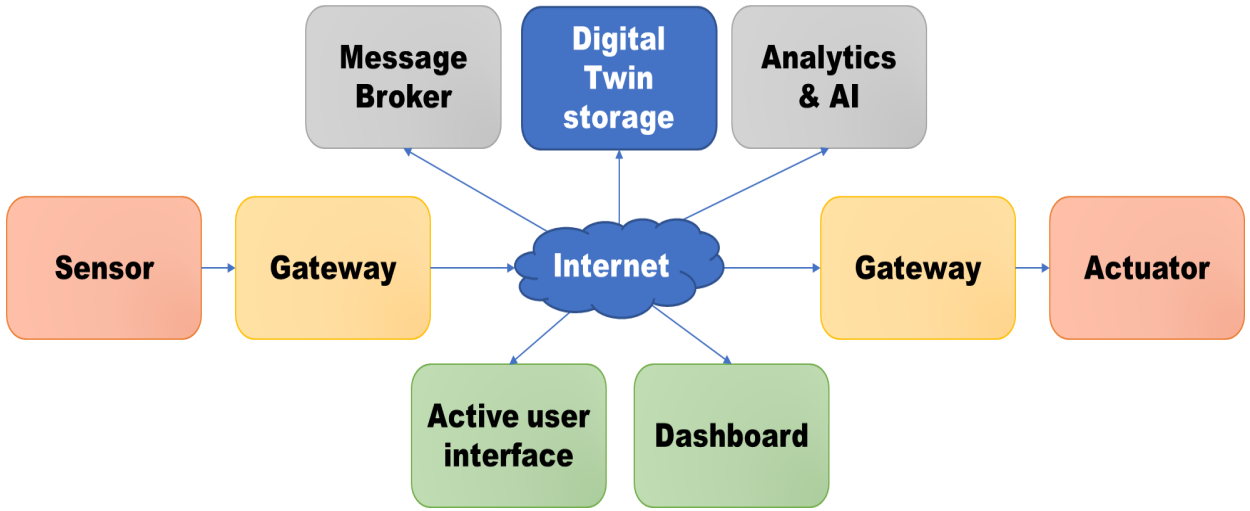


Key elements (4)

The central Digital Twin contains API for every action: synchronization, update, query, access to temporal series, and so on

Analytics & AI components can directly interact with the Digital Twin

The same API could be used by machines and humans (through wrappers and GUI/dashboards)



Demo time – a real «use case»

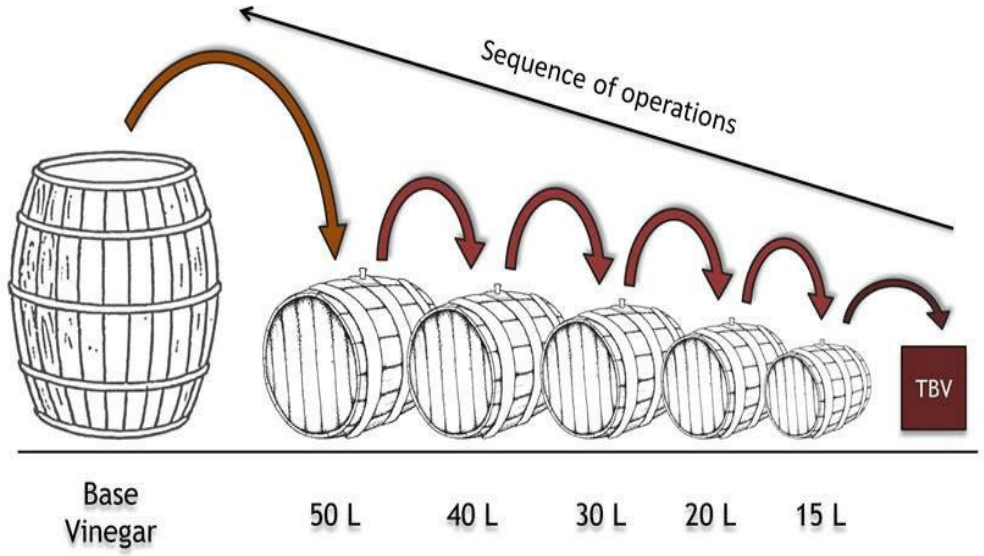
The real problem

We want to improve the production of traditional balsamic vinegar

Every year, a sequence of «travaso e rincalzo» operations should be done.

The prediction of the evaporation level could be useful

Evaporation is related to the current level inside each barrel and the temperature



Demo time – a real «use case»

Basic solution:

- create a digital twin of each barrel

- physical model vs machine learning for the evolution

- insert temperature and level sensors on each barrel to synchronize the digital twin with the real object

