SAPIENZA
UNIVERSITÀ DI ROMA

PLASTER SAFE

THIRD DELIVERY

Gianluca Circelli Giovanni Fiordeponti

Corrections after comments

Why not make the system based on edge computing? → **We put all the computations inside the Gateway, the component which receives data from the statues**

Focus on the IoT Components instead of the fancy *WebApplication* → **We deployed our solution on real hardware, using an STM32 Board and a Raspberry Pi**

Changes made following comments

In the edge based system, the changes are as follows:

1	**Architecture**	$\longrightarrow$	The data read by the sensors are not sent directly to the cloud but are first **processed** and then **sent**. In this way, even if there is no line coverage during transport, the data is always processed and then queued for transmission as soon as the line becomes available again.
2	**Evaluation**	$\longrightarrow$	By processing the data directly on the boards the system is more efficient, in fact response time and bandwidth saving are improved. It also reduces battery consumption because sending data is less frequent.

Technical work

The main job was to adapt the previous **firmware** and **gateway**, used in the simulations, to work on real hardware. **STM32 Nucleo** has been used to deploy the firmware and **Raspberry Pi** has been used to host the gateway. FInally, a **Web Application** has been deployed to *GitHub Pages*

Technical work

One or more board should be attached to a statue, to measure temperature and acceleration variations. We developed a firmware using the *RIOT-OS* operating system.

There exists two versions of the **firmware:**

- ***serial**:* The connection between the board and the Raspberry takes place via serial. The board reads digital values of temperature and accelerations from the *LSM303DHLC* and prints them on the console as a formatted string.

- ***native:*** temperature and accelerations along the three-axis are randomly generated and sent through the *MQTT-SN Protocol.*

Technical work

The ***Gateway***, written in *Python*, receives measurements from all the statues connected and performs some computation to detect whether some values are beyond a given threshold. If so, a message is sent to *AWS* using the *MQTT Protocol*

It has three operating modes:

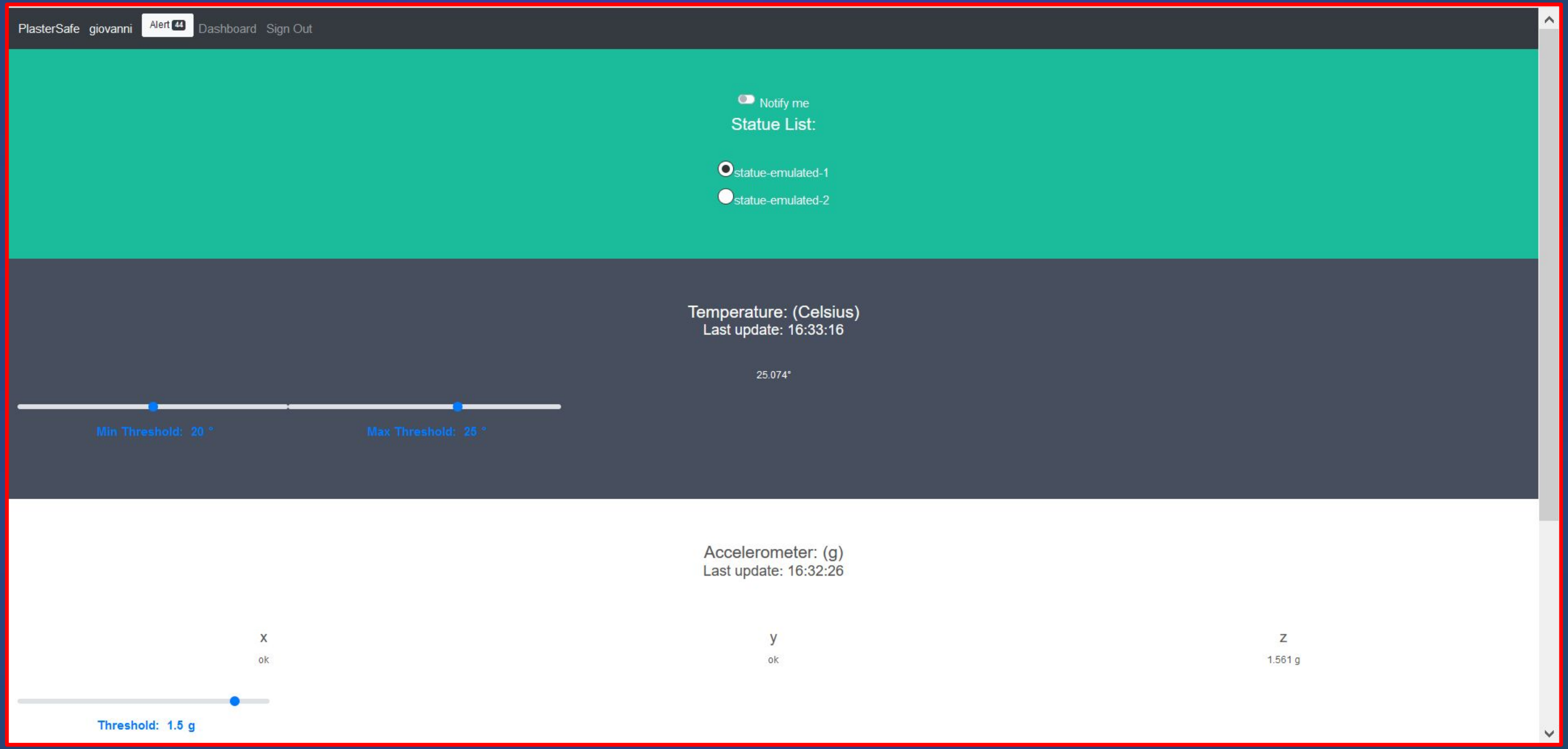
- ***serial:*** receives data from the STM32 via serial port;
- ***emulated:*** simulates all the data locally;
- ***local-mqttsn***: reads data from a local *MQTT-SN Broker*.

