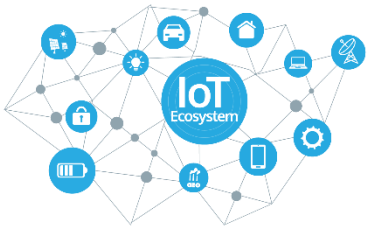
Parc national des Ecrins

Développer un réseau LoRaWAN pour l'Internet des Objets



LoRaWAN®

Juin 2021

GARCIN Baptiste

https://www.ecrins-parcnational.fr/

Préambule

Ce document a pour objectif de présenter un nouveau moyen de communication qu'est le LoRaWAN, et d'en comprendre le fonctionnement pour pouvoir par la suite développer son propre réseau.
Nous évoquerons la notion de **bits** et d'**octets** tout au long de ce document. Un octet équivaut à huit bits.

$$1\ octet\ (o) = 8\ bits\ (b)$$
$$1\ Go = 8\ Gb$$

De plus la bande passante, bandwidth chez nos amis anglophones, désigne dans le monde de l'informatique le débit binaire d'une voie de transmission. Elle représente la quantité d'informations pouvant être transmises simultanément sur une voie de transmission, et s'exprime en bits/seconde.
Aujourd'hui, le terme de bande passante est principalement utilisé dans la mesure de la qualité et des performances des accès à Internet à haut débit. En choisissant de **tester leur bande passante** en ligne, les internautes peuvent identifier très facilement la qualité de leur accès à Internet et la rapidité avec laquelle ils accèdent à des informations en ligne.

Glossaire

ABP : **A**ctivation **B**y **P**ersonnalisation

ADR : **A**daptive **D**ata **R**ate

AppEUI : **App**lication **E**xtended **U**nique **I**dentifier

AppKey : **App**lication **Key**

AppSKey : **App**lication **S**ession **Key**

BDD : **B**ase **d**e **D**onnées

BW : **B**and**W**idth

CHIRP : **C**ompressed **H**igh **I**ntensity **R**adar **P**ulse

CR : **C**oding **R**ate

CRC : **C**heck **R**edundancy **C**ycle

DevAddr : **Dev**ice **Addr**ess

DevEUI : **Dev**ice **E**xtended **U**nique **I**dentifier

IoT : **I**nternet **o**f **T**hings

JSON : **J**ava**S**cript **O**bject **N**otation

JoinEUI : **J**oin **E**xtended **U**nique **I**dentifier

LoRa : **Lo**ng **Ra**nge

LoRaWAN : **Lo**ng **Ra**nge **W**ide **A**rea **N**etwork

LTE-M : **L**ong **T**erm **E**volution Cat **M**1

MIC : **M**essage **I**ntegrity **C**ode

MQTT : **M**essage **Q**ueuning **T**elemetry **T**ransport

NwkSKey : **Nw**ork **S**ession **Key**

OTAA : **O**ver **T**he **A**ir **A**ctivation

PoE : **P**ower **o**ver **E**thernet

QoS : **Q**uality **o**f **S**ervice

RSSI : Received Signal Strength Indication

SDR : Software Digital Radio

SF : **S**preading **F**actor

SNR : Signal over Noise Ratio

TOA : Time On Air

TTN : **T**he **T**hings **N**etwork

TTS : **T**he **T**hings **S**tack

Table des matières

1. Internet of Things (IoT)

1.1. Définition

Le monde de l'IoT représente tous les capteurs connectés à Internet qui permettent d'obtenir diverses informations à distance tels que l'électro-ménager connecté par exemple. Cela permet notamment de pouvoir récolter des données en temps réel à plusieurs endroits différents sans devoir forcément s'y rendre pour chaque mesure. Avec le développement des appareils de domotique, l'Internet des objets est présent en permanence dans notre quotidien. Cette technologie est fleurissante et ne cesse de croître.

Comme nous le verrons par la suite, elle possède des avantages que les autres moyens de télécommunications n'ont pas : faible consommation d'énergie, longue portée, faible débit de données. Les technologies de communication cellulaires comme la 4G émettent rapidement mais consomment énormément en contrepartie.

Ces nouveaux moyens de communication sont appelés LPWAN pour **L**ong **P**ower **W**ide **A**rea **N**etwork (réseau étendu à basse consommation).

1.2. Intérêts

Comme dit précédemment, l'avantage majeur est la très faible consommation de ces capteurs qui leurs confèrent une autonomie allant jusqu'à dix ans, voire plus. Cela s'explique par l'émission de petits fichiers (entre 10 et 50 octets par envoi) et d'un faible débit d'émission. Toutefois, la portée est supérieure à tous les moyens de communication existant. La *figure 1* classe ces derniers en fonction de leur bande passante et de leur portée.

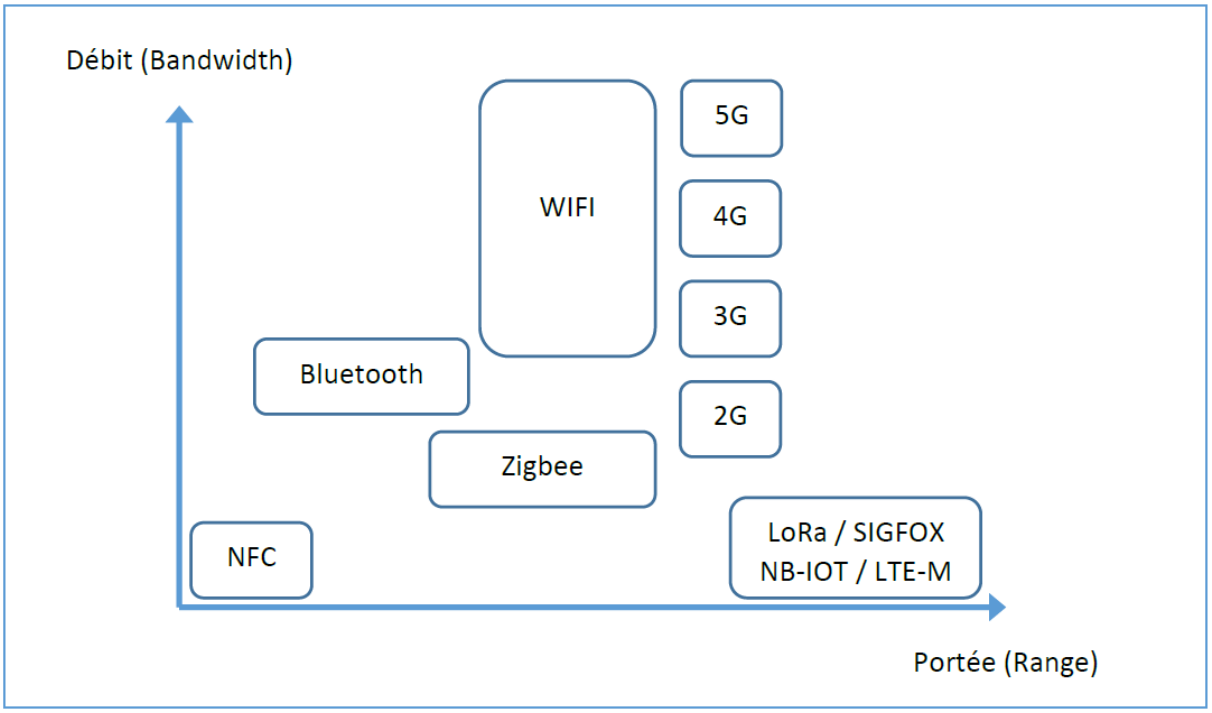


Figure 1 : Comparaison des protocoles utilisés dans l'IoT en fonction du débit et de la portée

1.3. Domaines d'applications

Mais puisque ce moyen de communication semble si performant, pourquoi ne pas remplacer la 4G par du LoRaWAN ? Tout simplement à cause du faible débit des LPWAN. En LoRaWAN, le débit moyen se trouve entre 40 octets et 2,5 ko/s, contrairement à la 4G avec ses 9,3 Mo à 19 Mo/s. Néanmoins, ce type de communication est très prisé dans certains cas.

- Bâtiment intelligent : capteur permettant de mesurer le taux de passages, de vérifier que les lumières soient éteintes, de surveiller la consommation d'eau ou d'électricité ainsi que le taux de CO2, de savoir quelle pièce est occupée etc.
- Agriculture : contrôler l'irrigation, la température des sols, suivre par GPS le bétail, surveiller le remplissage des silos et/ou des cuves etc.
- Ville intelligente : détecter l'occupation des places de parking par des véhicules, mesurer la qualité de l'air etc.

2. LoRaWAN

2.1. Présentation

Tout d'abord, il est important de différencier le terme « LoRa » avec celui de « LoRaWAN ».
Le protocole LoRaWAN régit la communication depuis le capteur jusqu'au serveur d'application, et la transmission LoRa (ou modulation LoRa) est un type de communication qui se limite au périmètre de communication radio bas débit (du capteur jusqu'à la passerelle).

La bande de fréquence légale utilisée en Europe est 863-870 MHz, plus couramment appelé bandes des 868 MHz. Mais cette dernière est interdite dans de nombreux pays hors Europe. Malgré tout, nous avons choisi cette bande du fait du développement en local.
Nous pouvons utilisés aussi la bande des 433 MHz (de 433,05 à 434,79 MHz) qui elle est utilisée à l'échelle mondiale mis à part quelques exceptions près.
D'autres canaux de fréquences dans le monde existent, mais nous ne nous attarderons pas sur ce sujet.

2.2. Structure du réseau

Le réseau est composé de capteurs qui envoient leur données à un serveur via une ou plusieurs passerelle(s). Nous pouvons ensuite récupérer ces données afin de concevoir une interface utilisateur (site web, application mobile etc.). La *figure 2* illustre cette structure.

