MONITORING OF THE CAMPUS IOT PLATFORM

Final Report - Student Project

INFO4 – Group 18

BLANQUET Antoine

LAMBERT Paul

YUNG Kevin

Tutor

DONSEZ Didier

Overview

During our fourth year of study at Polytech Grenoble, as INFO students, we managed to complete a group project. Supervised by a tutor, we carry out all the steps of a project.

This report gathers all the work done during this project and can be used as a starting point to continue this project.

All the documentation and code are available on the gitlab below :
https://gricad-gitlab.univ-grenoble-alpes.fr/Projets-INFO4/20-21/18/

INTRODUCTION

Context

The **CampusIoT** platform is a 'Learn-by-doing' platform for teaching Internet of Things technologies and businesses. With its private LoRaWAN network and university fablabs, CampusIoT accelerates the testing of Internet of Things application prototypes in the Grenoble and Valence metropolitan areas.

Objective

The objective of the INFO4 2020-2021 project is to develop tools for monitoring the CampusIoT platform.
These tools will be used :

- to alert administrators of object, gateway or server failures (and their return to normal).
- to alert the administrators of maintenance operations on the objects (change of object battery for example).
- produce regular (HTML) reports giving metrics on gateways and objects of an organization (including compact graphs like Sparkline).

Organisation

We used github/gitlab to manage our organization. Infact, we used issues in our project to give us guidelines with tasks. We created a Milestone for our project; and we put issues on it.

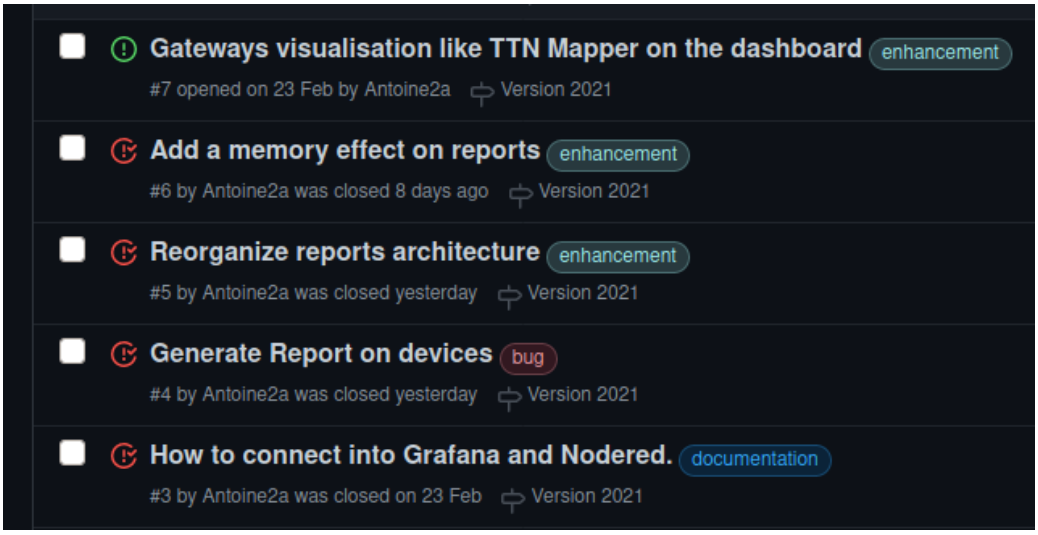


Figure 1 : Example of issues in our project

Each Issue is a specific task of our project.

Technologies

Scripts Bash (Curl JQ)

curl is used in command lines or scripts to transfer data.

jq is a filter that allows you to manage data on json files.

doc : https://gricad-gitlab.univ-grenoble-alpes.fr/Projets-INFO4/20-21/18/docs/-/blob/master/notice_curl_jq.md

CampusIoT Network Server's REST API

ChirpStack Application Server provides an API interface that can be used for building integration. Both interfaces provide *exactly* the same functionality.

link : https://lns.campusiot.imag.fr/api#/

doc : https://gricad-gitlab.univ-grenoble-alpes.fr/Projets-INFO4/20-21/18/docs/-/blob/master/notice_api_rest.md

Mail PostFix

Allows you to send emails via a shell script. Your pc is used as a local server. We use postfix for this.

doc : https://gricad-gitlab.univ-grenoble-alpes.fr/Projets-INFO4/20-21/18/docs/-/blob/master/notice_mail_postfix.md

Sparkline

Allows to create mini-graphs to compact the data.

link : https://omnipotent.net/jquery.sparkline/#s-about

NodeRED

NodeRED is a graphical programming language for connecting different web services, APIs or devices together. It provides a browser-based editor using a very wide range of nodes making it quite simple and very intuitive to use. We use the data we are interested in that goes through the Lora Gateway on NodeRed and make the transition to our InfluxDB database.

