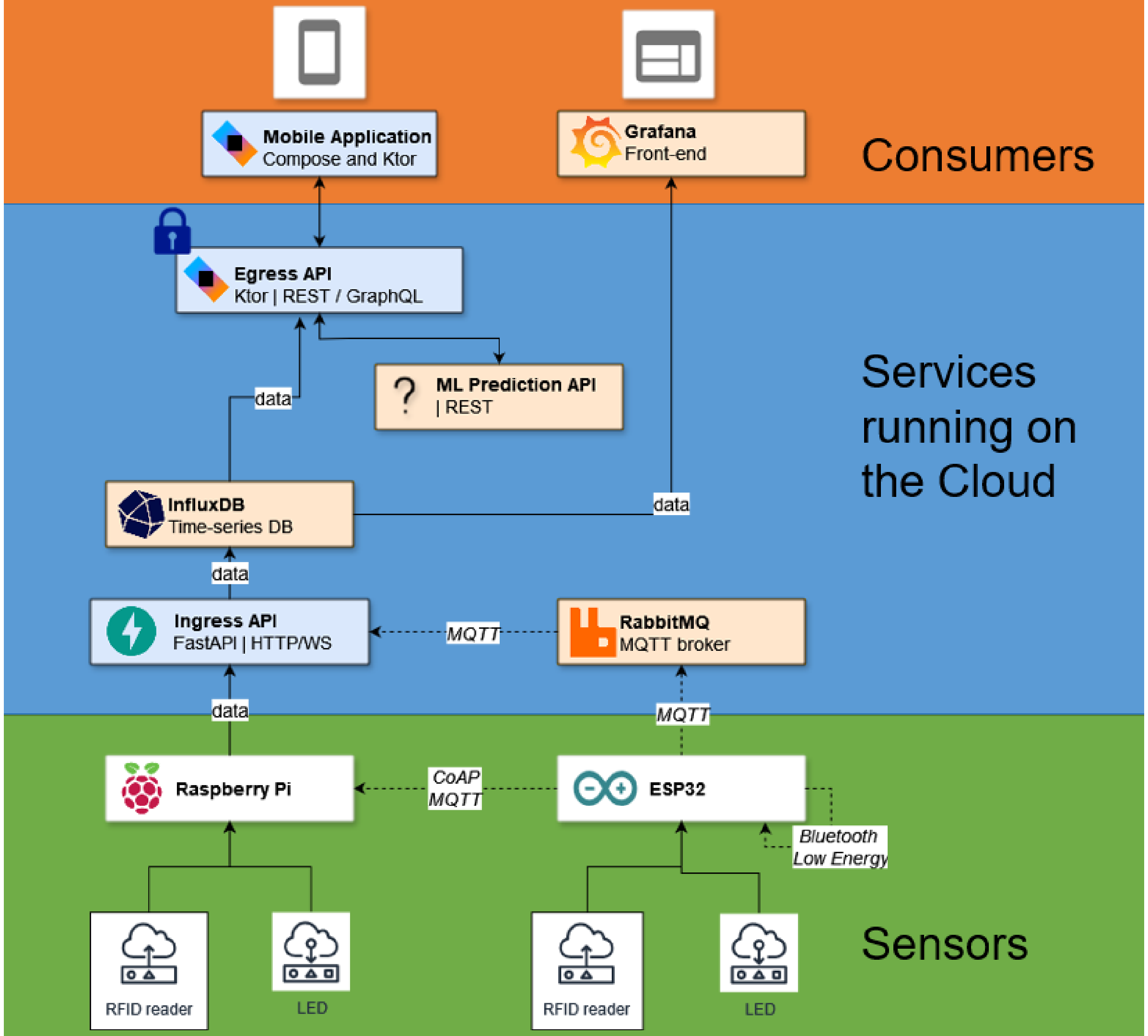
FACULTY OF ENGINEERING
AND ARCHITECTURE

LAB3 - Microservices

Cloud and mobile applications

GHENT
UNIVERSITY



Consumers

Mobile Application
Compose and Ktor

Grafana
Front-end

Services running on the Cloud

Egress API
Ktor | REST / GraphQL

ML Prediction API
| REST

InfluxDB
Time-series DB

data

data

data

Ingress API
FastAPI | HTTP/WS

MQTT

RabbitMQ
MQTT broker

data

MQTT

Raspberry Pi

CoAP
MQTT

ESP32

Bluetooth
Low Energy

RFID reader

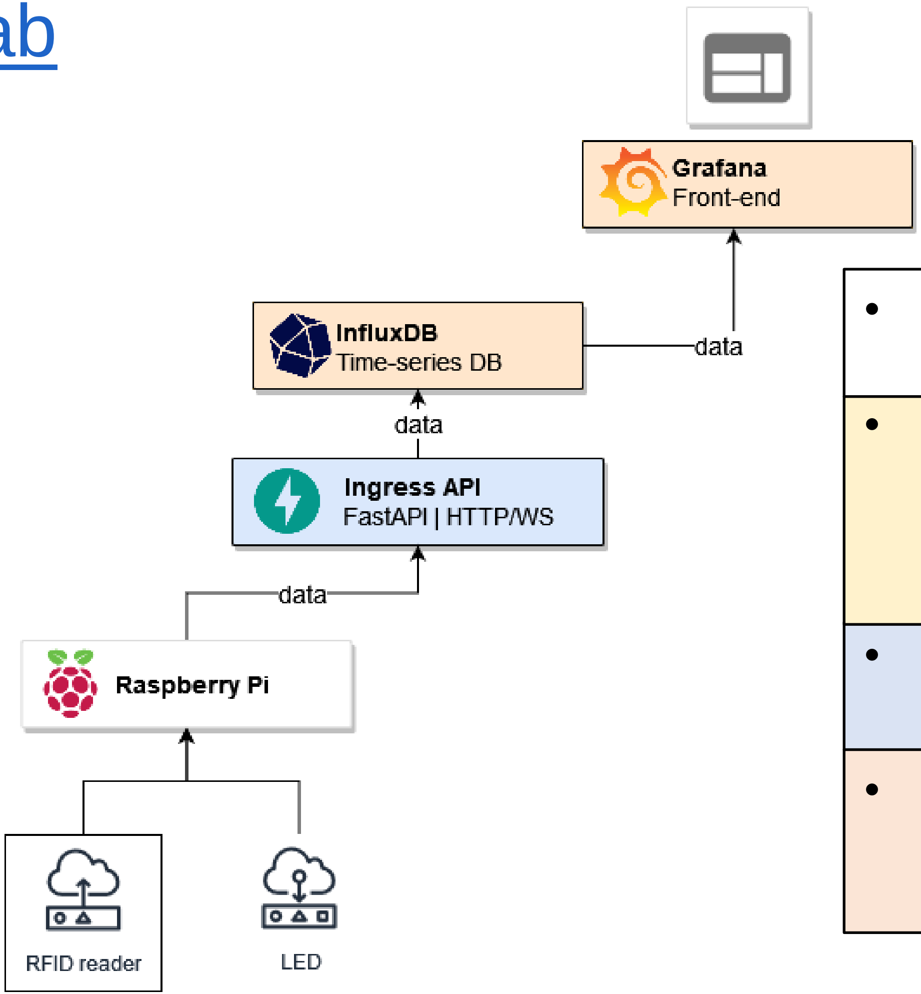
LED

RFID reader

LED

Sensors

This Lab



- Outlined boxes are microservices
- Colored boxes are hosted on Kubernetes cluster in the cloud
- Blue boxes you implement
- Orange boxes are shared services in the cloud

Goals

1. Implement a REST API using FastAPI and InfluxDB
2. Hands-on experience with the microservice architecture concept using Docker and Kubernetes in the cloud
3. Persistence and visualization of time-series (sensor) data with InfluxDB and Grafana

@Home: Download and Install

- Docker Desktop
- Kubectl
- Helm

@Home: Preparation Section

- Access the Kubernetes cluster
  - Use your *personal* config file
  - Connect to UGent VPN
  - Execute kubectl command
- Access Grafana dashboard

In Lab: Tutorials

- Interacting with the cloud environment
  - Creating a basic microservice
  - Virtualization using Docker
  - Deploying resources using Helm
  - Debugging your code
- REST API powered by FastAPI
- Ingress API & InfluxDB