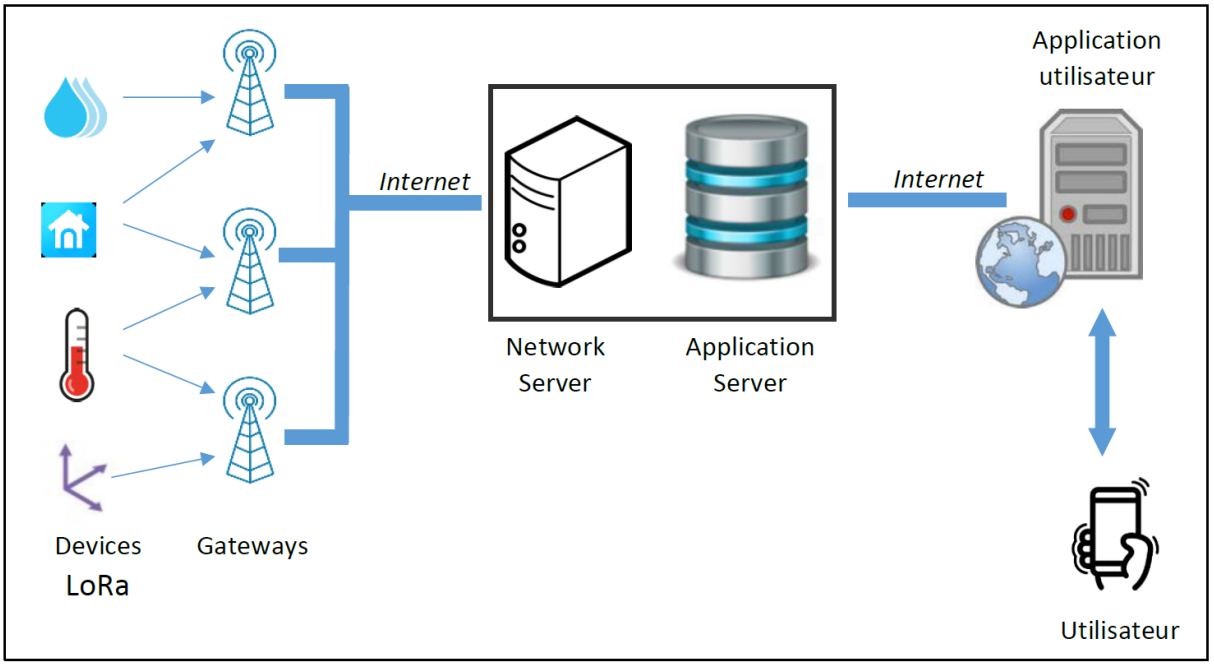


Figure 2 : Structure globale d'un réseau LORAWAN

2.2.1. Capteur, device, nœud

Des capteurs (aussi appelés « devices » ou « nœuds ») émettent des données chiffrées par ondes radios, soit sur la bande des 433 MHz soit sur celle des 868 MHz selon ce qu'il aura été choisi au préalable. Ils émettent dans une zone de couverture et n'adressent pas leur message à une antenne en particulier. Ils ont une faible consommation d'énergie, une faible taille, un faible coût et une faible puissance. Ils sont alimentés par une batterie interne et leur autonomie est parfois supérieure à dix ans selon les réglages.
Chaque capteur possède un identifiant différent les uns des autres appelé **DevEUI**, écrit sur 64 bits.

Indice de Protection (IP)

Afin de protéger les équipements de l'environnement extérieur plus ou moins de la poussière et/ou de l'eau, ils sont soumis à une norme d'étanchéité, appelé IP (Indice de Protection). L'IP maximum est l'IP69 (à prononcer six-neuf et non pas soixante-neuf). Le premier chiffre indique le taux d'herméticité, le second l'étanchéité. Un « x » à la place d'un chiffre signifie « aucune protection » pour l'herméticité et/ou l'étanchéité. Plus le chiffre est élevé, plus le critère de protection associé est efficace.

2.2.2. Passerelle, gateway

Concrètement, pour déployer un réseau LoRaWAN, les opérateurs publics ou privés installent des stations dotées d'antennes fabriquées par des équipementiers télécoms. Ce sont les passerelles.

Ces dernières, aussi appelées « Gateways », écoutent sur tous les canaux et reçoivent les informations des devices. Elles les réémettent ensuite par l'Internet vers un Network Server. Elles peuvent parfois émettre un message en provenance du Network Server et à destination des capteurs.

Elles nécessitent une alimentation électrique allant d'une dizaine de volts à 30 V, ainsi que d'une connexion Internet. Cette dernière peut se faire grâce à un câble Ethernet, par réseau téléphonique ou encore par PoE (**P**ower **o**ver **E**thernet). Le PoE se présente sous la forme d'un boîtier qui prend en entrée de l'énergie électrique ainsi que de l'Ethernet (prise RJ45). Un seul câble (RJ45) en sort pour alimenter la gateway en énergie et en Internet. Ainsi, comme il y a qu'un seul câble qui alimente la passerelle, le risque de perte d'étanchéité est diminué.

Chaque passerelle possède un identifiant différent des unes des autres et de ceux des capteurs appelé aussi **Gateway EUI**, écrit sur 64 bits.

2.2.3. Network Server

Le composant logiciel en charge d'établir le raccordement avec les objets et d'animer le cœur de réseau est le « Network Server ».

Le Network Server reçoit les messages des passerelles et supprime les doublons, car plusieurs passerelles peuvent avoir reçu le même message.

Si la méthode de chiffrement choisi est l'ABP, alors une clé AES 128 bits appelé **NwkSKey** (**N**et**w**or**k** **S**ession **Key**) est inscrite dans le device et les messages sont authentifiés grâce à cette dernière.

Si la méthode de chiffrement choisie est l'OTAA, alors la **NwkSKey** sera généré automatiquement entre le device et le Network Server.

Dans les deux cas, il ne s'agit pas de chiffrement mais d'authentification.

Nous verrons cette partie d'authentification et de « Key » plus en détail dans la partie 4.

2.2.4. Application Server

Souvent appelé « AppServer », il est souvent sur le même support physique que le Network Server. Il permet de dissocier les « applications » (différent d'une application de type utilisateur) les unes des autres. Chaque application enregistre des Devices LoRa qui auront le droit de stocker leurs données (Frame Payload). Les messages transmis à l'Application Server sont chiffrés grâce à une clé AES 128 bits appelée **AppSKey** (**App**lication **S**ession **Key**). Contrairement au **NwkSKey (N**et**w**or**k** **S**ession **Key**), il s'agit bien ici d'un chiffrement.

Nous verrons cette partie de chiffrement et de « Key » plus en détail dans la partie 4.

2.2.5. Interface utilisateur

Après la réception des données sur le serveur, nous pouvons les récupérer afin de créer des « dashboards » pour les visualiser. La figure 3 illustre un exemple de dashboard.

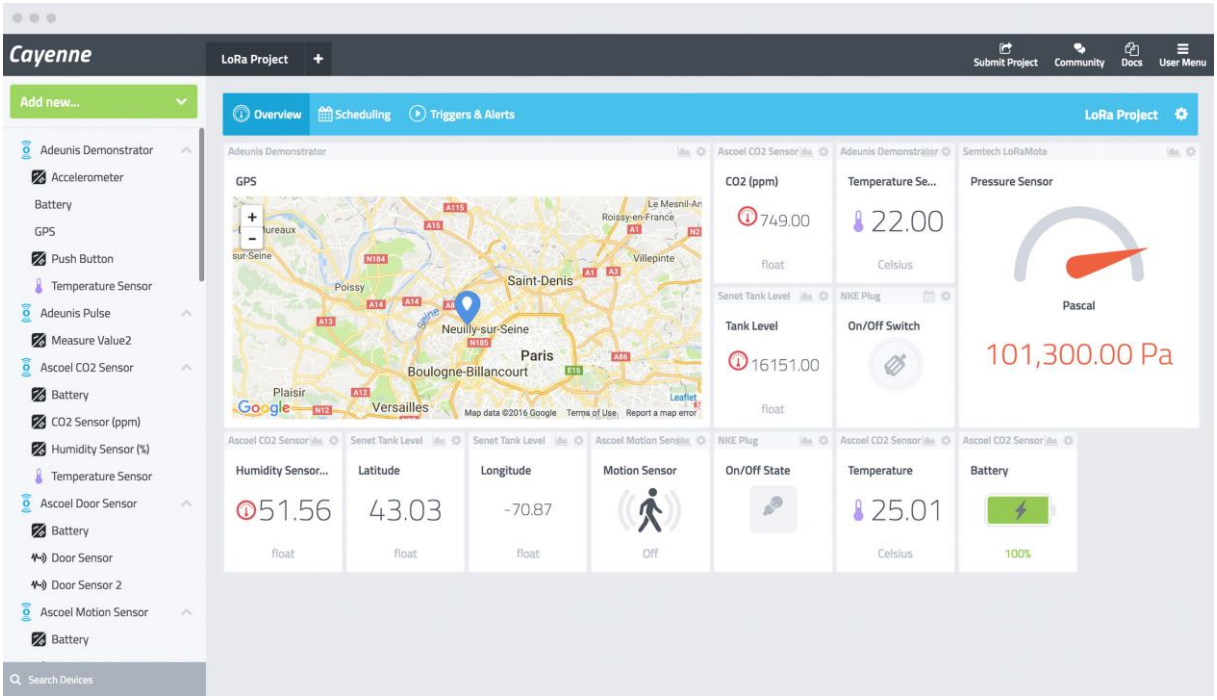


Figure 3 : Exemple de dashboard

2.3. Les différentes classes des capteurs

2.3.1. Classe A

Un device LoRa de classe A peut recevoir à l'unique condition d'avoir émis. Mais il ne peut recevoir que très peu de temps après l'émission. L'ordre de grandeur de la réception est d'environ de quelques secondes après l'émission.
Il n'est donc pas joignable facilement.

2.3.2. Classe B

Ils sont plus facilement joignables que les classes A. Des balises (ou Beacon) sont émises par la passerelle afin de se synchroniser avec le device et d'augmenter le nombre de phase d'écoute.
Ils consomment en revanche plus qu'un capteur de classe A.

2.3.3. Classe C

C'est la classe de device la plus énergivore. Elle est en permanence joignable.

3. Transmission des messages

3.1. La modulation LoRaWAN et le Spreading Factor

En LoRa, chaque groupe contenant un certain nombre de bits transmis est appelé **SF** (Spreading Factor) et est représenté par ce que nous appelons **symbole** ou **chirp** :

$$\boldsymbol{Nombre\ de\ bits\ transmis\ dans\ un\ symbole\ ou\ chirp = Spreading\ Factor}$$

Le spreading factor est noté SF, avec à la suite sa valeur comprise entre 7 à 12 (exemple : SF 8, spreading factor de 8).

Exemple : si la transmission utilise un SF 10, alors chaque chirp représente 10 bits. C'est-à-dire qu'à l'émission, les bits sont regroupés par paquet de 10 bits, puis chaque groupe est représenté par un chirp différent parmi les 2^{10} (2^{10} = 1024) formes de chirp possibles.

3.2. Trame complète en LoRaWAN

Lorsqu'un capteur ou une passerelle émet un message, celui-ci est divisé en plusieurs parties qui sont détaillées ci-dessous. Nous mettrons de côté le chiffrement des données par l'AppSKey que nous verrons plus tard (partie 4).