For each statue:

- it stores a set of samples to compute a statistical measure of the acceleration magnitude using the *Signal Magnitude Area (SMA)*,
- it tells whether acceleration, temperature or SMA are beyond a given threshold, that can be changed by the users of the system,
- check if every statue is communicating properly. If not, an alert is raised

Technical work

The ***WebApplication*** gives to archivists a tool to see data in real time coming from the statues and view an history of all movements by a dashboard.



Notify me

Statue List:

(●) statue-emulated-1
( ) statue-emulated-2

Temperature: (Celsius)
Last update: 16:33:16

25.074°

Min Threshold: 20 °
Max Threshold: 25 °

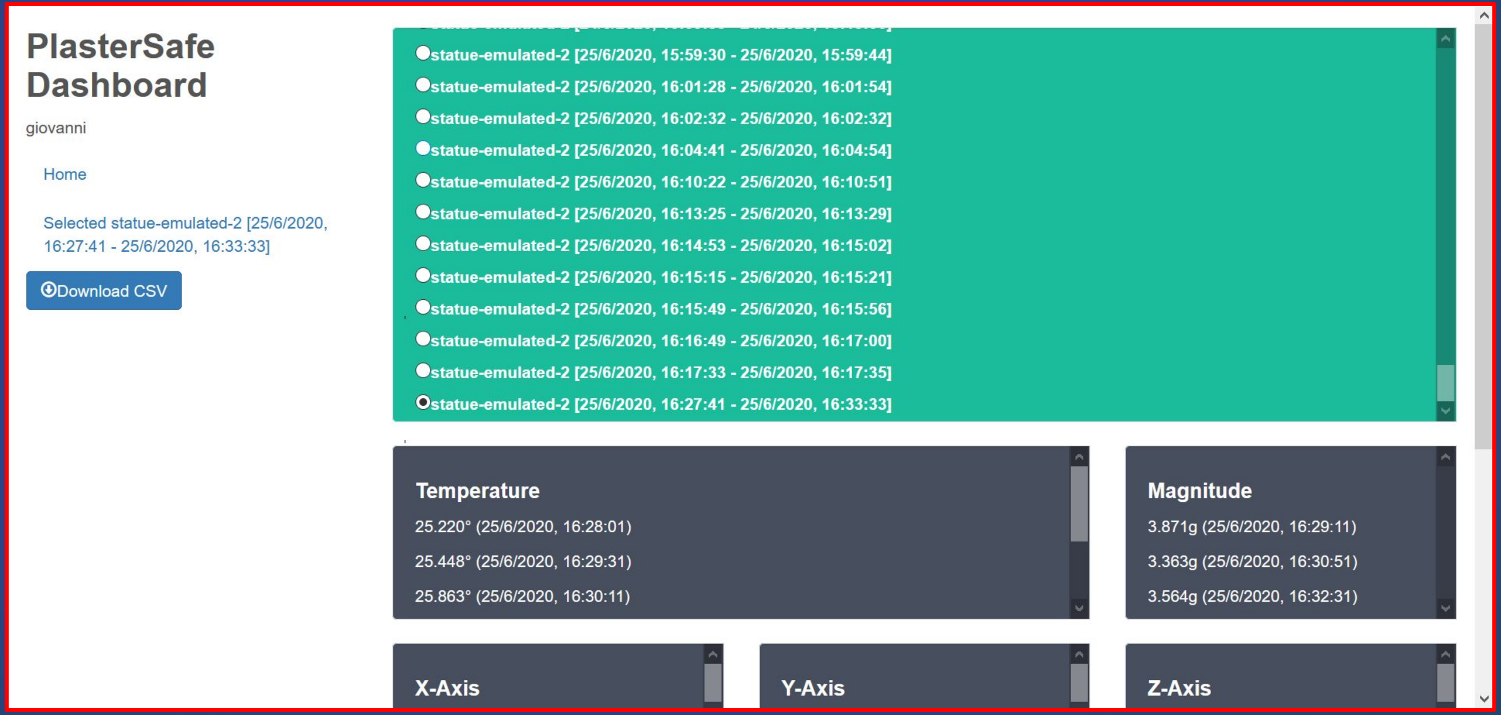
Accelerometer: (g)
Last update: 16:32:26