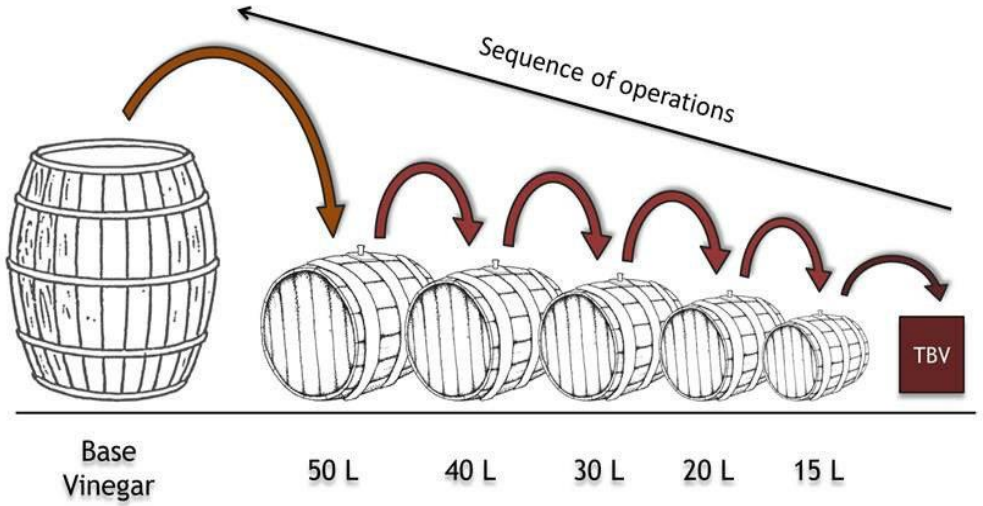
- local processing and local «storage» of the digital twin (without internet) → this is not IoT!!

It becomes IoT:

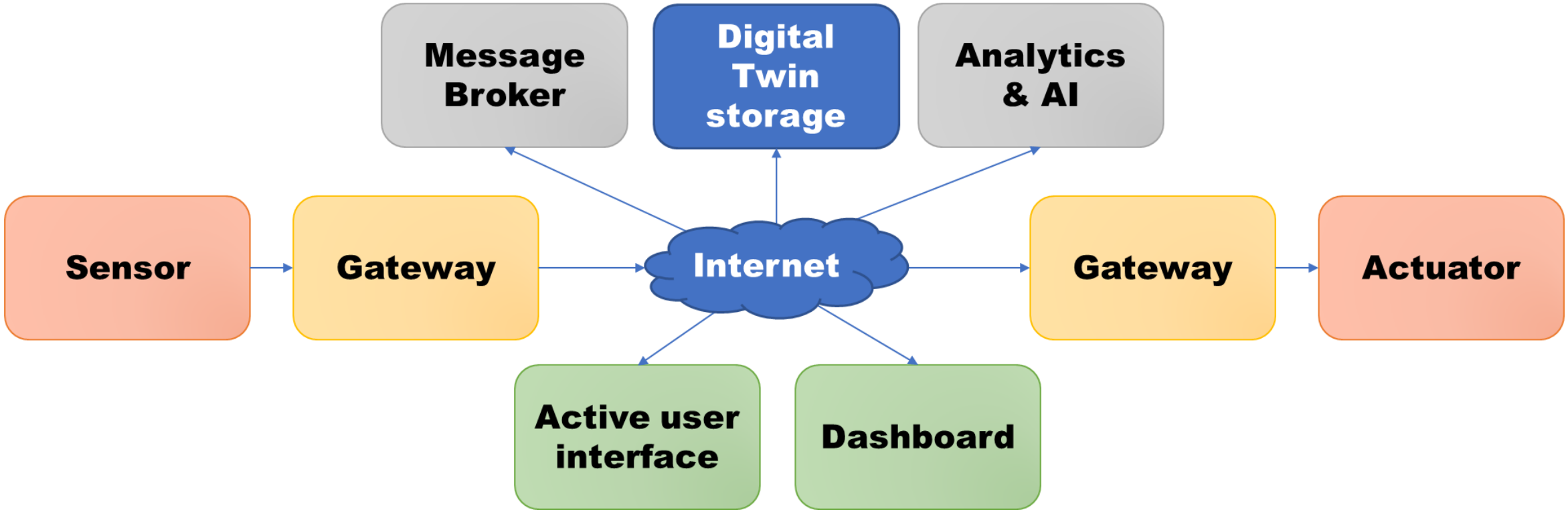
- if we exploit weather forecasts (digital twin of weather)

- if we want to exploit more data for ML (Confederation of Producers)

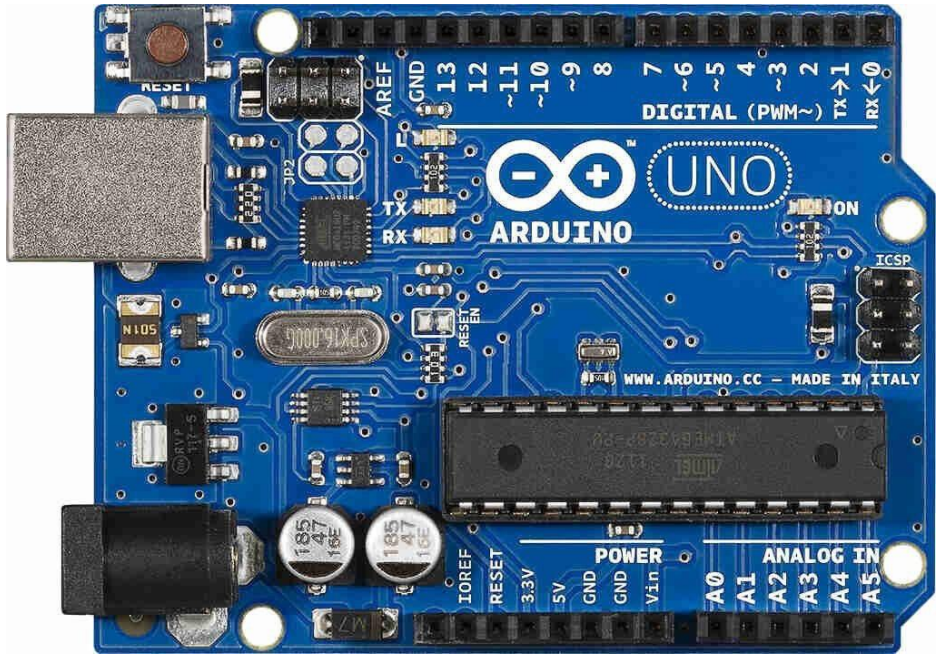
Start today to collect data for the algorithms of tomorrow!