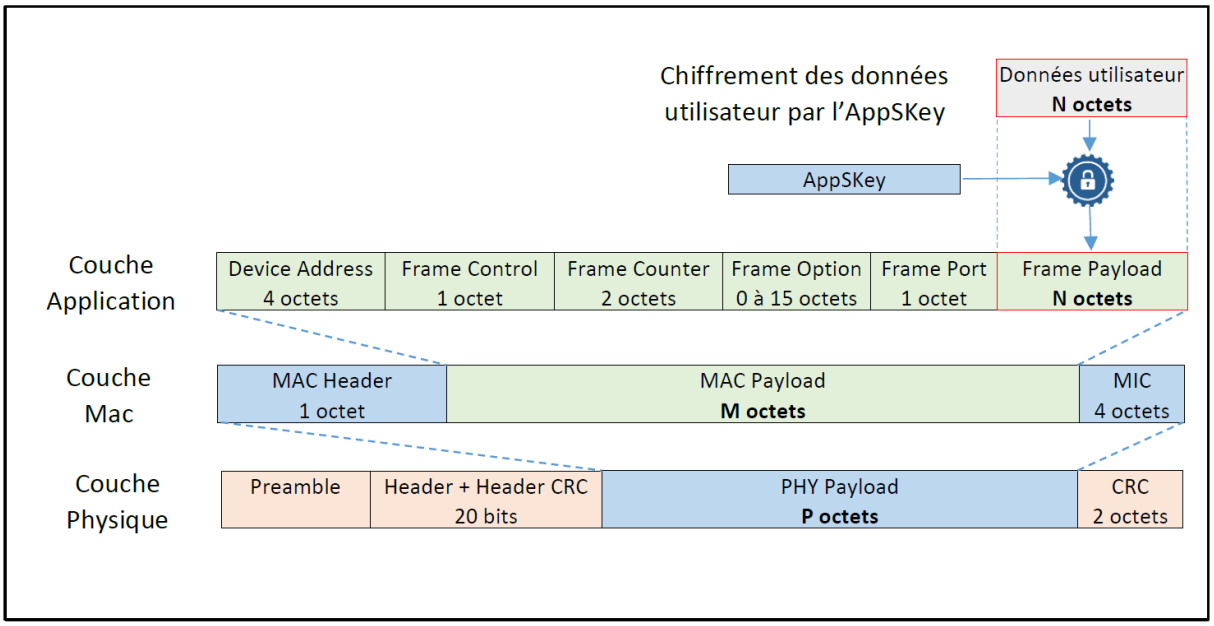


Figure 4 : LORAWAN complète par couche protocolaire

3.2.1. Couche Application

La couche Application héberge les données de l'utilisateur. Avant de les encapsuler, elles sont chiffrées avec l'AppSKey afin de sécuriser la transaction.

- L'ensemble **Frame Header** spécifie le :
  - Device Address : identifiant unique sur 32 bits au sein du réseau LoRaWAN. C'est l'adresse du device
  - Frame Control : octet de control
  - Frame Counter : 16 bits pour le compteur de trame
  - Frame Option : entre 0 et 120 bits pour les options

- **Le Frame Port** : dépend du type de l'application, c'est à l'utilisateur de choisir. Son rôle est d'informer si le message contient des commandes MAC (sa valeur est 0 dans ce cas), ou si les données sont spécifiées pour une application (il prend alors la valeur de son numéro).
- **Le Frame Payload** : contient les données chiffrées (avant le calcul du MIC) grâce à l'AppKey. Le nombre d'octet maximum pouvant être transmis en fonction de la bande-passante est décrit dans le tableau suivant.

Le DR (Data Rate) est une valeur associée à un couple débit-BW (SF-MHz).

Data Rate	Spreading Factor	Bandwidth	Max Frame Payload (nombre N)
DR 0	SF 12	125 KHz	51 octets
DR 1	SF 11	125 KHz	51 octets
DR 2	SF 10	125 KHz	51 octets
DR 3	SF 9	125 KHz	115 octets
DR 4	SF 8	125 KHz	242 octets
DR 5	SF 7	125 KHz	242 octets
DR 6	SF 7	250 KHz	242 octets

Tableau 1 : Taille maximum du Frame Payload en fonction du Data Rate

3.2.2. Couche Mac

- **MAC Header** : il est écrit sur 8 bits. Il indique la version de protocole et le type de message. Ce dernier est codé sur les 3 premiers bits qui sont décrits dans le tableau suivant :

Bits	Type de message	Signification
000	Join-Request	Le device demande à rejoindre le Server (OTAA)
001	Join-Accept	Le Server approuve l'envoi de données de ce device (OTAA)
010	Unconfirmed Data Up	Confirmation de réception d'un message montant et aucune confirmation de réception à envoyer
011	Unconfirmed Data Down	Confirmation d'envoi d'un message et aucune confirmation de réception demandée
100	Confirmed Data Up	Confirmation de réception d'un message montant et confirmation de réception envoyée
101	Confirmed Data Down	Confirmation d'envoi d'un message et une confirmation de réception est demandée
110	Re-Join-request	Le device demande à rejoindre le Server (OTAA et version 1.1 et plus)
111	Proprietary	Utilisé pour implémenter des formats de message non standard

Tableau 2 : Les types de messages transmis en LoRaWAN

- **MAC Payload** : contient tout le protocole applicatif.
- **MIC** : Message Integrity Code, pour l'authentification de la trame. Il est calculé à partir de la concaténation du Mac Header et du Mac Payload.

3.2.3. Couche Physique

- Le **Préambule** est représenté par 8 symboles + 4,25. Le temps du Préambule est donc de 12,25 $T_{symbole}$ *(voir partie 3.2 pour la définition d'un symbole)*.
  Contient 8 octets de 0x34 pour la bande passante de 868 MHz.
- L'en-tête (**Header optionnel**) est seulement présent dans le mode de transmission par défaut (explicite). Il est transmis avec un Coding Rate de 4/8. Il indique la taille des données, le Coding Rate pour le reste de la trame et il précise également si un CRC sera présent en fin de trame. C'est une division modulo 2 du Header dont le reste est le CRC.
- Le **PHY Payload** contient toutes les informations de la Couche LoRa MAC.
- Le **CRC (Cyclic Redundancy Check)** sert à la détection d'erreur de la trame LoRa.

3.3. Coding Rate, temps d'émission, débit et « Duty cycle » en LoRaWAN

3.3.1. Coding Rate

Le Coding Rate est un ratio qui augmente le nombre de bits à transmettre afin de réaliser de la détection/correction d'erreurs. Dans le cas d'un CR=4/8, il y aura réellement 8 bits transmis à chaque fois que 4 bits devront être transmis. Dans cet exemple, cela augmente la transmission du nombre de bits par deux. Le tableau 3 montre, en fonction du Coding Rate choisi, le ratio de bits à envoyer entre le message à transmettre et le message transmis.

Coding Rate	Coding Rate Ratio	Facteur multiplicatif
1	4/5	1,25
2	4/6	1,5
3	4/7	1,75
4	4/8	2

Tableau 3 : Influence du Coding Rate sur le nombre de bits ajoutés

3.3.2. Temps d'émission

Le temps d'émission est la durée totale qu'il faut au Device pour transmettre l'intégralité de la trame LoRaWAN. Il dépend du SF (Spreading Factor), et plus ce dernier est élevé, plus le temps d'émission d'un chirp sera long. Pour une même bande-passante, le temps d'émission d'un chirp en SF8 est deux fois plus long que pour envoyer ce même chirp en SF7.
C'est l'une des raisons pour lesquelles il faut privilégier un SF faible.

Le temps d'émission est inversement proportionnel à la bande-passante (ou bandwidth) :

$$T_{chirp} = \frac{2^{SF}}{Bande\text{-}passante}$$

Voici les temps d'émissions en fonction du Data Rate :

Data Rate	Spreading Factor	Temps d'émission d'un chirp	Nombre de bits par chirp
DR 6	SF7	512 ms	7
DR 5	SF 7	1,024 ms	7
DR 4	SF 8	2,048 ms	8
DR 3	SF 9	4,096 ms	9
DR 2	SF 10	8,192 ms	10
DR 1	SF 11	16,384 ms	11
DR 0	SF 12	32,768 ms	12

Tableau 4 : Temps d'émission pour une bande-passante de 125 Mz

Pour rappel, la bande-passante des DR 0 à DR 5 sont de 125 kHz et celle du DR 6 est de 250kHz.

Plus le SF sera faible, et donc plus le DR sera élevé, et plus le débit sera élevé.

3.3.3. Débit

Le débit des symboles est donc :

Avec : T_{chirp} la période d'émission du chirp

F_{chirp} la fréquence d'émission du chirp

$$\frac{1}{T_{chirp}} = F_{chirp} = \frac{Bande\text{-}passante}{2^{SF}}$$

Donc plus la bande-passante est élevée, plus le débit est élevé.

Le débit binaire (nombre de bits par seconde, bps) s'écrit :

$$Debit_{binaire} = SF.\frac{Bande\text{-}passante}{2^{SF}} = SF.\frac{1}{T_{chirp}} = SF.F_{chirp}$$

3.3.4. Duty cycle

La norme LoRaWAN impose qu'un device LoRa ne transmette pas plus de 1% du temps. Cela est appelé le **Duty Cycle**. Un Duty Cycle de 1% signifie que si j'émets pendant 1 (temps sans unité), je ne dois plus émettre pendant 99, quel que soit l'unité de temps utilisée.

$$Temps\ d'attente\ avant\ prochaine\ emission = 99 \times temps\ d'emission$$

Exemple : si mon device émet pendant 10 secondes, il devra attendre 990 secondes (16,5 minutes) avant d'avoir le droit d'émettre à nouveau.

4. Chiffrement, authentification et décryptage des données

4.1. Deux méthodes pour chiffrer les données

Deux méthodes de chiffrement existent. La première est l'OTAA (Over The Air Activation) et nous verrons que ce sera celle à privilégier. La seconde est l'ABP (Activation By Personnalisation). Dans les deux cas, nous devrons d'abord connaître **l'AppEUI** (unique identifiant pour l'AppServer), le **DevEUI** (similaire à une @MAC sur Ethernet), et **l'AppKey**. Puis nous devrons les enregistrer sur le serveur LoRaWAN. A noter que toute les clés qui seront évoquées seront codées sur 128 bits, et tous les EUI sont codés sur 64 bits.

A la suite de chaque processus, nous obtiendrons deux clés :

- AppSKey (Application Session Key) : permet de chiffrer les données de l'utilisateur dans le frame payload *(cf. figure 5)*.
- NwkSKey (Network Session Key) : sert au capteur afin qu'il puisse s'identifier au Network Server. En se basant sur les données chiffrées du Mac Header et du Mac Payload, un MIC est généré.

4.1.1. ABP (Activation By Personnalisation)

En plus de l'AppEUI, du DevEUI et de l'AppKey, il faut créer et enregistrer en dur dans le device et dans le serveur LoRaWAN le DevAddr, l'AppSKey et le NwkSKey. Cet ensemble de données est propre à chaque device.

