

# IoT Module

## List of projects

**November-December 2024**

### Organisation :

- Groupes 2 de personnes

### Calendrier :

- Du 7.11.24 au 12.12.24 : Travail individuel sur le projet. Professeur ou assistant à disposition durant les sessions du jeudi matin.
- 19.12.24 : Présentation par groupe

### Evaluation et rendu :

- Rapport décrivant le projet, l'architecture et les challenges du projet. Si demandé dans le projet, analyse des coûts et de la scalabilité.
- Pour les projets utilisant le cloud AWS, possibilité de rendre à la place du rapport, un document écrit sous la forme d'un post blog  
exemple : <https://aws.amazon.com/fr/blogs/iot/serverless-iot-anomaly-detection>
- Vidéo de la solution réalisée montrant le fonctionnement global du système (1" idéalement)
- Note individuelle basée entre autre sur la présentation, la participation, la capacité de chaque personne à répondre correctement aux questions de façon globale sur le projet

### Bonus Incentive :

- 1 des projets rendu sous la forme d'un blog AWS aura peut être la chance d'être publié sur le site de AWS avec mention personnelle.

**Projets 1 à 6** supervisés par **Nabil Abdennadher et Raoul Dupuis**

**Projets 7 à 11** supervisés par **Fabien Vannel**.

Dans le cadre des projets 7 à 11, il sera demandé d'étudier des solutions complémentaires du cercle AWS IoT (dernière page). Une ou plusieurs solutions d'analyse pourront par exemple être choisies et étudiées dans le projet. Une discussion avec le professeur sera nécessaire pour choisir les produits à intégrer. Pour les projets avec AWS, un audit de l'architecture devra être mené à l'aide des outils WAR IoT Lens.

## Projects 1, 2 and 3: Using edge computing to deploy an IoT self-adaptive applications on an IoT-Edge-Cloud platform

The aim of this project is to classify images from a camera and improve the classification as the application runs. ML techniques are used to generate the classifier. The type of classification is left up to the students: recognising digits, animals, objects, etc.

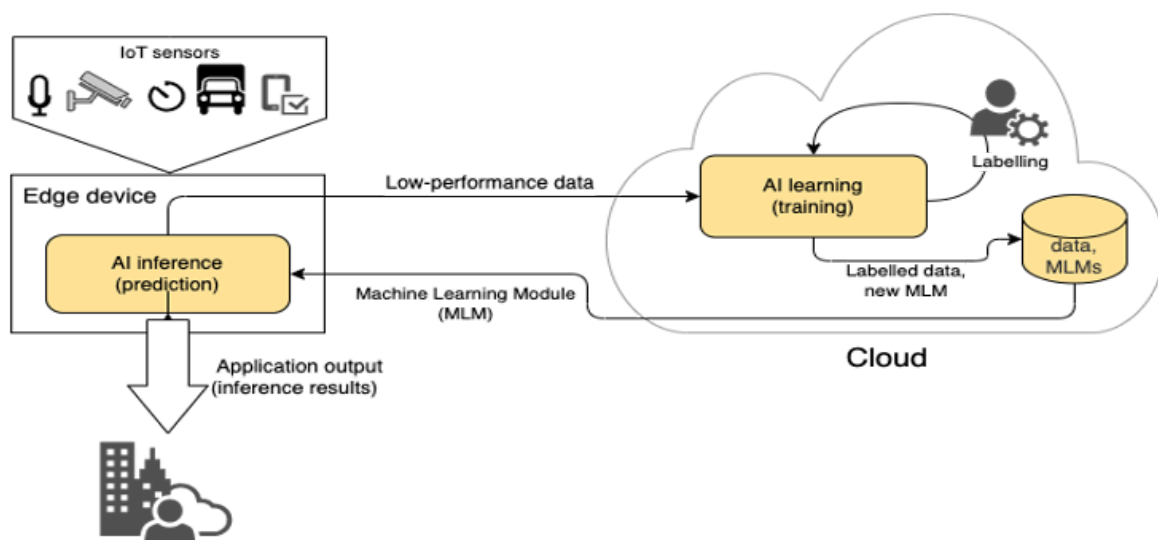
The hardware system is composed of a camera, an edge device (represented by a raspberry) and a cloud infrastructure.

