**Directions to Final Workshop**

The final workshop is a key activity in the GSL, where it is the opportunity for the groups to show the technical result of the cooperation. It follows the basic structure of an Academic Workshop and has two different moments. The first one is the presentation, where the students will have until 20 minutes to present

Given that some groups do not solve the Mininet integration, Figure 1 shows the minimum deliverable for this project.

Diagram, timeline

Description automatically generated

**Android App**

The first step is the Android app. This client will send every 10 seconds a publish message to the topic named **rescue/severity** using broker1, located at IP2 and port 1883. This message follows this pattern: ***id:severity:position***, where:

* ID: is the node id.
* Severity: Low||Medium||High.
* Position: LAT-LONG.

An example is “*node-1:Low:-23.22488,-45.232”*.

If the user changes the severity, the publish message will update its content.

**Observation 1:** If the students do not have an Android device, the solution can be run using an iPhone.

More information can be checked at: <https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/android/ANDROID.md>.

**Bridge Broker Server**

This server uses a virtual machine (VMWare or VirtualBox) or a physical machine. It must use a bridge-mode NIC emulation if it runs in a Virtual Machine. This machine runs a Mosquitto Broker instance in bridge mode. The configuration of the broker needs to follow the directions and examples defined in these materials:

* <https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/mosquitto/Unit%206.5%20MQTT%20Advanced%20(WITH%20ERRATA).pdf>
* <https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/mosquitto/mosquitto_bridge.conf>

Remember, JavaScript uses websockets; consequently, you must define the broker using the websockets protocol. Set the connection name as a **drone**.

**Central Broker Server**

This server runs in a cloud (Azure, AWS, or another). The creation and configuration of an Ubuntu server in the Azure environment are presented in **MQTT Foundations.pdf**. This server will run multiple services. The first one is Mosquitto as a central Broker. An example of a configuration file is **mosquitto-central.conf**.

* <https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/mosquitto/MQTT_Foundations.pdf>
* <https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/mosquitto/mosquitto_central.conf>

More information can be checked at: <https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/mosquitto/MOSQUITTO.md>.

The second service is a python MQTT client subscribing to topic **rescues/severity**. An example of this python file is [mqtt\_subscriber.py](https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/cloud/mqtt_subscriber.py) **and** [mysql-utils.py](https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/cloud/mysql-utils.py). The challenge is to store the messages received from the topic in a SQL database. The provided file has a complete code to subscribe to the topic (remember to set the parameters), where you need to implement the SQL part. Also, previously, you needed to create the database and the tables.

* <https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/cloud/mqtt_subscriber.py>
* <https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/cloud/mysql-utils.py>

More information can be checked at: <https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/cloud/MYSQL.md>.

Finally, it would be best if you created the dashboard using Grafana or Freeboard. The approach is the same one used in the class. The minimum dashboard to implement are:

* A dynamic map where you identify the risk situation colors the node positions.
* The severity situation colors a dynamic map where you identify the node positions and then.
* A bar chart shows the number of nodes classified by risk.
* A bar chart shows the number of nodes classified by severity.

More information can be checked at: <https://github.com/kabartsjc/gsl-iot-2022/blob/main/templates/cloud/DASHBOARD.md>.