Grafana

Grafana is a practical and open source platform for monitoring, analyzing and visualizing system data in real time. The objective of this solution is to easily and intuitively present a large amount of data from different sources.

link : https://grafana.com/docs/grafana/latest/panels/queries/

https://blog.ruanbekker.com/cheatsheets/grafana/

InfluxDB

InfluxDB is a time series oriented database management system. It was designed to manage time-stamped data.

link : https://air.imag.fr/index.php/InfluxDB

cheatsheet : https://github.com/CampusIoT/tutorial/blob/master/influxdb/influxdb_cheatsheet.md

Docker

Docker is a software which allows you to test and deploy applications. Docker integrates software into containers that collect all dependencies.

link : https://air.imag.fr/index.php/Docker

APPLICATION

Architecture of Reports folder

At the beginning of the project, there was no architecture in the project. We made one, it looks like this :

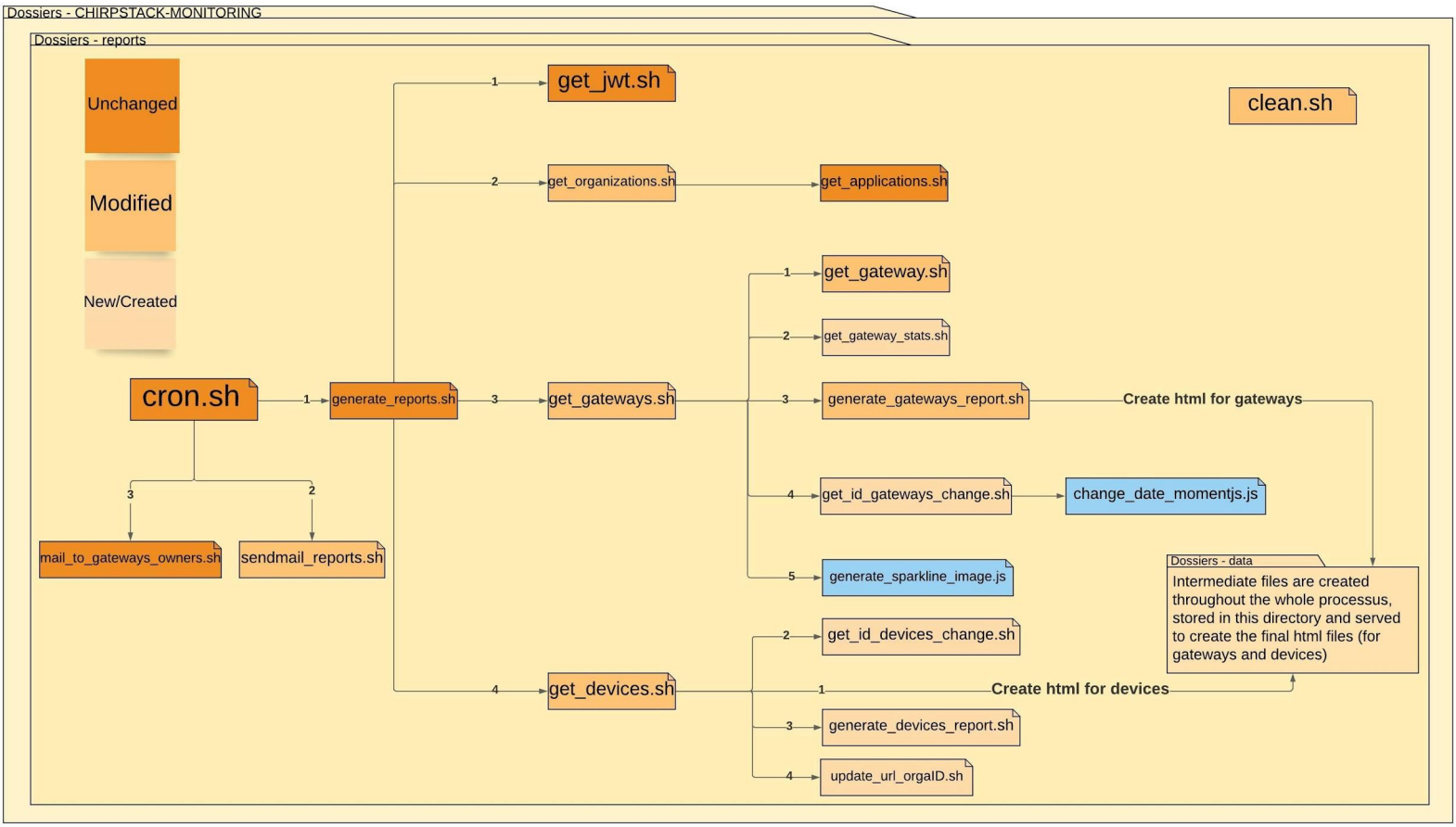


Figure 2 : Architecture of Reports folder

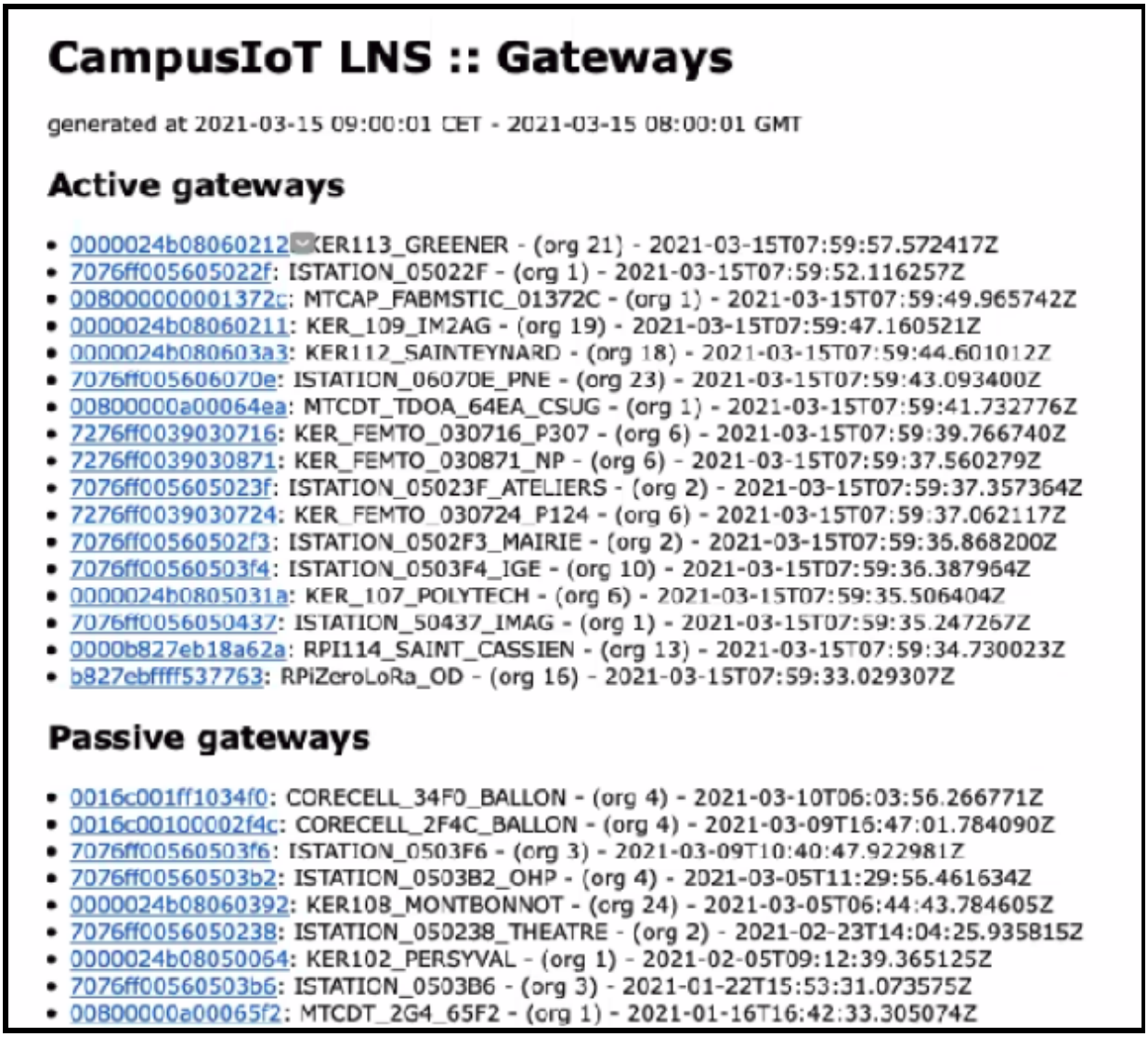
The reports folder already had a simple architecture when we started the project. *cron.sh* was calling *generate_reports.sh* , *sendmail_reports.sh* and *mail_to_gateways_owners.sh*. *generate_reports.sh* was only using *get_jwt.sh* to get an access token, *get_applications.sh*, *get_devices.sh* and *get_gateways.sh*. In order to do what we wanted and to still have a clear