Cette méthode est moins sécurisée car les clés de chiffrement sont stockées en dur dans le capteur contrairement à la méthode OTAA.

4.1.2. OTAA (Over The Air Activation)

On choisit une AppKey que nous stockons en dur dans le capteur et dans le serveur LoRaWAN.

Au démarrage du capteur, ce dernier envoie une join-request (chiffrées avec l'AppKey) au serveur LoRaWAN. Si ce dernier l'accepte, deux clés sont négociées puis générées entre les deux entités grâce à l'AppKey : l'AppSKey et la NwkSKey. Ces dernières sont gardées jusqu'à réinitialisation.

Il est important d'avoir une AppKey différente par device.

4.1.3. Bilan

Le tableau 4 ci-dessous résume ce qui a été énoncé précédemment.

Identifiant et clés	ABP	OTAA
AppEUI (64 bits)	Connu au préalable	Connu au préalable
DevEUI (64 bits)		
AppKey (128 bits)		
AppSKey (128 bits)		Généré automatiquement
NwkSKey (128 bits)		

Tableau 5 : Récapitulatif des caractéristiques de chaque clés/EUI en fonction du mode de chiffrement choisi

4.2. Processus complet : chiffrement puis authentification

4.2.1. Chiffrement avec l'AppSKey dans le capteur

Nous venons de voir que le device récolte des données qui sont ensuite chiffrées par l'AppSKey.

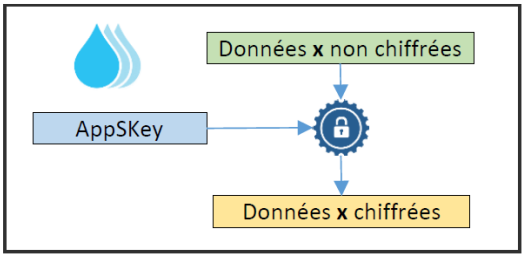


Figure 5 : Chiffrement des données dans le device par l'Application Session Key

4.2.2. Le MIC et authentification par le Network Serveur

Une fois les données chiffrées, la Network Session Key se base sur le Mac Header et le Mac Payload pour générer un MIC (Message Integrity Code) qui est implanté dans la trame. Celle-ci est ensuite envoyée et est captée par une passerelle qui transmet la trame chiffrée au Network Server. Ce dernier va utiliser sa propre NwkSKey ainsi que les données chiffrées pour encoder un nouveau MIC. Si les deux MIC correspondent, cela signifie que la trame reçue provient d'un capteur appartenant à notre réseau, et est ensuite envoyée à l'AppServer comme le montre la figure 6. Dans le cas contraire, la trame ne sera pas transmise et restera bloqué.

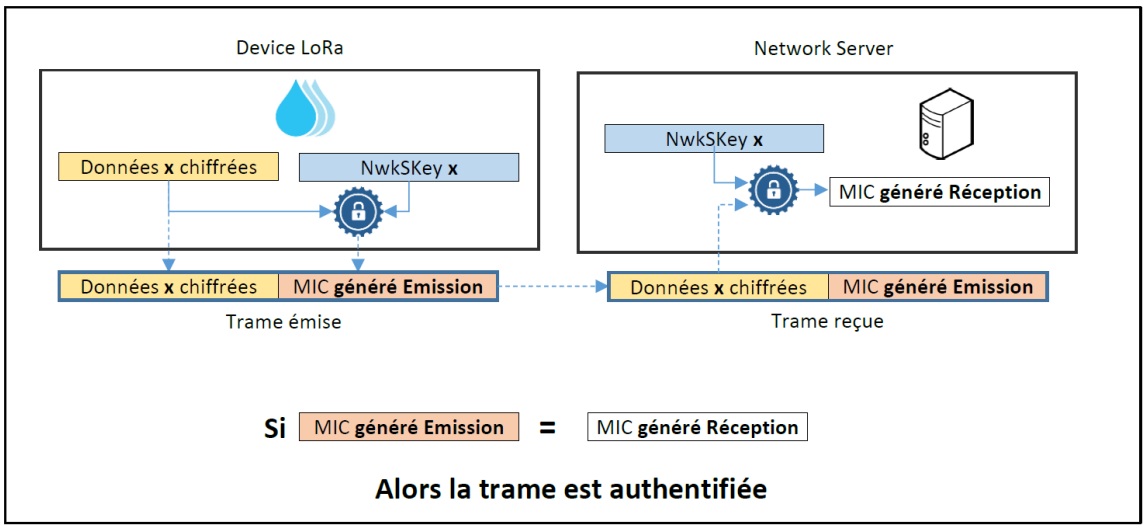


Figure 6 : Authentification d'un Device LoRa par le Network Server

4.2.3. Décryptage des données avec l'Application Server

L'AppServer possède lui aussi l'AppSKey qui lui permet de déchiffrer les données après authentification du device.

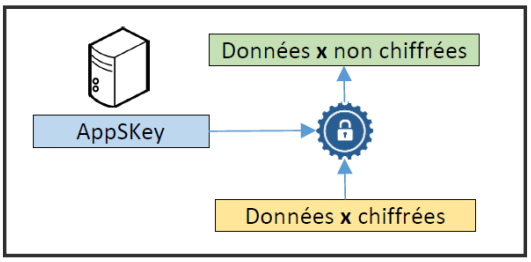


Figure 7 : Déchiffrement des données dans l'AppServer par l'Application Session Key

5. Choisir son serveur LoRaWAN

Il y a plusieurs façons de créer son serveur et/ou son réseau. Pour cela, de nombreux outils et acteurs permettent d'obtenir un réseau personnel.

- La première possibilité qui est la plus simple, est de passer par des opérateurs télécoms
- Une autre possibilité est de monter soi-même son propre réseau privé grâce à certains outils que nous développerons par la suite
- Une dernière possibilité est de mixer les deux : ce sont les réseaux dit dédiés

Nous nous pencherons plus précisément sur deux outils, ChirpStack (privé) et The Things Network (dédié). Leur installation et leur paramétrage sont détaillés en annexe respectivement C et D.

5.1. Orange et Objenious (Bouygues Telecom)

Cette possibilité est la plus simple car il n'est pas nécessaire d'avoir des connaissances en informatique. En effet, l'utilisateur a juste à connecter ses devices et à gérer ses applications côté utilisateur. Les Gateways, le Network Server et l'Application Server sont gérés par l'opérateur.

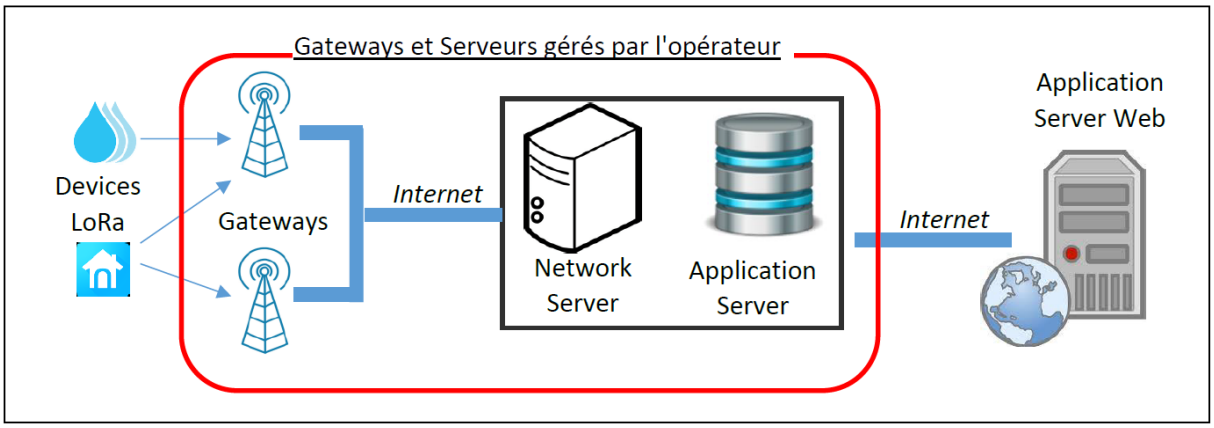


Figure 8 : Infrastrucutre d'un réseau LoRaWAN opéré

A titre d'exemple, voici en 2020 les abonnements proposés par Bouygues et Orange pour avoir accès à leur réseau LoRaWAN :

- Bouygues Objenious :
  - 144 messages par jour en Uplink
  - 6 messages par jour en Downlink
  - Le prix de l'abonnement par Device LoRa : 20 € TTC / capteur / an
- Orange :
  - Illimité en Uplink (dans le respect du Duty Cycle)
  - Prix de chaque message Uplink de 5 cts
  - Le prix de l'abonnement par Device LoRa est présenté dans le Tableau 6

36 mois	1€/mois
24 mois	1,2€/mois
12 mois	1,5€/mois
sans engagement	2€/mois

Tableau 6 : Tarification d'un abonnement par Device LoRaWAN chez Orange

5.2. Réseau privé – ChirpStack

Chacun est libre de réaliser son propre réseau privé en implémentant son propre réseau de Gateways, ainsi que sa propre infrastructure serveur pour communiquer avec ses Devices LoRa. L'entreprise, ou du moins l'utilisateur, doit prendre en charge la mise en place d'un Network Server et Application Server privés. Il existe des serveurs (Network et Application) mit à disposition en ligne, c'est le cas par exemple de ChirpStack ou TTS (**T**he **T**hings **S**tack). L'installation et le paramétrage de ChirpStack est détaillé en annexe C. Nous ne parlerons pas de TTS car ce dernier est payant.