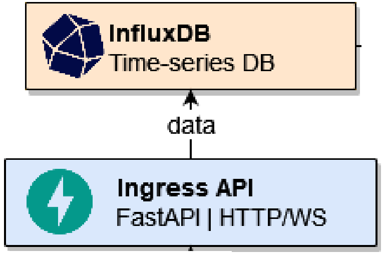
GHENT
UNIVERSITY

Task 1 – Extending the Ingress API

Components

- FastAPI
- InfluxDB

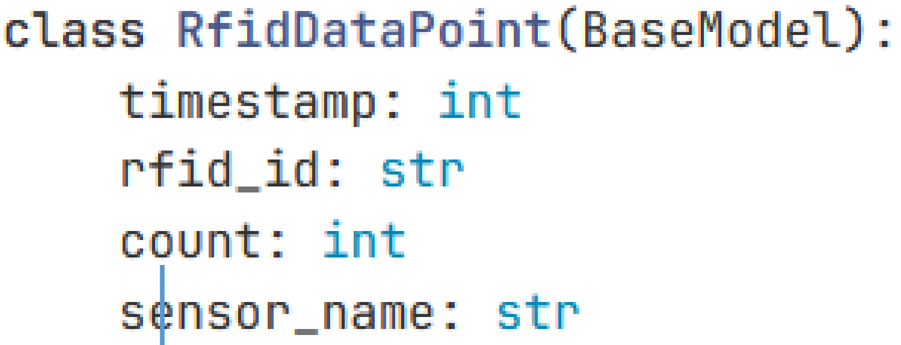
Extend Ingress API to post data to InfluxDB, and then query database for data.



Use Flux query language

Task 1 - Overview

- **/data/ -** POST - send new sensor measurement to the database
- **/data/ -** GET – query the database for all data within a given timespan (last hour)
