

---

# Dédicaces

Je dédie ce travail

A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort pour me rendre heureuse : mon adorable mère .

A l'homme, mon précieux offre du dieu, qui doit ma vie, ma réussite et tout mon respect : mon cher père

À tous mes frères qui m'avez toujours soutenu et encouragé durant ces années d'études.

À toute ma famille et Mes amis,

À mes professeurs merci d'avoir enrichi mes connaissances et de me guidé durant toute cette année.

A tous ceux que j'aime,

Nadine Sahli.

---

# REMERCIEMENT

Au terme de ce projet de fin d'étude, mes vifs remerciements sont dédiés à tous ceux qui ont contribué directement ou indirectement à l'élaboration de ce projet. J'exprime toute ma gratitude à Monsieur MOEZ BALTI , mon encadrant à l'institut Supérieur des études technologiques en Communications de Tunis pour ses conseils prodigués, sa persévérance dans le suivi et pour la qualité de son encadrement.

J'adresse également mes sincères remerciements à Monsieur KARIM

FATHALLAH mon encadrant du stage à la Start up Smart for green, pour ses précieux conseils, sa patience et sa disponibilité et pour ses critiques et ses conseils enrichissants.

J'adresse aussi mes remerciements aux membres de jury pour avoir accepté de juger, d'évaluer et d'enrichir ce travail.

Par la même occasion, je présente mes sincères reconnaissances à tous mes enseignants qui m'ont soutenu au long de mon cursus étudiant et à tous ceux qui m'ont porté aide et soutien durant mes études.

Merci pour toute personne qui a participé de près ou de loin pour l'accomplissement de ce modeste travail.

---

# TABLE DES MATIÈRES

<b>Liste des tableaux</b>	<b>vii</b>
<b>LISTE DES ABRÉVIATIONS</b>	<b>viii</b>
<b>Introduction générale</b>	<b>ix</b>
<b>1 Contexte général du projet et les notions utilisées</b>	<b>2</b>
1.1 Présentation de l'organisme d'accueil . . . . .	3
1.1.1 Objectif de Smart for Green . . . . .	3
1.2 Présentation du projet . . . . .	3
1.2.1 Problématique . . . . .	4
1.3 Planification et déroulement du projet . . . . .	5
1.4 L'internet des objets . . . . .	6
1.5 Les réseaux à courte portée . . . . .	7
1.6 Les réseaux à longue portée et basse consommation (LPWAN) . . . . .	7
1.7 Protocole de communication : MQTT . . . . .	17
1.7.1 Description . . . . .	17
1.7.2 Architecture . . . . .	17
<b>2 Analyse des besoins et Conception</b>	<b>20</b>
2.1 Étude de l'existant . . . . .	21
2.1.1 Critique de l'existant . . . . .	22
2.1.2 Solution proposée . . . . .	23
2.2 Spécification des besoins . . . . .	23
2.2.1 Besoins fonctionnels . . . . .	23
2.2.2 Besoins non fonctionnels . . . . .	24
2.3 Identification des acteurs et des cas d'utilisation . . . . .	25
2.3.1 Les acteurs . . . . .	25
2.3.2 Diagramme des cas d'utilisation global . . . . .	26

2.3.3	Diagramme de cas d'utilisation gestion des composteurs . . . . .	27
2.3.4	Diagramme de cas d'utilisation gestion des projets de composteurs . . .	28
2.3.5	Diagramme de cas d'utilisation gestion des clients . . . . .	29
2.3.6	Diagramme de cas d'utilisation gestion des noeuds . . . . .	30
2.4	Architecture générale de la solution proposée . . . . .	31
2.5	Conception détaillée . . . . .	31
2.5.1	Vue statique . . . . .	32
2.5.1.1	Diagramme de classe . . . . .	32
2.5.2	Vue denamique . . . . .	33
2.5.2.1	Diagramme de séquence . . . . .	33
<b>3</b>	<b>Réalisation</b>	<b>35</b>
3.1	Environnement de travail . . . . .	36
3.1.1	Environnement matériel . . . . .	36
3.1.2	Environnement logiciel . . . . .	40
3.2	Réalisation et développement de la solution IoT . . . . .	46
	<b>CONCLUSION GÉNÉRALE</b>	<b>51</b>
	<b>BIBLIOGRAPHIE</b>	<b>51</b>
	<b>ANNEXES</b>	<b>53</b>
A.1	TITRE ANNEXE 1 . . . . .	53
A.2	TITRE ANNEXE 2 . . . . .	53

---

# LISTE DES FIGURES

1.1	l'émission d'oxyde de carbone en tnisie . . . . .	4
1.2	Diagramme de gantt . . . . .	6
1.3	Occupation spectrale d'une transmission LoRa . . . . .	9
1.4	Architecture de LoraWAN . . . . .	13
1.5	Les couches protocolaires LoRa . . . . .	14
1.6	Les classes LoRaWAN . . . . .	15
1.7	Combinaison de l'authentification et du chiffrement . . . . .	16
1.8	Architecture de mqtt . . . . .	19
2.1	Diagramme de cas d'utilisation globale . . . . .	26
2.2	Diagramme de cas d'utilisation gestion des composteurs . . . . .	27
2.3	Diagramme de cas d'utilisation gestion des projets . . . . .	28
2.4	Diagramme de cas d'utilisation gestion des clients . . . . .	29
2.5	Diagramme de cas d'utilisation gestion des noeuds . . . . .	30
2.6	Architecture générale de la solution proposée . . . . .	31
2.7	Diagramme de classe . . . . .	32
2.8	Diagramme de séquence : "S'authentifier" . . . . .	33
3.1	la Wio-E5 Development Kit . . . . .	36
3.2	La carte lora E5 . . . . .	37
3.3	Le Dragino gateway LoRa . . . . .	38
3.4	Capteur de température et d'humidité Grove - DHT22 . . . . .	39
3.5	Logo du Stm32CubeIDE . . . . .	40
3.6	Logo du Stm32CubeProgrammer . . . . .	41
3.7	Logo du langage du programmation C . . . . .	41
3.8	Logo de python . . . . .	42
3.9	Logo de la framework django . . . . .	43
3.10	Logo de Postgresql . . . . .	43
3.11	Logo de leaflet . . . . .	44

3.12 Logo HTML . . . . .	44
3.13 Logo CSS . . . . .	45
3.14 Logo Javascript . . . . .	45
3.15 Logo git . . . . .	46
A.1 Logo De Smart for Green . . . . .	53
LISTE DES FIGURES	

---

## Liste des tableaux

1.1	Technologies à courtes portées . . . . .	7
1.2	Technologies à longues portées . . . . .	7
2.1	Cas d'utilisation : gestion des composteurs . . . . .	27
2.2	Cas d'utilisation : gestion des projets des composteurs . . . . .	29
2.3	Cas d'utilisation : Ajouter un client . . . . .	30
2.4	Cas d'utilisation : gestion des noeuds . . . . .	31

---

# **LISTE DES ABRÉVIATIONS**



---

# Introduction générale

Ces dernières années, l'IoT<sup>1</sup> est devenu l'une des technologies les plus prometteuses du 21ème siècle. Actuellement nous pouvons connecter des objets du quotidien à internet par l'intermédiaire de terminaux intégrés. Plusieurs communications sont devenues possibles en toute transparence entre les personnes, les processus et les objets.

L'IoT offre aussi un potentiel énorme et s'impose dans de nombreuses activités publiques et industrielles. Il mène à des villes plus intelligentes, à des industries beaucoup plus efficaces, à des transports plus sûrs et à une meilleure qualité de vie en général.

Dans le monde agricole, les solutions IoT prennent la forme de capteurs reliés à Internet pour collecter des mesures environnementales et mécaniques.

L'agriculture intelligente basée sur les technologies IoT offre par ailleurs l'opportunité aux agriculteurs et aux producteurs de réduire les déchets et d'améliorer la productivité[1].

Le projet concerne la conception et développement d'un système de surveillance du processus de compostage des déchets organique . L'internet d'objet est la technique retenue pour réaliser cette tâche.

---

1. IoT :Internet of things (Internet des Objets)

Divisé en quatre grandes parties, ce rapport replace dans un premier temps le contexte du projet ainsi que ses enjeux. Ensuite, dans le chapitre "État de l'art de l'internet des objets", nous commencerons par un aperçu général de l'internet des objets où nous définirons certaines notions clés utilisées dans ce projet. puis dans le troisième chapitre intitulé " Conception " nous entrons davantage dans les détails sur lesquels se base notre solution à l'aide des diagrammes.

Finalement, dans le dernier chapitre intitulé " Mise en œuvre d'un système de surveillance du processus de compostage des déchets organique" nous présentons les environnements matériels et logiciels utilisés et les étapes d'implémentation de notre solution en se basant sur des imprimes d'écran.

---

# Contexte général du projet et les notions utilisées

## Sommaire

---

<b>1.1</b>	<b>Présentation de l'organisme d'accueil . . . . .</b>	<b>3</b>
1.1.1	Objectif de Smart for Green . . . . .	3
<b>1.2</b>	<b>Présentation du projet . . . . .</b>	<b>3</b>
1.2.1	Problématique . . . . .	4
<b>1.3</b>	<b>Planification et déroulement du projet . . . . .</b>	<b>5</b>
<b>1.4</b>	<b>L'internet des objets . . . . .</b>	<b>6</b>
<b>1.5</b>	<b>Les réseaux à courte portée . . . . .</b>	<b>7</b>
<b>1.6</b>	<b>Les réseaux à longue portée et basse consommation (LPWAN)</b>	<b>7</b>
<b>1.7</b>	<b>Protocole de communication : MQTT . . . . .</b>	<b>17</b>
1.7.1	Description . . . . .	17
1.7.2	Architecture . . . . .	17

---

## **Introduction**

Dans ce chapitre, nous exposons le cadre général du projet. Nous présentons en premier lieu l'organisme d'accueil. En deuxième lieu, nous expliquons la problématique ainsi que le diagramme de gantt aussi les notions utilisées.

### **1.1 Présentation de l'organisme d'accueil**

Smart for Green est une Startup créée en 2022 au sein de centre local d'innovations de Ras Jebel-Bizerte, son activité principale était un bureau d'études qui propose une multitude de services concernant l'agriculture intelligente .

#### **1.1.1 Objectif de Smart for Green**

L'objectif de smart for green est d'améliorer la productivité, la qualité des produits agricoles, la durabilité et la rentabilité de l'agriculture toute en préservant l'environnement.

L'équipe Smart for Green propose des solutions intelligente pour la protection contre les risques agricoles.

### **1.2 Présentation du projet**

Cette partie explique la problématique et la solution proposée.