view of our code, we decided to create new files. Now *get_gateways.sh* is using not only *get_gateway.sh*, but also *get_gateway_stats.sh* and *get_id_gateways_change.sh*. *get_devices.sh* uses *get_id_devices_changes.sh* and *change_url_orgaID.sh*. To maintain a clear code we decided to put every generated file into a data folder which is organized in 6 different folders (applications, configuration, devices, gateways, generated_files, images and organizations).

We've managed to keep a simple architecture and organized even more the project by dividing code files from generated files.

Development Reports

Here was an example of the HTML report at the beginning :



CampusIoT LNS :: Gateways

generated at 2021-03-15 09:00:01 CET - 2021-03-15 08:00:01 GMT

Active gateways

- 0000024b08060212: KER113_GREENER - (org 21) - 2021-03-15T07:59:57.572417Z
- 7076ff005605022f: ISTATION_05022F - (org 1) - 2021-03-15T07:59:52.116257Z
- 008000000001372c: MTCAP_FABMSTIC_01372C - (org 1) - 2021-03-15T07:59:49.965742Z
- 0000024b08060211: KER_109_IM2AG - (org 19) - 2021-03-15T07:59:47.160521Z
- 0000024b080603a3: KER112_SAINTEYNARD - (org 18) - 2021-03-15T07:59:44.601012Z
- 7076ff005606070e: ISTATION_06070E_PNE - (org 23) - 2021-03-15T07:59:43.093400Z
- 00800000a00064ea: MTCDT_TDOA_64EA_CSUG - (org 1) - 2021-03-15T07:59:41.732776Z
- 7276ff0039030716: KER_FEMTO_030716_P307 - (org 6) - 2021-03-15T07:59:39.766740Z
- 7276ff0039030871: KER_FEMTO_030871_NP - (org 6) - 2021-03-15T07:59:37.560279Z
- 7076ff005605023f: ISTATION_05023F_ATELIERS - (org 2) - 2021-03-15T07:59:37.357364Z
- 7276ff0039030724: KER_FEMTO_030724_P124 - (org 6) - 2021-03-15T07:59:37.062117Z
- 7076ff00560502f3: ISTATION_0502F3_MAIRIE - (org 2) - 2021-03-15T07:59:36.868200Z
- 7076ff00560503f4: ISTATION_0503F4_IGE - (org 10) - 2021-03-15T07:59:36.387964Z
- 0000024b0805031a: KER_107_POLYTECH - (org 6) - 2021-03-15T07:59:35.506404Z
- 7076ff0056050437: ISTATION_50437_IMAG - (org 1) - 2021-03-15T07:59:35.247267Z
- 0000b827eb18a62a: RPI114_SAINT_CASSIEN - (org 13) - 2021-03-15T07:59:34.730023Z
- b827ebffff537763: RPiZeroLoRa_OD - (org 16) - 2021-03-15T07:59:33.029307Z