x	y	z
ok	ok	1.561 g

Threshold: 1.5 g

Technical work

The ***WebApplication*** gives to archivists a tool to see data in real time coming from the statues and view an history of all movements by a dashboard.



PlasterSafe Dashboard

giovanni

Home

Selected statue-emulated-2 [25/6/2020, 16:27:41 - 25/6/2020, 16:33:33]

Download CSV

○ statue-emulated-2 [25/6/2020, 15:59:30 - 25/6/2020, 15:59:44]
○ statue-emulated-2 [25/6/2020, 16:01:28 - 25/6/2020, 16:01:54]
○ statue-emulated-2 [25/6/2020, 16:02:32 - 25/6/2020, 16:02:32]
○ statue-emulated-2 [25/6/2020, 16:04:41 - 25/6/2020, 16:04:54]
○ statue-emulated-2 [25/6/2020, 16:10:22 - 25/6/2020, 16:10:51]
○ statue-emulated-2 [25/6/2020, 16:13:25 - 25/6/2020, 16:13:29]
○ statue-emulated-2 [25/6/2020, 16:14:53 - 25/6/2020, 16:15:02]
○ statue-emulated-2 [25/6/2020, 16:15:15 - 25/6/2020, 16:15:21]
○ statue-emulated-2 [25/6/2020, 16:15:49 - 25/6/2020, 16:15:56]
○ statue-emulated-2 [25/6/2020, 16:16:49 - 25/6/2020, 16:17:00]
○ statue-emulated-2 [25/6/2020, 16:17:33 - 25/6/2020, 16:17:35]
● statue-emulated-2 [25/6/2020, 16:27:41 - 25/6/2020, 16:33:33]

Temperature

25.220° (25/6/2020, 16:28:01)
25.448° (25/6/2020, 16:29:31)
25.863° (25/6/2020, 16:30:11)

Magnitude

3.871g (25/6/2020, 16:29:11)
3.363g (25/6/2020, 16:30:51)
3.564g (25/6/2020, 16:32:31)

X-Axis

Y-Axis

Z-Axis

Technical work

The ***WebApplication*** gives to archivists a tool to see data in real time coming from the statues and view an history of all movements by a dashboard.

Main Functionalities:

- ***Authentication*** using *Cognito User Pool*
- **Notification API** to inform the user whether some alert has been received by the gateway
- ***MQTT WebSocket*** to receive data and to send threshold variations for each statue
- Dashboard that receives all movements from the statue from ***API Gateway REST Endpoint***
- Download a given session in *CSV Format*.

