

# sens@campus reloaded

IoT end-devices management

F.Thiebolt, MP.Gleizes

{thiebolt,gleizes}@irit.fr

Edge Computing

# AIoT

towards the next level



D  
A  
T  
A  
  
L  
A  
K  
E



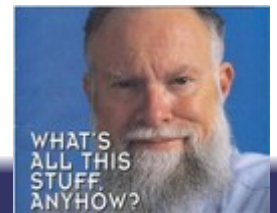
ne@campus



# What's all this **end-devices** stuff, Anyhow\* ?

- ne**OC**ampus overview,
- sens**OC**ampus: end-devices management,
- neo-sens**OC**ampus :-)

*\* Remember Ti's great analog engineer, Bob Pease*



- 2013, June, kick-off,
- ECO-CAMPUS Toulouse,
- B. Monthubert President



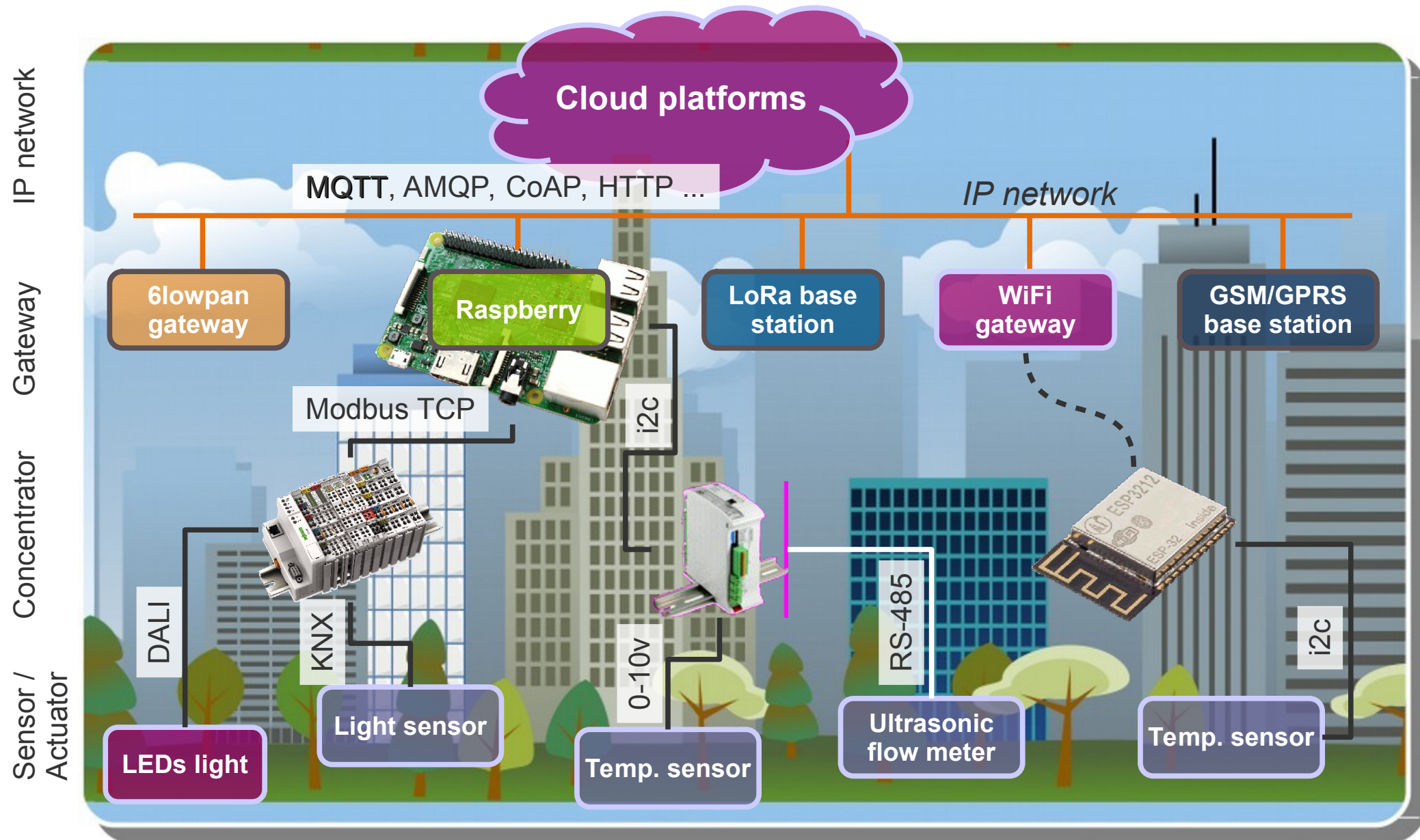
- Buildings ~ 407 000 m<sup>2</sup>,
- People ~ 36 000

- Well-being for users in the university community,
- Improvement of the ecological footprint of our buildings,
- Reduction of operating costs, especially for fluids.



