LAB 4 – Exploring Integration Patterns with Bluetooth, CoAP, & MQTT

Introduction

UNIVERSITEIT GENT

CoAP

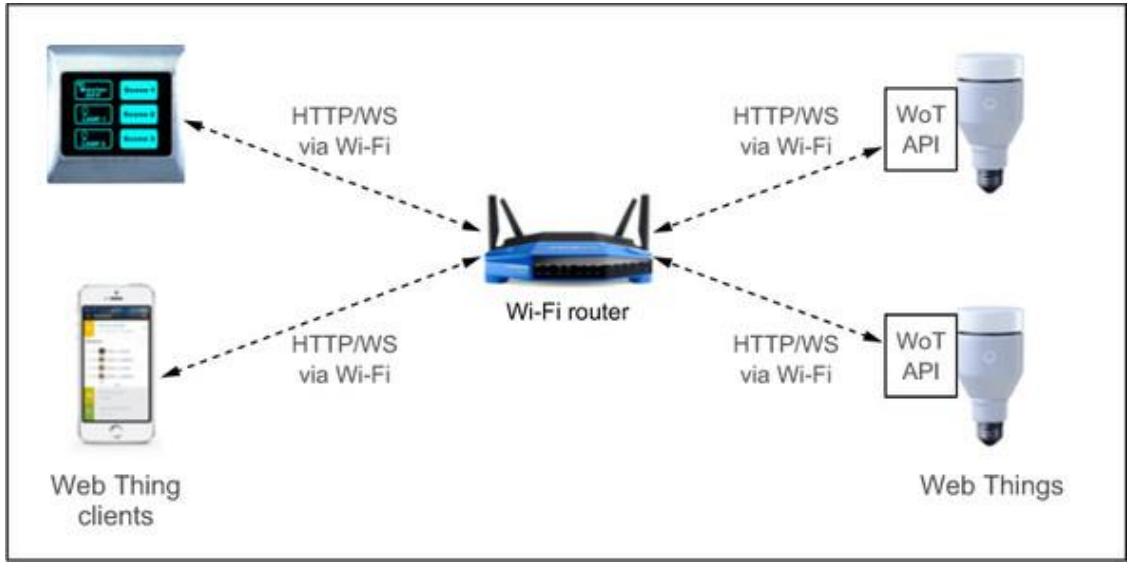
MQTT

Integration patterns

1. Direct integration pattern
2. Gateway integration pattern
3. Cloud integration pattern

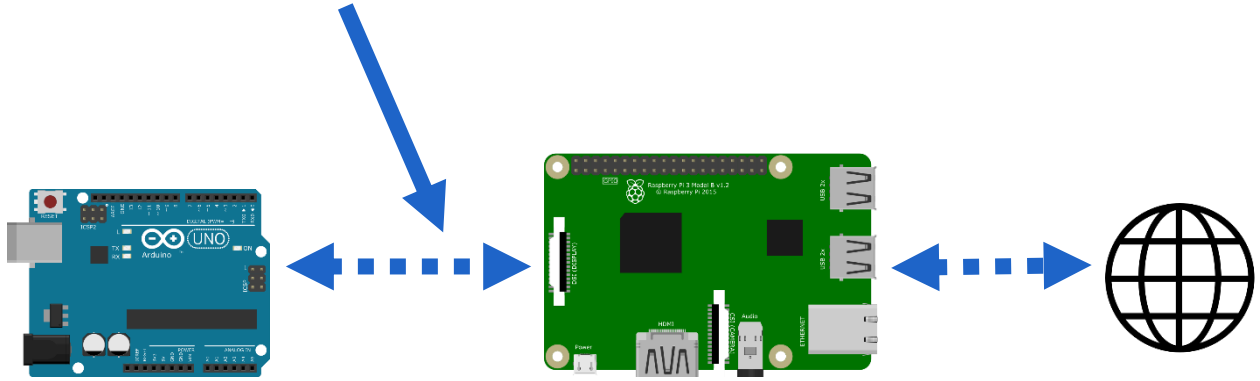
Direct integration pattern

- Direct connection to internet
- Support HTTP and TCP/IP
- Expose web API directly
- Independent of internet
- Type of **device**
  - Not battery powered
  - Short latency needed
  - Decent bandwidth
  - Capable of running web service
  - Direct access from clients required

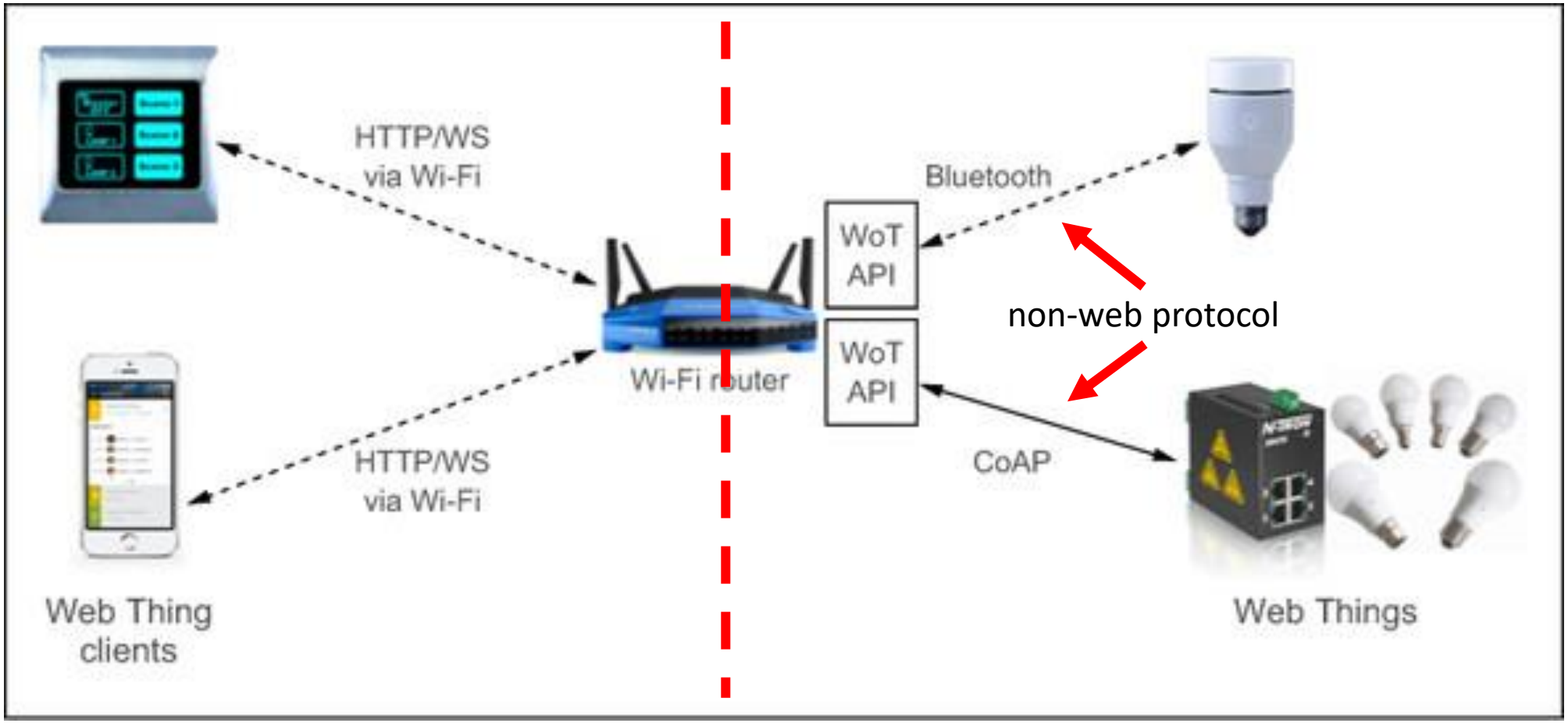


Gateway integration pattern

- Resource constrained devices
- Talk to more powerful intermediary device (**WoT gateways**)
- Communicate via low power, non-web protocol
- **Gateway**
  - Exposed to the internet
  - Translates incoming messages
  - Sometimes extra functionality
- Type of **device**
  - Limited resources (**battery**, memory, processor, …)
  - Can’t serve HTTP requests directly