The IoT application is composed of three modules:

1. Camera Reader: reads the stream coming from the camera
2. Classifier: Receive an image from the camera reader and perform the classification
3. Learning tool: a Learning algorithm which generates the classifier (inference)

The Learning takes place in the cloud, while inference (the classifier, called: Machine Learning Model - MLM, in Figure 2) is performed on the edge. The deployment of MLMs on the edge devices roughly follow three steps:

1. The Learning tool generates an initial MLM from a set of training data. The MLM is then deployed on the edge device.
2. The performance of the deployed MLM is continuously monitored: each time the MLM is unable to take a reliable decision, it sends the data it failed to process to the cloud.
3. The cloud then collects these datasets from different edge devices and triggers a new learning cycle, resulting in a new MLM which will then be pushed back to the edge devices.



**Figure 2: IoT applications deployed on IoT-Edge-Cloud solutions**

You are expected to deploy this solution using one of the following edge computing tools: AWS Green Grass, IoT Edge Azure and Nuvla/NuvlaEdge. Three groups will work on this project. They will use each a different edge computing tool

Here are the steps to follow to successfully complete this project:

1. Deploy the selected edge computing solution on a virtual edge device (VM instance deployed on the cloud) and implement the closed loop of Figure 2.
2. Deploy a “Hello World Application” using the selected edge computing solution. The goal here is to test the closed loop detailed in Figure 2.
3. Develop the camera reader software (on raspberry)
4. Redo points 1 and 2 using a raspberry (instead of the virtual Edge device) and the classification application (instead of the Hello World Application). The camera is connected to the raspberry. The “camera reader” software, deployed on the raspberry, retrieves images from the camera.
5. Use the Web of Things Standard to define the interface between the camera and the raspberry.
6. Documentation + Presentation

The project will be evaluated following four milestones:

1. M1 (Thursday 14 Nov. 2024, 11h15). Step 1 and 2 are completed.
2. M2 (Thursday 28 Nov. 2024, 11h15). Step 3 and 4 are completed.
3. M3 (Thursday 12 Dec. 2024, 11h15), Step 5 is completed
4. M4 (Thursday 19 Dec. 2024, 8h45), Step 6 is completed

## Project 4, 5, 6: Using edge computing to deploy an IoT self adaptive energy application on an IoT-Edge-Cloud platform

The goal of this project is to use the [Clemap edge device](#) (Figure 3.1 and Figure 3.2) as a backbone to set-up an IoT infrastructure applied to the power energy market. The platform will be composed of:

1. digital infrastructure with 3 layers: home appliances (heat pumps, boilers, EV charging stations, etc.), Clemap edge devices and a cloud infrastructure,
2. a specific module that collects data from home appliances
3. a specific module that predicts/forecasts the consumption of the home appliances (To be developed according to instructions to be given later)

The goal of the project is to:

- Collect data from the Sensors, transfer and store them on the cloud (point 2 above)
- Develop a draft version of the forecast module
- Deploy the whole solution (forecast and collection modules) using one of the following edge computing tools: AWS Green Grass, IoT Edge Azure and Nuvla/NuvlaEdge. Three groups will work on this project. They will each use a different edge computing tool.
- Improve the intelligence (forecast) by implementing the closed loop described in Figure 2