Passive gateways

- 0016c001ff1034f0: CORECELL_34F0_BALLON - (org 4) - 2021-03-10T06:03:56.266771Z
- 0016c00100002f4c: CORECELL_2F4C_BALLON - (org 4) - 2021-03-09T16:47:01.784090Z
- 7076ff00560503f6: ISTATION_0503F6 - (org 3) - 2021-03-09T10:40:47.922981Z
- 7076ff00560503b2: ISTATION_0503B2_OHP - (org 4) - 2021-03-05T11:29:56.461634Z
- 0000024b08060392: KER108_MONTBONNOT - (org 24) - 2021-03-05T06:44:43.784605Z
- 7076ff0056050238: ISTATION_050238_THEATRE - (org 2) - 2021-02-23T14:04:25.935815Z
- 0000024b08050064: KER102_PERSYVAL - (org 1) - 2021-02-05T09:12:39.365125Z
- 7076ff00560503b6: ISTATION_0503B6 - (org 3) - 2021-01-22T15:53:31.073575Z
- 00800000a00065f2: MTCDT_2G4_65F2 - (org 1) - 2021-01-16T16:42:33.305074Z

Figure 3 : Initial HTML report of gateways

We implemented the Sparkline feature. We used the **API REST of Network Server of CampusIoT** to recover statistics data of gateways. Then, we used **JQuery Sparkline** to display these graphs on the report.