- **/count/ -** GET – query the database for its latest count



```
class RfidDataPoint(BaseModel):
    timestamp: int
    rfid_id: str
    count: int
    sensor_name: str
```

- Count of active people/RFID tags
- Stored locally on RPi

InfluxDB Data Model

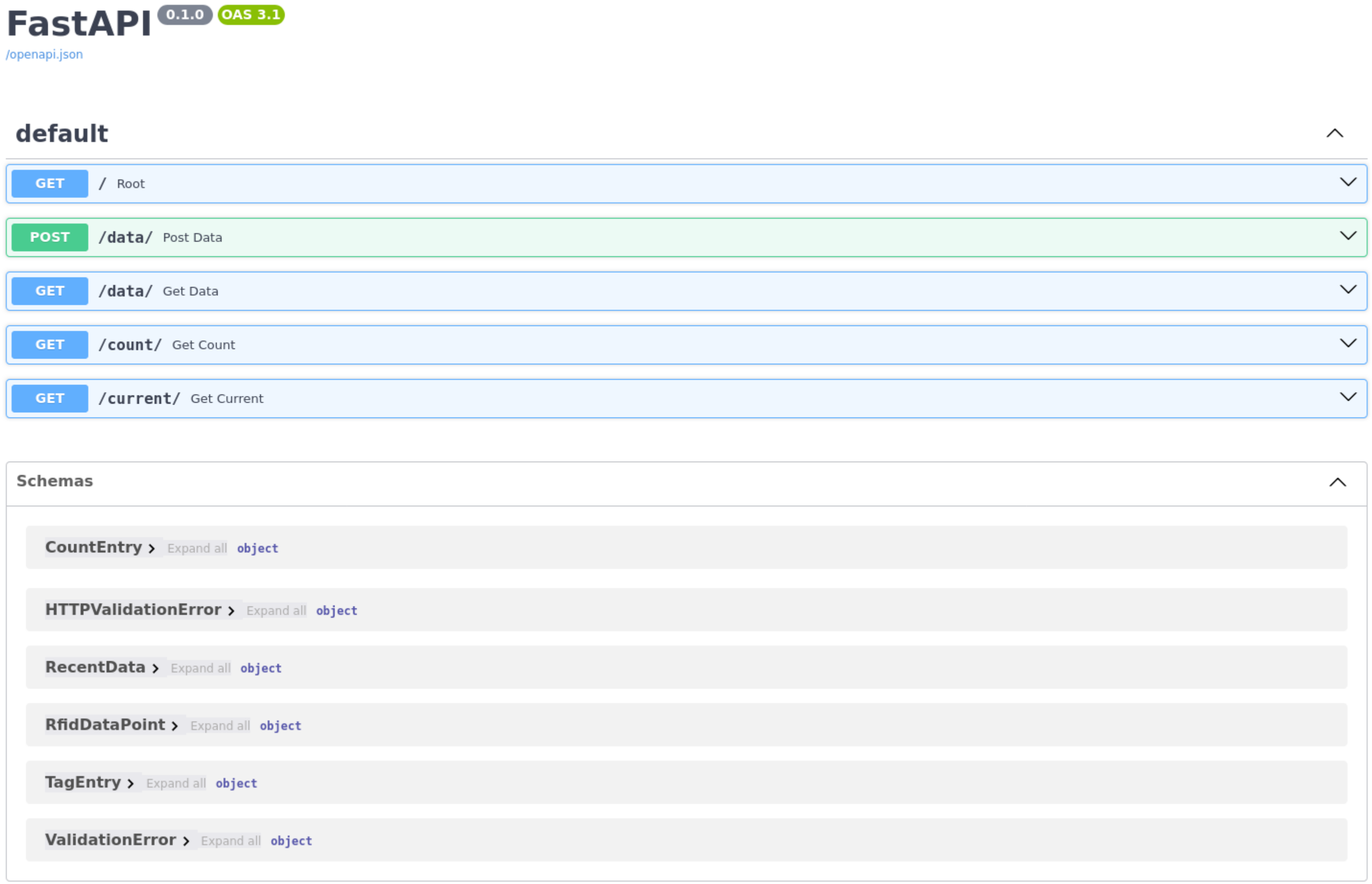
- measurement: people
  - tags
    - source: sensor_name
  - fields
    - value: count
  - time