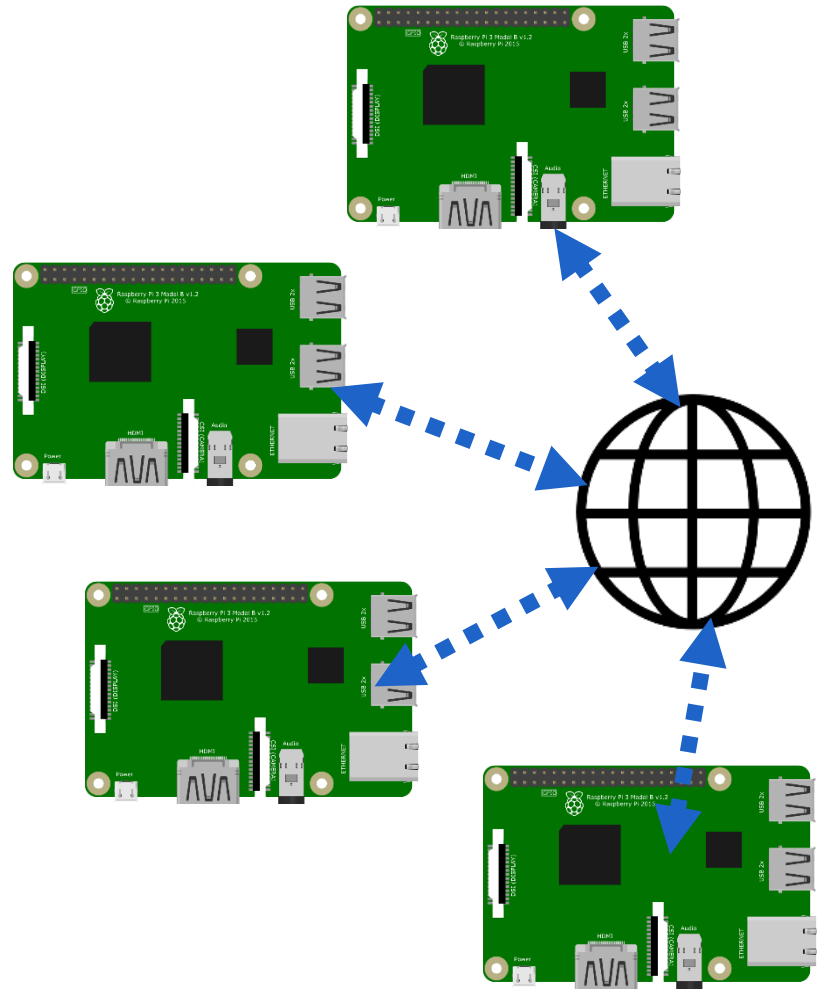


Gateway integration pattern

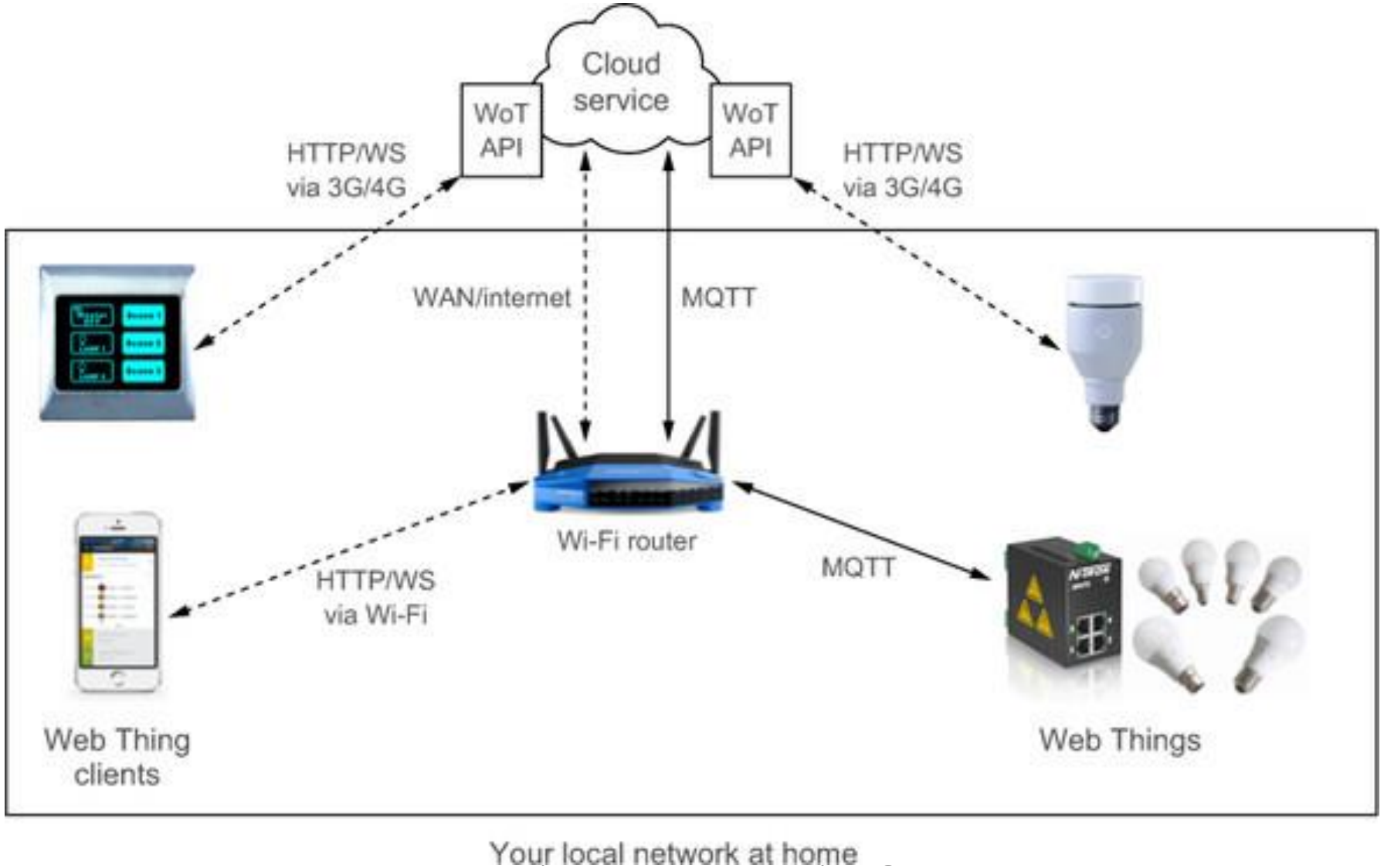


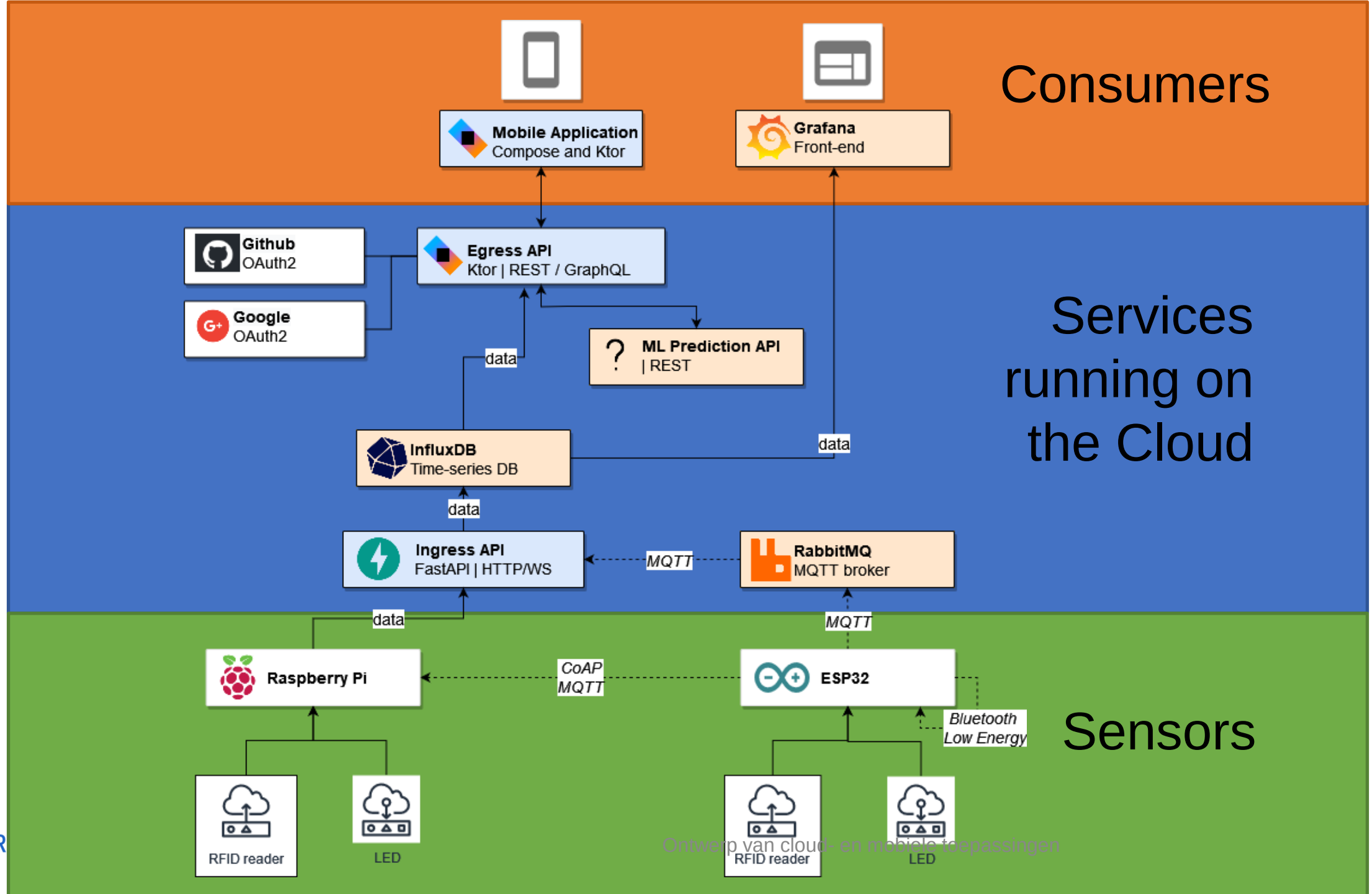
Cloud integration pattern

- Extension of gateway pattern
- Large quantities of devices and data
- More powerful and scalable platform
- More features
- Easily integrated into third-party systems

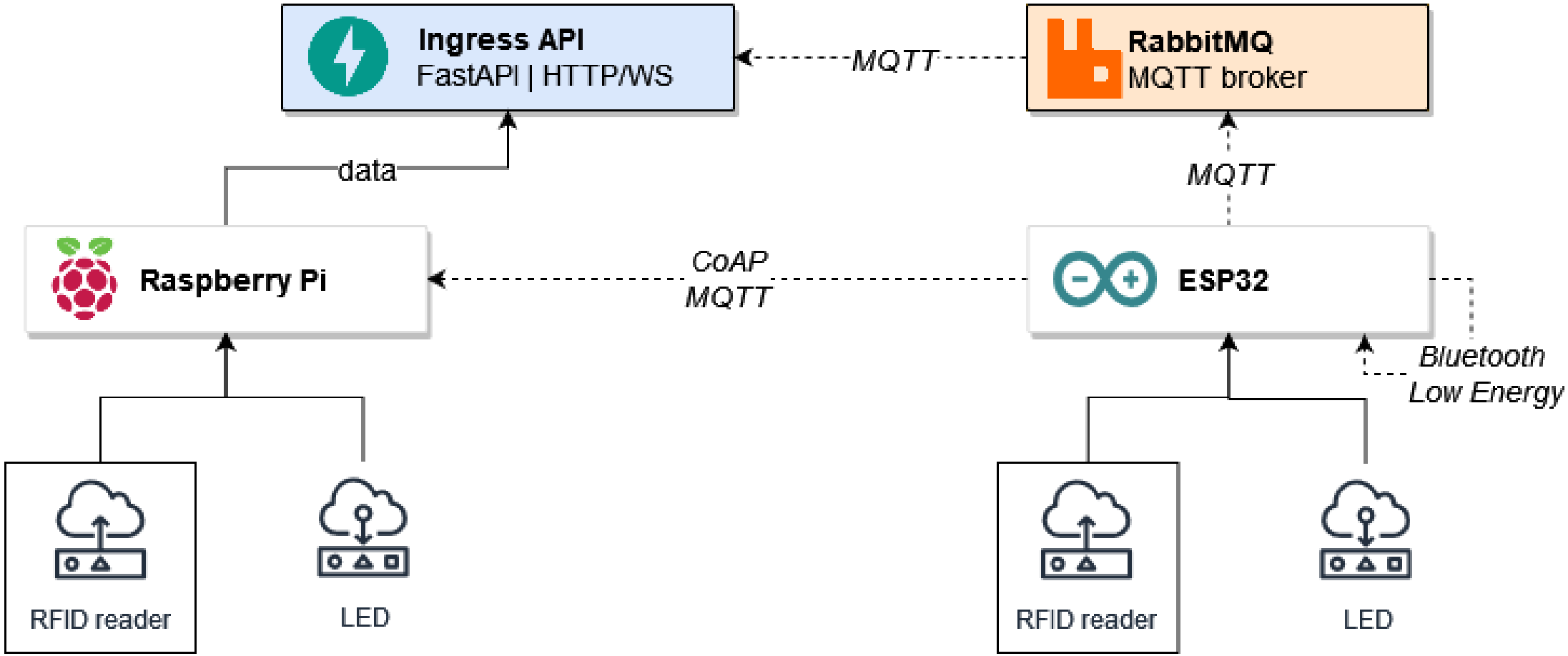


Cloud integration pattern




Consumers
Mobile Application
Compose and Ktor
Grafana
Front-end
Services
running on
the Cloud
Github
OAuth2
Google
OAuth2
Egress API
Ktor | REST / GraphQL
ML Prediction API
| REST
data
InfluxDB
Time-series DB
data
data
Ingress API
FastAPI | HTTP/WS
MQTT
RabbitMQ
MQTT broker
data
MQTT
Raspberry Pi
CoAP
MQTT
ESP32
Bluetooth
Low Energy
Sensors
RFID reader
LED
RFID reader
LED