Our campus is a small-sized town like Périgueux.

B. Monthubert





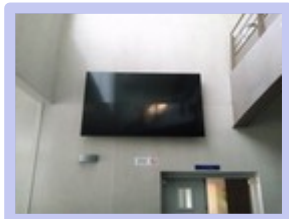


# neOCampus infrastructure

~ 40 x Raspberry Pi  
~ 3 x Concentrators  
(RPI + industrial Arduinos)



neOCom  
(connected displays)



myGates  
(automatic fences for  
autonomous vehicles)

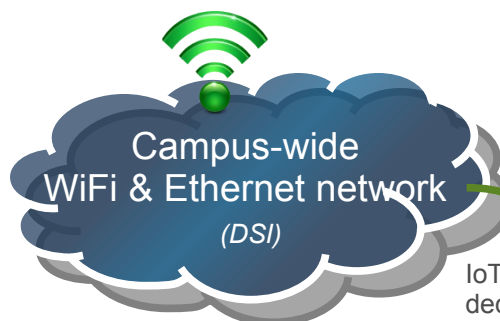


Federated  
LoRaWAN network  
(868MHz, 3 gateways, BU  
santé & sciences, Ecolab)



Note: LoRa deployment is a  
joint work with Pr R. Kacimi.

neOTraffic@BU,U4  
(attendance measurement)



IoT end-devices  
dedicated vlan

<https://neocampus.univ-tlse3.fr>

LoRaWAN server

<https://lorawan.univ-tlse3.fr>

~100 million data

Comms  
(MQTT)

~500 access / s

Apps.



1 x Dell R730 @ DSI

Connected hives  
(Rucher université)



neOSensors @BU  
(noise, temp., lux sensors)