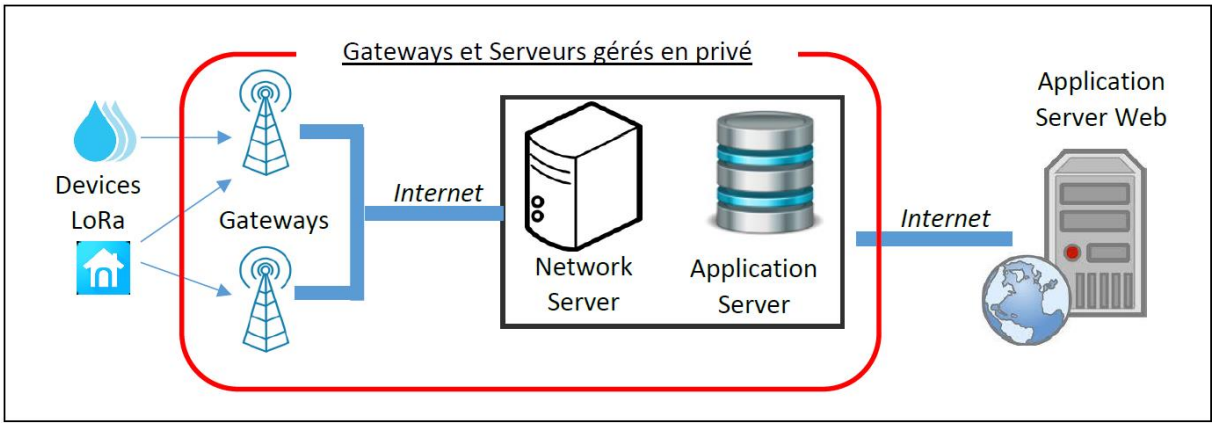


Figure 9 : Infrastructure d'un réseau LoRaWAN privé

ChirpStack

ChirpStack est un outil gratuit qu'il faut au préalable installer sur son serveur ou sur une Raspberry PI. Cette procédure est détaillée en annexe C. L'utilisation de cet outil se fait donc en local et est propre à chaque utilisateur. Il s'agit d'une interface personnalisée, où il n'y a aucune réglementation de débit dans le respect du Duty Cycle, aucune limitation dans le nombre de devices, de gateways et d'utilisateurs enregistrables. Cependant le support technique est très rare voire inexistant mis à part sur le forum de la communauté ChirpStack qui se fait uniquement en anglais.

Contrairement à The Things Network que nous verrons plus tard, l'utilisation de ChirpStack n'est pas à vocation d'étendre le réseau d'une communauté. Son usage reste dans un cadre privé.

5.3. Bilan comparatif

Nous pouvons résumer les avantages et les inconvénients de ces deux différents type de réseau dans le tableau ci-dessous.

	Réseau privé	Réseau opéré
Coût d'abonnement	Gratuit	Environ 1,5 € / mois par Device LoRa
Coût de l'infrastructure	Investissement important au début (Gateways et Serveurs)	Compris dans l'abonnement
Compétences requises	Demande des compétences en interne à l'entreprise pour la mise en place, et pour la maintenance.	Tout est géré par l'opérateur
Couverture	Optimisée suivant les besoins	Dépendante de l'opérateur choisi. Possibilité de Roaming avec l'international
Flux Uplink	Illimité dans le respect du Duty-Cycle	Limité suivant l'abonnement
Flux Downlink	Illimité dans le respect du Duty-Cycle	Limité en nombre ou payant à l'unité

Tableau 7 : Choix entre un réseau privé et un réseau opéré

5.4. Mixte des deux : dédié – The Things Network

Dans le cas où aucune des solutions extrêmes (réseau opéré ou réseau privé) ne convient, il est possible d'avoir une solution intermédiaire appelée réseau dédié. Elle a l'avantage de gérer la couverture réseau LoRa en utilisant ses propres Gateways, tout en confiant l'infrastructure du Serveur LoRaWAN à un prestataire afin de limiter les investissements et la maintenance.

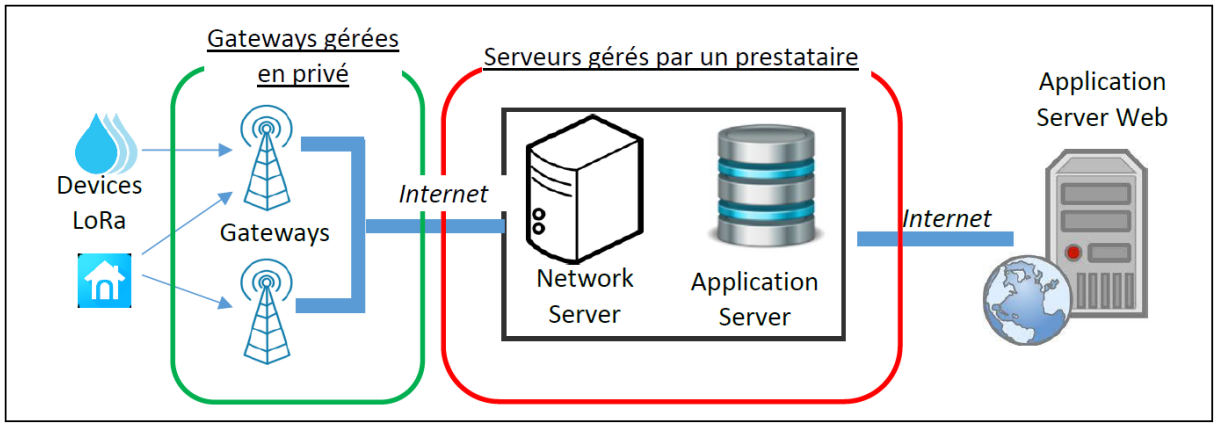


Figure 10 : Infrastructure d'un réseau LoRaWAN dédié

Network Server et Application Server en ligne : de très nombreux Network Server et d'Application Server sont proposés. Ce sont des services payants ou avec des contreparties :

- **Actility** : www.actility.com
- **Loriot** : www.loriot.io
- **ResIoT** : www.resiot.io
- **The Things Network** : www.thethingsnetwork.org

The Things Network (TTN)

Contrairement à ChirpStack, TTN ne peut pas être installé. Il s'agit uniquement d'une interface graphique sur le web. Pour s'y connecter, il faut au préalable créer un compte TTN pour chaque utilisateur. Son utilisation n'est pas complexe puisque l'utilisateur n'a que ses devices et de ses gateways à gérer. Mais cette interface n'est pas personnalisable. En effet, la gestion des Network et Application Server étant gérés par TTN, il est impossible d'avoir des droits équivalents au statut de « global admin » sur ChirpStack.

Néanmoins, ce système n'a pas que des inconvénients. Il permet de créer son propre réseau sans acheter nécessairement une gateway. En effet, s'agissant d'un réseau communautaire, chacune des gateways déployées est au bénéfice de tous les utilisateurs dans le but d'enrichir et d'étendre le réseau. Toutes les passerelles enregistrées et activées sous TTN sont donc rendues publiques à tous ceux qui possèdent et posséderont un compte TTN.

6. Récupérer les données du serveur vers l'interface utilisateur

6.1. Objectif d'une application

Jusqu'à maintenant, nous savons comment envoyer des données de nos devices LoRaWAN vers un serveur via des passerelles. Mais ces données ne sont pas encore exploitables par l'utilisateur. Il nous reste maintenant à voir comment récupérer, traiter, stocker et afficher les données.

Ce que nous allons voir dans cette partie est totalement indépendante du protocole LoRa et de la technologie LoRaWAN et de ce que nous avons vu jusqu'ici. Ces informations peuvent s'adapter à des protocoles de l'Internet des Objets autre que celui du LoRaWAN. Nous avons donc d'un côté le protocole LoRaWAN avec l'envoi des données des devices jusqu'au serveur via les passerelles. Et de l'autre la récupération de ces dernières vers une application.

Il est important de faire la différence entre « Application Server » et « Application ». Nous parlons d'Application Server lorsque nous évoquons du serveur LoRaWAN relié au Network Server. Et nous parlons d'Application quand nous définissons le serveur côté utilisateur (interface Web, Android, serveur de traitement etc…)

Le sens **Uplink** de notre application correspond au trajet de la trame des devices vers cette dernière. Le sens **Downlink**, quant à lui, définit le sens inverse, c'est-à-dire de l'application vers les devices.

Dans le sens Uplink, notre Application utilisateur aura pour rôle :

1. De gérer la récupération des données.
2. De stocker les données (sauvegarde).
3. De les mettre en forme (graphiques, tableaux, …).
4. De les mettre à disposition (interface utilisateur)

Dans le sens Downlink, notre Application utilisateur devra :

1. Présenter une interface utilisateur (Bouton, champ texte, …).
2. Traiter la requête de l'utilisateur.
3. Stocker la requête (sauvegarde).
4. Envoyer la requête au Serveur LoRaWAN.

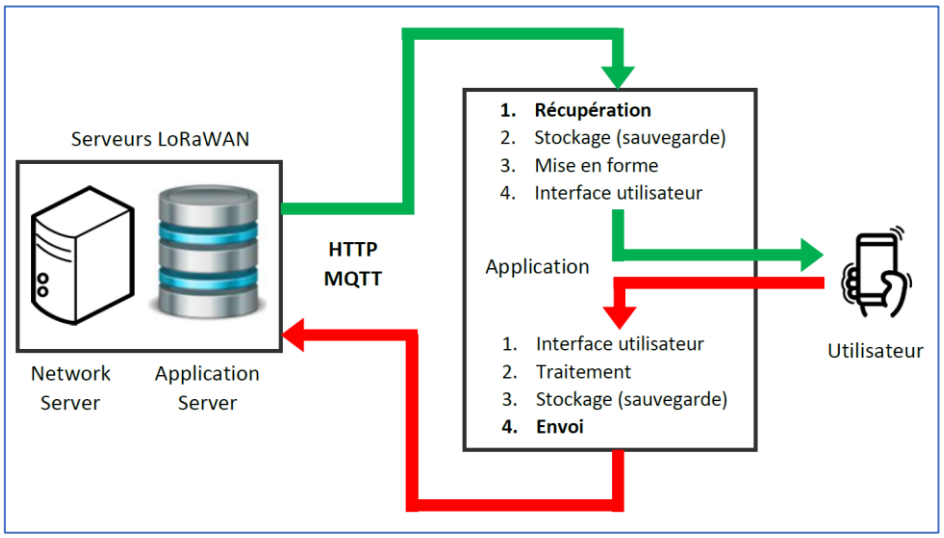


Figure 11 : Détail de l'application

Les parties 6.2 et 6.3 ci-dessous ne traiteront que de la « récupération » dans le sens « uplink » et de l'envoi dans le sens « downlink ».
Pour réaliser ces deux étapes, il existe deux types de langages différents : le HTTP et le MQTT.

6.2. HTTP et MQTT

6.2.1. Méthode HTTP

6.2.1.1. Récupérer les données avec le HTTP GET

6.2.1.2. HTTP POST

6.2.2. MQTT

6.2.2.1. Introduction

MQTT est un protocole léger qui permet de s'abonner à des flux de données. Plutôt que l'architecture Client/Serveur classique qui fonctionne avec des Requêtes/Réponses, le protocole MQTT est basé sur un modèle Publisher/Subscriber. La différence est importante, car cela évite d'avoir à demander (Requête) des données dont on n'a aucune idée du moment où elles vont arriver. Une donnée sera donc directement transmise au Subscriber dès lors que celle-ci aura été reçue dans le **Broker** (serveur central). ChirpStack sert déjà de Broker MQTT, nous n'avons donc pas besoin d'en installer un.