### 1.2.1 Problématique

La concentration de dioxyde de carbone en mai 2020, il était de 417 ppm (l'unité de mesure utilisée pour quantifier la pollution atmosphérique). En mai 2021, le taux était de 419 ppm, et en 2022, dépassait le seuil de 420. La terre se réchauffe sous l'effet des activités humaines. L'émission de gaz carbonique (CO<sub>2</sub>) est énorme dans l'atmosphère : 2.5 tonnes par an et par tunisien en moyenne, selon le centre d'analyses des informations relatives au dioxyde de carbone.

Cela a pour effet de renforcer l'effet de serre et d'augmenter les températures globales. et ça peut rendre une partie de la planète invivable. la figure 1.1 montre l'émission d'oxyde de carbone en Tunisie.

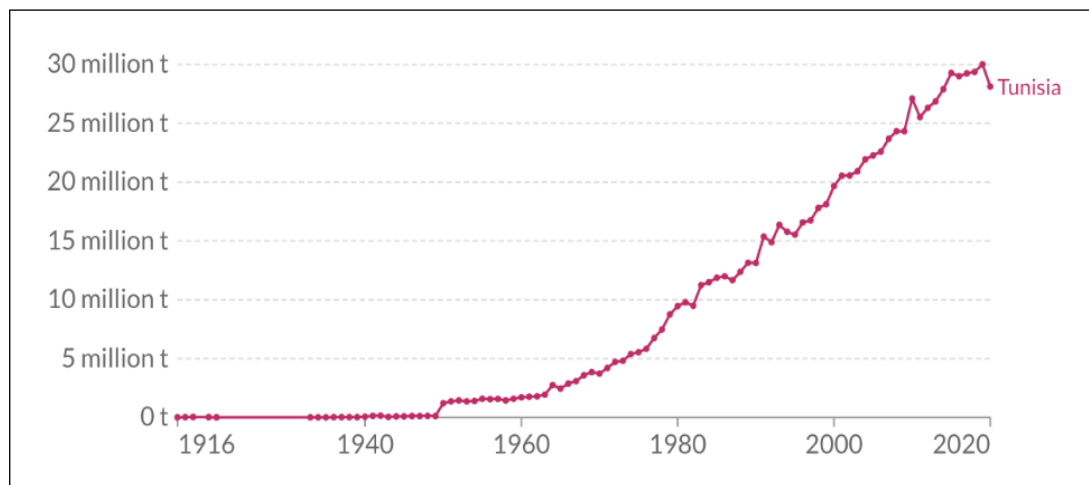


FIGURE 1.1 – l'émission d'oxyde de carbone en tnisie

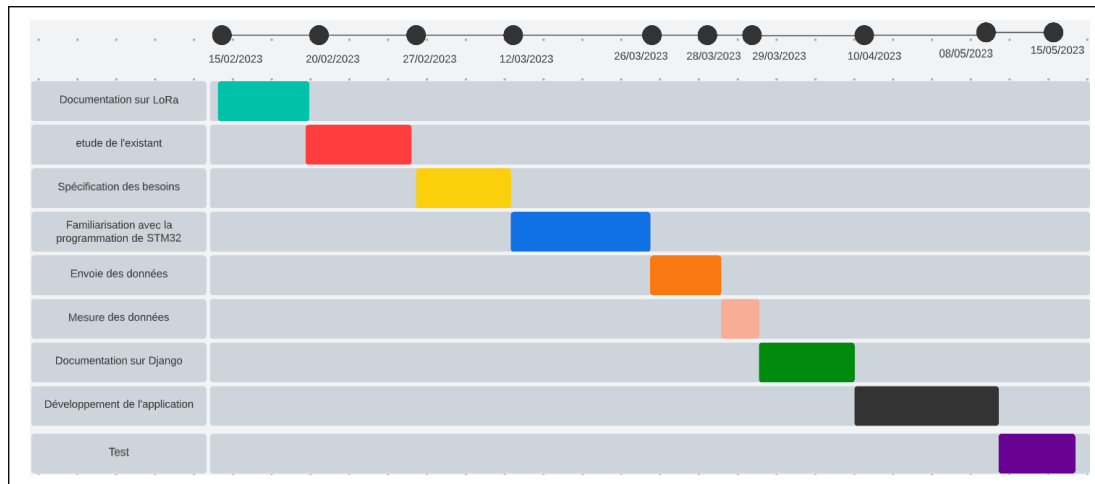
Notre négligence envers plusieurs techniques efficaces pour réduire les émissions de CO<sub>2</sub>, comme le compostage, peut avoir des conséquences désastreuses sur le climat. En effet, en ne mettant pas en pratique des méthodes simples comme la transformation de nos déchets organiques en compost, nous laissons échapper

des gaz à effet de serre dans l'atmosphère, contribuant ainsi au réchauffement climatique.

Il est important de souligner que le compostage sans contrôle de sa qualité peut avoir des effets inverses à ceux recherchés. En effet, si le compost contient des éléments toxiques ou des plastics qui ne se dégradent pas, cela peut non seulement détruire la qualité du compost, mais également contribuer à la libération de gaz à effet de serre. Ainsi, il est crucial d'adopter des pratiques de compostage appropriées et de contrôler la qualité du compost pour s'assurer qu'il est utilisable et bénéfique pour l'environnement. L'objectif de réduire les émissions de CO<sub>2</sub> ne peut être atteint que si nous adoptons des pratiques écologiques qui respectent l'environnement et contribuent à la préservation de notre planète.

### **1.3 Planification et déroulement du projet**

La planification de projet consiste à déterminer et ordonnancer les tâches du projet, et estimer la charge de chacune de ces tâches. Le diagramme de Gantt (figure 1.1) permet de représenter graphiquement l'ensemble des tâches de ce projet et de suivre sa progression.



**FIGURE 1.2 – Diagramme de gantt**

**Nous présentons maintenant l'état de l'art qui est essentiel pour définir les principaux axes du projet.**

## 1.4 L'internet des objets

L'Internet des Objets (IoT) est un concept révolutionnaire qui décrit l'interconnexion intelligente des objets physiques à travers un réseau mondial. Dans l'univers de l'IoT, des dispositifs de toutes sortes, tels que des capteurs, des véhicules, des appareils ménagers et des infrastructures urbaines, sont connectés et échangent des informations de manière transparente et autonome.

Pour permettre cette connectivité, deux familles de réseaux jouent un rôle essentiel dans l'IoT :

- Les réseaux à courte portée.
- Les réseaux à longue portée et basse consommation (LPWAN).

## 1.5 Les réseaux à courte portée

Les réseaux à courte portée comme le bluetooth, le Z-Wave. . . , qui permettent le transfert de données sur de faibles distances. Ils sont utilisés dans les montres, les bracelets, etc... Le tableau 1.1 montre les technologies à courtes portées.

Technologies sans fil	Topologie de réseau	Débit	Réseau	Bandes de fréquences	Norme	Couverture	Consommation d'énergie
Bluetooth	Etoile	700Kbps	PAN	2.4GHz	IEEE 802.15.4	<30m	Faible
BLE	Etoile	1Mbps	PAN	2.4GHz	IEEE 802.15.4	50-150m	Très faible
Zigbee	Mesh, Etoile, Arbre	250Kbps	LAN	2.4GHz	IEEE 802.15.4	10-300m	Très faible
z-Wave	Maillée (mesh)	100 kb	LAN	868,42 MHz Europe	IEEE 802.11	30 mètres	Faibles

TABLE 1.1 – Technologies à courtes portées

## 1.6 Les réseaux à longue portée et basse consommation (LPWAN)

Les réseaux LPWAN offrent une longue portée de communication tout en maintenant une faible consommation d'énergie. Ils sont conçus pour permettre la connectivité de périphériques IoT sur de grandes distances, allant de plusieurs kilomètres à plusieurs dizaines de kilomètres.

Technologies sans fil	Topologie de réseau	Débit	Réseau	Bandes de fréquences	Norme	Couverture	Consommation d'énergie
LoRaWAN	Etoile des Etoiles	0,2-27 Kbps	LPWAN	Divers, sub-gigahertz	LoRaWAN	5Km urbain, 15Km rural	Très faible
Sigfox	Etoile	100 bits/s	LPWAN	868 MHz	Sigfox	10Km urbain 50Km rural	Très faible
NB-IoT	Etoile	100b/s-250 kb/s	LAN	1800MHz	IEEE 802.15.4	2kmurbain-10km rural	Très faible
eMTC	point à multipoint	<1Mb	LPWAN	800Mhz	4G LTE	2Km rural	faible

TABLE 1.2 – Technologies à longues portées

**Les réseaux LPWAN peuvent être classés en deux catégories principales :**



### — les réseaux s'appuient sur les réseaux cellulaires existants

**NB-IoT :** (Narrowband IoT) est un standard de communication sans fil. NB-IoT est plus adapté aux applications IoT à faible bande passante, avec une longue durée de vie de la batterie et une meilleure couverture en intérieur.

**eMTC :** L'eMTC utilise les bandes de fréquences réservées pour les réseaux cellulaires et peut coexister avec les services de voix et de données existants sur ces réseaux. Cela permet une intégration facile des appareils eMTC dans l'infrastructure des opérateurs de télécommunications qui déploient des réseaux LTE.

### — Les réseaux utilisent l'internet :

**Sigfox :** Sigfox utilise une bande de fréquences non sous licence pour transmettre des données entre les appareils IoT et le cloud. Les appareils Sigfox sont généralement équipés de modules de communication sans fil qui transmettent des données sous forme de messages courts (jusqu'à 12 octets).

**LoRa<sup>2</sup> :** Il s'agit d'une technologie qui permet aux objets connectés d'échanger des données de faible taille en bas débit. La technologie LoRa permet notamment la transmission de données de faible volume par des capteurs fixes. Le principe est de transmettre des données par liaison hertzienne depuis des capteurs à faible puissance d'émission, fonctionnant sur batterie pour 5 à 10 ans [4] . Elle utilise l'étalement de spectre pour transmettre ses informations. Mais au lieu d'utiliser des codes d'étalement (CDMA), elle utilise une méthode appelée Chirp Spread Spectrum. Plusieurs transmissions dans le même canal. La conséquence sur le

---

2. LoRa :Long Range

spectre est aussi la même : cela provoque un étalement du spectre. LoRa utilise le protocole LoRawan.

**Le Chirp :** Le signal émis par la modulation LoRa est un symbole Son nom (Chirp) vient du fait que ce symbole est utilisé dans la technologie Radar (Chirp : Compressed High Intensity Radar Pulse). La figure 1.3 présente un Chirp LoRa dans le domaine fréquentiel.

