

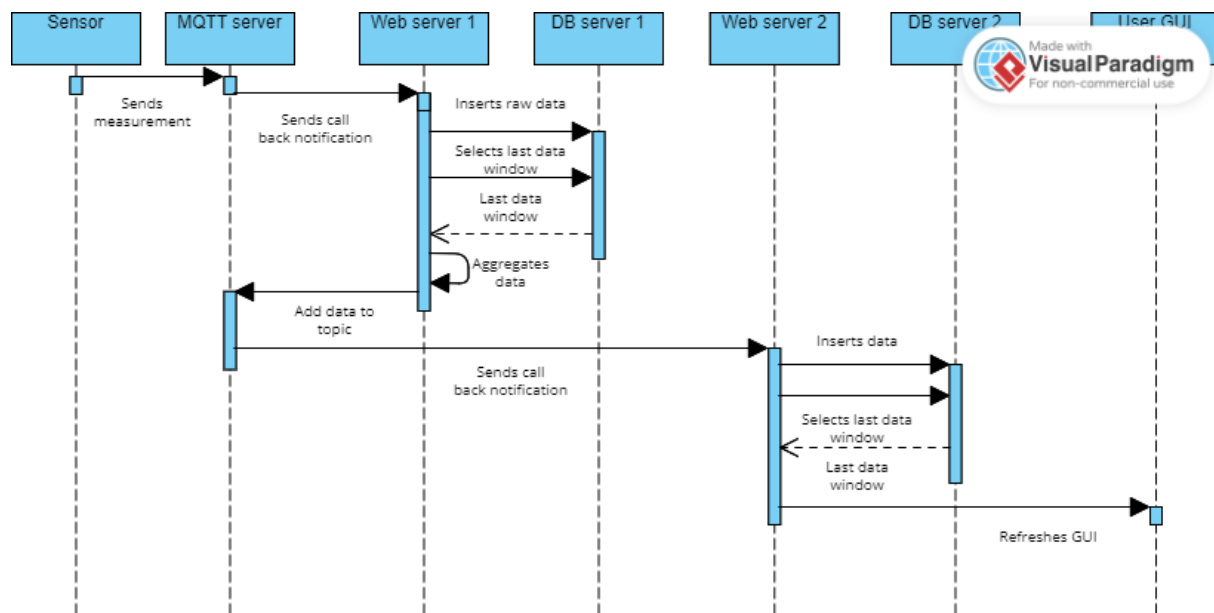
## Assignment 2

### The context

This assignment is a mini project promoting skills in IoT software development, debugging and self-working.

### 1. The need

Develop a multi-server system that enables logging of sensor data into a DB, aggregating it and present to the user. Thus a fog architecture is simulated for a better understanding of its concepts. An indicative UML sequence model of the potential solution is presented in the following figure:



It presents the following parties:

- **Sensor** – Students self-developed sensors or use an existing sensor, which produces data for test purposes.
- **MQTT server**  
IP: 158.38.67.229  
WebSocket port: 9001  
tcp, port: 1883  
User name: ntnu  
password: ntnuais2103
- **Web Server 1** – subscribes and receives data submission notifications from MQTT, logs data and does some aggregation. After aggregation, it sends a message to MQTT for other subscribers.
- **DB server 1** – stores raw and aggregated data into separate tables

- **Web Server 2** – subscribes and receives notifications from MQTT on aggregated data sent. Provides GUI functionality for user interaction.
- **DB server 2** – stores aggregated data for presentation purposes
- **GUI** – implements GUI functionality

## 2. The task

Using the code examples and visualization tutorial, implement the following:

- 1) Two different Postgres DB servers using Docker images. One for the fog server needs another for the presentation needs;
- 2) Two different web servers (Python + Flask) – fog server that responds to sensor data received and GUI server for serving user interface needs;
- 3) At least two different aggregation functionalities – one that implements the 2<sup>nd</sup> or 3<sup>rd</sup> order digital filter and the other of your choice.
- 4) Implement GUI with designed presentation and interaction elements, including charts, graphs or other means. Use Jinja or another template processor + free templates to make your design look nice. In case of using pre-made graphical or dashboard interfaces, they should connect directly to the DB server 2.
- 5) Deliver project documentation, which is composed of the following items:
  - a. Video presentation of the working system and code (walkthrough 5 min. max).
  - b. A report – reflection statement (PDF or Word).

## 3. The assessment method

Evaluation approach (200 pts at max.)

- Two DB servers are running and used = 30 pts each (60 max)
- Two web servers are running and used = 30 pts each (60 max)
- The digital filter implemented and operationally used = 10 pts each (20 max)
- Template processor or pre-made graphical or dashboard interface used = 30 pts (30 max)
- Interactive scenario implemented and working with nice looking design = 30 pts. (15pts without a design)

## 4. Tools

- PyCharm or VSCode for Python programming
- Docker for running DB images
- Internet connection to MQTT and DBs
- The example project in PyCharm
- Example tutorial on visualization