Lab4


Ingress API
FastAPI | HTTP/WS
MQTT
RabbitMQ
MQTT broker
data
MQTT
Raspberry Pi
CoAP
MQTT
ESP32
Bluetooth
Low Energy
RFID reader
LED
RFID reader
LED

Goals

1. Send data from the ESP32 via **Bluetooth** Classic and Bluetooth Low Energy (BLE)
2. Send data from the ESP32 to the RPi using **CoAP**
3. Let the RPi act as a **gateway** and pass incoming data from CoAP to your Ingress API
4. Publish data **directly** from the ESP32 to a broker via **MQTT** and have the RPi subscribe to that data
5. Modify your Ingress API to receive data from an MQTT broker in the **cloud**, which receives data from the ESP32.
6. Implement **integration patterns** for connecting Things to the web

Material

Own material

- Laptop & charger
- Smartphone & charger
- Ethernet cable (optional)
- **Micro-USB cable for the ESP32**

Material available in classroom

- (mobile) Screen with HDMI (& power cable (USB to barrel jack))
- HDMI cable
- Keyboard
- Mouse

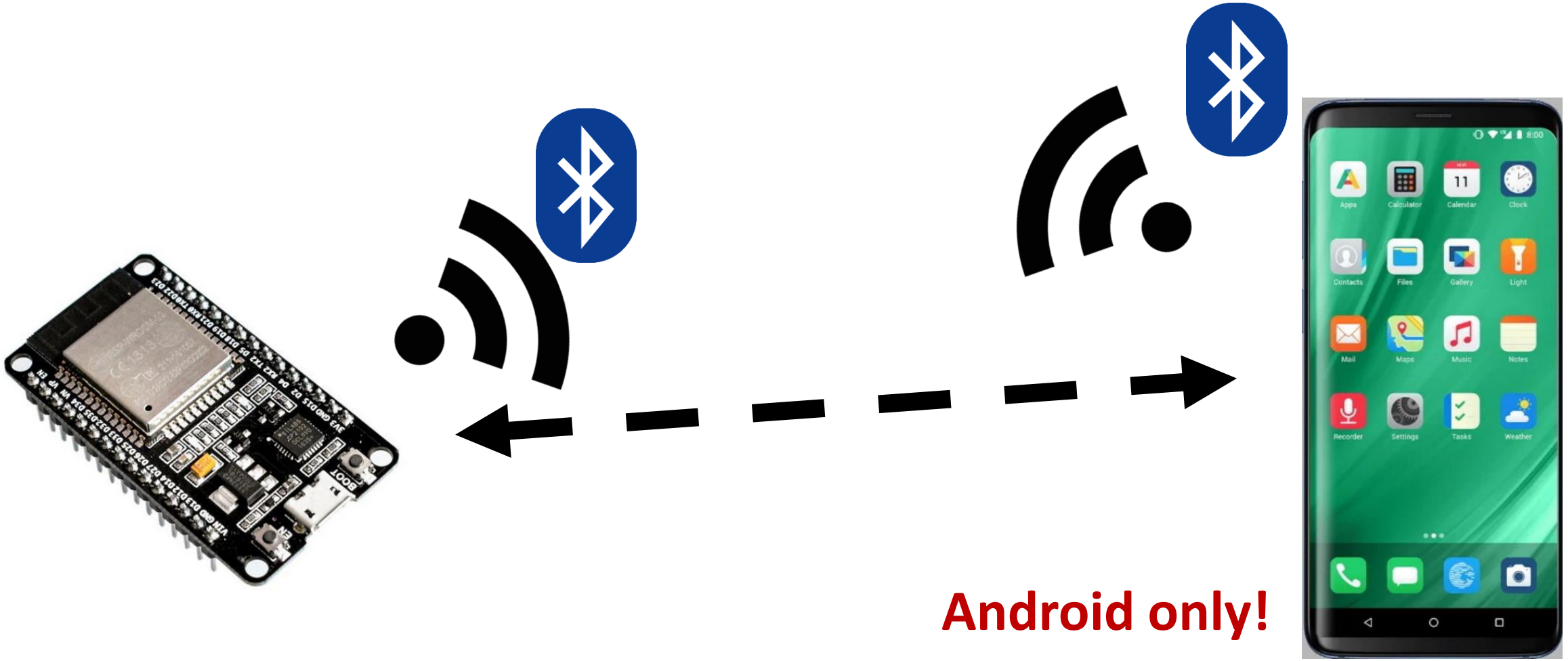
@Home: Preparation Work: Installation

- CoAP and MQTT on RPi
  - CoAPthon3
  - Mosquitto broker
- Arduino IDE
  - CoAP simple library
  - PubSub client (MQTT)
- Bluetooth on Rpi (optional)
  - Bluetooth set of libraries
  - Bluepy (to connect with BLE)
- Bluetooth app on smartphone