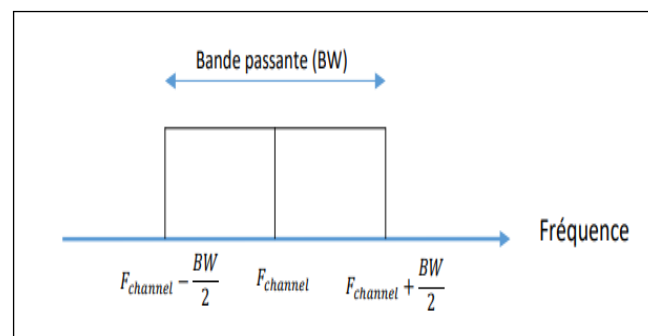


FIGURE 1.3 – Occupation spectrale d’une transmission LoRa

**Les paramètres de LoRa :** La consommation réelle d’un système LoRa dépend de plusieurs paramètres :

\_ **Le Spreading Factor :** En LoRa, le temps d’émission de chaque symbole ( $T_{\text{symbole}}$ ) dépend du Spreading Factor utilisé. Plus le SF est élevé, plus le temps d’émission sera long . Le Spreading Factor (SF) est lié au mécanisme qui adapte la puissance d’émission et la vitesse de transmission aux conditions du réseau dans lesquelles se trouve l’objet. Le SF permet ainsi une meilleure gestion des ressources radio et une optimisation de la consommation énergétique. Le débit et les performances radio varient en fonction du facteur d’étalement utilisé. Le facteur d’étalement le plus grand (SF 12) permet la plus longue portée du signal (meilleure sensibilité de la technologie) et donc potentiellement la

meilleure. Cela permet en outre d'améliorer la géolocalisation de l'objet par le réseau car le signal émis atteindra le plus grand nombre de gateways [5] .

\_ **Influence du Duty-cycle sur le débit** : Le duty cycle fait référence à la proportion de temps pendant laquelle un dispositif LoRa peut transmettre des données par rapport à la durée totale de l'intervalle de temps. IL est souvent réglementée par les organismes de régulation en matière de fréquences radio pour éviter les interférences avec d'autres dispositifs utilisant les mêmes fréquences

\_ **Le taux de transmission (TX)** : Le taux de transmission (TX) fait référence à la vitesse à laquelle les données sont transmises sur le réseau LoRaWAN. Plus le taux de transmission est élevé, plus la quantité de données transmises est grande et plus la durée de transmission est courte. Cependant, un taux de transmission élevé peut entraîner une diminution de la portée du signal, ce qui peut limiter la couverture du réseau.

\_ **La puissance consommée en veille entre deux transmissions** : Pour optimiser la consommation d'énergie en veille et répondre aux besoins de l'application. Il est important de prendre en compte ces facteurs lors de la conception et de la configuration du dispositif LoRa.

\_ **Stop mode** : Dans ce mode, la plupart des composants et fonctionnalités du dispositif sont complètement arrêtés pour minimiser la consommation d'énergie en veille. Le processeur principal, les interfaces de communication, les capteurs et d'autres circuits non essentiels sont désactivés. Le dispositif entre dans un état de basse consommation d'énergie où il ne peut pas réagir rapidement aux

événements externes. Le réveil du mode Stop nécessite généralement une réinitialisation complète du dispositif.

\_ Sleep mode : Dans le mode Sleep, certains composants et fonctionnalités du dispositif sont mis en veille pour réduire la consommation d'énergie, mais une partie du dispositif reste actif pour permettre un réveil rapide. Le processeur principal peut être mis en veille, certaines interfaces de communication peuvent être désactivées, et certains capteurs ou circuits de puissance peuvent être éteints. Cela permet de réduire la consommation d'énergie tout en conservant la capacité de réagir rapidement à des événements prédéfinis, tels que des interruptions ou des minuteries.

\_ Standby mode : Le mode Standby se situe entre le mode Sleep et le mode Stop. Dans ce mode, certains composants essentiels du dispositif restent partiellement actifs pour détecter et réagir rapidement à des événements prédéfinis, tout en réduisant la consommation d'énergie globalement. Les interfaces de communication, les capteurs ou les circuits de détection d'événements restent actifs. Cela permet au dispositif de répondre rapidement à des stimuli spécifiques, tout en conservant une consommation d'énergie réduite par rapport au mode actif.

### **Le protocole LoRaWAN :**

LoRaWAN, en tant qu'extension du protocole LoRa, assure la connectivité sécurisée entre le dispositif et un serveur, permettant ainsi la transmission de données vers l'utilisateur final. L'architecture complète du réseau, comprenant les dispositifs, les passerelles (Gateways) et les serveurs, ainsi que le format de

la trame utilisé par un dispositif LoRaWAN, garantit une transmission sécurisée des données vers un serveur LoRaWAN.

**Architecture :** Les réseaux LoRaWAN sont déployés selon une topologie en et se compose des éléments suivants comme montre la figure 1.4 :

\_ **Les devices ou les terminaux** (peuvent être des capteurs ou actionneurs)  
. Ils disposent d'un transceiver radio LoRa pour atteindre les Gateways.

\_ **Gateways :** Elles écoutent sur tous les canaux et sur tous les Spreading Factor en même temps. Lorsqu'une trame LoRa est reçue, elle transmet son contenu par Internet au serveur LoRaWAN réseau qui a été préalablement renseigné dans la Gateway. D'un côté, la Gateway reçoit un signal modulé en LoRa, et de l'autre côté, elle est connectée à internet via toutes les connexions possibles : 3G, 4G, 5G, Ethernet, LTE-M...

\_ **Serveur de réseau :** un logiciel fonctionnant sur un serveur qui gère tout le réseau. Le Serveur de réseau ou Network Server reçoit les messages transmis par les Gateways et supprime les doublons (plusieurs Gateways peuvent recevoir le même message et le transmettre au même Network Server). Ensuite, il authentifie le message grâce à une clé AES 128 bits appelée NwkSKey (Network Session Key). Nous parlons bien ici d'authentification et non pas de chiffrement.

\_ **Serveurs d'application :** Le serveur d'application reçoit des messages chiffrés d'un Serveur de réseau. Le chiffrement et le déchiffrement se font grâce à une clé AES 128 bits appelée AppSKey (Application Session Key).

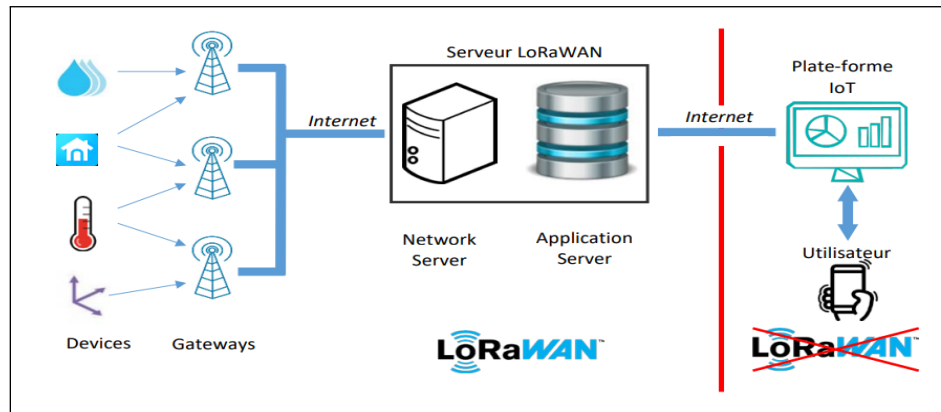


FIGURE 1.4 – Architecture de LoraWAN

La couche protocolaire de réseau LoRa est organisée en trois couches principales :

- \_ **La couche physique** : Elle définit la modulation radio utilisée pour transmettre les données, ainsi que les paramètres de transmission tels que la puissance de transmission, la largeur de bande et le débit binaire. La couche physique de LoRa utilise une modulation à spectre étalé chirp pour transmettre des signaux sur de longues distances.
- \_ **La couche liaison de données** : Elle gère la communication entre les noeuds LoRa et les passerelles LoRa. Cette couche est responsable de la gestion de l'accès au canal radio partagé, de la gestion des collisions et de l'assurance qualité de la transmission des données. Elle est basée sur le protocole LoRaWAN qui utilise la technique d'évitement de collisions ALOHA.
- \_ **La couche application** : Cette couche est responsable de la gestion des données d'application qui sont échangées entre les noeuds finaux et les serveurs de réseau, elle utilise un format de message appelé « payload » pour encapsuler les données de l'application. Le payload peut contenir différents types de données, tels que des capteurs, des commandes, des notifications, etc. Ces données sont envoyées sous forme de trames radio LoRa, qui sont transmises à travers le

réseau LoRaWAN jusqu'à leur destination. La figure 1.5 montre Les couches protocolaires LoRa.

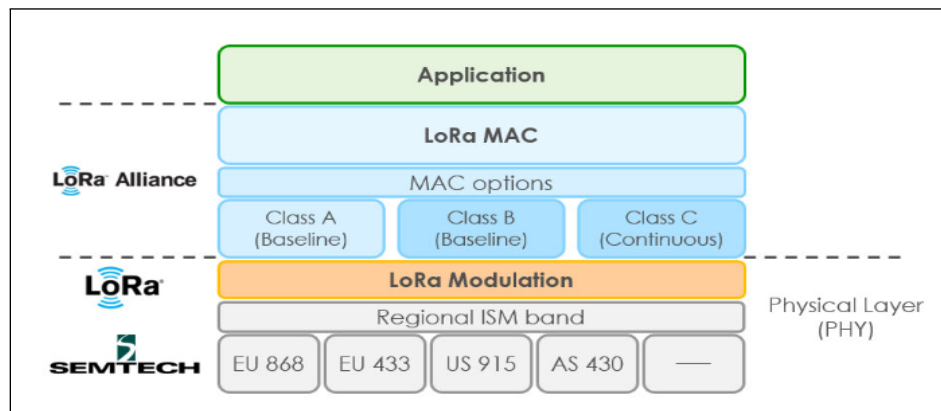


FIGURE 1.5 – Les couches protocolaires LoRa

Le protocole LoRaWAN catégorise les équipements de réseau en trois types :

- Classe A : Cette classe est la plus courante et est implémentée par défaut sur tous les terminaux LoRaWAN. Elle offre un compromis entre la durée de vie de la batterie et la latence des communications. Les terminaux de classe A écoutent les canaux de réception pendant une période de temps définie après chaque transmission. Si le terminal ne reçoit pas de réponse de la passerelle, il retourne en mode de veille pour économiser l'énergie.
- Classe B : Cette classe offre une meilleure synchronisation entre les terminaux et les passerelles, permettant des communications plus rapides et plus précises. Les terminaux de classe B écoutent également des plages de temps prédéfinies appelées "fenêtres de réception" à des intervalles réguliers, ce qui permet aux passerelles de leur envoyer des données à des moments précis. Cela améliore la latence des communications, mais réduit la durée de vie de la batterie. la figure 1.6 montre les classes LoRaWAN

- Classe C : Cette classe est la plus gourmande en énergie, mais offre la latence la plus faible. Les terminaux de classe C sont constamment en mode d'écoute, ce qui leur permet de recevoir des données à tout moment, même pendant une transmission. Cela permet une communication en temps réel, mais réduit considérablement la durée de vie de la batterie. Cette classe est donc adaptée aux applications nécessitant une réponse immédiate, telles que les systèmes d'alarme.