Demo time


Message Broker
Digital Twin storage
Analytics & AI
Sensor
Gateway
Internet
Gateway
Actuator
Active user interface
Dashboard

Demo time – simulation



Serial communication

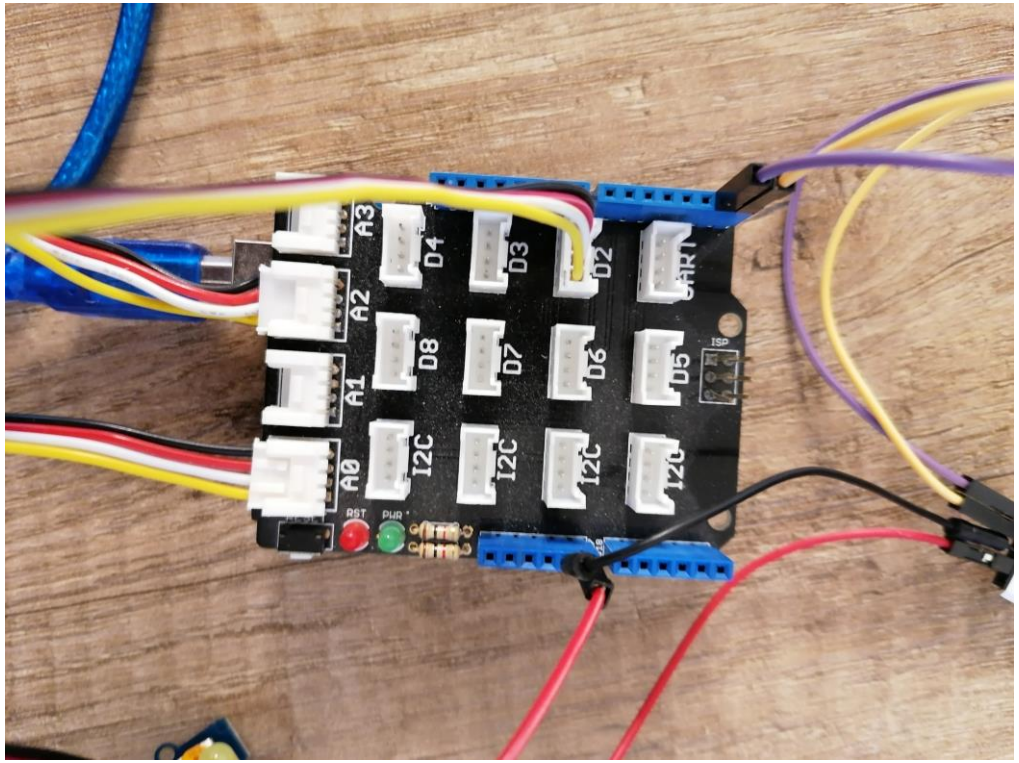
Custom protocols:

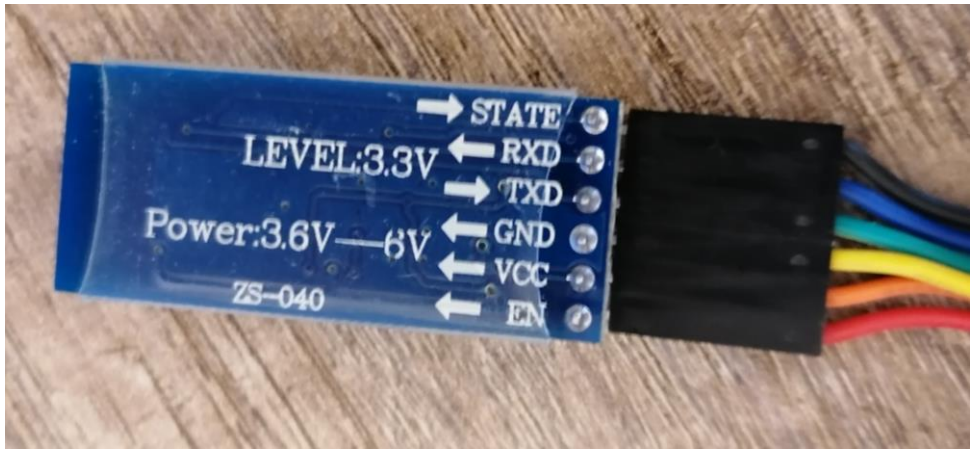
Sensor mc to gateway: nfields;f1;..;fn \n