~ 10 x neOSensors

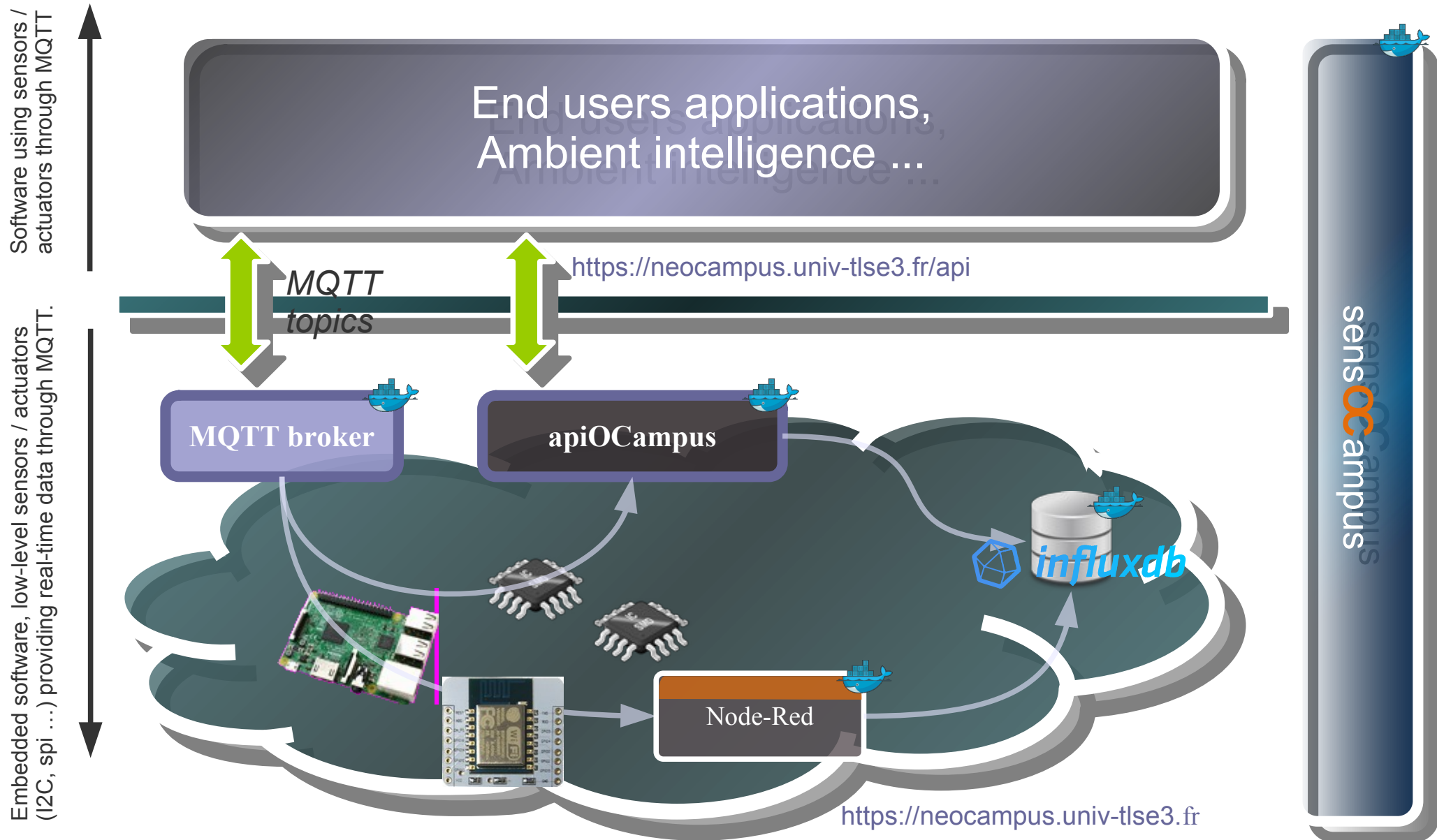
Amilab

(IRIT2-366)