****InfluxDB has nanosecond precision by default!**

- measurement: raw_ids
  - tags
    - source: sensor_name
  - fields
    - value: rfid_id
  - time

GHENT UNIVERSITY

Task 1 - Result



FastAPI 0.1.0 OAS 3.1

/openapi.json

default

GET / Root

POST /data/ Post Data

GET /data/ Get Data

GET /count/ Get Count

GET /current/ Get Current

Schemas

CountEntry › Expand all object

HTTPValidationError › Expand all object

RecentData › Expand all object

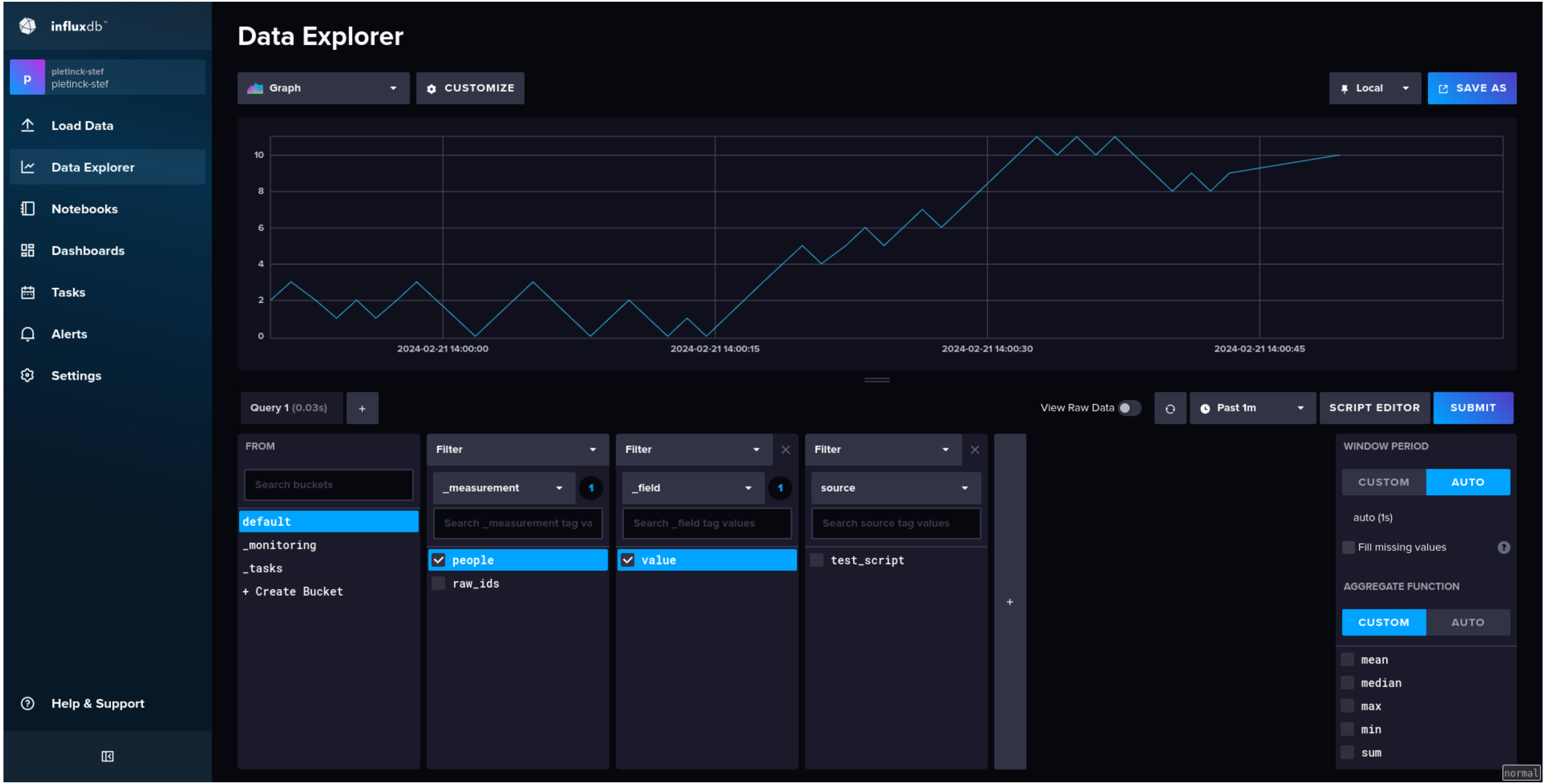
RfidDataPoint › Expand all object

TagEntry › Expand all object

ValidationError › Expand all object

normal

Task 1 - Result



FROM: default ☑ (_monitoring, _tasks)

_measurement 1: ☑ people □ raw_ids

_field 1: ☑ value

source: □ test_script

WINDOW PERIOD: AUTO, auto (1s)