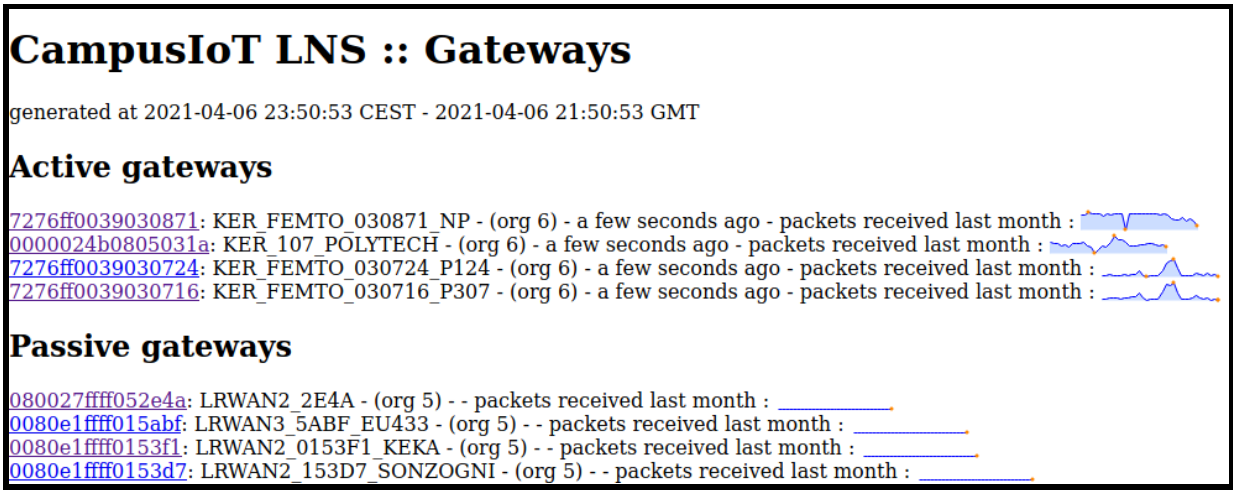


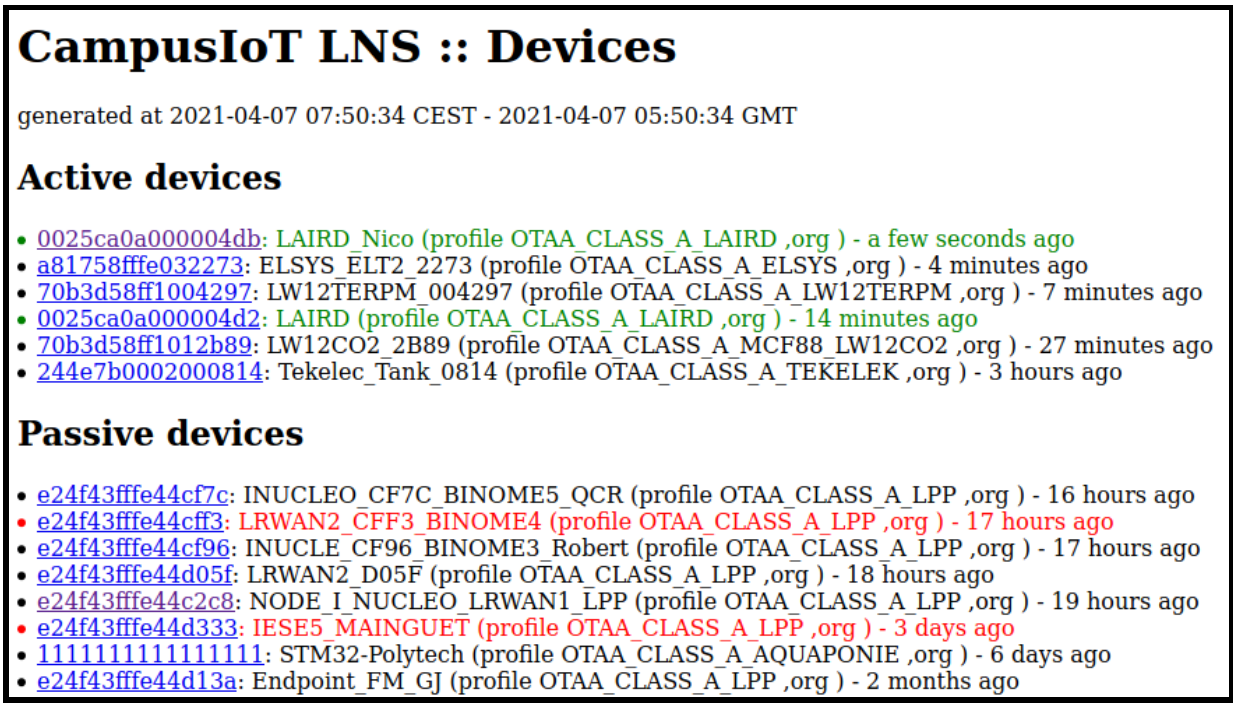
Figure 4 : The result of HTML report of gateways

You can also see that we change the unreadable time format for better understanding.

For the implementation of devices, we had a bug at the beginning because the actual code was working if the user was admin on Campuslot. To get devices, we had to use the **API** by recovering the organization and application to which the user belongs. Then we re-create the devices json file from there.

We add another feature on the report which consists in adding a memory effect. It's the delta between passive and active gateways. We want to know if, compared to the last report, some gateways turned on or off. We put a simple code color on this : If a gateway is green, then she went from a passive to an active state. Otherwise the color code is red.

We implemented the Sparkline feature. We used the **API REST of Network Server of CampusIoT** to recover statistics data of gateways. Then, we used **JQuery Sparkline** to display these graphs on the report.



CampusIoT LNS :: Devices

generated at 2021-04-07 07:50:34 CEST - 2021-04-07 05:50:34 GMT

Active devices

- 0025ca0a000004db: LAIRD_Nico (profile OTAA_CLASS_A_LAIRD ,org) - a few seconds ago
- a81758fffe032273: ELSYS_ELT2_2273 (profile OTAA_CLASS_A_ELSYS ,org) - 4 minutes ago
- 70b3d58ff1004297: LW12TERPM_004297 (profile OTAA_CLASS_A_LW12TERPM ,org) - 7 minutes ago
- 0025ca0a000004d2: LAIRD (profile OTAA_CLASS_A_LAIRD ,org) - 14 minutes ago
- 70b3d58ff1012b89: LW12CO2_2B89 (profile OTAA_CLASS_A_MCF88_LW12CO2 ,org) - 27 minutes ago
- 244e7b0002000814: Tekelec_Tank_0814 (profile OTAA_CLASS_A_TEKELEK ,org) - 3 hours ago