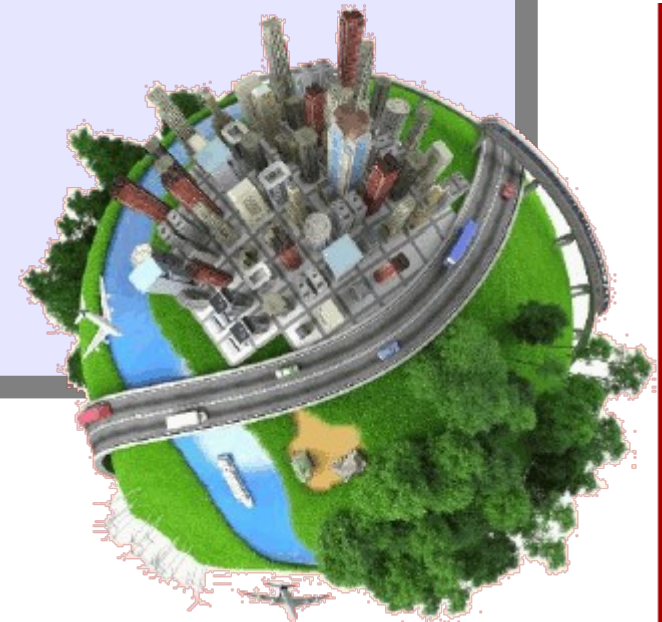
neOCampus R&D area



# neOCampus infrastructure

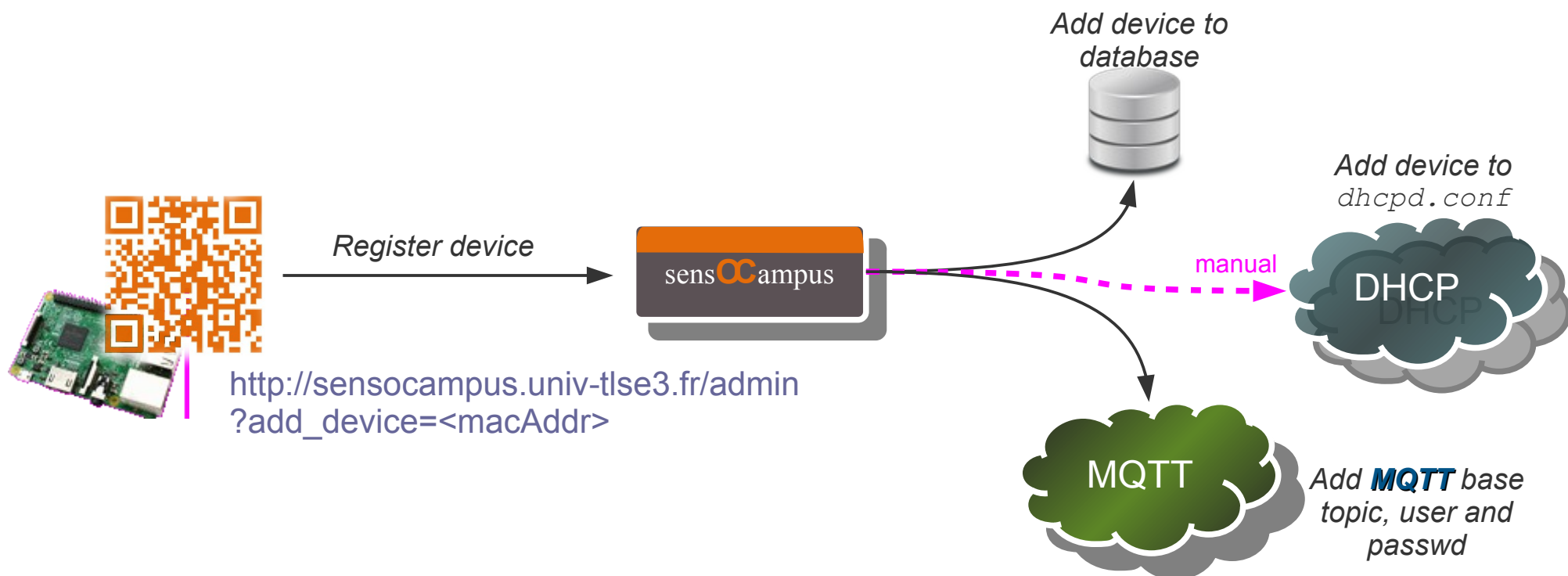


- neOCampus overview,
- sensOCampus: end-devices management,
- neo-sensOCampus :-)



- Device registration | sensOCampus web. App. (Django)

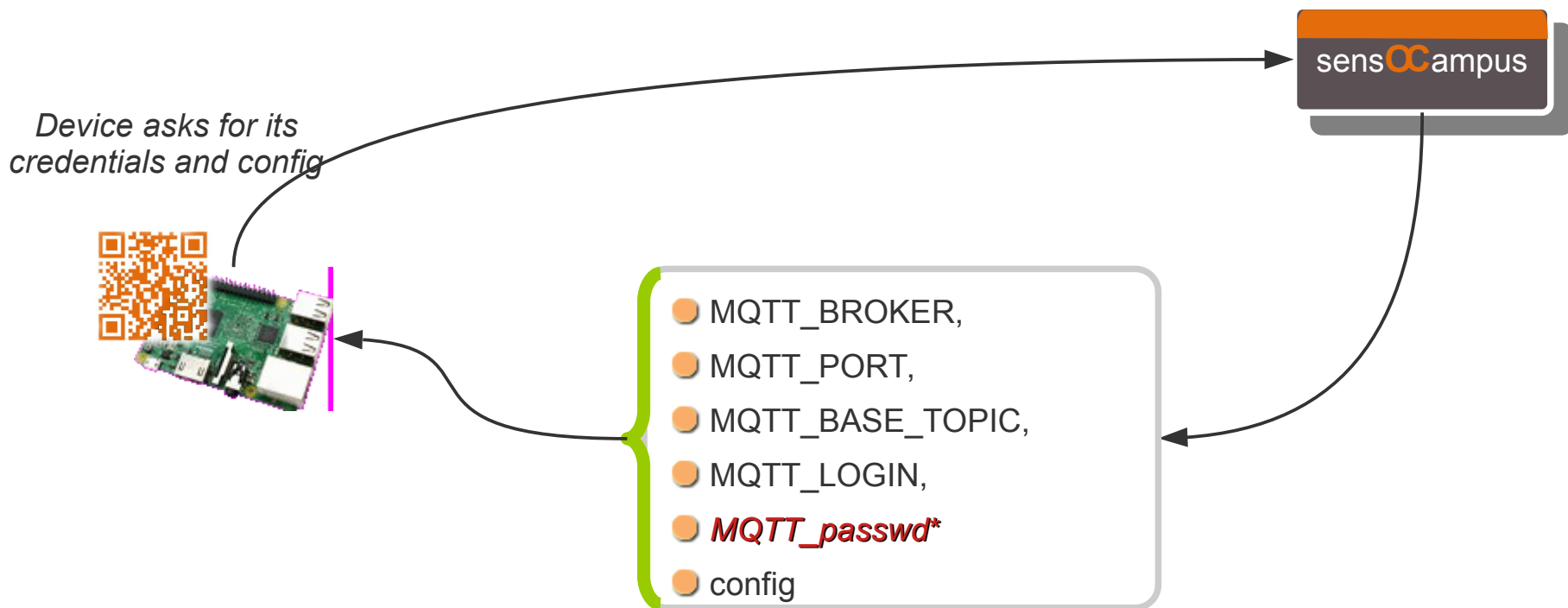
A **device** is a physical embedded system connected to a network (eg. Raspberry Pi, esp8266, stm32 ...).