□ Fill missing values

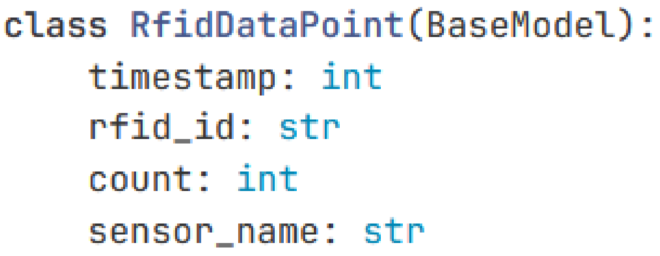
AGGREGATE FUNCTION: CUSTOM

□ mean
□ median
□ max
□ min
□ sum

Task 2 – Collecting sensor data

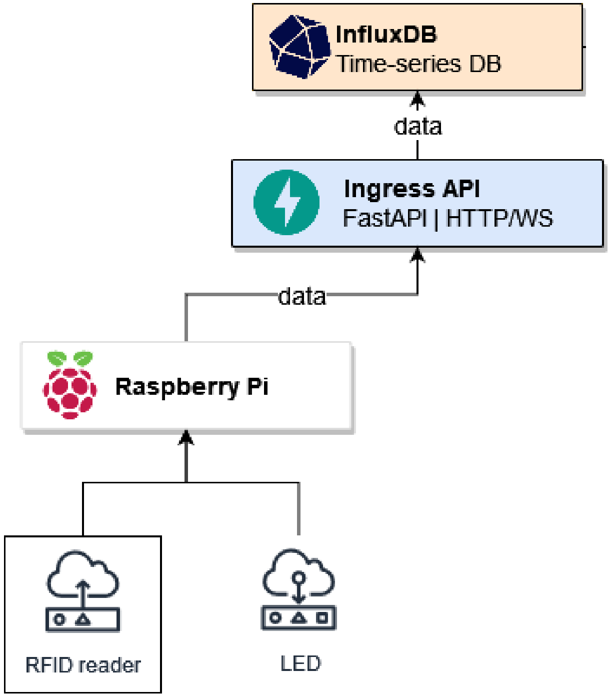
Components

- FastAPI
- InfluxDB
- RPi

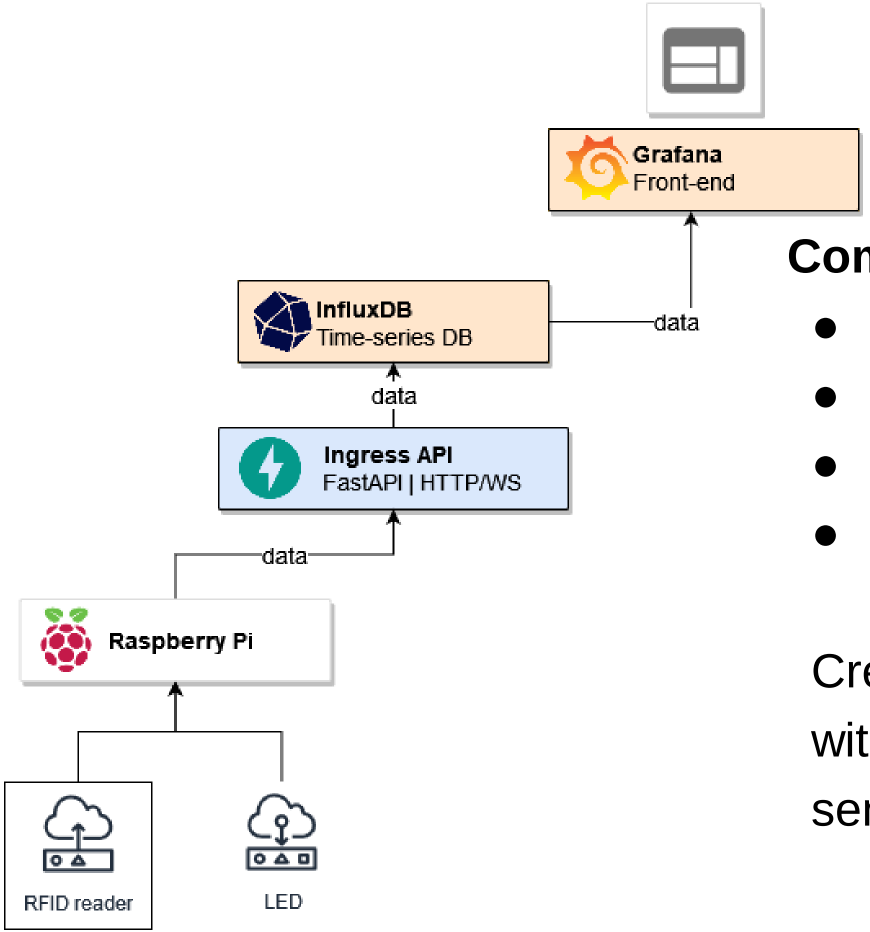


```python
class RfidDataPoint(BaseModel):
    timestamp: int
    rfid_id: str
    count: int
    sensor_name: str
```

Python app that reads sensor values and pushes them to Ingress API.