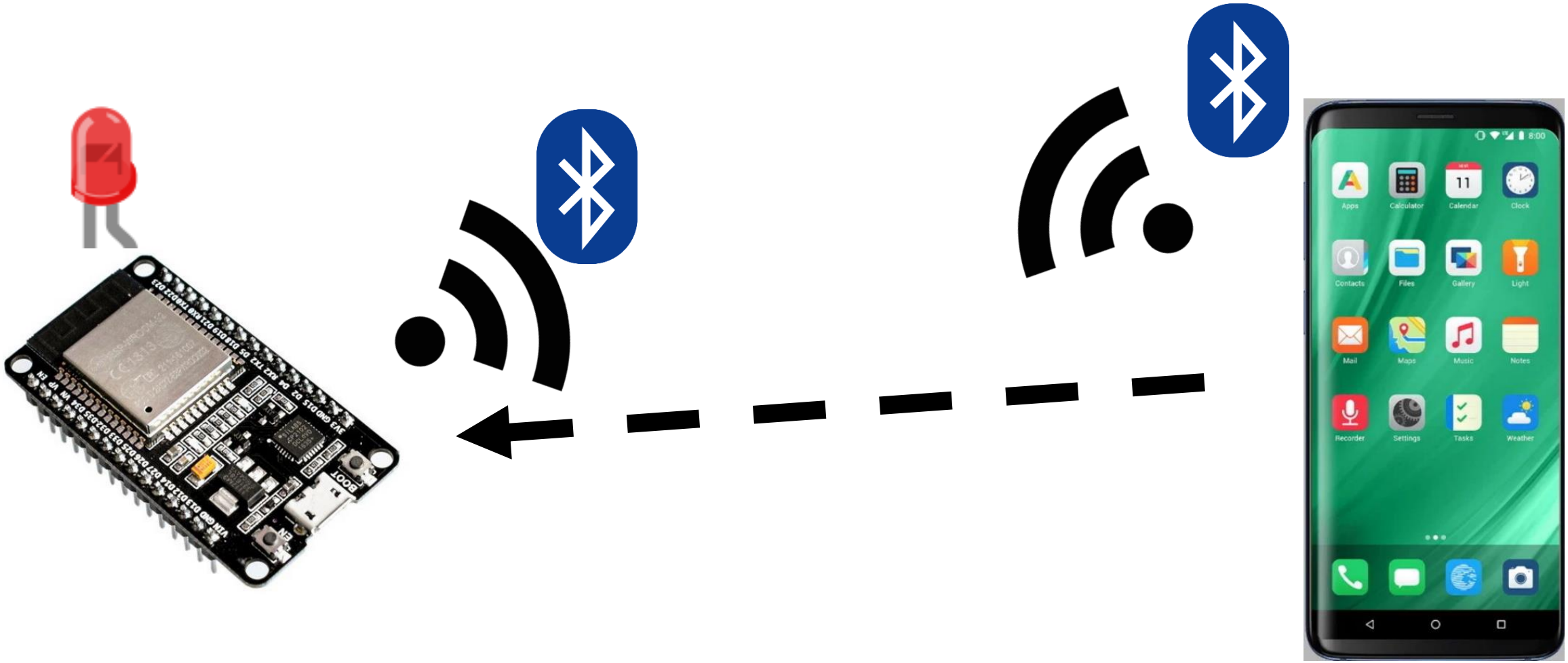
In Lab: Installation

- VNC Viewer
  - 2 week free trial period
  - Install only in lab when we come together

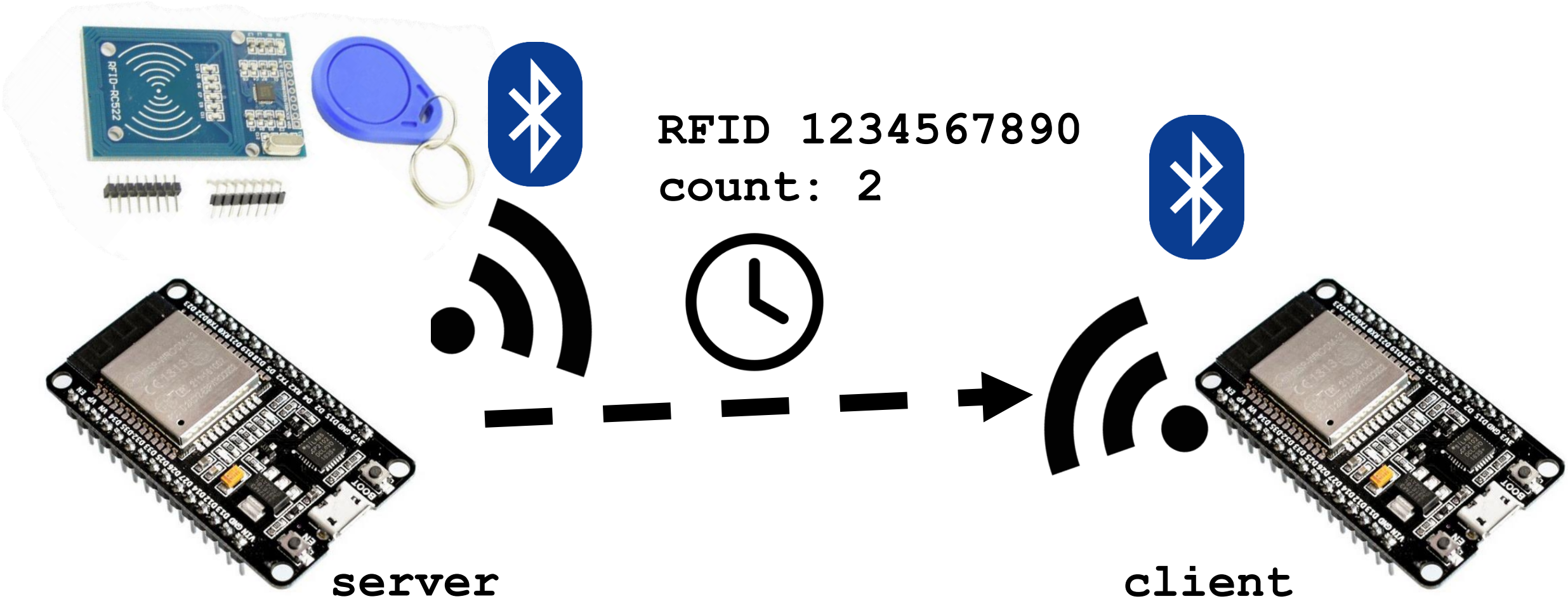
In Lab: Task1: Send data with Bluetooth



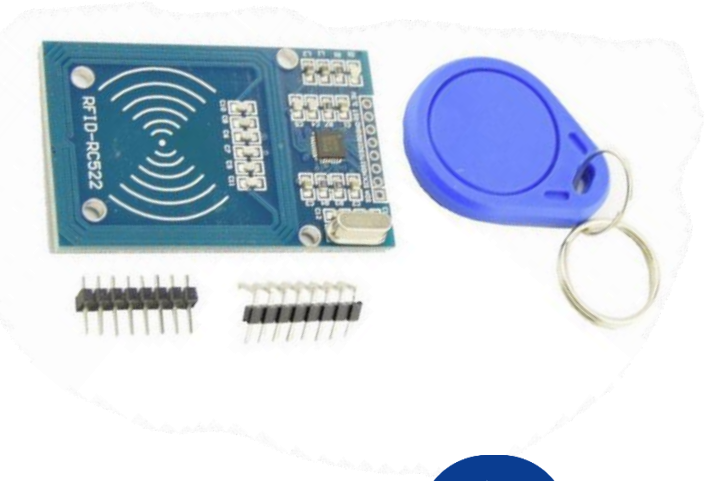
Bonus Task2: Control LED with Bluetooth

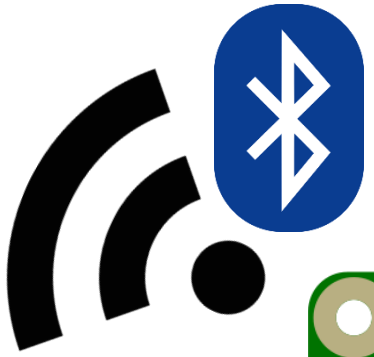


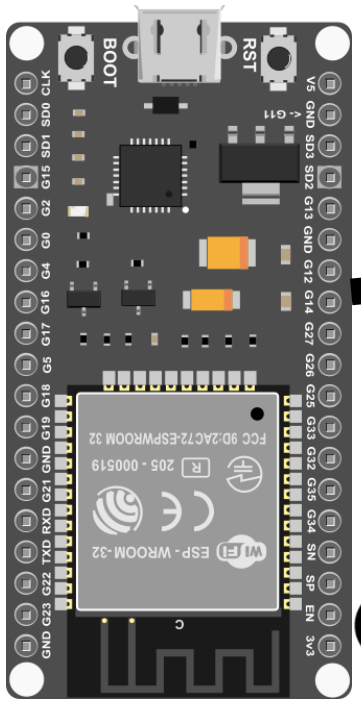
In Lab: Task3: Send data with Bluetooth Low Energy