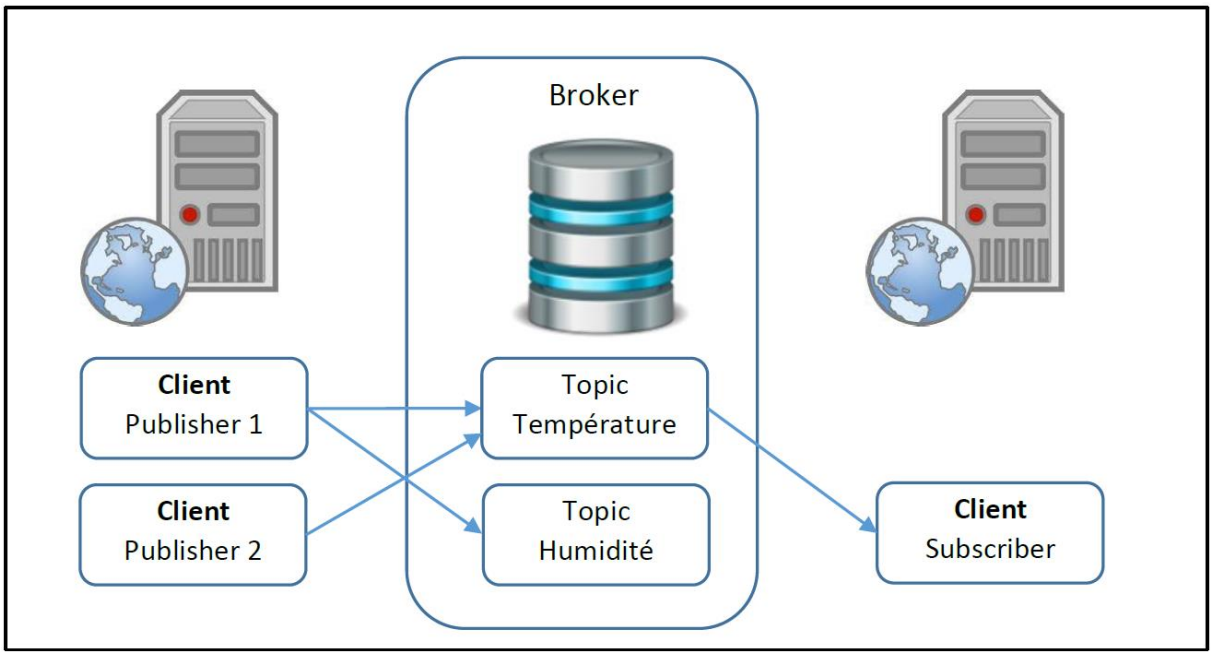


Figure 12 : Modèle Publisher-Subscriber du protocole MQTT

Dans cette figure, il y a deux publishers, ce sont des entités qui publient sur un certain lien que l'on appelle « topic ». Ici, le device 1 publie sur les topics « température » et « humidité ». Le device 2 publie uniquement sur le topic « température ». Le subscriber lui, récupère les informations se trouvant dans le topic « température » mais pas de celui de « humidité ». A noter qu'il récupère les données de températures des devices 1 et 2.

- **Les Publishers et les Subscribers n'ont pas besoin de se connaître.**

- **Les Publishers et les Subscribers ne sont pas obligés de s'exécuter en même temps.**

6.2.2.2. Topic du MQTT

Les topics sont construits sur une chaîne de caractères en utilisant la barre oblique « / » comme caractère de séparation.

La figure 13 et le tableau 8 montrent cette hiérarchie :

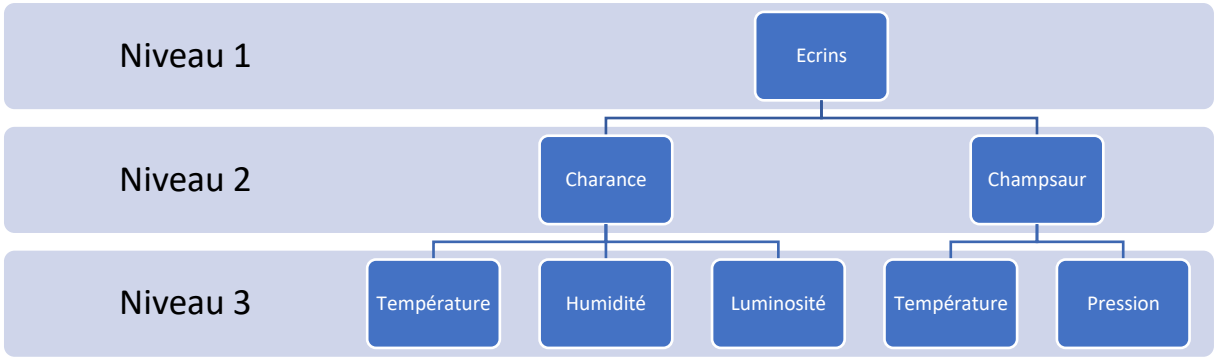


Figure 13 : Exemple de hiérarchie de topic MQTT

Nom du topic	Détail du topic
Ecrins/Charance/Température	La **Température** à **Charance** aux **Ecrins**
Ecrins/Champsaur/Pression	La **Pression** au **Champsaur** aux **Ecrins**

Tableau 8 : Exemple de topic

Toutefois, il est possible grâce à des « jokers » qu'un subscribers puissent s'abonner à plusieurs branches de l'arborescence en même temps et en un seul topic. Les deux jokers existant sont le « + » et le « # ».

- Le signe « + » remplace n'importe quelle chaînes de caractères du niveau où il est placé
- Le signe « # » remplace toutes les chaînes de caractères de tous les niveaux suivants (il doit obligatoirement être placé à la fin)

Nous allons reprendre le tableau 8 (Fait référence a tableau précédent « Exemple de topic » et non pas au tableau « exemple de topics avec jokers ») en rajoutant ces jokers :

Nom du topic	Détail du topic
Ecrins/+/Température	La **Température** à **Charance** et au **Champsaur** aux **Ecrins**
Ecrins/Champsaur/#	La **Pression** et la **Température** au **Champsaur** aux **Ecrins**

Tableau 9 : Exemple de topics avec jokers

Nos topics auront plutôt la forme « application/*applicationID*/device/*DevEUI* » quand nous utiliserons ChirpStack. Les données en italiques seront à modifiées en fonction du topic que l'on souhaitera obtenir.

6.2.2.3. Qualité de Service au cours d'un échange

La qualité du service, ou Quality of Service (QoS),

6.3. Node-RED, InfluxDB, Grafana

Nous serons amenés à utiliser ces interfaces afin d'optimiser la récupération des données.
Node-Red sera le centre de ce procédé. En effet il nous permettra de récupérer les données, de les stocker temporairement (environ 48h) et de les afficher.
Mais pour le long terme nous aurons besoin de stocker les données dans une vraie base de données. C'est pourquoi nous installerons InfluxDB qui est une base de données temporelle et parfaitement adaptée à l'IoT. Puis avec Grafana dont les fonctionnalités d'affichage sont plus importantes et plus personnalisables qu'avec Node-Red, nous afficherons les données stockées dans InfluxDB.

Ces procédés sont expliqués en détail en annexe.

ANNEXE A : Installer sa/son serveur/RasberyPI/Virtual Machine (VM)

Avant toute chose, il nous faut un serveur installé sur une VM ou sur une carte RasberryPi afin d'y installer toutes les applications dont nous aurons besoin.

1. Serveur Debian sur une RaspBerry Pi 4

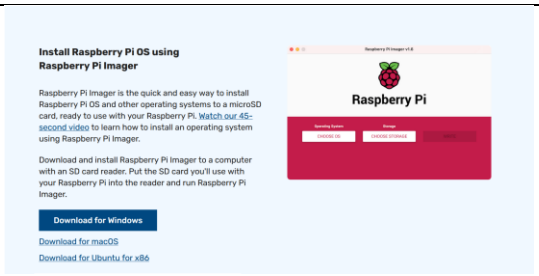
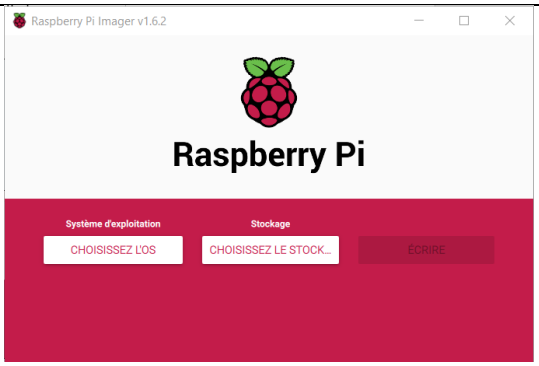
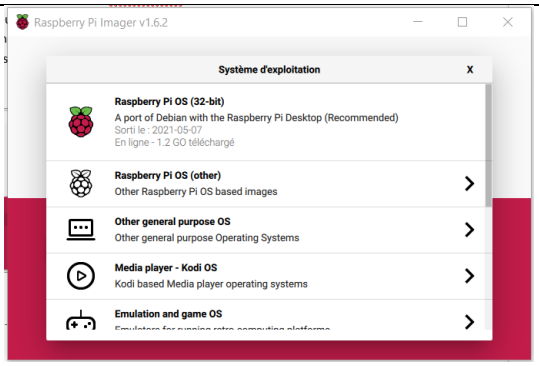
Nous allons faire le choix d'installer un serveur Debian sur une Raspberry Pi4 avec une carte SD de 256 Go. Une carte avec un stockage moindre suffit largement. La carte SD devra être vierge de toute donnée.

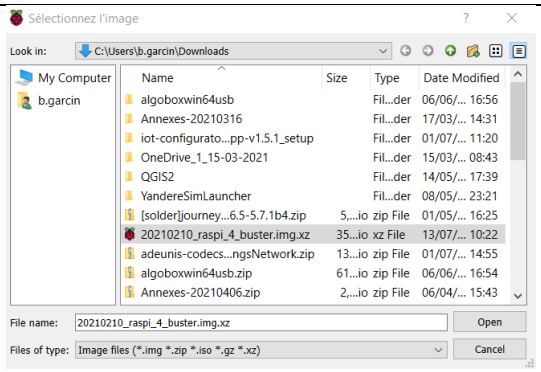
Tout d'abord il faut télécharger la version Debian pour Raspberry sur le site : https://raspi.debian.net/tested-images/

Télécharger la version la plus ancienne disponible car ce sera la version la plus stable. Lors de la création de cette documentation, nous avons choisis cette version :

2021.02.10	10 (Buster)	4	4 (4GB)	xz-compressed image 356.58 MB, 2021-02-10 14:16:-0600 sha256sum GPG-signed sha256sum

Pendant que l'explorateur Linux se télécharge, aller sur le site suivant pour télécharger le flash Raspberry : https://www.raspberrypi.org/software/

1. Une fois sur le site, descendre dans la page web pour trouver l'interface ci-contre, puis cliquer sur « Download for Windows ». Une fois téléchargé, procéder à l'installation.	
2. Une fois l'installation terminée, cliquer dans « Choose OS » (« Choisir l'OS ») Note : OS pour Operating System	
3. Une pop-up s'ouvre. Descendre tout en bas et sélectionner « Utiliser image personnalisée ».	