Passive devices

- e24f43fffe44cf7c: INUCLEO_CF7C_BINOME5_QCR (profile OTAA_CLASS_A_LPP ,org) - 16 hours ago
- e24f43fffe44cff3: LRWAN2_CFF3_BINOME4 (profile OTAA_CLASS_A_LPP ,org) - 17 hours ago
- e24f43fffe44cf96: INUCLE_CF96_BINOME3_Robert (profile OTAA_CLASS_A_LPP ,org) - 17 hours ago
- e24f43fffe44d05f: LRWAN2_D05F (profile OTAA_CLASS_A_LPP ,org) - 18 hours ago
- e24f43fffe44c2c8: NODE_I_NUCLEO_LRWAN1_LPP (profile OTAA_CLASS_A_LPP ,org) - 19 hours ago
- e24f43fffe44d333: IESE5_MAINGUET (profile OTAA_CLASS_A_LPP ,org) - 3 days ago
- 1111111111111111: STM32-Polytech (profile OTAA_CLASS_A_AQUAPONIE ,org) - 6 days ago
- e24f43fffe44d13a: Endpoint_FM_GJ (profile OTAA_CLASS_A_LPP ,org) - 2 months ago

Figure 5 : The result of HTML devices of gateways (with Memory Effect)

We managed at the end to send through emails, the html report with sparkline generated into. It was a hard task because we cannot inject javascript on emails. We had to take an image of our html page.

Development Dashboard

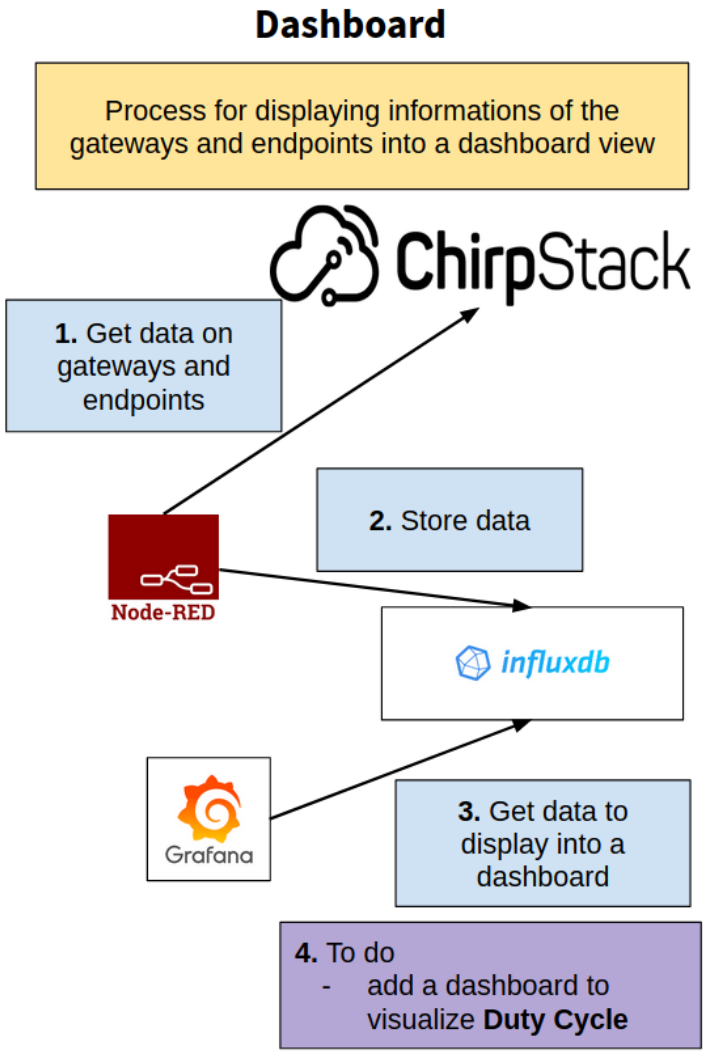


Figure 6 : Dashboard Architecture

If we wanted to develop some new dashboards we had to understand how the 3 technologies were working together. After learning how to use docker-compose (1 docker images for each technology) and after getting every account login we could finally understand how those three were working together. Finally NodeRED is asking Chirpstack for gateways and endpoints data and sending them to an influxdb database. Grafana then requests this database and displays the answer in a dashboard shape. Unfortunately we didn't have time to create new dashboards because report development took us more time than we thought. If we had time we would have searched to display a duty cycle dashboard of gateways. We should have done a request to Influxdb and some calculation (Time On Air).