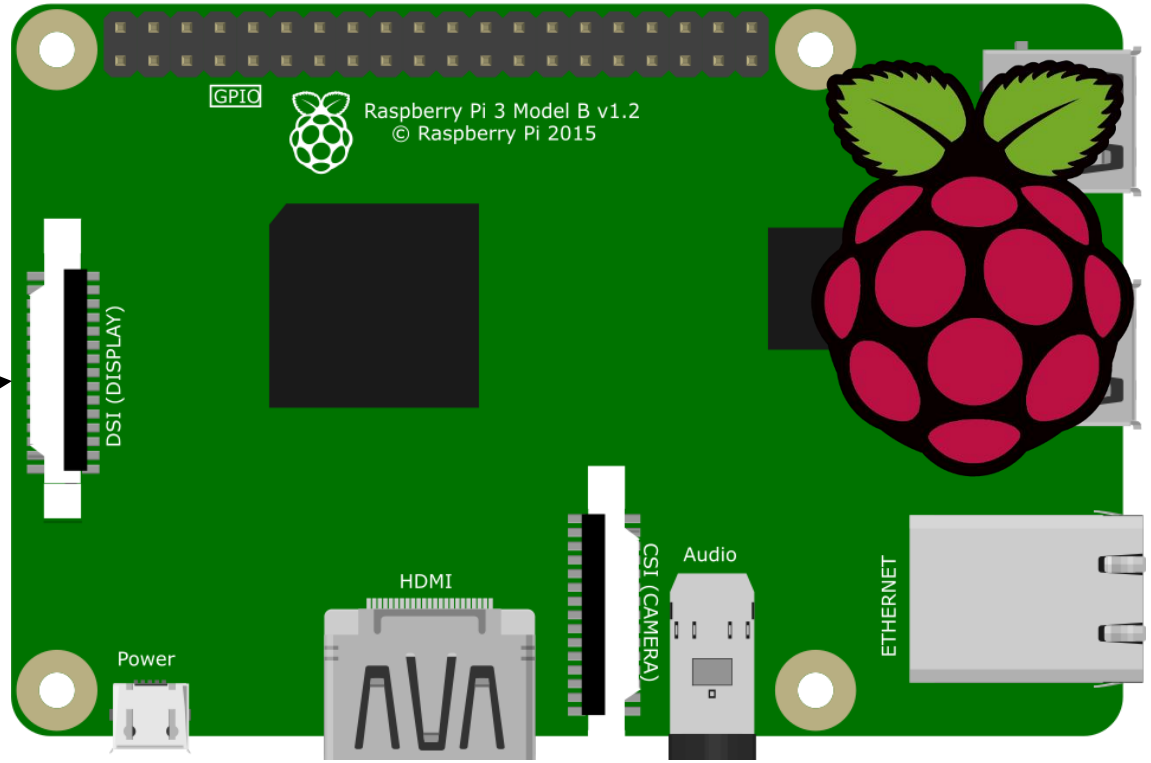
Bonus Task4: Send data via BLE to RPi



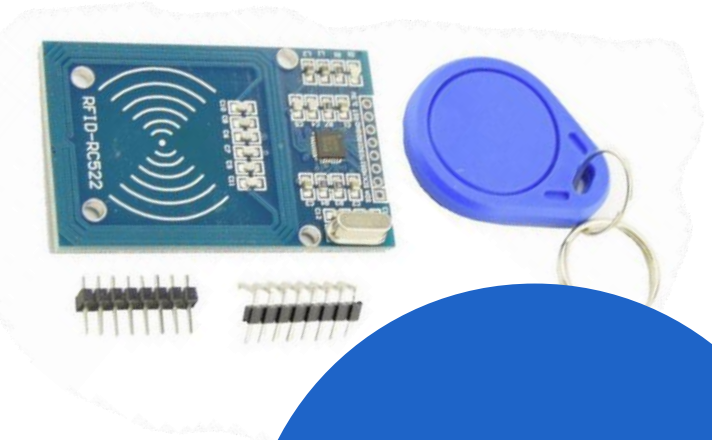


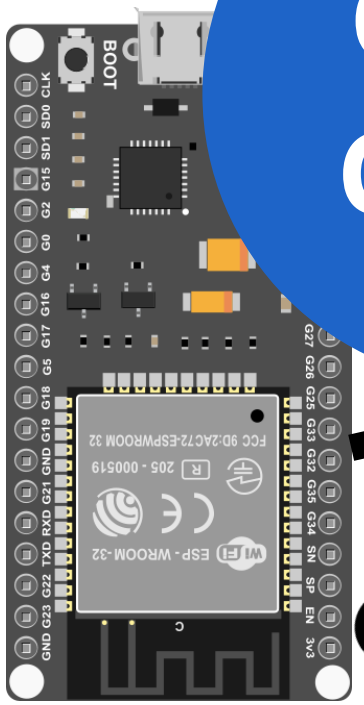


RFID 1234567890
count: 2



In Lab: Task5: Send data via CoAP to RPi

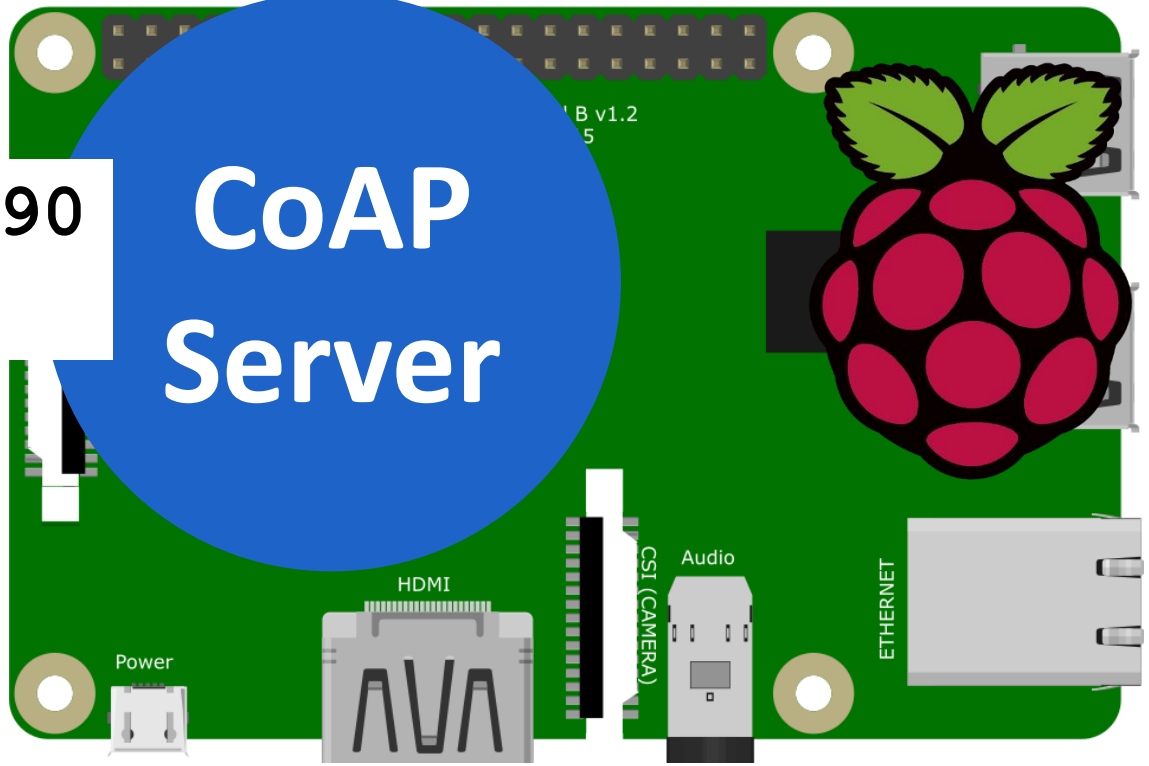




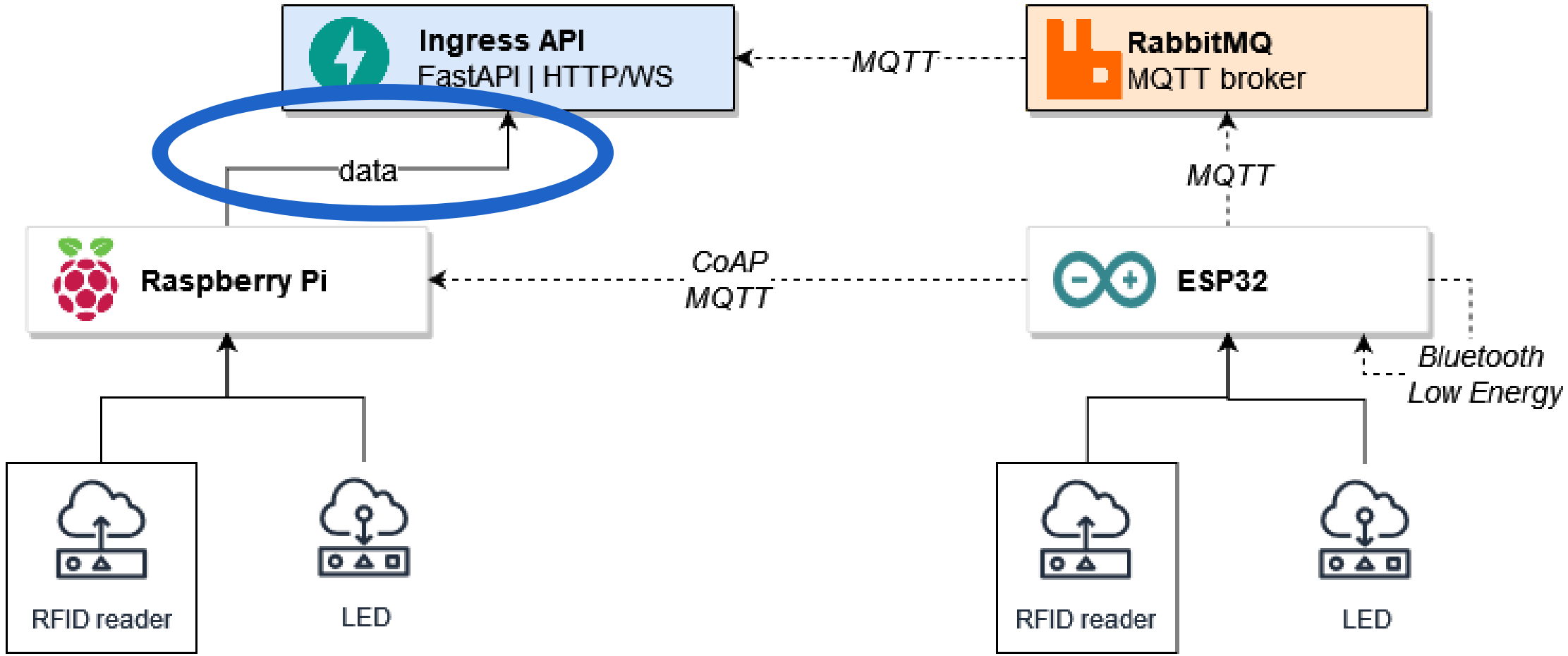
CoAP Client

RFID 1234567890
count: 2

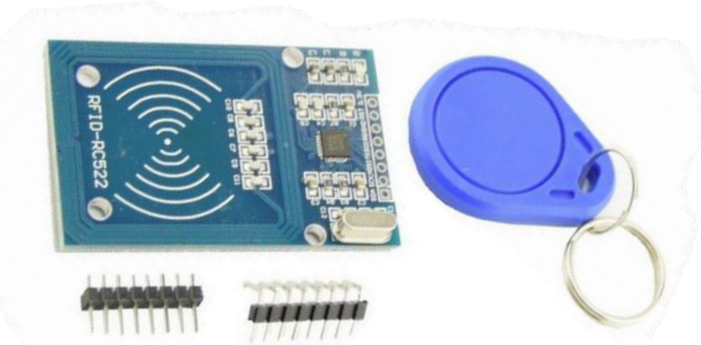
CoAP Server

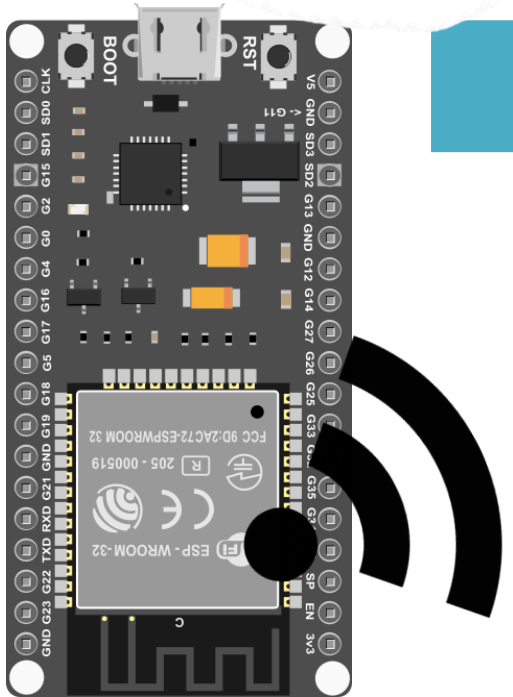