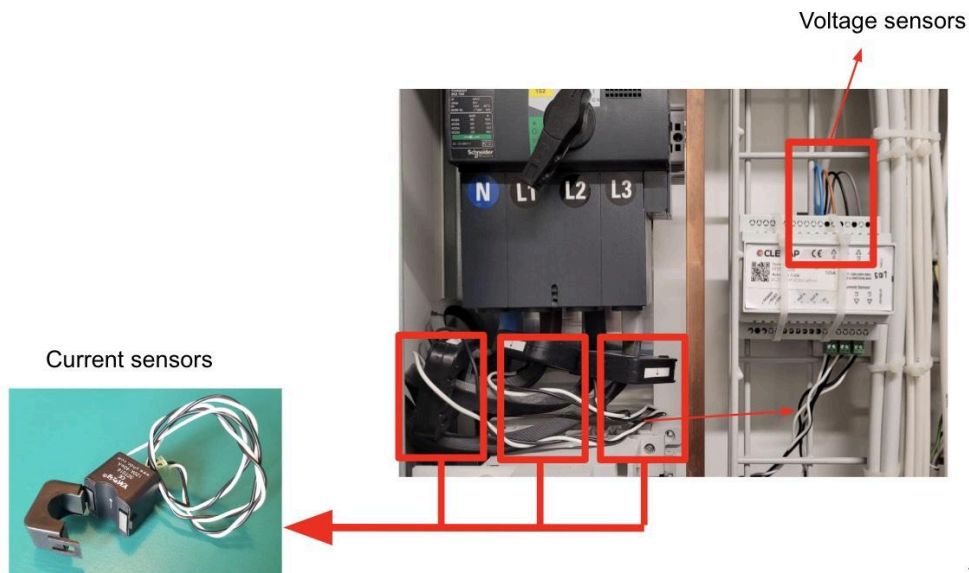


Figure 4.1: Clemap installation

Here are the steps to follow to successfully complete this project:

1. Deploy the selected edge computing solution on a virtual edge device (VM instance deployed on the cloud) and implement the closed loop of Figure 2.
2. Deploy a “Hello World Application” using the selected edge computing solution. The goal here is to test the closed loop detailed in Figure 2.
3. Deploy and test the data collection module (on the CLEMAP)

4. Redo points 1 and 2 using the CLEMAP device (instead of the virtual Edge device) and the forecast application (instead of the Hello World Application). The sensors are connected to the CLEMAP. The data collection module, deployed on the CLEMAP, retrieves measures from the sensors and retrieves them in a database.
5. Use the Web of Things Standard to define the interface between the sensors and the CLEMAP.
6. Documentation + Presentation

The project will be evaluated following four milestones:

5. M1 (Thursday 14 Nov. 2024, 11h15). Step 1 and 2 are completed.
6. M2 (Thursday 28 Nov. 2024, 11h15). Step 3 and 4 are completed.
7. M3 (Thursday 12 Dec. 2024, 11h15), Step 5 is completed
8. M4 (Thursday 19 Dec. 2024, 8h45), Step 6 is completed

## Project 7: Gestion de flotte de vélo



Nous souhaitons développer une solution d'alarme et tracker de vélo pour une flotte de vélo en libre service dans une ville. Des vélos sont équipés de tracker LoRa basé sur la carte WIO Tracker 1110.

Le système basse énergie utilise la technologie LoRa pour communiquer. Le module (LR1110) WiFi, GNSS, en mode scanner de réseau, sera aussi utilisé pour réaliser une géolocalisation plus précise.

- Au travers d'une application (page web) sur mobile vous devrez activer ou désactiver la surveillance du vélo. (location en cours ou pas)
- Si celui-ci bouge alors qu'il est sous alarme, vous devrez remonter immédiatement un message sur l'application.
- Régulièrement (5-10 min) nous souhaitons obtenir la géolocalisation approximative du vélo. Pour ce faire vous allez utiliser les réseaux possibles de localisation de la carte Wio Tracker 110 pour obtenir une identification de cette position. Une cartographie permettra de localiser le vélo.
- En cas d'absence de message pendant un certain temps une alarme devra être remontée.