Task 3 – Monitoring sensor data

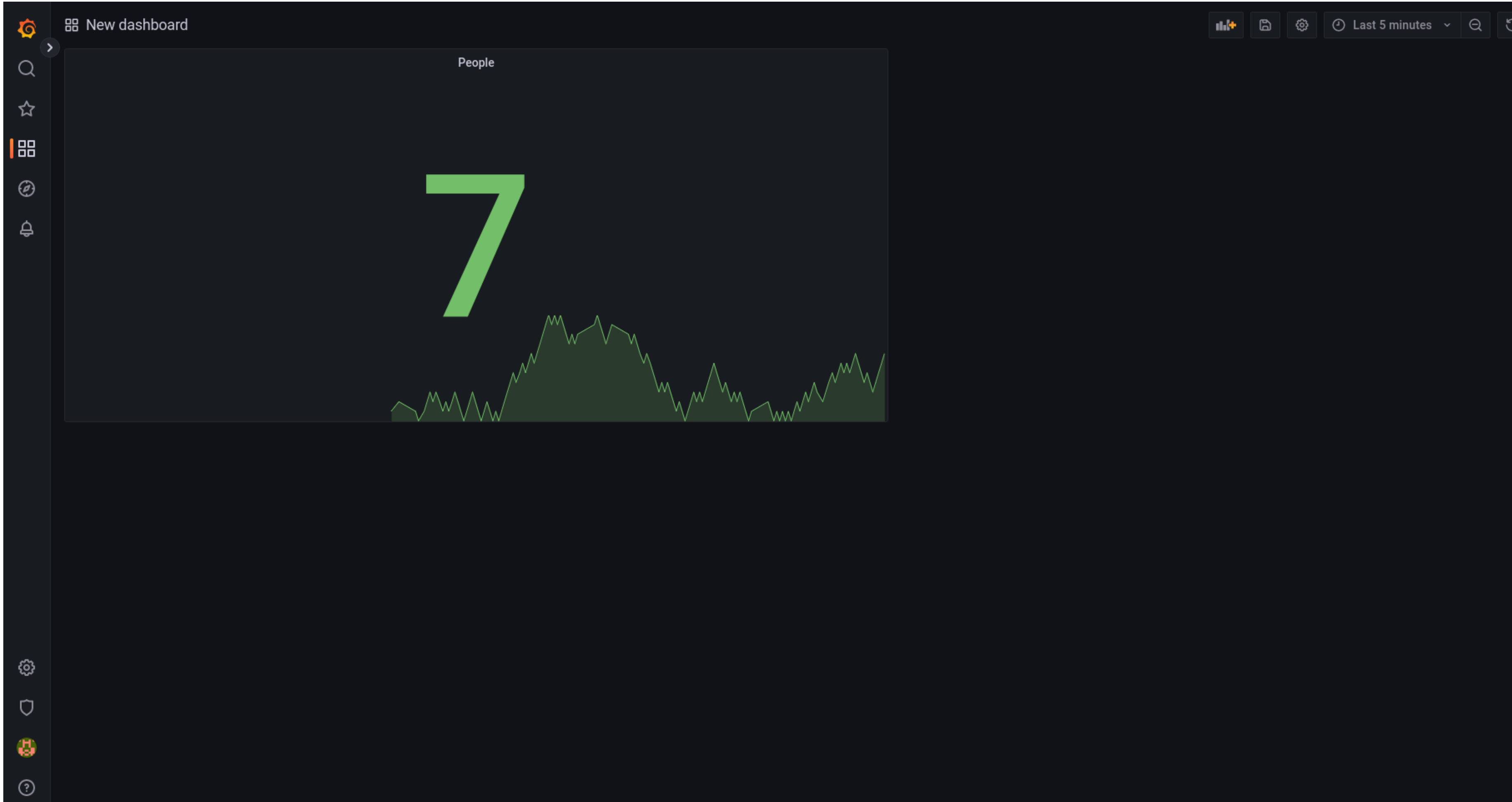


Components

- Ingress API
- InfluxDB
- Grafana
- RPi

Create Grafana dashboard with InfluxDB to visualize sensor data

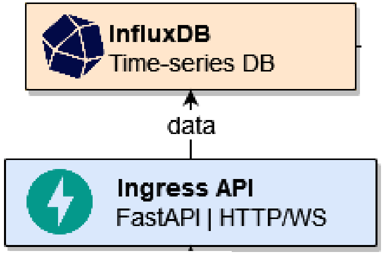
Task 3 - Results



Optional Task 4: Query for active RFID IDs

Components

- FastAPI
- InfluxDB



Extend Ingress API with endpoint:

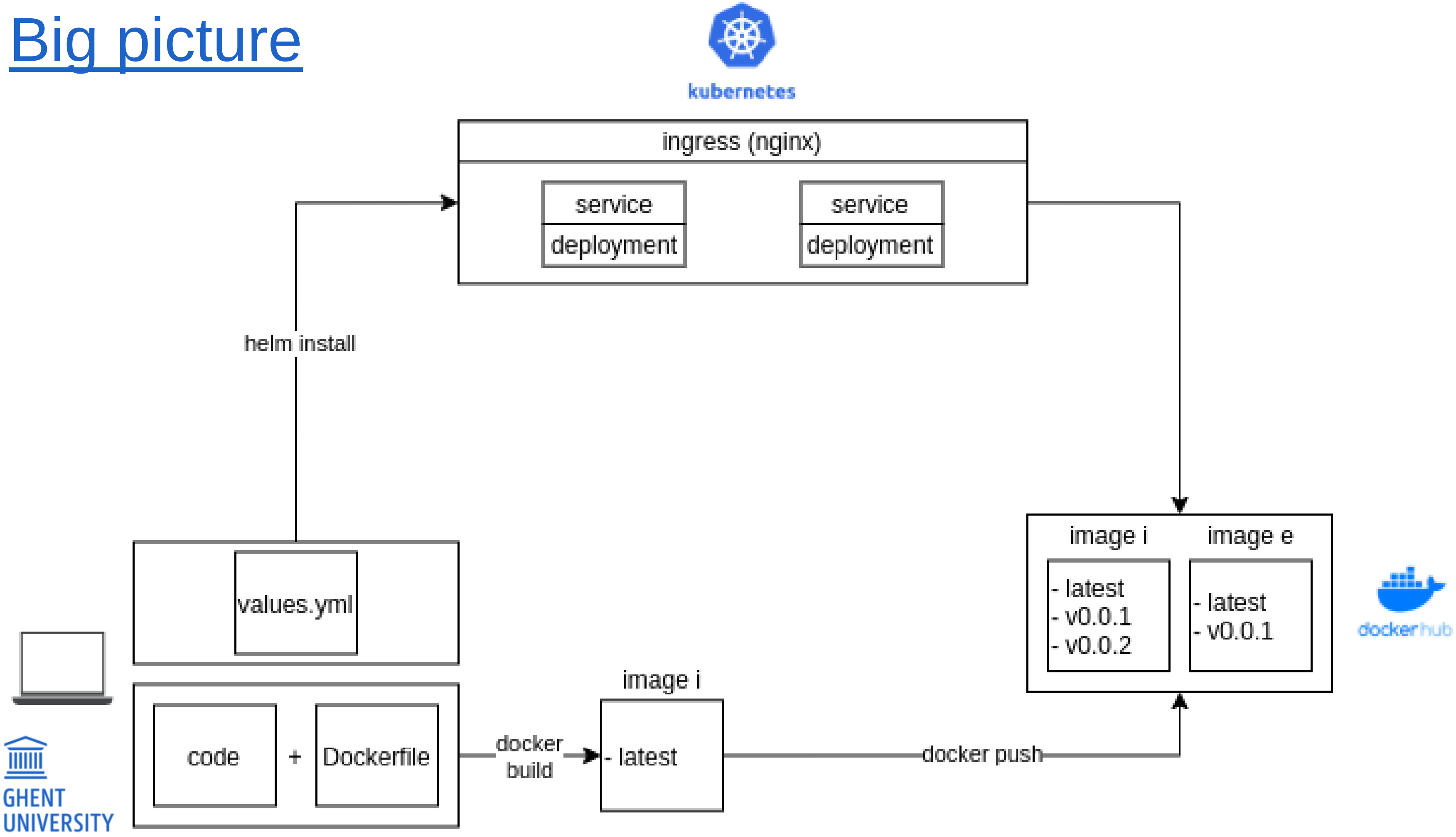
/current/ - GET – query the database for the active rfid_ids, or those scanned an odd number of times. Return list of strings.

Note: logic to find active rfid_ids should be in Flux query, not Python code

Material to submit

- Preparation part at home: due **next Thursday 27 February at 10:00**
    - Checklist on Ufora
- Archive (Lab3_FamilyName_FirstName.zip): **due 13 March at 10:00**
    - Lab report in .pdf
        - Problems with preparation or deploying Lab2 code to the cloud, screenshot of result
        - Explanation of code that you wrote for tasks
        - Screenshots of task results
        - Questions
    - Source code
        - no *.idea* or __*pycache*__ folders!
    - Videos of task results (if necessary)

Big picture


kubernetes
ingress (nginx)
service
deployment
service
deployment
helm install
values.yml
code
+
Dockerfile
docker build
image i
- latest
docker push
image i
- latest
- v0.0.1
- v0.0.2
image e
- latest
- v0.0.1
docker hub
GHENT UNIVERSITY

FACULTY OF ENGINEERING
AND ARCHITECTURE

Ing. Stef Pletinck
Developer/Engineer

Stef.Pletinck@UGent.be

Dr. Jennifer B. Sartor
Onderwijsbegeleider

Jennifer.Sartor@ugent.be

GHENT
UNIVERSITY