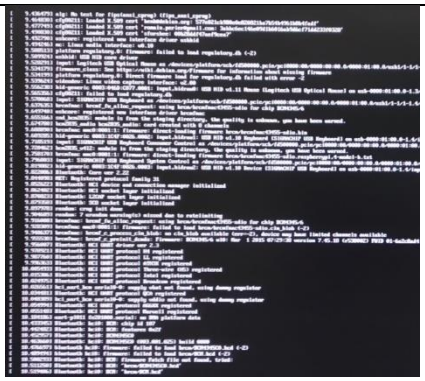
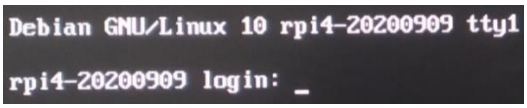
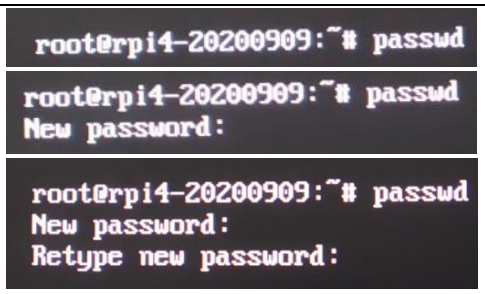
4. Aller dans ses fichiers et retrouver le fichier de l'OS.	
5. L'OS est défini. Choisir maintenant la carte SD du Raspberry en cliquant sur « Choisissez le stockage ». Sélectionner la carte SD souhaitée.	
6. Puis cliquer sur « Ecrire ». Un message s'affichera vous informant que votre carte SD sera formatée.	
7. Après avoir cliqué sur « Oui », retirer la carte SD et insérer la dans votre Raspberry Pi.	

Une fois ces opérations terminées, connecter votre Raspberry à un ordinateur :

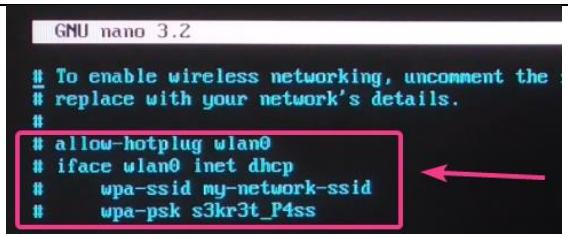
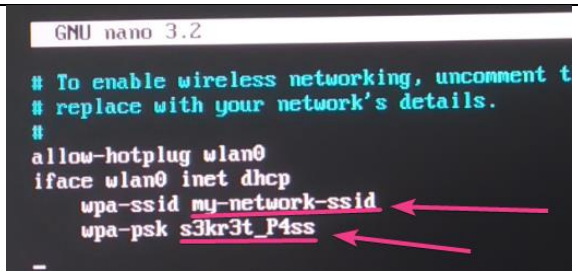
- Brancher le câble d'alimentation
- Brancher l'adaptateur SD-HDMI dans lequel il faut insérer le câble HDMI relié à l'écran (ce dernier ne sera branché sur aucun ordinateur, le Raspberry en faisant déjà l'office)
- Brancher un clavier
- Brancher une souris (optionnel)

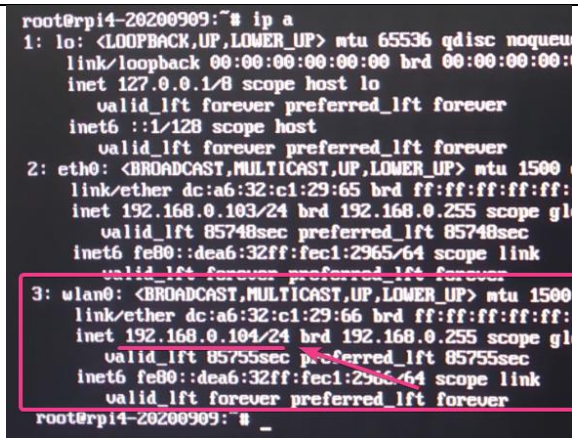
Avertissement : pour le Raspberry Pi, votre clavier est un qwerty. Il est donc conseiller de suivre ces étapes avec une image de clavier qwerty, ou dans l'idéal, d'avoir un clavier qwerty.

Configurer son identifiant de session Debian :

1. Pendant un certain temps, Debian se met en route.	
2. Une fois la mise en route de Debian terminée, enregistrer le nom de l'utilisateur de session, par exemple « root ». Appuyer sur Entrée.	
3. Par défaut, aucun mot de passe n'est demandé. Si vous souhaitez en mettre un un, taper la commande `passwd` Ecrire le mot de passe souhaité puis appuyer sur Entrée. Réécrire son mot de passe puis Entrée.	

Configurer le réseau Wi-Fi :

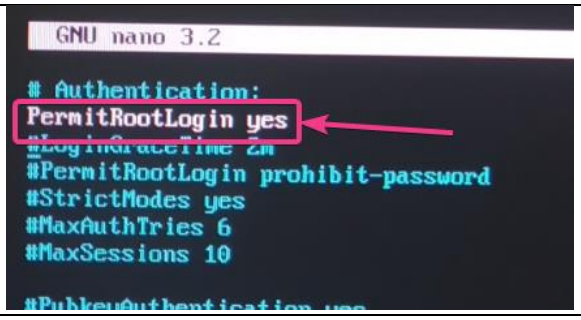
1. Pour commencer, ouvrer le fichier de configuration avec la commande suivante : `nano /etc/network/interfaces.d/wlan0`	
2. Retirer les # des lignes présentent dans l'encadré rouge ci-contre	
3. Rentrer le nom de votre accès Wi-Fi puis le mot de passe de connexion.	
4. Redémarrer votre Raspberry avec la commande suivante : `systemctl reboot`	

5. Une fois le redémarrage terminé, lancer la commande suivante afin de tester le réseau : `ip a` L'adresse IP de votre connexion s'affiche. Celle présente sur l'image sera différente de la vôtre.	

A noter que ces étapes seront à refaire intégralement à chaque fois que :

- Le nom d'accès et/ou le mot de passe de la Wi-Fi change
- Une nouvelle connexion Wi-Fi est à effectuer

Paramétrer son Raspberry en SSH :

1. Comme pour le paramétrage Wi-Fi, ouvrez le fichier de configuration du SSH : `nano /etc/ssh/sshd_config`	
2. Ajouter la ligne **PermitRootLogin Yes** comme indiqué ci-contre Procéder à la sauvegarde du fichier en faisant Ctrl+X, puis Y, et enfin Entrée.	
3. Redémarrer le SSH avec la commande suivante : `systemctl restart sshd`	
4. Allumer un autre ordinateur et ouvrir l'invite de commandes, peu importe que son OS soit un Windows ou un Linux. Lancer la commande suivante : `ssh root@192.168.xxx.xxx` Vous devriez avoir une IP commençant par les mêmes préfixes.Les « xxx » correspondent à la fin de votre propre adresse IP. A noter que la connexion Internet de cette ordinateur est sans importance. Elle peut très bien être différente de celle de votre Raspberry.	
5. Taper yes et appuyer sur Entrée. Vous êtes maintenant connecté à votre Raspberry à distance.	

Votre Raspberry est prêt à l'emploi. Néanmoins, le Raspberry ainsi installer est loin d'être au même niveau qu'un serveur clé en main comme ceux d'OVH par exemple.
Il faut donc installer des fichiers et des extensions pour éviter les messages d'erreurs lors de l'installation de ChirpStack, Node-Red etc…

A chaque nouvelle installation, Debian vous demandera confirmation. Ecrire « Y » si vous êtes en accord, écrire « n » dans le cas contraire.

```
Do you want to continue? [Y/n]
```

Installer l'extension SUDO

```
apt install sudo
```

Mettre à jour l'apt

```
apt update
```

Appliquer les mises à jour à l'apt

```
apt upgrade
```

Installer l'extension NPM

```
apt install npm
```

Mettre à jour l'apt

```
apt update
```

Appliquer les mises à jour à l'apt

```
apt upgrade
```

```

```

2. Serveur Debian sur une VM

Nous allons faire le choix d'installer un serveur Debian sur une VM.

ANNEXE B : paramétrer un capteur/une passerelle

Comme le paramétrage des capteurs et des passerelles varie d'une marque à une autre, et parfois d'un modèle à un autre, ces informations seront dans un autre document afin de ne pas surchargé inutilement celui-ci.

C:\Users\b.garcin\Desktop\Estia\2A\Stage 2A\StageParcEcrins\Récap infos\ 08_Paramétrer un capteur et_ou une passerelle en LoRaWAN

ANNEXE C : paramétrer ChirpStack

ChirpStack est à la fois un Network Server et un Application Server.

Pour se connecter à son compte ChirpStack il faut d'abord l'installer sur son serveur. Pour ce faire il faut suivre les instructions de cette page internet : https://www.chirpstack.io/project/guides/debian-ubuntu/

```
apt install mosquitto mosquitto-clients redis-server redis-tools postgresql
```

```
sudo -u postgres psql
```

```
-- set up the users and the passwords
-- (note that it is important to use single quotes and a semicolon at the end!)
create role [login appserver] with login password '[mot de passe associé]';
create role [login nwkserver] with login password '[mot de passe associé]';

-- create the database for the servers
create database [name chirpstack_as] with owner [login appserver];
create database [name chirpstack_ns] with owner [login nwkserver];

-- change to the ChirpStack Application Server database
\c chirpstack_as

-- enable the pq_trgm and hstore extensions
-- (this is needed to facilitate the search feature)
create extension pg_trgm;
-- (this is needed to store additional k/v meta-data)
create extension hstore;

-- exit psql
\q
```

```
sudo apt install apt-transport-https dirmngr
```