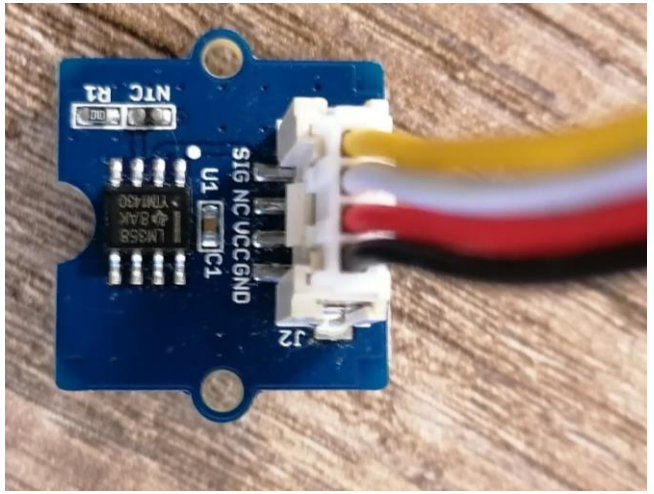
Gateway to actuator mc: single byte ‘0’/’1’

Wired vs wireless

Demo time


A3
A2
A1
A0
D4
D3
D2
UART
D8
D7
D6
D5
ISP
I2C
I2C
I2C
I2C
RST
PWR


STATE
LEVEL:3.3V
RXD
TXD
Power:3.6V—6V
GND
VCC
ZS-040
EN


NTC R1
SIG NC VCC GND
U1
C1
J2

MQTT

MQTT was invented by Andy Stanford-Clark (IBM) and Arlen Nipper (Arcom, now Cirrus Link) back in 1999, when their use case was to create a protocol for **minimal battery loss and minimal bandwidth** connecting oil pipelines over satellite connection.

They specified the following goals, which the future protocol should have:

- Simple to implement
- Provide a Quality of Service Data Delivery
- Lightweight and Bandwidth Efficient
- Data Agnostic
- Continuous Session Awareness

MQTT: Pub/Sub

Ecosystem based on a broker and a constellation of clients

Clients start the connection to the Broker (the only component in cloud)

Topics defined with a hierarchical structure, like paths

Clients can publish messages about topics:

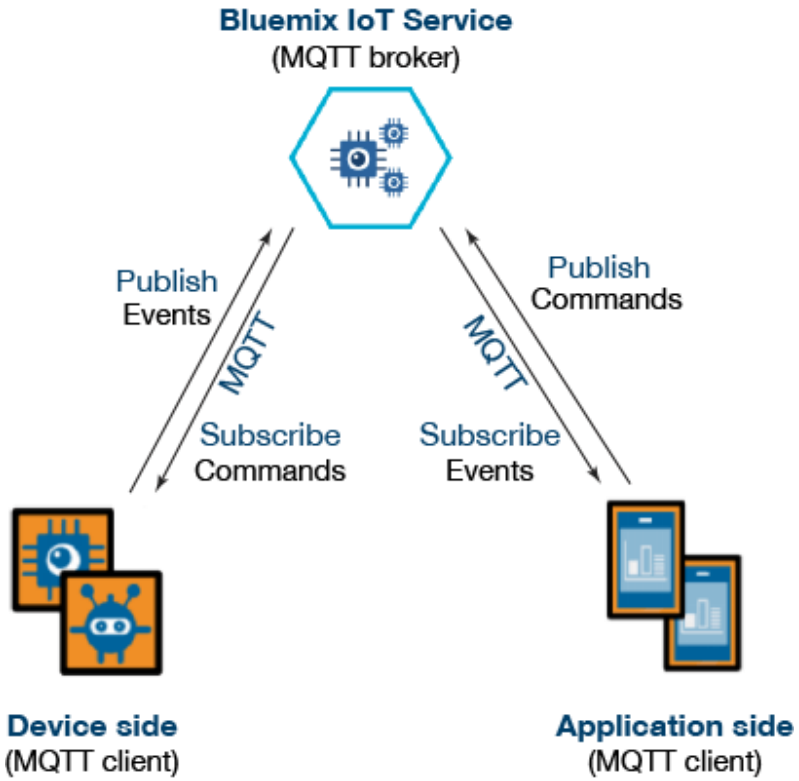
- client.publish('home/firstfloor/lightswitch/1', 'ON');

Clients can subscribe to specific topics or to set of topics by means of wildcard filters:

- client.subscribe('home/firstfloor/lightswitch/1')
- client.subscribe('home/secondfloor/lightswitch/#')
- client.subscribe('home/+/lightswitch/+')

All clients receive "immediately" all messages published on topics to which they have subscribed