- ... then registered device fetches credentials and configuration from sensOCampus

get credentials : <http://sensocampus.univ-tlse3.fr/device/credentials?mac=<macAddr>>  
get config (makes use of credentials) : <http://sensocampus.univ-tlse3.fr/device/config>

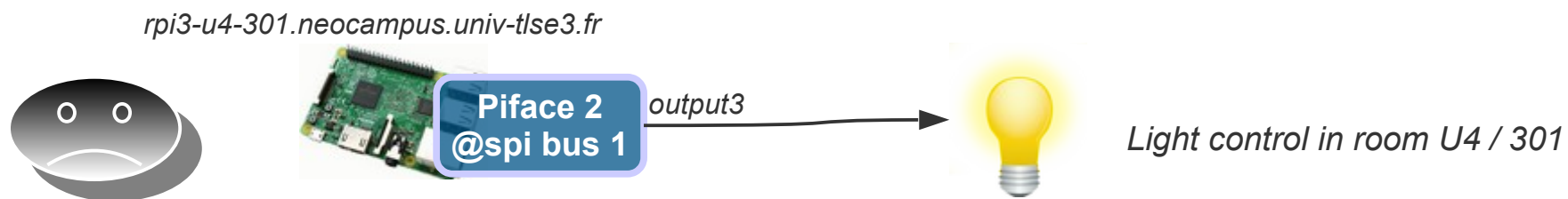


Minimum configuration sent from sensOCampus to a device

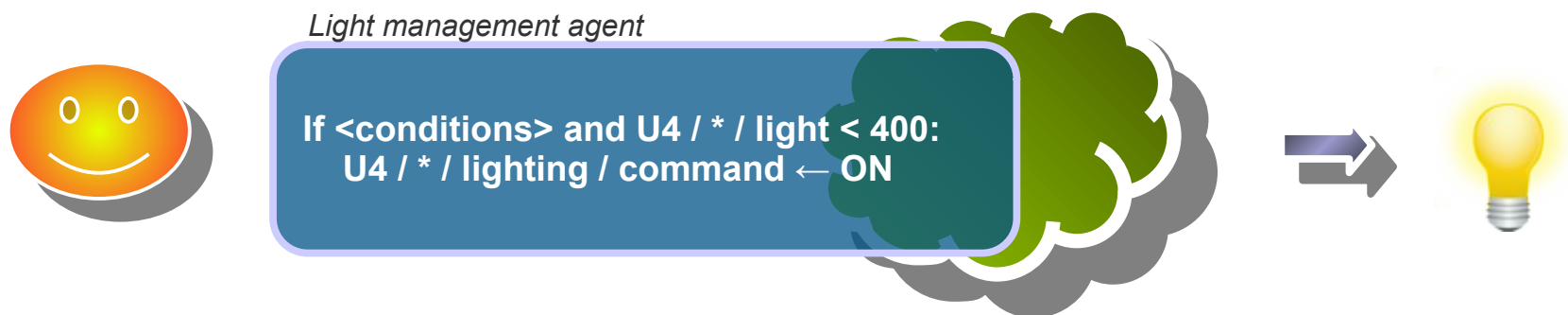
\*MQTT passwd is only sent on **first call** (admin action required to create a new one otherwise)

neOCampus gives users / applications access to useful data without hassle about networks, sensors technology or underlying embedded systems.

✗ High level of hardware details



✓ Useful data





# Data acquisition

- MQTT, almost the 'de facto' IoT protocol ;)
  - Network addressing independent,
  - Works behind firewalls,
  - publish / subscribe paradigm (no more pooling),
  - embedded security (login / passwd) + TLS,
  - Support for WebSockets,
  - Bindings for almost all languages,
  - Paho-mqtt (python client), Mosquitto (C written client & server),
  - MQTT bridges for multiples brokers setup,
  - Topics based real-time exchanges,
  - Topics are arbitrary tokens separated with '/'
  - Payload agnostic (mainly json).



MQTT data trade-off:  
topic vs payload content.

*You may also have a look to AMQP (e.g RabbitMQ) or CoAP (Constrained Application Protocol)  
Adafruit provides a free MQTT broker with a data visualisation GUI on a per-user basis.*