In Lab: Task 6 - RPi is gateway to Ingress



In Lab: Task7: Publish data directly to RPi via MQTT



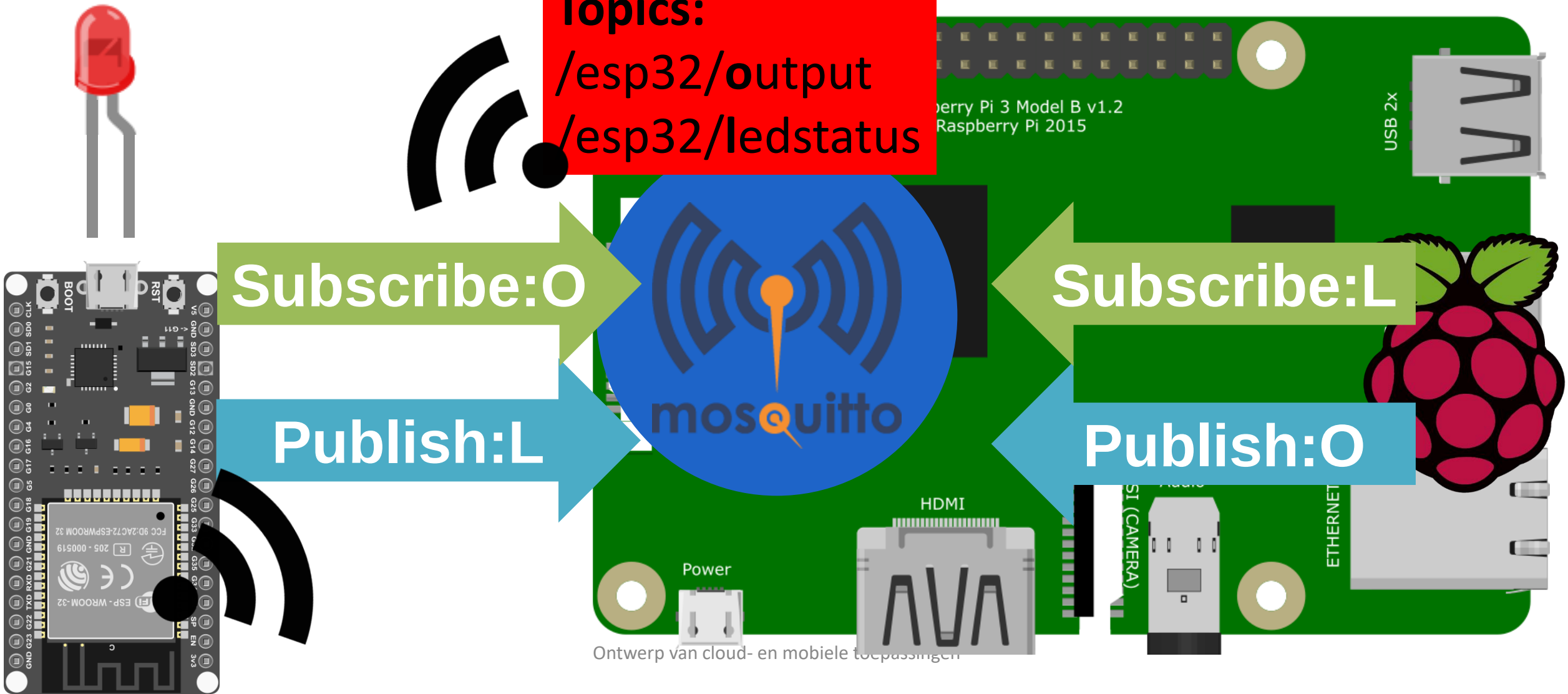


Topic:
/esp32/tagscan

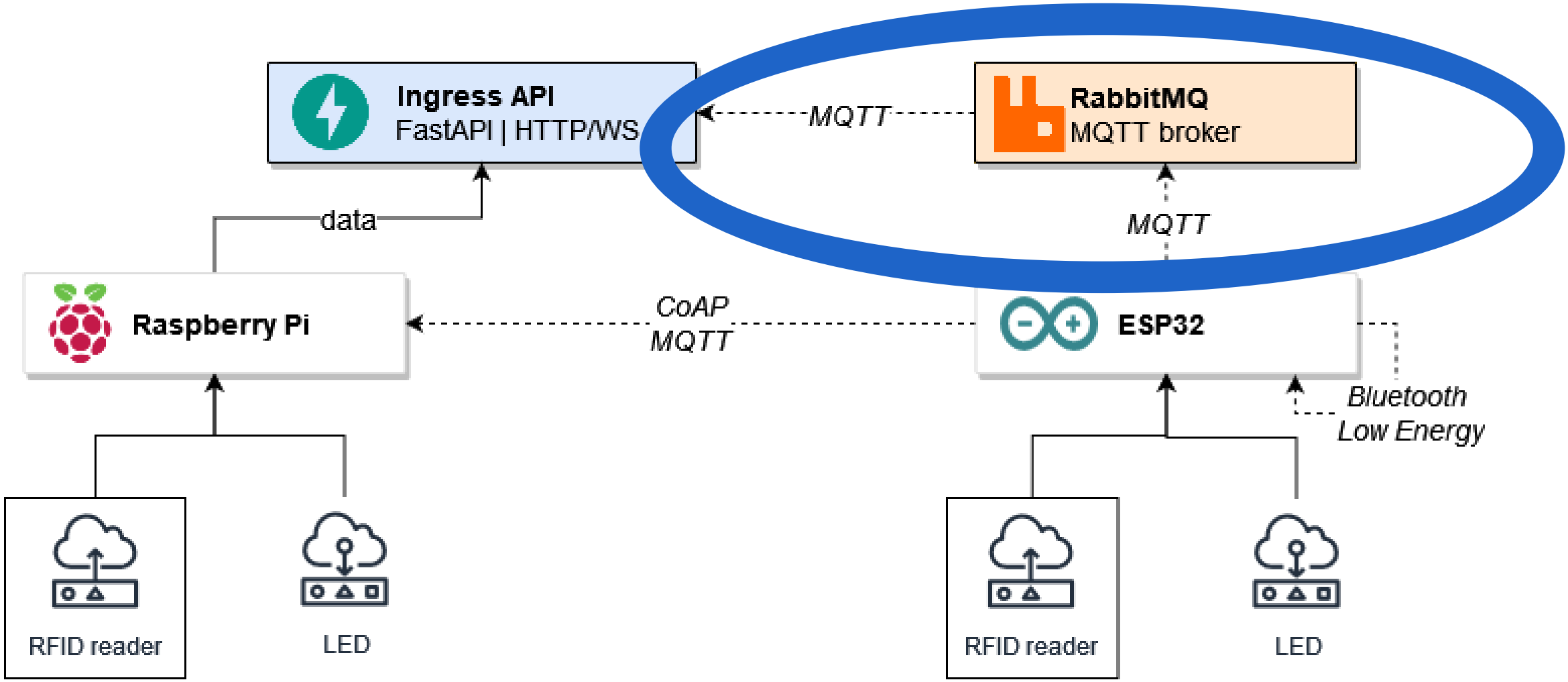
Publish

Subscribe

Bonus Task8: Control a(n) LED via MQTT



In Lab: Task 9 - ESP32 Talk to Ingress in Cloud



Material to submit

- Preparation part at home: due **Thursday 13 March at 10:00**
  - Checklist on Ufora
- Archive (Lab4_FamilyName_FirstName.zip): **due Thursday 27 March at 10:00**
  - Lab report in .pdf
    - Explanation of code
    - Screenshots and/or pictures of task results
    - Questions
  - Source code
    - no *.idea* or *__pycache__* folders!
  - Videos of task results (if applicable)

FACULTY OF ENGINEERING AND ARCHITECTURE

Ing. Cedric Bruylandt
PhD student

Cedric.Bruylandt@UGent.be

Dr. Jennifer B. Sartor
Onderwijsbegeleider

Jennifer.Sartor@ugent.be

GHENT UNIVERSITY