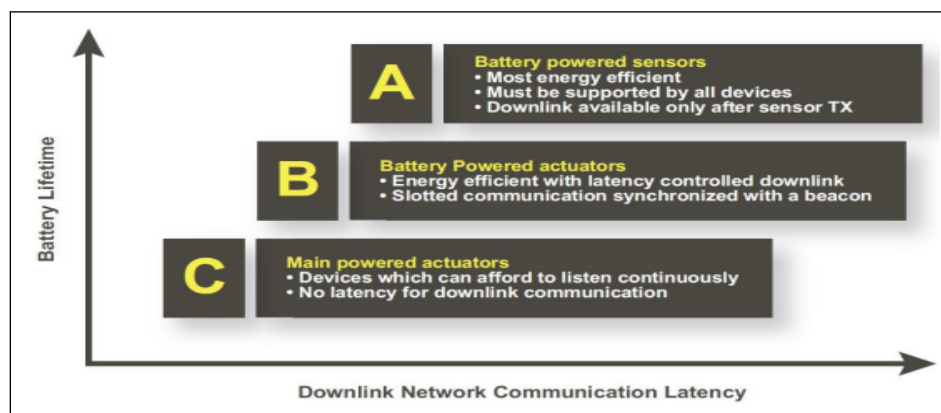


FIGURE 1.6 – Les classes LoRaWAN

### La sécurité de LoraWAN :

LoRaWAN spécifie un certain nombre de clés de sécurité : NwkSKey, AppSKey et AppKey. Toutes les clés ont une longueur de 128 bits. L'algorithme utilisé est AES-128, similaire à l'algorithme utilisé dans la norme 802.15.4

- **La Network Session Key (NwkSKey) :** La Network Session Key (NwkSKey) est utilisée pour l'authentification entre le Device LoRaWAN et le Network Server. Afin de réaliser cette authentification, un champ MIC<sup>3</sup> est ajouté à la trame. Le MIC dépend des données transmises cryptées et de la NwkSKey.

3. MIC :Message Integrity Control



- **Application Session Key (AppSKey) :** L'Application Session Key (AppSKey) est utilisée pour chiffrer les données de l'utilisateur sur le Device LoRaWAN. Les données seront déchiffrées sur l'Application Server. Il s'agit d'un chiffrement symétrique : l'AppSKey du Device doit être la même que celle stockée sur l'Application Server.
- Il n'y a aucun moyen pour la Gateway et le Network Server de comprendre la valeur réelle des données de l'utilisateur. Le canal est sécurisé (confidentiel).
- **Combinaison de l'authentification et du chiffrement :** la figure 2.3 montre la combinaison entre l'authentification et le chiffrement.

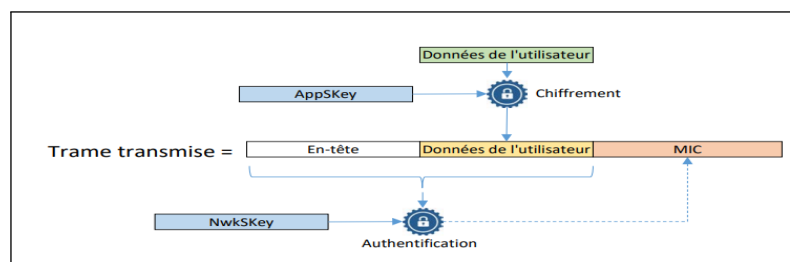


FIGURE 1.7 – Combinaison de l'authentification et du chiffrement

**Activation des Devices LoRaWAN : ABP et OTAA :** En LoRaWAN, les trois éléments essentiels à la communication sont le DevAddr pour l'identification du Device, ainsi que deux clés : la NwkSKey pour l'authentification et la AppSKey pour le chiffrement. Deux méthodes sont possibles pour fournir ces informations à la fois sur le Device et sur le serveur LoRaWAN :

- **ABP<sup>4</sup> :** En ABP, toutes les informations nécessaires à la communication sont déjà connues par le Device, le Network Server et l'Application Server

4. ABP : Activation By Personalization

— **OTAA** <sup>5</sup> : le DevAddr, l'AppSKey et le NwkSKey seront générés lors d'une procédure de Join. Pour réaliser cette procédure, le Device LoRaWAN doit être configuré avec les éléments suivants :

- DevEUI
- AppEUI/JoinEUI
- AppKey

Tous les éléments nommés "EUI" <sup>6</sup> sont toujours uniques et ont une taille de 8 octets. Les clés sont toujours sur 128 bits (16 octets).

## **1.7 Protocole de communication : MQTT**

### **1.7.1 Description**

MQTT <sup>7</sup> est un protocole de communication léger et distribué qui permet la transmission de messages entre des appareils connectés à Internet des objets ou à d'autres réseaux. MQTT est souvent utilisé dans les applications IoT pour la collecte de données, la surveillance à distance et la commande d'appareils.

### **1.7.2 Architecture**

L'architecture de MQTT est basée sur un modèle de publication/abonnement (publish/subscribe) où les appareils connectés peuvent envoyer des messages à des canaux appelés "topics", et les autres appareils peuvent s'abonner à

---

5. OTAA : Over The Air Activation

6. EUI :Extended Unique Identifier

7. MQTT :Message Queuing Telemetry Transport

ces topics pour recevoir les messages.

Le modèle de publication/abonnement est composé de trois éléments principaux : le broker, les clients publish et les clients subscribe.

- **Broker** : Le broker MQTT est un serveur qui gère la communication entre les clients publish et les clients subscribe. Le broker reçoit les messages publiés par les clients publish et les transmet aux clients subscribe qui sont abonnés aux mêmes topics.
- **Clients publish** : Les clients publish sont des appareils qui envoient des messages à des topics spécifiques. Les messages peuvent contenir des données telles que des informations de capteurs, des alertes ou des commandes à exécuter.
- **Clients subscribe** : Les clients subscribe sont des appareils qui s'abonnent à des topics spécifiques pour recevoir les messages publiés sur ces topics. Les messages peuvent être utilisés pour afficher des informations, déclencher des actions ou mettre à jour des données.

Les clients MQTT peuvent être des appareils physiques tels que des capteurs, des actionneurs ou des passerelles IoT, ou des applications logicielles qui utilisent le protocole MQTT pour communiquer avec des appareils IoT. La figure 2.4 montre l'architecture de MQTT

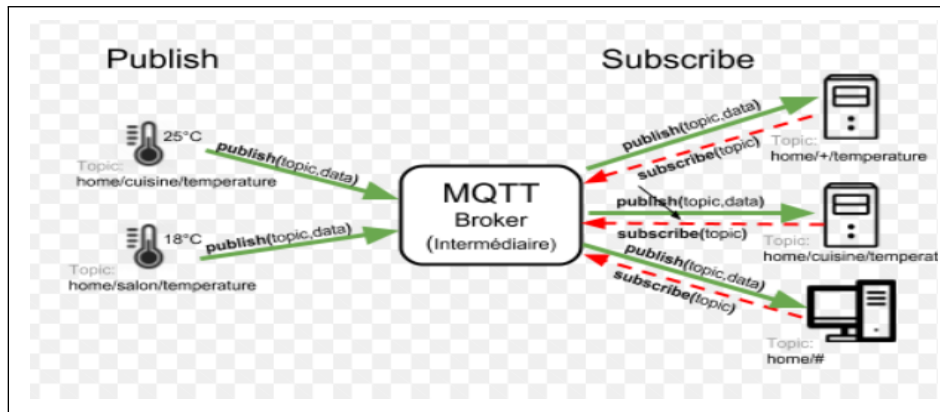


FIGURE 1.8 – Architecture de mqtt

## Conclusion

Dans ce chapitre, Nous avons présenté l'organisme d'accueil ainsi que la problématique et certains concepts fondamentaux nécessaires à la réalisation du projet tels que : les réseaux à longue portée, les réseaux à courte portée, les réseaux cellulaires et non cellulaires etc .. Le chapitre suivant sera consacré à l'analyse des besoins et Conception qui sera indispensable pour comprendre mieux la méthodologie de travail.

---

# Analyse des besoins et Conception

## Sommaire

---

<b>2.1</b>	<b>Étude de l'existant . . . . .</b>	<b>21</b>
2.1.1	Critique de l'existant . . . . .	22
2.1.2	Solution proposée . . . . .	23
<b>2.2</b>	<b>Spécification des besoins . . . . .</b>	<b>23</b>
2.2.1	Besoins fonctionnels . . . . .	23
2.2.2	Besoins non fonctionnels . . . . .	24
<b>2.3</b>	<b>Identification des acteurs et des cas d'utilisation . . . . .</b>	<b>25</b>
2.3.1	Les acteurs . . . . .	25
2.3.2	Diagramme des cas d'utilisation global . . . . .	26
2.3.3	Diagramme de cas d'utilisation gestion des composteurs . . . . .	27
2.3.4	Diagramme de cas d'utilisation gestion des projets de composteurs . . . . .	28
2.3.5	Diagramme de cas d'utilisation gestion des clients . . . . .	29
2.3.6	Diagramme de cas d'utilisation gestion des noeuds . . . . .	30
<b>2.4</b>	<b>Architecture générale de la solution proposée . . . . .</b>	<b>31</b>
<b>2.5</b>	<b>Conception détaillée . . . . .</b>	<b>31</b>
2.5.1	Vue statique . . . . .	32
2.5.2	Vue denamique . . . . .	33

---

## **Introduction**

Dans ce chapitre, nous décrirons d'abord les besoins fonctionnels et non fonctionnels de notre sujet, puis nous allons présenter en détail quelques diagrammes pour donner un aperçu clair du comportement de notre solution.

### **2.1 Étude de l'existant**

L'étude de l'existant dans le domaine du contrôle du processus de compostage du produit organique révèle qu'il existe déjà plusieurs systèmes de contrôle et de surveillance qui ont été développés pour améliorer l'efficacité et la qualité du compost produit. Parmi ces systèmes existants :

- monty mobile co propose une solution pour étudier la façon de protéger la qualité du compost .
- La société française WeCoTech a développé une solution de monitoring en temps réel pour surveiller le processus de compostage. Le système est équipé de capteurs qui mesurent la température, l'humidité et le taux d'oxygène dans le compost, afin de garantir des conditions optimales pour la décomposition des déchets.
- La start-up américaine WISerg a conçu un système de compostage intelligent pour collecter les déchets alimentaires dans les supermarchés et les restaurants. Les déchets sont ensuite traités dans une unité de compostage équipée de capteurs qui surveillent le processus de décomposition et ajustent les paramètres pour maintenir des conditions optimales.