Messages can be anything: Text, Numbers, Images, …



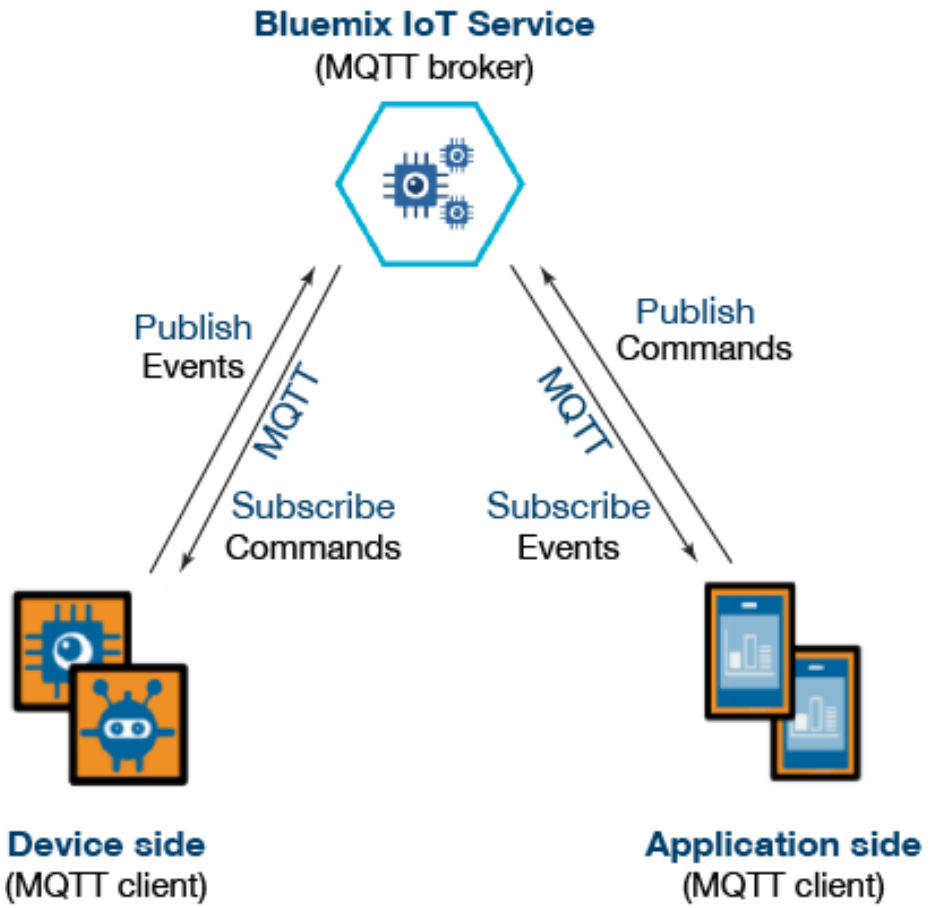
MQTT – Additional features

Clients can setup a last-will topic and message

Clients should have an almost stable connection (TCP)

Three Quality of Service levels are allowed

Authentication / encryption are available



Request / response – HTTP REST

Client / server architectures

When a client makes a request, the server answer with a response

The server cannot start a conversation

The server cannot send data to a client if the client does not start a request

HTTP - *(HyperText Transfer Protocol)*

An application-level protocol for distributed, collaborative, hypermedia information systems

HTTP is a generic, stateless client-server protocol used not only for hypertext document exchange, but for a multitude of applications also, including IoT.

Key features:

- the negotiation of the data format, for the independence of the system from the data representation format.
- Sophisticated caching policy, depending on the type of connection
- User authentication specifications

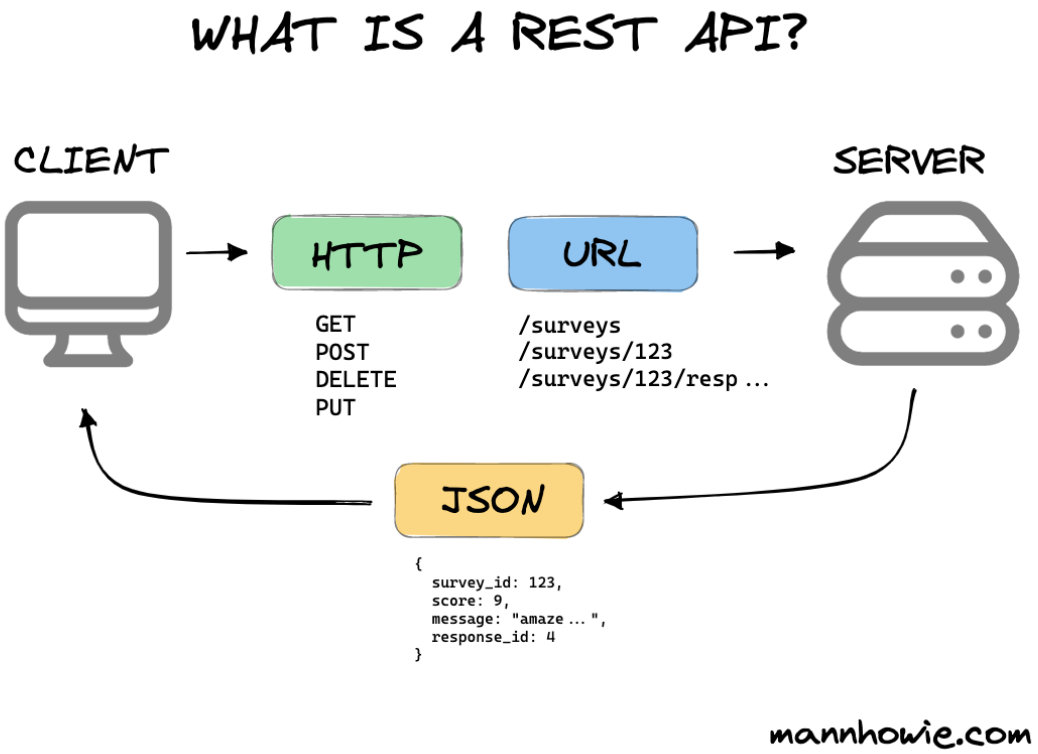
REST

Resources (digital twins or their components) are identified by uniform resource identifiers (URIs)

Resources are manipulated through their representations (JSON, XML,…)