- Topics segmentation in neOCampus

u4 / campusfab / shutter / command



Base | type | [optional] command

**Base** : defined at device registration time according to location

*e.g u4 / 300 or bu / hall ...*

**Type** : kind of sensor / actuator (module) defined by sensOCampus or automatically detected

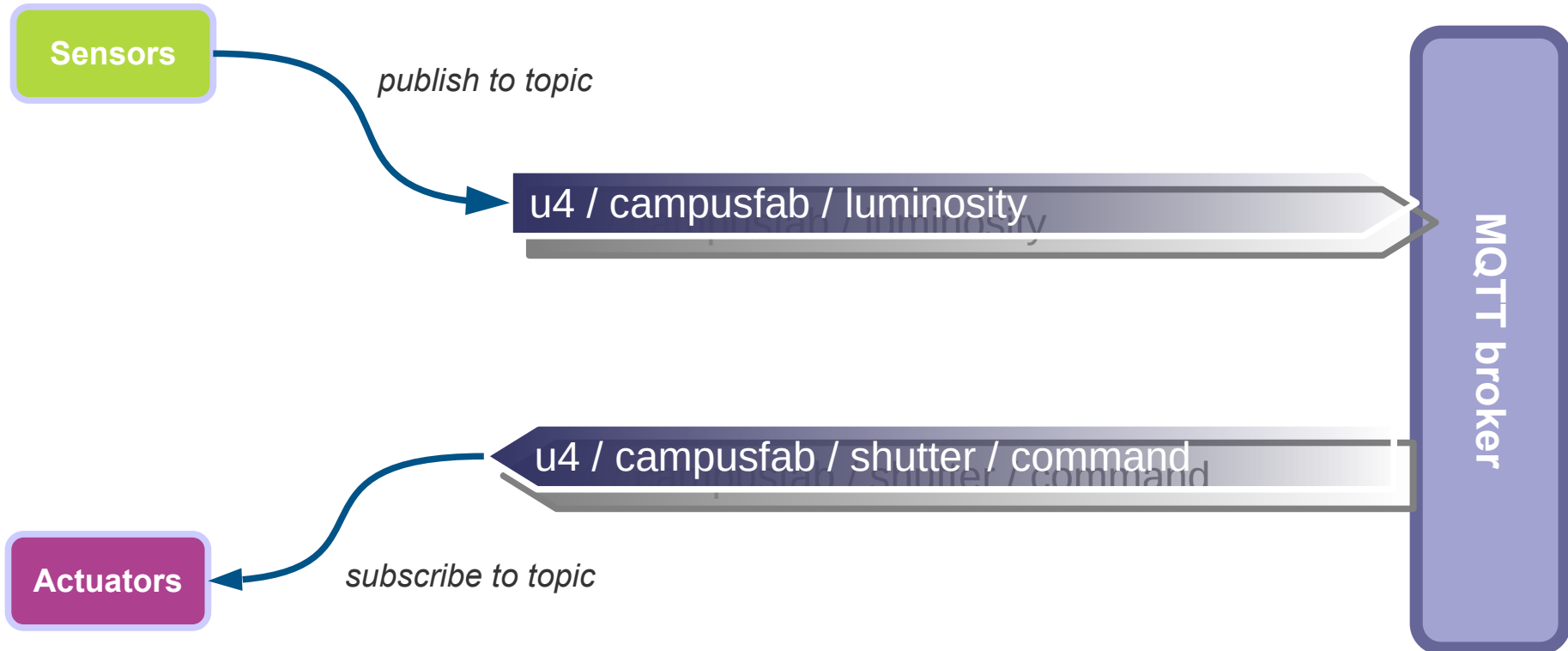
*e.g shutter, luminosity, temperature, sound, lighting ...*

**Command** : to send orders to a sensor / actuator (module)

*e.g orders to shutter like UP, STOP, DOWN*



- Real-time data exchange through topics



Since actuators initiate a TCP connection to the broker, they can be sent data back from the broker even when they are located behind a firewall (e.g Internet box).

## MQTT payloads @ neOCampus

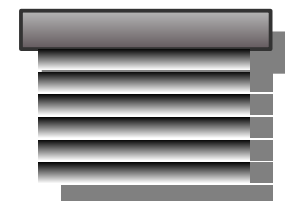
- ✓ Sending order to a shutter (with proper mqtt login / passwd)

```
order: "up"  
dest: "all"  
      or "<shutter_ID>"
```

*Json frame as mqtt payload*

u4 / campusfab / shutter / command

Shutter



- ✓ ... then shutter publish its status back

```
order: "idle"  
unitID: "<shutter_ID>"  
status: "open"
```

*Json frame as mqtt payload*

u4 / campusfab / shutter

One caveat is that you can't send an order to a single module (shutter), hence the `dest` field.

- Wildcards for multi-topics subscribing

u4 / # / temperature

*multi-level subscribing (e.g u4 / campusfab / temperature, u4 / hall / box1 / temperature)*

u4 / + / temperature

*single-level subscribing (e.g u4 / campusfab / temperature, u4 / 301 / temperature)*

- Single topic publishing



MQTT specifications does not allow to publish to topics containing wildcards.

*The `multiple` method enables you to publish multiple data to multiple topics in a one-shot way.*

- neOCampus overview,
- sensOCampus: end-devices management,
- neo-sensOCampus :-)







- To make things clear ...