- L'Université de Californie à Davis a mis en place un système de compostage de déchets verts équipé de capteurs pour surveiller la température, l'humidité et le pH du compost. Les données sont utilisées pour ajuster les paramètres et améliorer l'efficacité du processus.

### **2.1.1 Critique de l'existant**

La critique de l'existant dans le domaine du contrôle du processus de compostage du produit organique met en évidence certaines limites et faiblesses des systèmes de contrôle existants.

- Les coûts d'installation et de maintenance d'un système de surveillance de processus de compostage peuvent être très élevés. Cela peut être un obstacle pour les gouvernements et les organisations qui ne disposent pas d'un budget suffisant pour financer un tel projet
- La consommation d'énergie, le facteur le plus important dans les projets de l'Internet des objets . La majorité des solutions utilisent les cartes arduino ou d'autre carte qui consomment d'énergie et ça applique un problème.
- Les systèmes de contrôle existants peuvent ne pas être suffisamment précis pour fournir une surveillance complète du processus de compostage. Par exemple, les capteurs de température peuvent être affectés par des facteurs externes tels que la météo ou la lumière du soleil, ce qui peut affecter la fiabilité des données collectées.

### **2.1.2 Solution proposée**

Suite à l'étude que nous avons menée sur la solution existante et au regard des inconvénients recensés plus haut, il est primordial de proposer une solution qui pourra mieux répondre aux besoins actuels. Nous avons proposé un système basé sur l'IoT, notre système est un noeud capteur Lora Low-cost qui mesure les paramètres climatiques permettant le contrôle de température, humidité avec précisions. Nous utilisons le réseau LPWAN, non-cellulaire, long porté lora.

Ses caractéristiques peuvent ensuite être utilisées pour déclencher des alertes pour signaler un risque de perte de qualité de compost. Notre solution consiste à concevoir une application web et un dashboard contenant les courbes des mesures et une carte map de la forêt visée. Cela facilite aux utilisateurs de visualiser les caractéristiques qu'ils veulent. Aussi une application mobile pour recevoir les alertes.

## **2.2 Spécification des besoins**

Dans cette partie, nous présenterons les besoins fonctionnels et non fonctionnels de notre système.

### **2.2.1 Besoins fonctionnels**

Les besoins fonctionnels ou besoin métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait.



Notre système de surveillance de processus de compostage se répartit en deux modules fondamentaux :

- Module de système embarqué et IoT.
- Module d'application mobile.

Les besoins fonctionnels concernant le système embarqué et IoT sont liés à la carte utilisée se résument-en :

- L'alimentation en énergie pour pouvoir actionner les capteurs.
- La récupération des valeurs des différents capteurs, les paramètres climatiques (température , humidité).

Les besoins fonctionnels concernant l'application mobile sont :

- Informer la personne en cas d'élévation anormale de l'un des facteurs et envoyer à l'application .
- Connaitre la localisation du compost .
- gère les projets

### **2.2.2 Besoins non fonctionnels**

Dans cette nouvelle partie, nous allons traiter les besoins non fonctionnels de notre solution. Les besoins non fonctionnels concernant le système embarqué et IoT :

- Fiabilité : Le système doit être fiable et fournir des résultats précis et cohérents en tout temps.
- Performance : le système doit être capable de traiter rapidement les données collectées et d'envoyer des alertes en temps réel.

- Évolutivité : le système doit être évolutif et être capable de gérer des compost de différentes localisation .

Les besoins non fonctionnels concernant l'application web :

- Sécurité : l'application web doit être sécurisée pour protéger les données sensibles des utilisateurs et prévenir les attaques malveillantes.
- Maintenance : l'application web doit être facile à maintenir et à mettre à jour pour garantir sa stabilité et sa disponibilité à long terme.
- Facilité d'utilisation Le système doit avoir une interface conviviale et pratique .
- Temps de latence : Le temps d'exécution de l'application doit être rapide .

## **2.3 Identification des acteurs et des cas d'utilisation**

Dans cette partie nous présentons les acteurs ainsi que les diagrammes .

### **2.3.1 Les acteurs**

Administrateur : Cet acteur fait les fonctions suivantes

- S'authentifier : L'application doit garantir aux administrateurs l'accès à la base de donnée en faisant remplir un champ identifiant et un champ mot de passe.
- gérer les composteurs :
  - Ajouter de nouveaux composteurs
  - Supprimer des composteurs
  - Modifier les données des composteurs

- Consulter la liste des composteurs

composteur

- S'authentifier L'application doit garantir aux superviseurs l'accès à l'application en faisant remplir un champ pseudo et un champ mot de passe.
- gérer les projets de supervision :
  - Ajouter un nouveau projet de supervision
  - Supprimer un projet de supervision
  - Consulter un projet de supervision
- gérer les clients
  - Ajouter un nouveau client
  - Consulter un client
- gérer les noeuds :
  - Ajouter un nouveau noeud
  - Supprimer un noeud

client(greener) :fermier/citoyens : consulte son projet.

### 2.3.2 Diagramme des cas d'utilisation global

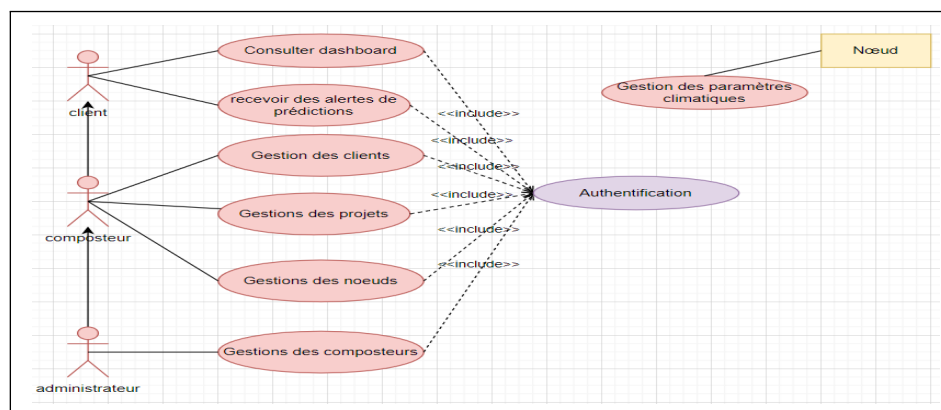
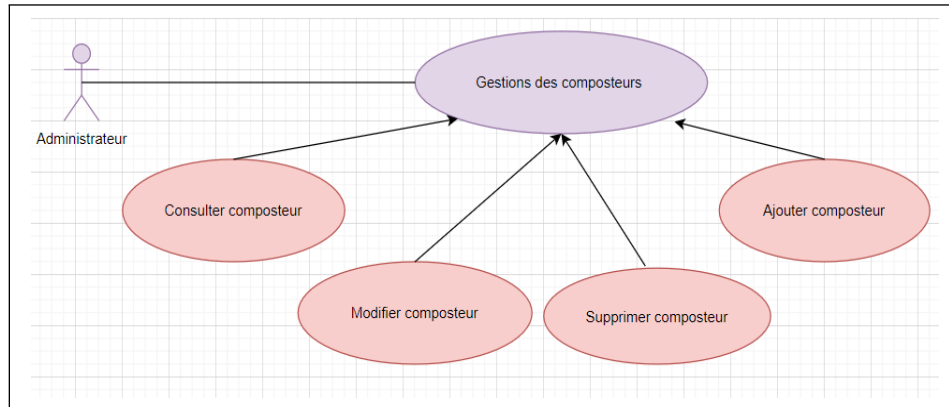


FIGURE 2.1 – Diagramme de cas d'utilisation globale

### 2.3.3 Diagramme de cas d'utilisation gestion des composteurs

La figure 2.2 montre le diagramme de cas d'utilisation gestion des composteurs.



**FIGURE 2.2 – Diagramme de cas d'utilisation gestion des composteurs**

Le tableau 2.1 représente la description textuelle du cas d'utilisation "gestion des superviseurs"

<b>Titre</b>	Gestion des composteurs
<b>Acteurs</b>	Administrateur
<b>Pré condition</b>	Administrateur authentifié
<b>Post condition</b>	Action de gestion effectuée sur le composteur : composteur créé ou supprimé ou modifié
<b>Scénario principal</b>	La possibilité de : 1- Ajout d'un nouveau composteur 2- Modification des données d'un composteur 3- Suppression d'un composteur 4- Consultation d'un composteur
<b>Exception</b>	1.1- Erreur de saisie lors de l'ajout 2.2- Erreur de saisie lors de la modification

**TABLE 2.1 – Cas d'utilisation : gestion des composteurs**

### 2.3.4 Diagramme de cas d'utilisation gestion des projets de composteurs

La figure 2.4 explique le cas d'utilisation "gestion des projets de composteurs".

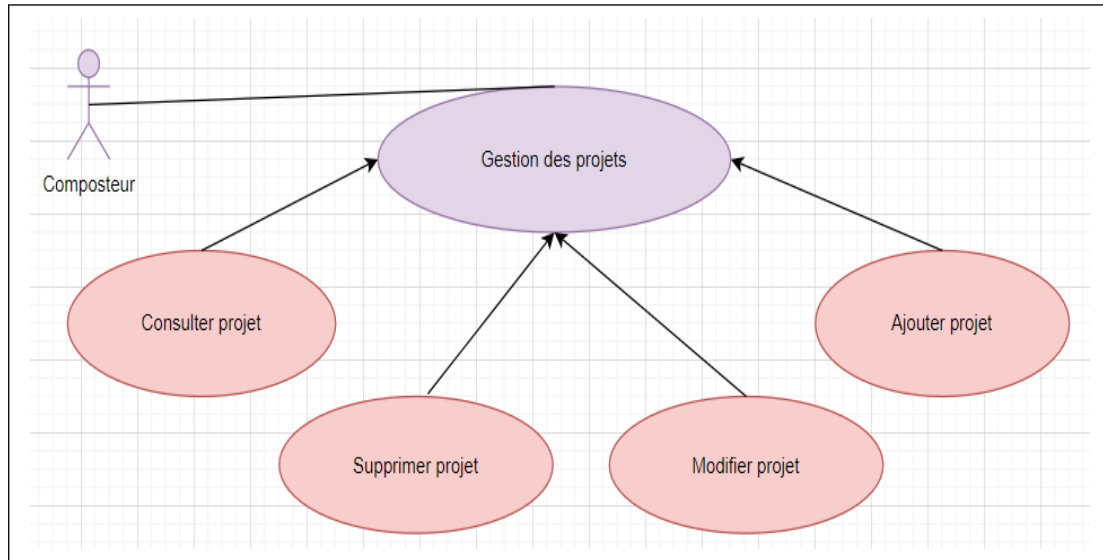


FIGURE 2.3 – Diagramme de cas d'utilisation gestion des projets

Le tableau 2.2 représente la description textuelle du cas d'utilisation "gestion des projets de composteurs".