Evaluation

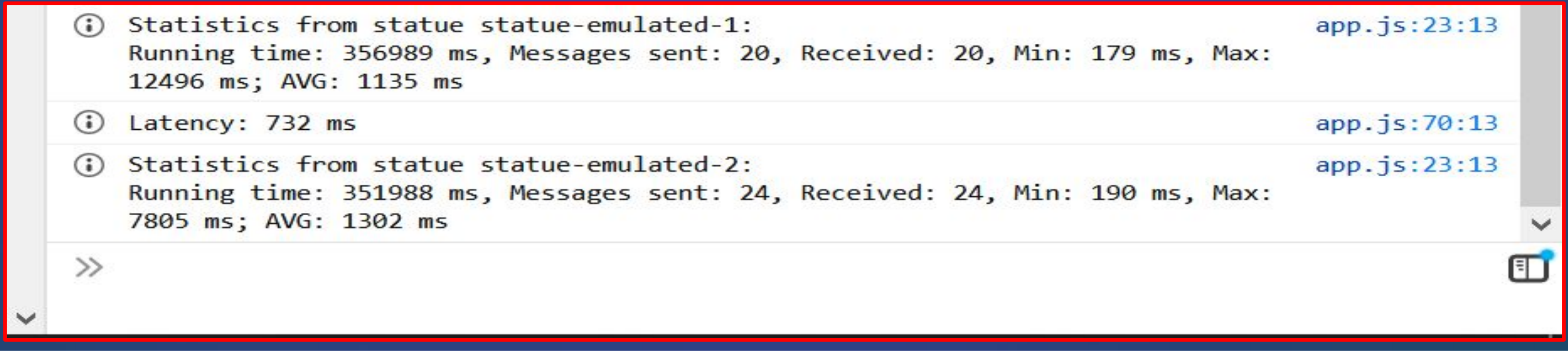
- Assess message latency
- Costs
- Museum Feedback

Evaluation

The ***latency of the system*** was calculated using timestamps linked to each message sent, making a difference with the timestamp of its reception.

The system performs well when the network connectivity of the gateway is stable:

- We achieved during a *five-minute session* an average of *1-second latency* for each statue involved.
- The connection used was deliberately slow, with 6.15 Mbps download rate and 1.25 Mbps upload rate



```
Statistics from statue statue-emulated-1:                                    app.js:23:13
Running time: 356989 ms, Messages sent: 20, Received: 20, Min: 179 ms, Max:
12496 ms; AVG: 1135 ms
Latency: 732 ms                                                              app.js:70:13
Statistics from statue statue-emulated-2:                                    app.js:23:13
Running time: 351988 ms, Messages sent: 24, Received: 24, Min: 190 ms, Max:
7805 ms; AVG: 1302 ms
>>
```

Evaluation

The cost is divided into ***Hardware*** and subscriptions to **AWS Services**:

Hardware:

- *Board STM32 Nucleo* considering 4 boards per statue	88€
- *Radio antennas (NRF24L01)* one for each board plus the gateway	45€
- *Sensors (LSM303DHLC)* one for each board	34€
- *Raspberry Pi 3* acts as a gateway	45€
Subtotal	**212€**

Aws Services:

- *Aws IoT Core* full service for million calls of: - device connectivity - messages exchanged - publications - activation of rules	9.65€
- *DynamoDB* for million writings and readings	1.90€
- *Cognito* up to 100,000 users	0.49€
- *API Gateway* for a million REST calls	3.29€
Subtotal	**24.98€**

The final total of the cost for millions of calls, for 100,000 users is **237€**

Evaluation

As final feedback we decided to send a demo of our work to the museum team, so as to have a direct opinion on the features implemented and on the ease of use.

We will update this section as soon as we’ll receive a response.

Functionality and evaluate missing

- Connect multiple boards, equipped with antennas, to the gateway via Mqtt-sn protocol;
- We were unable to calculate actual battery consumption and estimate a minimum amperage for long-lasting transportation.
- Perform more sophisticated analysis on the acceleration, such as detecting peaks through frequency domain analysis.