- ▶ we'll keep all of the existing features of the current release,  
we need some upgrade of existing features along with new ones (e.g a dashboard)
- ▶ back-end will evolve from Django1.X to the latest Django release,  
... probably through a complete rewrite of the existing code
- ▶ a real front-end ...  
... that will be based on the **React** framework
- ▶ monitoring,  
otherwise how could you monitor your whole infrastructure ??
- ▶ for dev/testing purposes, ability for users to **ssh** as **root** within the container !



- ... in a nutshell:

- ▶ additional kind of end-devices: **LoRaWAN**,

up to now, end-devices were identified by their MAC address, we'll now add UUIDs for LoRaWAN kind of end-devices.

- ▶ new 'site' location along with 'outdoor' base topics,

all end-devices were supposed to belong to the same **UT3** site. It will still remains the default site but we need to add new sites along with outdoors locations.

- ▶ a **dashboard**,

mandatory feature to keep track of what's going on; need for a dynamic front-end based on React.

- ▶ **Prometheus** monitoring,

As for all neOCampus services, we need the ability to monitor them ! These data will get grabbed by our future services dashboard :)

- ▶ interactions with the **new authentication plugin** from mosquitto2.0

The new mosquitto release now integrates an authentication plugin mechanism. We'll make use of **Mosquitto Go Auth\*** that behaves quite the same as jpmens' mosquitto-auth-plug we currently use.

\* <https://github.com/iegomez/mosquitto-go-auth>

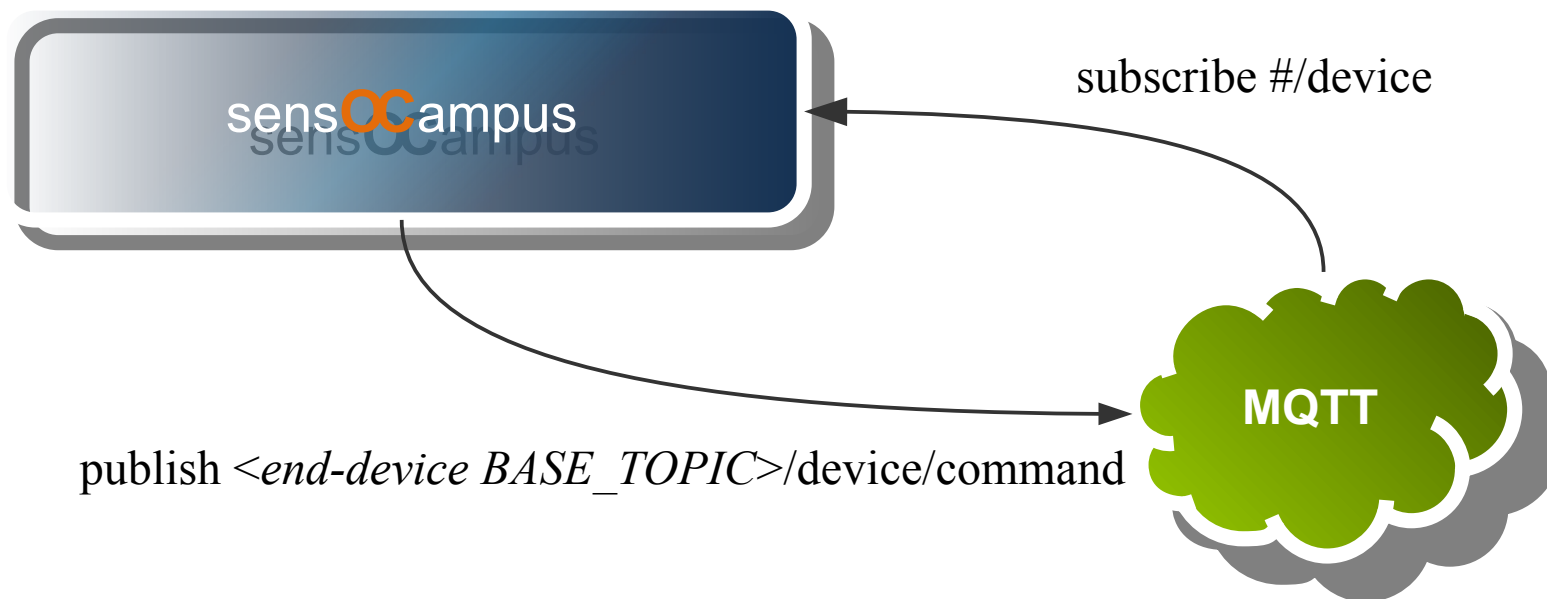


# end-devices attributes

- Ethernet (MAC address) or LoRaWAN UUID
- Device\_type: rpi3, esp32, jetson\_nano ...  
this list can get augmented with new devices types
- Location: site / building / room  
link to another table with declared sites, buildings from sites and rooms from building (see location slides farther)
- Configuration  
a json structure that needs to be well structured (e.g validictory)
- MQTT  
either defaults tick box or MQTT\_SERVER and MQTT\_PORT  
login/passwd with delivered flag (passwd sent first time)  
MQTT\_BASE will get determined by the end-device location  
LoRaWAN end-device do not need MQTT connectivity
- last\_seen  
date field last time the end-device was seen
- last\_status  
text field about last status sent by end-device