<b>Titre</b>	Gestion des projets
Acteurs	Administrateur , composteur
Pré condition	Administrateur ou composteur authentifié
Post condition	Action de gestion effectuée sur le projet : projet créé ou supprimé ou modifié
Scénario principal	La possibilité de : 1- Ajout d'un nouveau projet 2- Modification d'un projet 3- Suppression d'un projet 4- Consultation d'un projet
Exception	1.1- Erreur de saisie lors de l'ajout 2.2- Erreur de saisie lors de la modification

TABLE 2.2 – Cas d'utilisation : gestion des projets des composteurs

### 2.3.5 Diagramme de cas d'utilisation gestion des clients

La figure 2.5 explique le cas d'utilisation "gestion des clients".

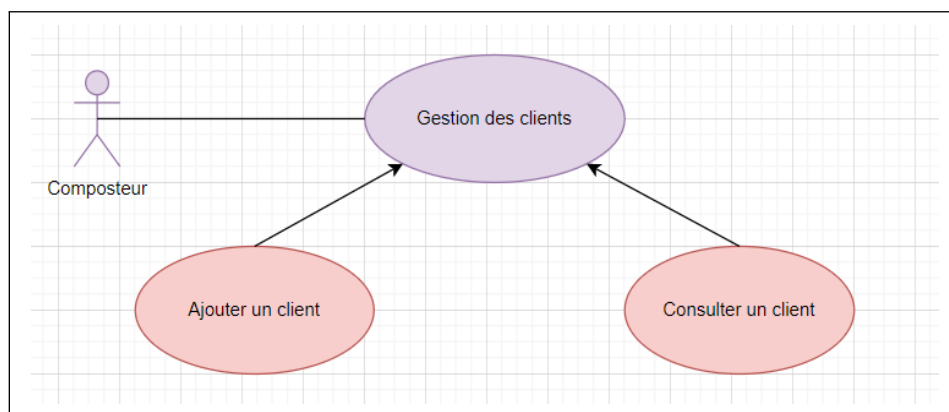


FIGURE 2.4 – Diagramme de cas d'utilisation gestion des clients

Le tableau 2.3 représente la description textuelle du cas d'utilisation "gestion des clients".

<b>Titre</b>	Ajouter un client(greener)
Acteurs	Administrateur , composteur
Pré condition	Administrateur ou composteur authentifié
Post condition	Client créé
Scénario principal	1- Le composteur remplit les champs nécessaires 2- Il confirme
Exception	1.1- Erreur de saisie lors 2.1- Client existe déjà

TABLE 2.3 – Cas d'utilisation : Ajouter un client

### 2.3.6 Diagramme de cas d'utilisation gestion des noeuds

La figure 2.6 explique le cas d'utilisation "gestion des noeuds".

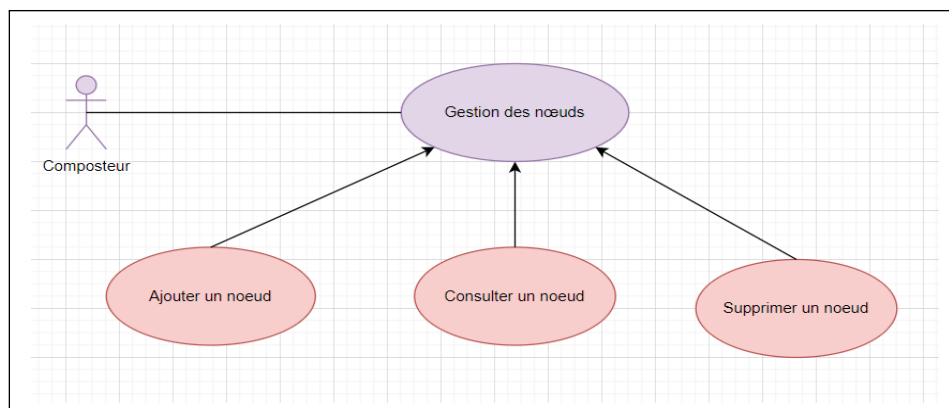


FIGURE 2.5 – Diagramme de cas d'utilisation gestion des noeuds

Le tableau 2.3 représente la description textuelle du cas d'utilisation "gestion des clients".

<b>Titre</b>	Consulter un noeud
Acteurs	Administrateur , composteur
Pré condition	Administrateur ou composteur authentifié
Post condition	Noeud consulté
Scénario principal	1- En cliquant sue le bouton "visualize" le composteur peut visualiser le noeud qu'il a créé.
Exception	

TABLE 2.4 – Cas d'utilisation : gestion des noeuds

## 2.4 Architecture générale de la solution proposée

Dans cette section, nous présentons l'architecture générale de notre solution qui nous permet de décrire de façon globale et détailler le fonctionnement de notre système. La figure 2.7 montre l'architecture générale de notre solution.

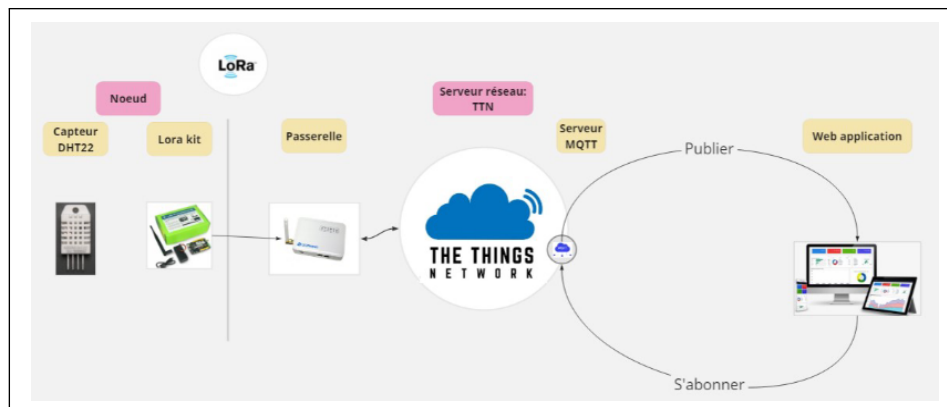


FIGURE 2.6 – Architecture générale de la solution proposée

## 2.5 Conception détaillée

En se basant sur les informations précédentes, nous avons eu une vision plus concrète des besoins de notre application. Nous détaillons les différents



éléments de conception, à savoir les diagrammes de séquences, et le diagramme de classe.

## 2.5.1 Vue statique

### 2.5.1.1 Diagramme de classe

Le diagramme de classe correspond à la représentation de la structure statique du système. Il représente les différentes entités intervenantes et les relations entre elles. La figure 2.7 suivante représente le diagramme de classe global :

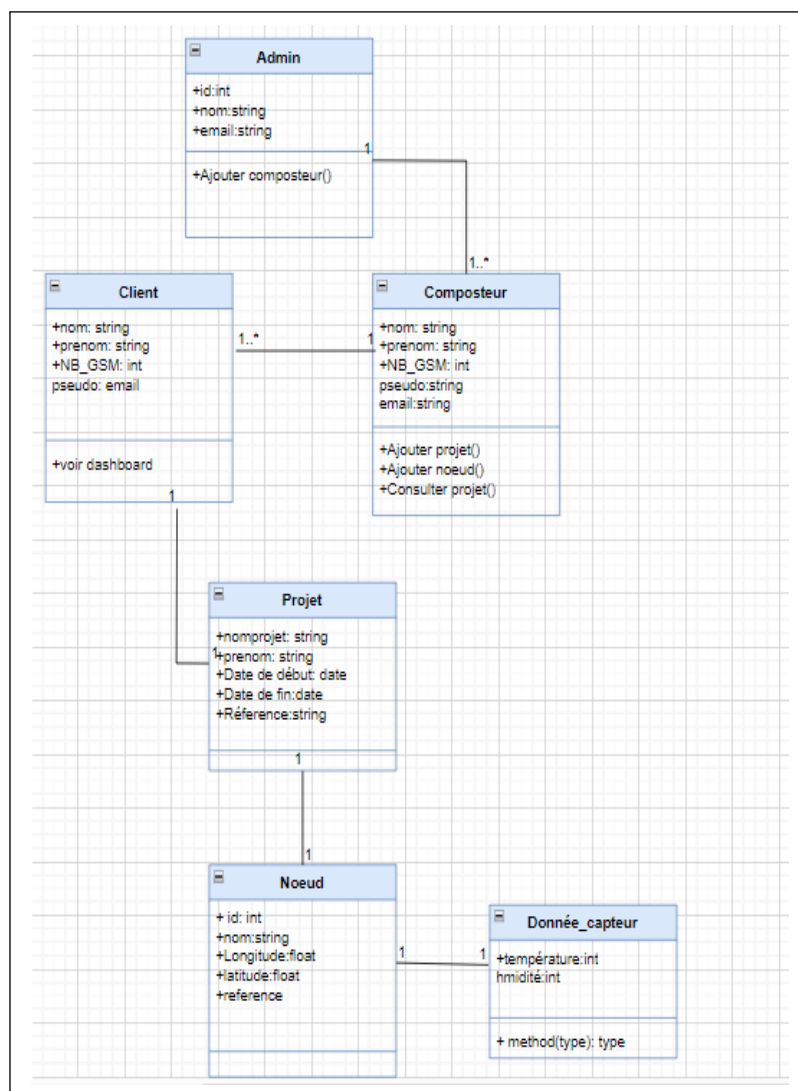


FIGURE 2.7 – Diagramme de classe

## 2.5.2 Vue denamique

### 2.5.2.1 Diagramme de séquence

- Diagramme de séquence : "S'authentifier"

La figure 2.8 illustre le diagramme de séquence "s'authentifier"