Des garages à vélo connectés seront déployés dans les villes. Ces garages seront équipés d'une porte contrôlable à distance et intègrent différents capteurs entre autres permettant de connaître le taux d'occupation du garage à vélo:

- Ces garages à vélo seront simulés par des modules Wi-Fi.
- La communication par WiFi avec un cloud devra être sécurisée avec des certificats.

Vous pourrez déployer la solution LoRa Network Servers soit sur TTN, soit sur AWS

La géolocalisation pourra se faire avec le service suivant:

<https://developers.google.com/maps/documentation/geolocation/overview>

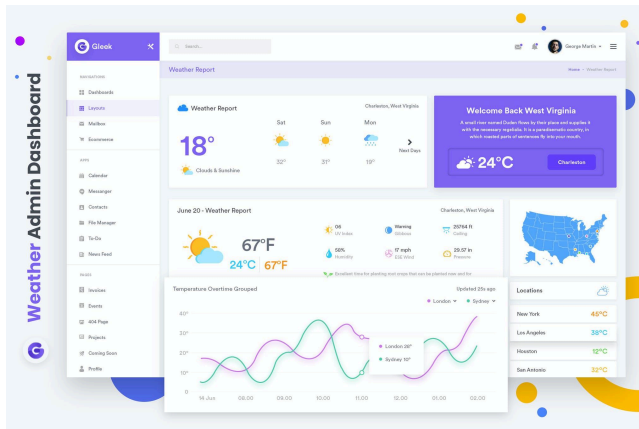
[https://wiki.seeedstudio.com/connect\\_wio\\_tracker\\_to\\_locator/](https://wiki.seeedstudio.com/connect_wio_tracker_to_locator/)

**Rendu attendu :**

- La solution devra être déployée dans un cloud en utilisant des solutions de type SaaS.
- Le système déployé devra être “scalable” avec la gestion de plusieurs vélos.
- Une analyse des coûts devra être faite en imaginant 3 types de tailles : small (~100 devices), medium (~5000 devices), large (~100'000 devices)
- Une étude sur une solution compatible “disaster recovery” est demandée

**Mot clés :** LoRaWAN, ESP32, AWS IoT Core, AWS location, Disaster recovery

## Project 8: Système météo et énergétique



Nous souhaitons développer une solution composée de plusieurs capteurs : Des capteurs météo et des capteurs énergétiques pour la maison (prises connectées, production solaire, etc..) Certains capteurs seront du type LoRa et d'autres de type WiFi. Dans la réalité il pourrait s'agir de capteurs à proximité d'une habitation et de capteurs déportés à l'extérieur.

- Un dashboard permettra de visualiser l'évolution des mesures
- La communication par WiFi avec un cloud devra être sécurisée avec des certificats.
- En cas d'absence de message pendant un certain de la part d'un capteur des alarmes devront être générées

### Rendu attendu :

- La solution devra être déployée dans un cloud en utilisant des solutions de type SaaS.
- Le système déployé devra être "scalable" avec la gestion de plusieurs capteurs.
- Une analyse des coûts devra être faite en imaginant 3 types de tailles : small (~100 devices), medium (~5000 devices), large (~100'000 devices)
- Le provisionnement des capteurs Wi-Fi devra se faire par un certificat intermédiaire de provisionnement.

<https://docs.aws.amazon.com/pdfs/whitepapers/latest/device-manufacturing-provisioning/device-manufacturing-provisioning.pdf#device-manufacturing-provisioning>  
<https://docs.aws.amazon.com/pdfs/whitepapers/latest/device-manufacturing-provisioning/device-manufacturing-provisioning.pdf#device-manufacturing-provisioning>

**Mot clés :** LoRaWAN, ESP32, AWS IoT Core, Certificate Provisioning



## Project 9: Détection d'anomalie sur des installations industrielles



Une usine dispose d'appareils à surveiller. Nous souhaitons développer une solution basée sur des capteurs de type WiFi ou LoRa à même de surveiller des paramètres (lumière, vibration, température...) Lorsque des mesures différentes de certains intervalles des alarmes doivent pouvoir être générées.