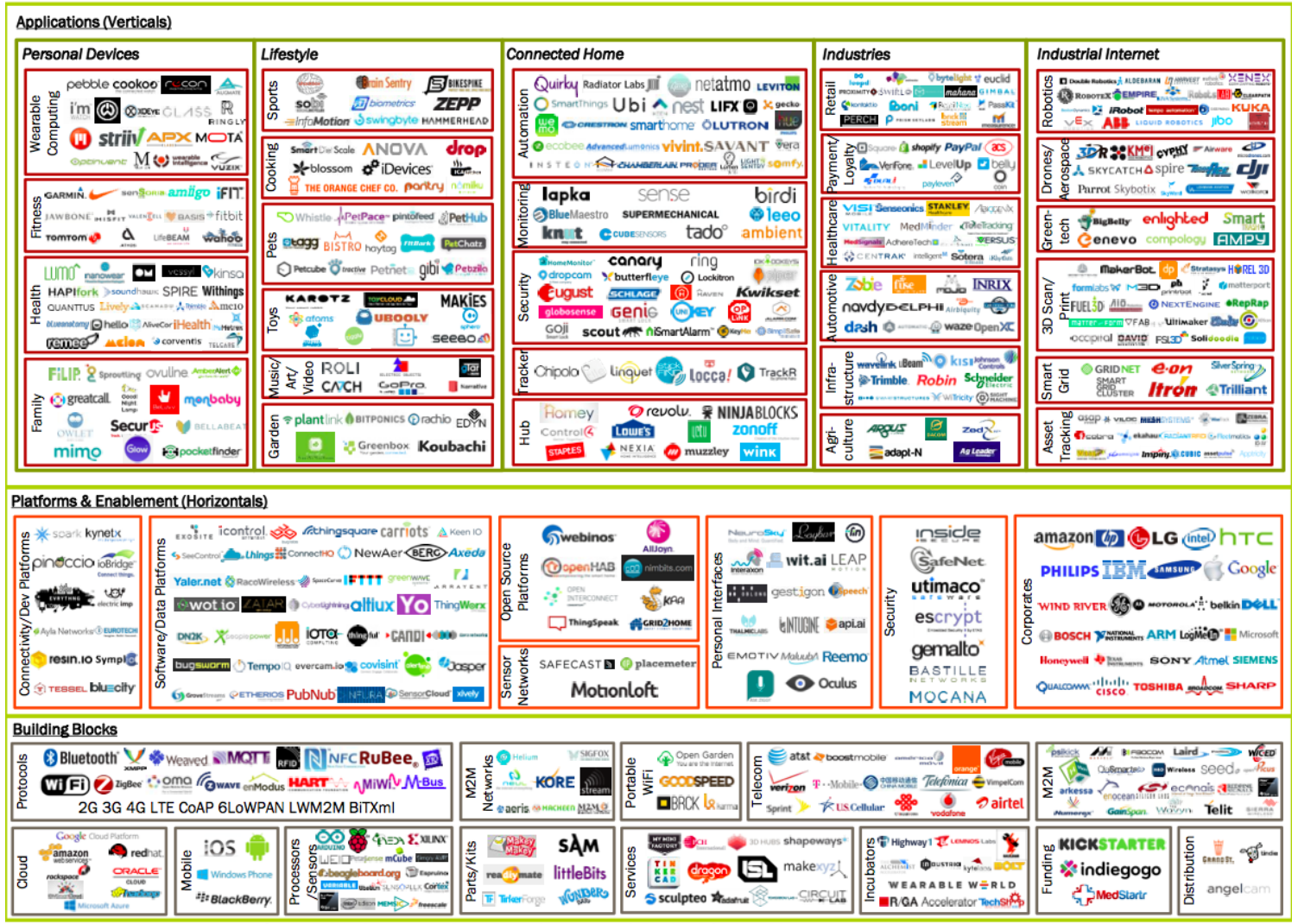
Requests are self-descriptive and stateless

Actions (Create, Read, Update, Delete) are related to HTTP verbs (POST, GET, PUT, DELETE)



A complete list of IoT tools…

… is not possible



© Matt Turck (@mattturck), David Rogg (@davidjrogg) & FirstMark Capital (@firstmarkcap) FIRSTMARK

Fast prototype

Usually, if you have a good idea, you need to create a first prototype as soon as possible

Basic, simple or even known tools are an excellent starting point

Once you have created a working prototype, you can move forward to improve each single element

Thingsboard

DIGITAL TWIN + BROKER+ DASHBOARDS+ …

Glossary - devices

Devices - basic IoT entities that may produce or use telemetry data. For example, sensors, actuators, switches;

Assets - abstract IoT entities that may be related to other devices and assets. For example, factory, field, vehicle;

Dashboards - visualization of your IoT data and ability to control particular devices through the user interface;

Entity Views - useful if you like to share only part of device or asset data to the customers;