UUID\_type: Ethernet / LoRaWAN / ...  
UUID: mac address or hweui(lorawan) or ...  
Device type: one among an expandable list  
Description (text filed)  
Location: one among a location list  
Enable/disable end-device flag  
Configuration: a json structure intended to an end-device  
MQTT (server, port, login, passwd)

- last\_seen & last\_status



- Each end-device publish its status every 30mn in BASE\_TOPIC/device
- sensOcampus will subscribe to #/device
- sensOcampus can send orders to end-devices (see reference doc)
- sensOcampus itself needs MQTT credentials (docker env. vars)





# end-devices special locations

- abroad sites (i.e **non UT3** site) and outdoor location specifications

outside	outdoor <b>UT3</b> end-devices e.g outside/ambient/rain or outside/access for our access control system
'abroad/<site>' MQTT topic prefix for <b>non UT3 end-devices</b>	
abroad	<b>non UT3</b> end-devices abroad/<site>/...  e.g for indoor sensors at Carcassonne site abroad/ <b>carcassonne</b> /<building>/<room>/<kind>  e.g for outdoor sensors at Carcassonne site abroad/ <b>carcassonne</b> /outside/...



# Dashboard

- ... or how to avoid Django admin static pages :|

- ▶ we need a dynamic presentation of our end-devices status,  
we don't need websockets but rather a more dynamic and synthetic presentation of end-devices status across our campus and all the others sites
- ▶ outdated end-devices (i.e unseens for > 30mn) will turn red and some intelligent alerts will pop-up on the interface and/or remotely,
- ▶ we'd like this **dashboard** to be React based,
- ▶ more and more embedded sensors are sending status as a json frame embedding many different informations (FW revision, free memory available, current number of sensors): we'd like the ability to customize the presentation of these information,
- ▶ we'd like the ability to easily send order to end-devices: e.g sending a reboot or a full reinstall order (in this case, we need to reset its credentials)
- ▶ ... more to come

- **Prometheus** monitoring of neo-sensOCampus service

- ▶ neo-sensOCampus service will provide a **prometheus** end-point,  
Prometheus is pull-based (http) request
- ▶ Later, each neOCampus service will get monitored through a **prometheus** end-point,
- ▶ Django – Prometheus integration seems quite easy  
<https://docs.timescale.com/latest/tutorials/tutorial-howto-monitor-django-prometheus>
- ▶ We'll be running both Prometheus and Grafana containers ... a bit later,
- ▶ TBD: Prometheus alerting mechanism,
- ▶ TBC: what about services behind a firewall ? **pushgateway** ??





# References

- sensOCampus wiki  
<https://neocampus.univ-tlse3.fr/projects/sensocampus>
- github repository ... coming soon
- sensOCampus reference document  
[https://neocampus.univ-tlse3.fr/\\_media/sensocampus\\_end-devices\\_api.pdf](https://neocampus.univ-tlse3.fr/_media/sensocampus_end-devices_api.pdf)
- existing Django1.x source code ... coming soon
- discord <https://discord.gg/KvZNqCEW>



# Livrables

- source code in our github repository (to come soon),

We'll provide our regular dev. env along with ssh connectivity within container.

- existing sensOCampus database imported into neo-sensOCampus,

TBC: maybe just a matter of applying migrations ...

- neo-sensOCampus software in a docker container able to manage both existing end-devices along with new LoRaWAN end-devices, either located outdoor, inside or outside the default UT3 site.

● Proposal for new buildings :



*Hey, that's the law!*

1% for art (**200K€**)

*Proposal for locals PME/PMI and labs !*

1% for **local** innovations



**Nvidia Jetson Nano**

- 4GB RAM
- 4 x ARM 64bits CPU
- 128 cores GPU up-to 472GFlops
- 10w max.

autOCampus Edge computing horsepower  
Jun.19 worldwide availability, #100€



**MyOpenCam project**

- custom built firmware
- M1 UE projet
- cheap open-source IP camera



**LoRaWAN BSFrance**

- ultra low power STM32
  - embedded LoRa module
  - display and battery support
- French company (Aude), 15€



**Google TPU**

- USB3 NN accelerator