CONCLUSION

Review Issues

For the code :

During the project, on the Milestone “Version 2021”, we created 9 issues and we have done 58 commits.

```
40 files changed, 878 insertions(+), 185 deletions(-)
```

It might seem a few but we have mainly written shell scripts. This is a powerful language but not easy to master.

For the documentation :

We have done 25 documentation commits. We have created 3 different documentations. We made a technical doc through notices.

We are satisfied about our first uses of issues on github. This is an amazing tool to follow task advancements. But we’ve lost rigor into the task following. We should have planned them earlier.

Discontinued Features

Duty Cycle on Grafana :

Duty cycle is the proportion of time during which a component, device, or system is operated. The duty cycle can be expressed as a ratio or as a percentage. In Europe there is a 0.1% and 1.0% duty cycle per day depending on the channel. To calculate the duty cycle of gateways, we had to get the Time On Air (amount of time before the receiver receives the signal) but we didn’t find any easy formula. There were only existing online calculators but with no explicit formula. We decided to stop the research because we decided to focus on the report generating.

TTN Mapper :

TTN Mapper is an existing tool that lets us see on a map the Lora gateways. We wanted to do the same with only gateways that were near from Saint-Martin-d’Heres campus. We decided to use leaflet because one member of the group already tried it. But we’ve come to a problem that had nothing to do with the map itself, we couldn’t read local files (for security reasons of course). We wanted to read local files to get coordinates of gateways stored in JSON files, to pin them up on the map. We lost a lot of time searching for solutions, only to reach the conclusion that it was impossible (from a web point of view which we had at the moment). It could maybe have been possible to read files using a scripting aspect but because we already had lost a lot of time we decided to pause the research and focus on the main part of the project which was mail reporting.

Possible improvements

Sparkline :

Instead of taking a capture of the whole page html, take a picture of each sparkline graph and replace it through the mail. It's possible to send an email integrating images attachment into html body. Indeed, we managed to get a result like this :

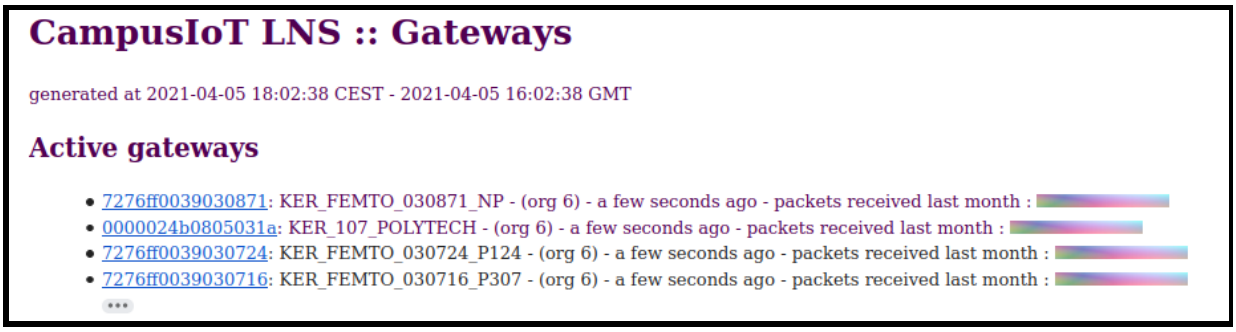


Figure 7 : Example of an email with images integrated on the html body

The hardest part is to capture each html sparkline element generated with javascript. All this without having to open the page from the internet. This might be possible with javascript libraries like **html2canvas** but we haven't been able to go deeper into the subject.

What the project brought us

Overall, this project was very interesting. We understood even better how to use technologies such as docker, npm, nodeJS, and the managing of JSON with shell script using JQ. It was a first step into the Internet of Things world which is a technology that will be ubiquitous in the near future. We also coded and well documented our work so that users could understand how to use the report generator. Also, other developers would have the resources to take over certain aspects of the project.