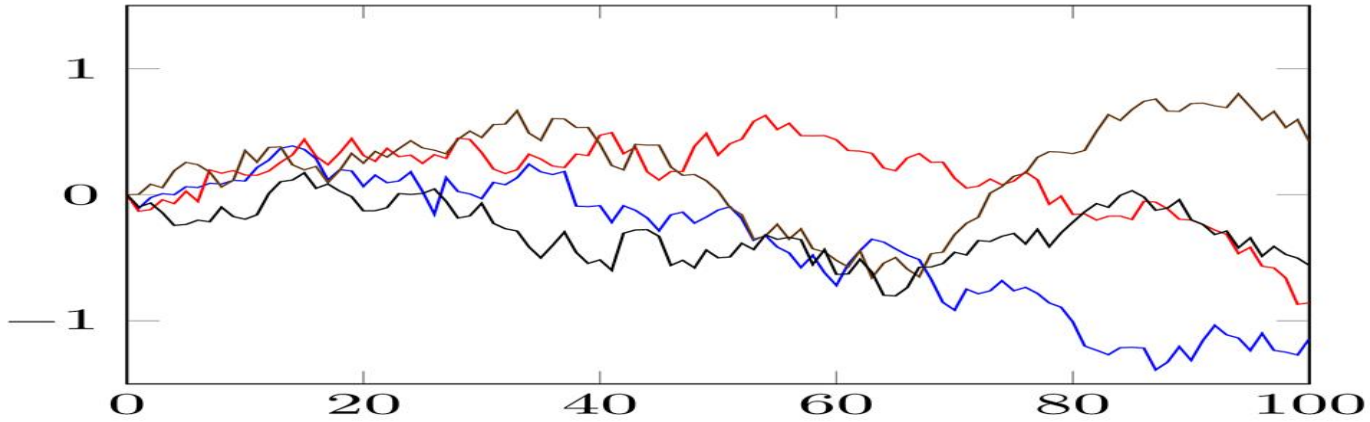
Alarms - events that identify issues with your assets, devices, or other entities;

Each entity supports:

Attributes - static and semi-static key-value pairs associated with entities. For example, serial number, model, firmware version;

Time-series data - time-series data points available for storage, querying and visualization. For example, temperature, humidity, battery level;

Relations - directed connections to other entities. For example, it contains, manages, owns, produces.



Bearer Authentication

Bearer authentication (also called token authentication) is an HTTP authentication scheme that involves security tokens called **bearer tokens**.

The name “Bearer authentication” can be understood as “give access to the bearer of this token.”

The bearer token is a cryptic string, usually generated by the server in response to a login request.

The client must send this token in the Authorization header when making requests to protected resources

Thingsboard Bearer Authentication

First, request a token for your account:

```
curl -X POST --header "Content-Type: application/json" --header
"Accept: application/json" -d "{'username’:’<EMAIL>',
'password’:’<PASSOWORD>'}" https://thingsboard.cloud/api/auth/login
```

Read the response fields and save the token:

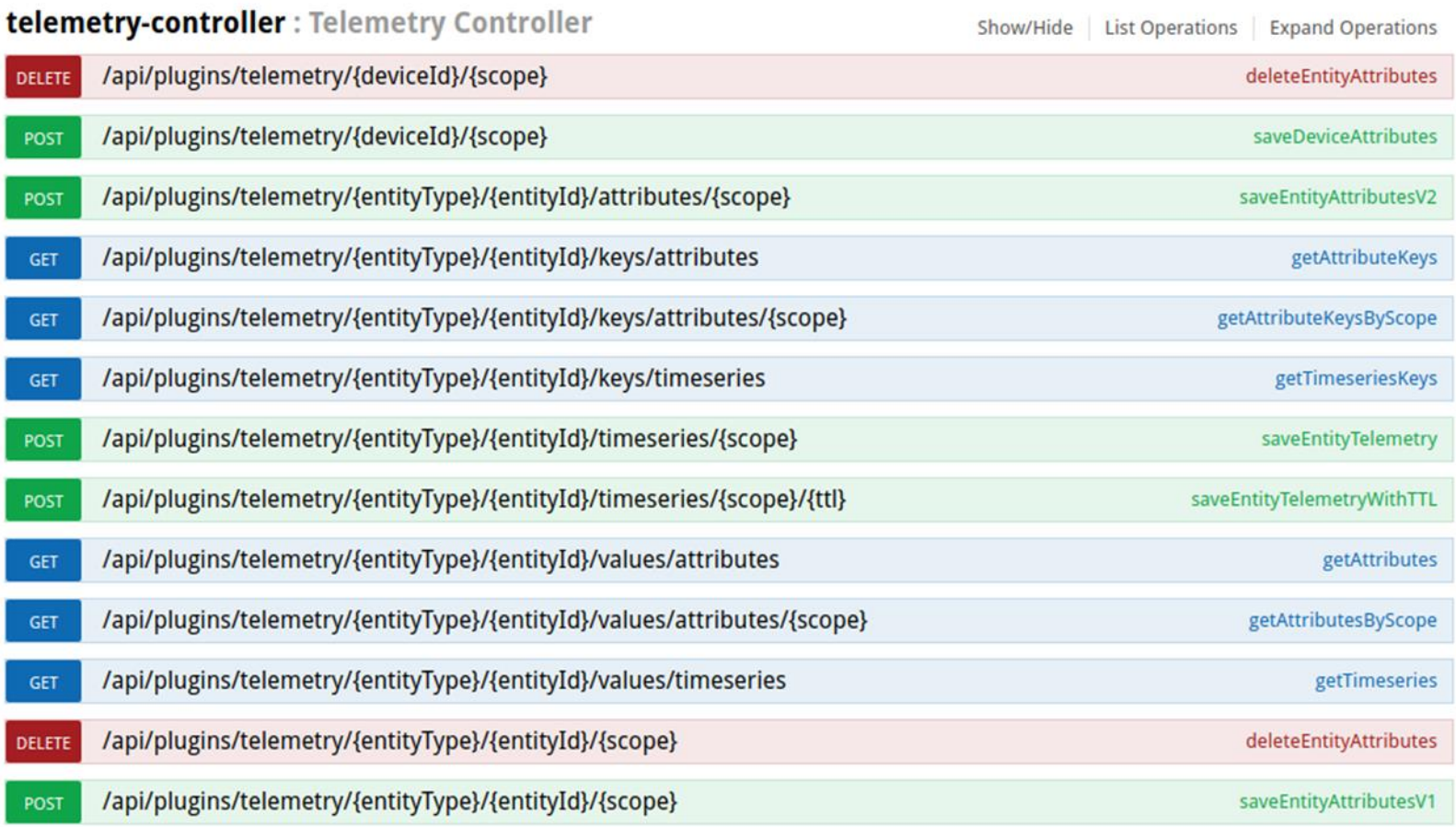
```
{"token":"$YOUR_JWT_TOKEN", "refreshToken":"$YOUR_JWT_REFRESH_TOKEN"}
```