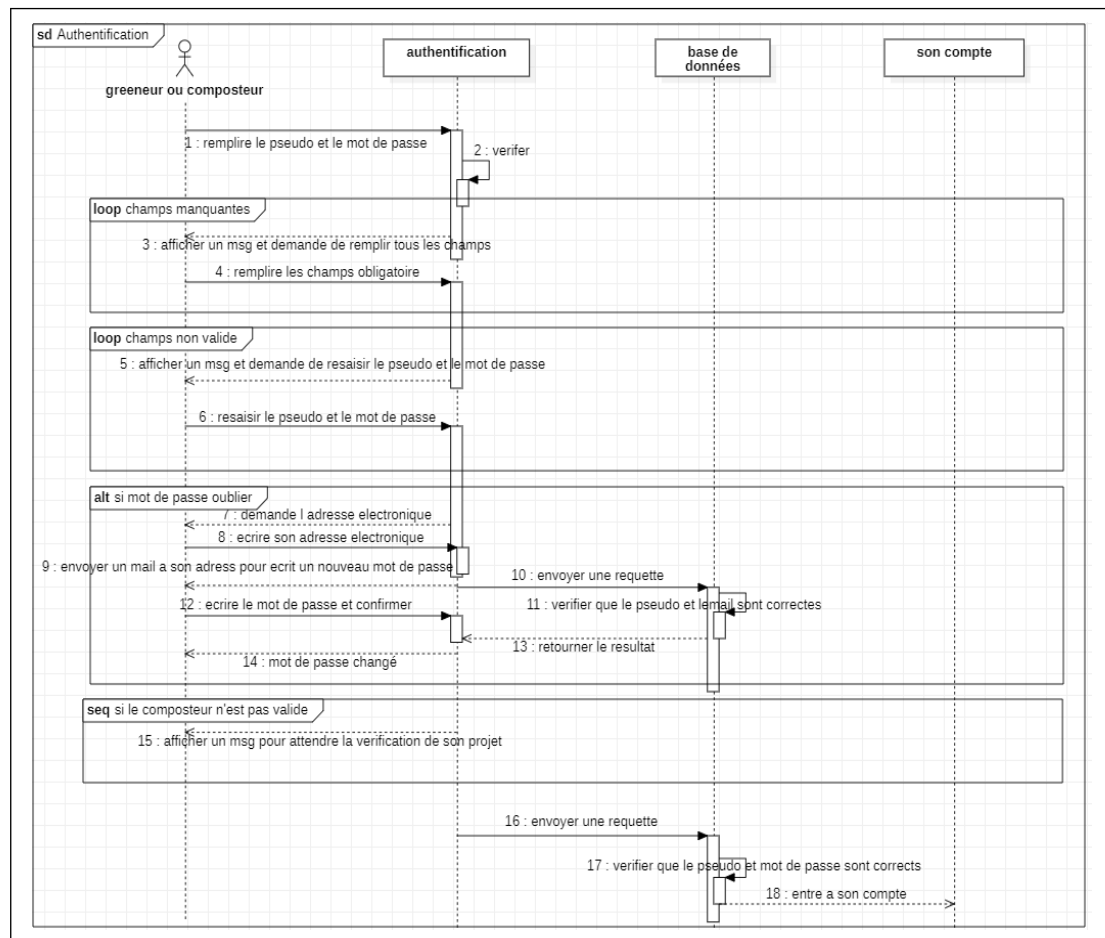


FIGURE 2.8 – Diagramme de séquence : "S'authentifier"

## Conclusion

Dans cette partie, nous avons entré dans les détails de conception dans notre solution en présentant le diagramme de cas d'utilisation. Le diagramme de séquence pour le scenario d'utilisation. Dans le chapitre suivant, nous

présentons l'environnement de travail nécessaire à la mise en œuvre des solutions proposées, ainsi que la mise en œuvre de ces solutions proposées, ainsi que l'implémentation et les résultats.

---

# Réalisation

## Sommaire

---

<b>3.1</b>	<b>Environnement de travail . . . . .</b>	<b>36</b>
3.1.1	Environnement matériel . . . . .	36
3.1.2	Environnement logiciel . . . . .	40
<b>3.2</b>	<b>Réalisation et développement de la solution IoT . . . . .</b>	<b>46</b>

---

# Introduction

Ce dernier chapitre est consacré à la description de la solution proposée et conçue dans le chapitre précédent. Pour cela : nous présenterons l'environnement matériel et logiciel utilisé pour la mise en place de l'infrastructure IoT de notre solution ainsi que dans la partie développement en fournissant également des captures d'écran de l'interface utilisateur et de la structure de code.

### 3.1 Environnement de travail

Dans cette partie, nous présentons les environnements matérielle et logiciels pour la mise en place de l'infrastructure IoT.

### 3.1.1 Environnement matériel

Pour la réalisation du partie IoT, nous utilisons la liste de matériels suivants :

- **kit de developpement wio-E5** : Le kit de développement LoRa-E5 STM32WLE5JC - LoRaWAN 868/915 MHz est une plateforme de développement conçue pour faciliter la création et le prototypage rapide d'applications IoT utilisant la technologie LoRaWAN. la figure 3.1 montre la Wio-E5 Development Kit



**FIGURE 3.1 – la Wio-E5 Development Kit**

Fabriqué par Seeed Studio, une entreprise spécialisée dans la conception et la fabrication de matériels et de logiciels pour l'IoT. Le kit de développement comprend :

- **une carte de développement LoRa-E5 STM32WLE5JC** équipée d'un microcontrôleur STM32WLE5JC intégré et d'un module LoRaWAN de haute performance permettant des communications longue portée à faible consommation d'énergie. La carte est compatible avec les réseaux LoRaWAN 868/915 MHz. la figure 4.2 montre la carte lora e5.

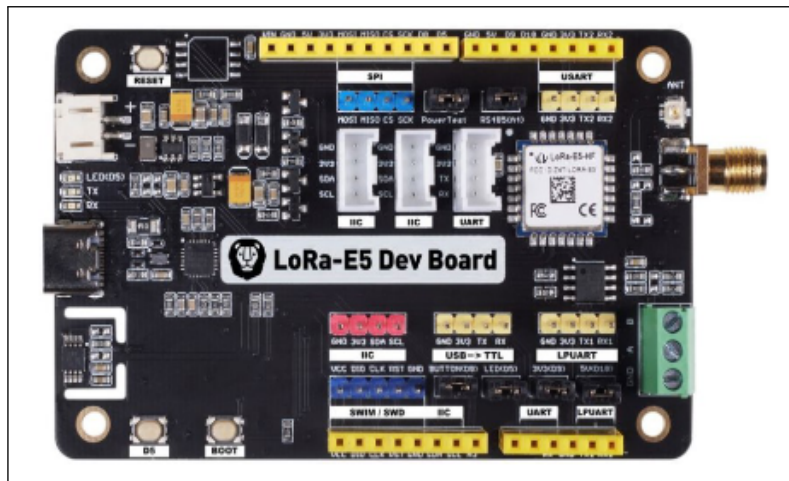


FIGURE 3.2 – La carte lora E5

- **une antenne LoRaWAN** : adaptée aux fréquences EU868 et US915
- **câble USB** : le cable USB est de type C de 20 cm pour la connexion à un ordinateur.
- **Porte-piles** : pour 2 piles AA 3V

*Motivation du choix :*

Voici quelques avantages à utiliser le kit de développement LoRa-E5 STM32WLE5JC - LoRaWAN 868/915 MHz :

- **Performances LoRa élevées :** La carte LoRa-E5 utilise une technologie LoRa de pointe qui permet des transmissions longue portée avec une faible consommation d'énergie. Cela en fait un choix idéal pour les applications IoT qui nécessitent une communication longue portée.
- **Puissant microcontrôleur STM32 :** La carte LoRa-E5 est basée sur le microcontrôleur STM32WLE5JC, qui offre des performances élevées et une faible consommation d'énergie. En fait, le microcontrôleur prend en charge plusieurs modes de sommeil, notamment le mode Stop, le mode Standby et le mode Hibernate, qui permettent de réduire considérablement la consommation d'énergie de la carte LoRa-E5 lorsque l'application est en veille.
- **Gateway :** Le Dragino Gateway LoRa est un équipement de passerelle qui permet de connecter des dispositifs LoRa à un réseau LoRaWAN. la figure 4.3 montre le dragino gateway



FIGURE 3.3 – Le Dragino gateway LoRa

Le Dragino Gateway LoRa prend en charge les fréquences de bande ISM (industrielle, scientifique et médicale) 433 MHz, 868 MHz et 915 MHz, et utilise le protocole LoRaWAN 1.0.2 pour la communication

avec les dispositifs LoRa. La passerelle dispose d'un module de communication sans fil LoRa concentrator, qui est capable de recevoir des données LoRa à partir de plusieurs dispositifs de nœud LoRa et de les transmettre à un serveur central LoRaWAN via une connexion Ethernet.

- **Capteur de température et d'humidité Grove - DHT22 :** Le capteur de température et d'humidité Grove - DHT22 est un module qui permet de mesurer à la fois la température et l'humidité relative de l'air ambiant. Ce capteur est basé sur le composant électronique DHT22 (également connu sous le nom d'AM2302), qui est un capteur de température et d'humidité numérique haute précision. La figure 4.4 montre la capteur DHT22 utilisée .

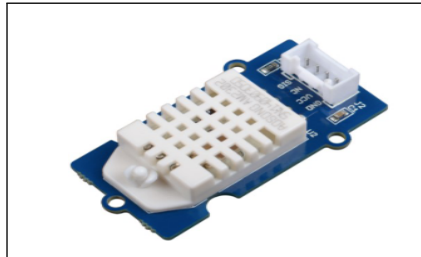


FIGURE 3.4 – Capteur de température et d'humidité Grove - DHT22

Pour la réalisation de l'application, nous utilisons :

**Pc msi avec les caractéristique suivants :**

- Microprocesseur : Intel core i7
- RAM : 8Go
- Système d'exploitation : Windows 10



### 3.1.2 Environnement logiciel

Nous avons utilisé une variété de logiciels et de bibliothèques pour réaliser les différentes tâches de partie IoT :


-  **Stm32CubeIDE** : STM32CubeIDE est un environnement de développement intégré (IDE) gratuit pour la programmation de microcontrôleurs STM32 de la société STMicroelectronics. Cet IDE est basé sur la plate-forme de développement Eclipse et offre une interface utilisateur conviviale pour développer, déboguer et déployer des applications sur les microcontrôleurs STM32. La figure 3.5 montre le logo Stm32CubeIDE



FIGURE 3.5 – Logo du Stm32CubeIDE



-  **Stm32CubeProgrammer** : STM32CubeProgrammer est un logiciel de programmation de microcontrôleurs STM32, développé par STMicroelectronics. Il est utilisé pour programmer et déboguer les microcontrôleurs STM32. La figure 3.6 montre le logo Stm32CubeProgrammer



FIGURE 3.6 – Logo du Stm32CubeProgrammer

- 

C embarqué est un langage de programmation couramment utilisé pour les systèmes embarqués, tels que les microcontrôleurs, les processeurs sans système d'exploitation et les systèmes en temps réel. Il offre une combinaison de faible surcharge, de fonctionnalités de bas niveau et de contrôle précis des ressources matérielles, ce qui le rend très adapté pour les applications en temps réel et à faible consommation d'énergie. La figure 3.7 montre Logo du langage de programmation C



FIGURE 3.7 – Logo du langage du programmation C

- The Thing Network :** The Thing Network fournit une infrastructure communautaire pour la connectivité IoT, permettant aux utilisateurs de déployer et de gérer leurs propres passerelles et appareils. Il fonctionne selon un modèle d'infrastructure partagée, où les utilisateurs fournissent des passerelles et partagent la couverture du réseau, créant ainsi un écosystème collaboratif.