- Un dashboard permettra de visualiser l'évolution des valeurs et notifier les alarmes
- La communication par WiFi avec un cloud devra être sécurisée avec des certificats.
- En cas d'absence de message pendant un certain de la part d'un capteur une alarme devra être visualisée
- Le déploiement de masse de capteurs et le provisionnement des certificats devra se faire de façon automatisable (sans passer par l'interface web du cloud)
- La solution devra privilégier une architecture de type IoT EDGE

### Rendu attendu :

- La solution devra être déployée dans un cloud en utilisant des solutions de type SaaS.
- Le système déployé devra être "scalable" avec la gestion de plusieurs capteurs.
- Une analyse des coûts devra être faites en imaginant 3 types de tailles : small (~100 devices), medium (~5000 devices), large (~100'000 devices)
- Le provisionnement des capteurs Wi-Fi devra se faire par un certificat intermédiaire de provisionnement.

<https://docs.aws.amazon.com/pdfs/whitepapers/latest/device-manufacturing-provisioning/device-manufacturing-provisioning.pdf#device-manufacturing-provisioning><https://docs.aws.amazon.com/pdfs/whitepapers/latest/device-manufacturing-provisioning/device-manufacturing-provisioning.pdf#device-manufacturing-provisioning>

**Mot clés :** ESP32, AWS IoT Core, AWS Greengrass, Certificate Provisionning

## Project 10: Supervision et contrôle d'une ruche



Une ruche instrumentée permet de surveiller le développement d'une colonie d'abeille.

Nous souhaitons simuler une ruche instrumentée par une communication LoRa. Plusieurs capteurs (température intérieure et extérieure, luminosité, hygrométrie) servent à suivre la santé de la ruche. Par exemple, en cas d'attaque par des insectes les abeilles peuvent augmenter volontairement la température pour épuiser l'intrus.

En cas de dispersion de pesticide dans la région, le propriétaire des ruches peut envisager de volontairement bloquer les abeilles à l'intérieur. Une trappe pilotable à distance doit empêcher aux abeilles de sortir dès le lever du soleil.

- Un dashboard permettra de visualiser l'évolution des valeurs et notifier les alarmes (par exemple anomalie sur la température ou autre cas)
- La communication par LoRa avec un cloud pourra se faire avec soit un réseau TTN soit un réseau propriétaire (AWS IoT LoRaWAN)
- A distance il devra être possible de bloquer les abeilles à l'intérieur
- Le déploiement de masse de ruches et la gestion multi-utilisateur devra être envisagée

Pour les ruches à proximité d'une habitation, une solution par Wi-Fi sera proposée à la place. D'autres capteurs pourront être ajoutés pour remonter plus d'information

### Rendu attendu :

- La solution devra être déployée dans un cloud en utilisant des solutions de type SaaS.
- Le système déployé devra être "scalable" avec la gestion de plusieurs ruches.
- Une analyse des coûts devra être faite en imaginant 3 types de tailles : small (~100 devices), medium (~5000 devices), large (~100'000 devices)
- Une étude sur une solution compatible "disaster recovery" est demandée

**Mot clés :** LoRaWAN, ESP32, AWS IoT Core, AWS location, Disaster recovery

**Project 11: Projet libre**

Ce projet devra utiliser des capteurs LoRa ou/et WiFi. Il devra être focalisé sur une solution Cloud de type SaaS

Le côté industrialisable du projet devra être pris en compte : par exemple déploiement de plusieurs capteurs, sécurité, surveillance du parc, ...

Le système déployé devra être “scalable” avec la gestion de plusieurs capteurs.

La solution devra être déployée dans un cloud en utilisant des solutions de type SaaS.

Une analyse des coûts devra être faite en imaginant 3 types de tailles : small (~100 devices), medium (~5000 devices), large (~100'000 devices)

# Ecosystème IoT AWS