For each API REST request, include this header field:

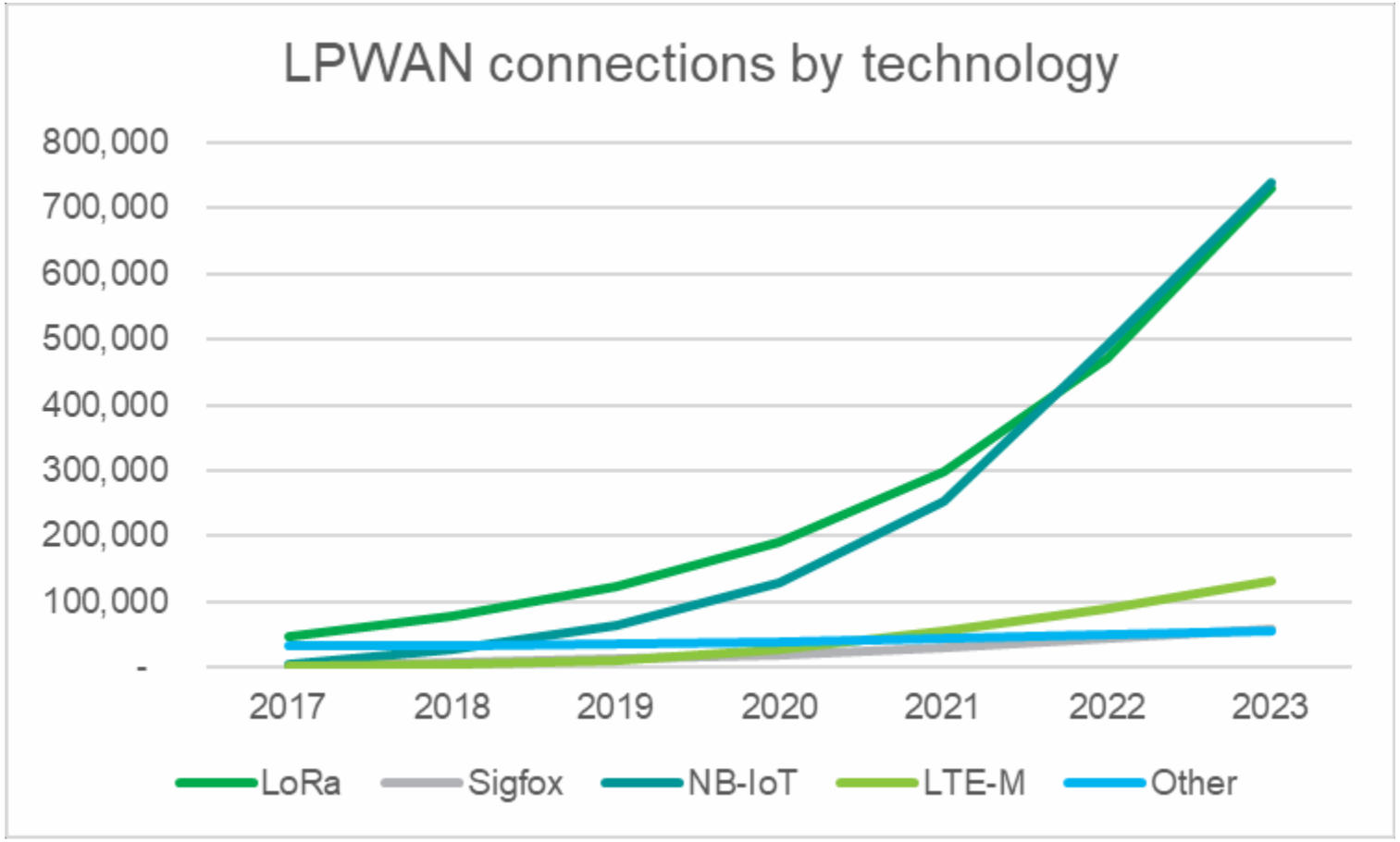
```
'X-Authorization':'Bearer TOKENHERE'
```

Swagger-ui

http://thingsboard.cloud/swagger-ui/



LoRa and NBIoT



Mit app inventor

https://appinventor.mit.edu/

MIT
APP INVENTOR

UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Internet of Things

a systemic perspective

Prof. Roberto Vezzani
University of Modena and Reggio Emilia
Email: roberto.vezzani@unimore.it

Milan, 10th May 2023