En utilisant The Thing Network, les développeurs et les entreprises peuvent connecter à l'internet une large gamme de dispositifs IoT, tels que des capteurs, des actionneurs et des traqueurs, et collecter des données à partir de ces dispositifs. Ces données peuvent ensuite être utilisées pour diverses applications, notamment les villes intelligentes, l'agriculture, le suivi des actifs et la surveillance de l'environnement.

- **Python :** Python 3.11 est un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions. La figure 4.8 montre l'icone python



**FIGURE 3.8 – Logo de python**

*Motivation du choix :* Elle dispose les bibliothèques les plus riches parmi les autre langages de programmation dans la communications avec le réseau capteur et internet objet.

- **Django :** Django est un framework de développement web écrit en Python, il est simple d'accès, rapide et il fournit les outils nécessaires à sécuriser notre application, à gérer la structure de nos modèles (base de données).C'est un framework idéal pour réaliser un projet collaboratif avec la moin sécurité nécessaire et qui permet de gérer les bases

des données spatiales en utilisant le module GeoDjango. Il y a une plus grande liberté dans la gestion du code et en plus un système d'administration y est déjà pré-construite. La figure 4.9 montre logo du django .



FIGURE 3.9 – Logo de la framework django

- **PostgreSQL** PostgreSQL : est un système gratuit de gestion de base de données relationnelle et objet (SGBDRO) robuste et puissant, aux fonctionnalités riches et avancées. Il peut stocker plus de types de données que les types simples traditionnels (entiers, caractères, etc). La figure 4.10 montre le logo du Postgresql.



FIGURE 3.10 – Logo de Postgresql

**PostGIS version 3.2** : C'est le module spatial qui confère à PostgreSQL le statut de système de gestion de base de données relationnelle (SGDBR) spatial.

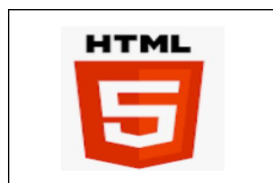
- **Leaflet** : Leaflet est la principale bibliothèque JavaScript open-source pour les cartes interactives adaptées aux téléphones portables. Avec un poids d'environ 42 Ko de JS, elle dispose de toutes les fonctionnalités

cartographiques dont la plupart des développeurs ont besoin. La figure 3.11 montre le logo du leaflet. La figure 3.10 montre le logo du Postgresql



**FIGURE 3.11 – Logo de leaflet**

- **Psycopg2** Psycopg est l'adaptateur de base de données PostgreSQL le plus populaire pour le langage de programmation Python. Ses principales caractéristiques sont l'implémentation complète de la spécification Python DB API 2.0 et la sécurité des threads (plusieurs threads peuvent partager la même connexion).
- **GDAL**<sup>8</sup> est une bibliothèque open source populaire pour la manipulation de données géospatiales. Elle permet de lire, d'écrire et de transformer de nombreux formats de données géospatiales, tels que les images satellites, les cartes vectorielles, les données lidar, etc.
- **HTML5**<sup>9</sup> : HTML signifie « HyperText Markup Language » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure. La figure 4.12 montre le logo Html



**FIGURE 3.12 – Logo HTML**

---

8. GDAL :Geospatial Data Abstraction Library

9. HTML :HyperText Markup Language

- **CSS**<sup>10</sup> :

Les CSS sont le code utilisé pour mettre en forme une page web. La figure 4.13 montre le logo de CSS.



FIGURE 3.13 – Logo CSS

- **JavaScript** Javascript désigne un langage de développement Web indépendant de HTML qui nous permet d'ajouter de l'animation et des fonctions dynamiques à nos pages web en accédant directement aux éléments HTML et les manipuler . La figure 4.14 montre le logo du javascript.



FIGURE 3.14 – Logo Javascript

- **vscode** : C'est un éditeur de code source gratuit et multi-plateforme développée par microsoft qui permet l'exécution des applications Web, bureautique et mobiles en utilisant une dizaine de langage de programmation.
- **Git** : C'est un éditeur de code source gratuit et multi-plateforme développée par Microsoft qui permet l'exécution des applications Web, bureautique et mobiles en utilisant une dizaine de langage de programmation.

---

10. CSS : Cascading Style Sheets



FIGURE 3.15 – Logo git

## 3.2 Réalisation et développement de la solution IoT

Le système de contrôle du processus de compostage des déchets en matières organiques a été développé pour optimiser et surveiller le processus de compostage, en garantissant des conditions idéales pour la décomposition des déchets. L'architecture globale du système comprend plusieurs composants clés qui interagissent entre eux pour assurer un compostage efficace.

### **La mise en oeuvre des noeuds :**

Pour la mise en œuvre des nœuds, le code est développé en utilisant les fichiers `main.c`, `lora_app.c`, `dht.h` et `dht.c`. Le fichier `main.c` sert de point d'entrée du programme, tandis que `lora_app.c` contient la logique de communication LoRaWAN avec TTN. Les fichiers `dht.h` et `dht.c` fournissent les fonctions nécessaires pour lire les données des capteurs DHT22.

Une fois le code développé, nous utilisons l'environnement de développement STM32CubeIDE pour compiler le code et l'environnement de développement STM32Cubeprogrammer pour programmer la carte STM32 avec l'aide de ST-Link. La carte STM32 est ensuite prête à être utilisée comme nœud LoRa dans le système de surveillance du compostage.

Lorsque le nœud est alimenté, il collecte en temps réel les données de température et d'humidité à l'aide du capteur DHT22. Ces données sont

préparées pour être transmises via LoRa en utilisant le microcontrôleur STM32WLE5JC. Le protocole LoRaWAN est utilisé pour envoyer les données du nœud à la passerelle LoRaWAN.

La passerelle Dragino Gateway LoRa reçoit immédiatement les données des capteurs de température et d'humidité via LoRaWAN et les transmet au réseau TTN. TTN agit comme une plateforme de connectivité en fournissant une infrastructure réseau LoRaWAN qui permet la communication bidirectionnelle entre les capteurs et la passerelle.

Pour garantir une authentification sécurisée entre le nœud LoRa et TTN, nous utilisons le mode d'activation OTAA. Les clés de sécurité générées, telles que DevAddr, AppKey et NwkSKey, sont intégrées dans le projet CubeIDE à l'aide du fichier de configuration "se-identity.h".

En parallèle, nous intégrons le protocole MQTT pour lier en temps réel les données collectées par les capteurs à une application web dédiée. Nous ajoutons une fonction de souscription MQTT à notre code en utilisant la bibliothèque Paho MQTT. Nous configurons les paramètres de connexion au broker MQTT pour établir une connexion sécurisée, et spécifions les topics de souscription pour recevoir les données des capteurs.

Lorsque nous recevons un message MQTT, nous extrayons les données pertinentes du payload JSON et les utilisons pour mettre à jour en temps réel les informations de température et d'humidité dans la base de données.

Nous utilisons un modèle de classe "Post" pour représenter les enregistrements de données, ce qui facilite leur intégration avec l'application web. À chaque



réception de données, nous créons une nouvelle instance de la classe "Post" et enregistrons les valeurs de température, d'humidité et de date de publication. Grâce à cette intégration MQTT, les données collectées par les capteurs sont transmises en temps réel à l'application web, ce qui permet de surveiller et de visualiser en direct le processus de compostage des déchets en matières organiques.

### **La réalisation du développement d'application web :**

La réalisation du développement de l'application web "Compost" a été effectuée en mettant en place différentes fonctionnalités et pages pour répondre aux besoins des utilisateurs. L'application se compose de plusieurs sections clés, dont la page d'accueil, la page "about us" et l'interface de connexion.

La page d'accueil présente une introduction sur les technologies utilisées et leur impact sur le réchauffement climatique. Elle vise à sensibiliser les utilisateurs sur l'importance du compostage et de la gestion des déchets organiques.

La page "about us" explique la mission de l'application et présente l'équipe de travail. Elle permet aux utilisateurs de comprendre les objectifs et les valeurs du projet.

L'interface de connexion propose deux options de connexion : en tant que client ou en tant que composteur. Lorsqu'un utilisateur choisit de se connecter en tant que composteur, une interface s'affiche avec une carte interactive. Cette carte contient des marqueurs représentant les emplacements des capteurs qui contrôlent la température et l'humidité du compost. En

cliquant sur un marqueur, une fenêtre contextuelle (popup) s'affiche, affichant les valeurs spécifiques de température et d'humidité de ce capteur, ainsi qu'un lien vers les détails du projet. En suivant ce lien, l'utilisateur peut accéder à toutes les informations spécifiques de ce projet, y compris les courbes d'humidité et de température.

L'interface du composteur comprend également une barre latérale (sidebar) avec deux cartes. En cliquant sur l'un des boutons de cette barre latérale, les étapes pour créer un projet s'affichent. Une fois toutes les étapes terminées, l'utilisateur peut placer un marqueur sur une autre interface qui s'ouvre, contenant une carte avec tous les marqueurs. L'utilisateur peut ajouter un marqueur et ajouter la référence du capteur dans la deuxième carte de la barre latérale. En cliquant sur un autre bouton, un tableau s'affiche, listant tous les projets créés. L'utilisateur peut consulter, supprimer ou modifier chaque projet à partir de ce tableau.

Lorsqu'un utilisateur se connecte en tant que client, il accède à une interface spécifique avec les détails du projet qui lui est assigné. Une alerte s'affiche dans des conditions spécifiques définies, fournissant des informations importantes ou des notifications aux clients.

L'administrateur est responsable de la création des comptes composteur en utilisant les informations fournies, ainsi que de la gestion générale de l'application.

Cette réalisation a impliqué l'utilisation des technologies Python, Django, PostgreSQL, Leaflet, Psycopg2, GDAL, HTML5, CSS et JavaScript. L'éditeur de code source utilisé était VSCode, et Git a été utilisé pour la gestion du

code source et des versions.

Grâce à cette conception et à ces fonctionnalités, l'application "Compost" offre une plateforme conviviale et pratique pour la gestion du compostage des déchets organiques, tant pour les composteurs que pour les clients, en contribuant à la réduction des déchets et à la préservation de l'environnement.

---

# CONCLUSION GÉNÉRALE

---

# BIBLIOGRAPHIE

[1]

---

# ANNEXES

## A.1 TITRE ANNEXE 1

La figure 1.1 présente le logo de Smart for green.



**FIGURE A.1 – Logo De Smart for Green**

## A.2 TITRE ANNEXE 2