

Team 25 - Homework 3

Fantasia Claudio
Politecnico di Torino
Student id: s319911

Lanzillotti Donato
Politecnico di Torino
Student id: s316001

Ngo Triet Minh
Politecnico di Torino
Student id: s309062

I. DATA COLLECTION, COMMUNICATION AND STORAGE

MQTT is a Publish/Subscribe communication protocol that is more suitable in the application considered for the following reasons:

- 1) **OVERHEAD:** MQTT is a lightweight protocol with reduced overhead compared to REST. In scenarios involving a large volume of transmitted messages, its usage can be beneficial. It helps to mitigate or even prevent congestion by transmitting less data over the network.
- 2) **STATEFULNESS:** One of the main properties of the REST communication protocol is its statelessness, implying that the interaction between the server and the client does not last longer than a single request. It would represent a negative aspect in applications with continuous monitoring, since it would mean re-establishing the connection a large number of times. On the other hand, MQTT being a stateful communication protocol maintains the connection without the need to establish it for each data update.
- 3) **ASYNCHRONOUS:** MQTT operates asynchronously, allowing the script to publish data without waiting for a response. In a continuous monitoring scenario like the application considered, where real-time data is crucial, the asynchronous nature of MQTT enables efficient communication without blocking the script's execution. REST instead, being synchronous by nature, might introduce delays in the script execution due to waiting for responses.
- 4) **SCALABILITY:** MQTT is designed for scalable and efficient communication. The publish/subscribe architecture allows for easy scaling, as new clients can subscribe to relevant topics without affecting the existing infrastructure. This scalability is particularly important when emulating a fleet of connected devices, as in the provided scenario.
- 5) **QUALITY OF SERVICE:** MQTT includes three QoS levels (0, 1, and 2), allowing devices to choose the appropriate degree of guarantee for message delivery. It can be a crucial aspect for devices where

optimizing power consumption is a priority, like PCs. The ability to tailor the Quality of Service level based on specific requirements provides flexibility without compromising reliability.

II. DATA MANAGEMENT AND VISUALIZATION

The following table shows the most HTTP suitable methods for each of the required action.

Method	Endpoint	Description
GET	/devices	Retrieve the list of MAC addresses of the monitored devices.
GET	/device	Retrieve battery status information of the device with the specified MAC address in the specified data range.
DELETE	/device	Delete the timeseries associated to the specified MAC address.

The first and the second choice presented in the table are motivated from the fact that the GET method is used for requesting a specified resource. In this case, it is the most appropriate method both for retrieving the list of the monitored devices and the battery status information of a device with a specified MAC address since it is not required to create a new resource or to update an existing one, scenarios in which the POST and PUT methods would be more suited.

The third choice presented in the table is motivated from the fact that the DELETE method deletes a specified resource. Since the aim is to delete the timeseries related to a specified MAC address it represents the most suitable HTTP method.