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 1CE2AFD36DBCCA00
```

```
sudo echo "deb https://artifacts.chirpstack.io/packages/3.x/deb stable main"

sudo tee /etc/apt/sources.list.d/chirpstack.list
```

```
sudo apt update
```

Dans ChirpStack, un *Network Server* se compose d'un ou plusieurs *Service-Profiles*, eux-mêmes divisés en *Application* et en *Gateway*. Chaque *Application* peut avoir un ou plusieurs *Device(s)*. Et un *Device-Profiles* peut-être associés à un ou plusieurs *Device(s)*. La figure 14 illustre cette hiérarchie.

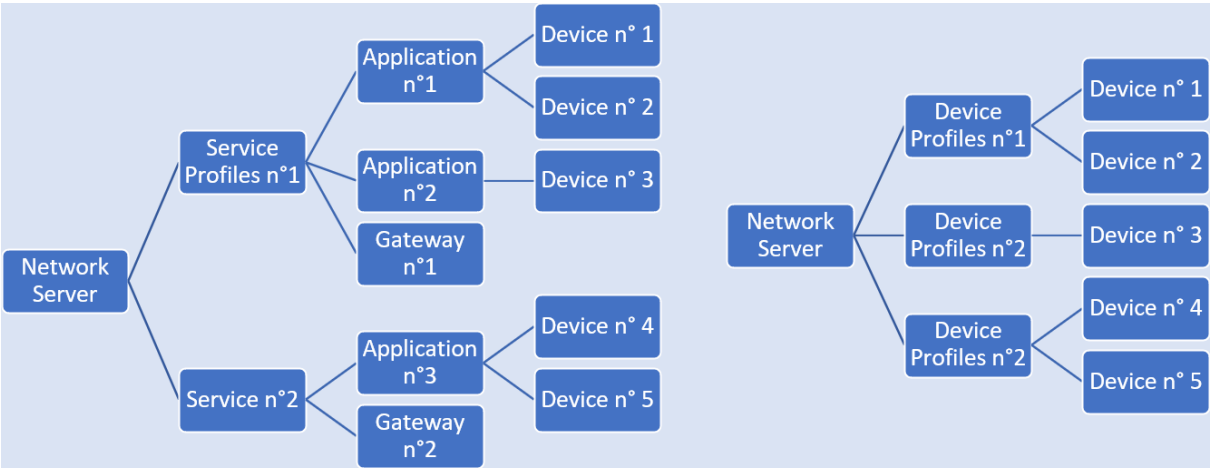


Figure 14 : Schéma de la hiérarchie dans ChirpStack

Pour vous connecter au ChirpStack, il faut aller au lien suivant : **@IPlocalhost :8080**. L'identifiant et le mot de passe sont « **admin** » par défaut.

Créer le Network Server :

- Cliquer sur « Network Server » dans le menu à gauche de l'écran.
- Cliquer ensuite sur « +ADD » en haut à droite
- Dans l'onglet « Général », inscrire le nom du Network Server que vous souhaitez lui donner.
  Renseigner l'adresse du Network Server, à savoir l'adresse IP de votre serveur ainsi que le port 8000 : @IPlocalhost :8000.
  Cliquer sur « Add Network Server » en bas à droite.
  Les autres onglets ne sont pas nécessaires pour l'instant.

Créer une organisation :

- Retourner dans le menu de gauche et sélectionner « Organization »
- Cliquer sur « Create organization »
- Renseigner le nom, le nom complet, si ce groupe peut avoir des passerelles ou non, ainsi que le nombre maximum de capteurs et/ou de passerelles que ce groupe peut avoir (si vous souhaitez que ce groupe possède un nombre illimité de capteur et/ou de passerelle, mettre la valeur 0).
- Puis cliquer sur « Create organization ».

Dans le menu de gauche, déplier le menu déroulant puis sélectionner votre organisation

Créer un Service-Profile :

- Sélectionner « Service-Profiles » puis « Create ».
- Renseigner les informations demandées, et liés votre Service-Profiles à votre Network Server.
  Il est conseillé de cocher la case « Add gateway meta-data » afin d'avoir les informations sur la connexion. La deuxième case est optionnelle mais il est possible de la cocher, cela n'affectera en rien les performances.

- Laisser à 0 les autres valeurs.
  La case « private gateways » signifie que ce « service profiles » sera privé. Cela signifie que ces gateways peuvent seulement être utilisées par des devices enregistrés sous le même Service-Profiles.

Enregistrer un nouveau profil pour un type de capteur :

- Aller dans le bandeau de gauche et cliquer sur « Device Profiles » puis sur « Create ».
- Dans l'onglet Général, renseigner les différents champs. Vérifier dans la datasheet des devices la version LoRa que ces derniers peuvent supporter.
  - Dans « LoRaWAN Regional Parameters revision » mettre la lettre A.
  - Laisser le reste avec les valeurs par défauts.
  - Dans l'onglet « JOIN(OTAA/ABP) cocher la case OTAA si vous voulez être en OTAA, sinon remplissez les champs pour l'ABP.
  - Ne rien mettre dans les onglets suivants.
  - L'onglet « CODEC » sera rempli plus tard car son contenu change d'une marque de capteur à une autre.
- Cliquer sur « Create device profiles » pour enregistrer votre profil type de capteur.

Enregistrer une application :

- Dans le menu de gauche cliquer sur « Applications » et compléter.
- Puis cliquer sur « Create application ».
- Dans la liste des applications apparaît maintenant la vôtre. Sélectionner là.
- Cliquer sur « Create » afin d'enregistrer des devices qui seront liées à un device-profile ainsi qu'à cette application.

Enregistrer une gateway :

- Pour enregistrer une gateway, cliquer sur « gateway » dans le menu de gauche, puis sur « Create ».
- Compléter les champs vides. Il n'est pas nécessaire de créer et relier la gateway à un « gateway profiles ». Cette option permet de mettre à jour la passerelle automatiquement mais nécessite ChirpStack Concentrator qui ne sera pas évoqué dans ce document.

Enregistrer de nouveaux utilisateurs :

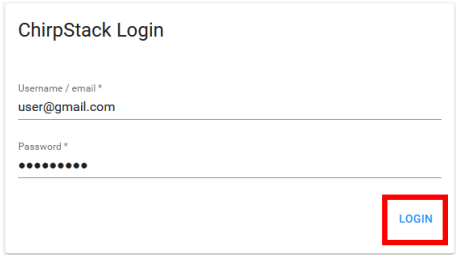
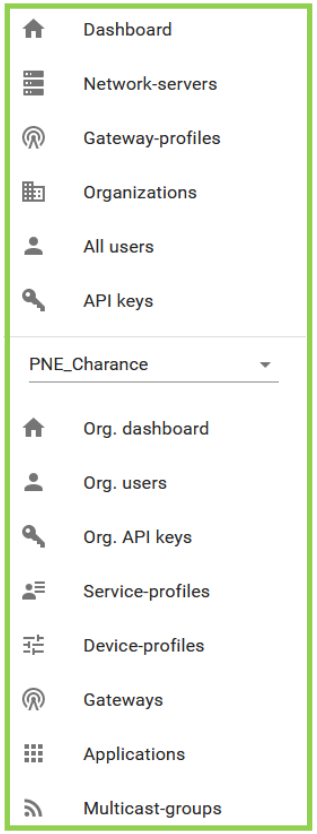
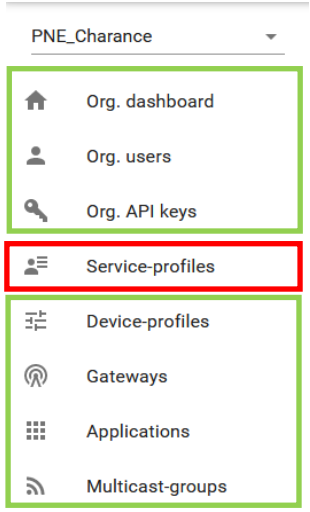
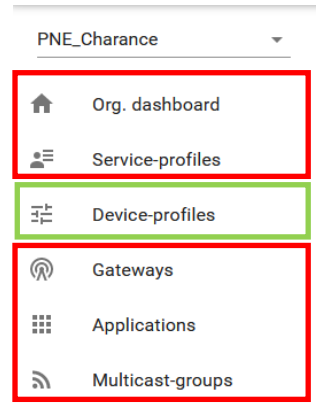
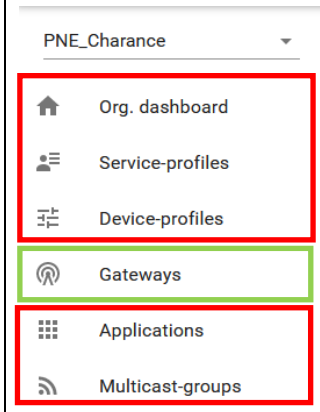
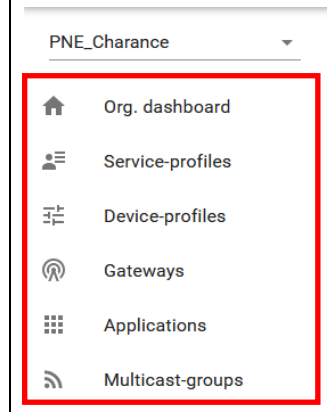
- Dans un premier temps, il faut aller dans « All Users » dans le menu de gauche. Il s'agit de la liste de tous les utilisateurs de toutes les organisations hébergées sous le même ChirpStack.
- Cliquer sur « Create ».
- Après avoir renseigné l'adresse mail de l'utilisateur qui lui servira d'identifiant de connexion, vous avez deux cases que vous pouvez cocher :
  - « Is active » pour autoriser un utilisateurs à se connecter (à cocher sauf si vous voulez bloquer l'accès à quelqu'un)
  - « Global admin » permet à un utilisateur de créer, modifier, supprimer des organisations, et à créer, modifier, supprimer des devices, gateways, service-profiles etc… de n'importe quelle organisation.

Pour affecter un utilisateur à une organisation, il n'est pas nécessaire de le mettre en « global admin ». En effet, il aura déjà accès à toutes les données de l'organisation. Sélectionner juste « Is active » lors de la création de son profil. Les restrictions d'accès dans sa propre organisation sont expliquées ci-dessous.

Sélectionner l'organisation dans le menu déroulant situé dans le bandeau de gauche puis en tant que global admin :

- Aller dans « Org. Users » et rajouté un utilisateurs qui aura été préalablement enregistré dans « All Users ».
  - « organization admin » permet à l'utilisateur de gérer uniquement son organisation et tout ce qu'elle contient et contiendra
  - « device admin » permet à l'utilisateur de gérer uniquement sa propre flotte de capteurs
  - « gateway admin » permet à l'utilisateur de gérer uniquement sa propre flotte de passerelles (uniquement valable si l'organisation a eu l'autorisation d'avoir des passerelles)
  - Ne rien cocher permet à l'utilisateur d'avoir un accès uniquement visuel au données de son organisation mais ne permet aucune modification.

Nota Bene : un service-profile est uniquement créable par un global admin. Un « organisation admin » n'a pas accès aux Service-Profiles de son organisation. Si l'une de vos organisations souhaitent avoir un service-profile, elle devra en faire la demande à un « global admin ».

« Is active »	Global Admin	Organisation Admin	Device Admin	Gateway Admin	User
					

Lorsque cette case **n'**est **PAS** cochée, l'utilisateur se verra bloqué à l'écran de connexion comme ci-dessus.

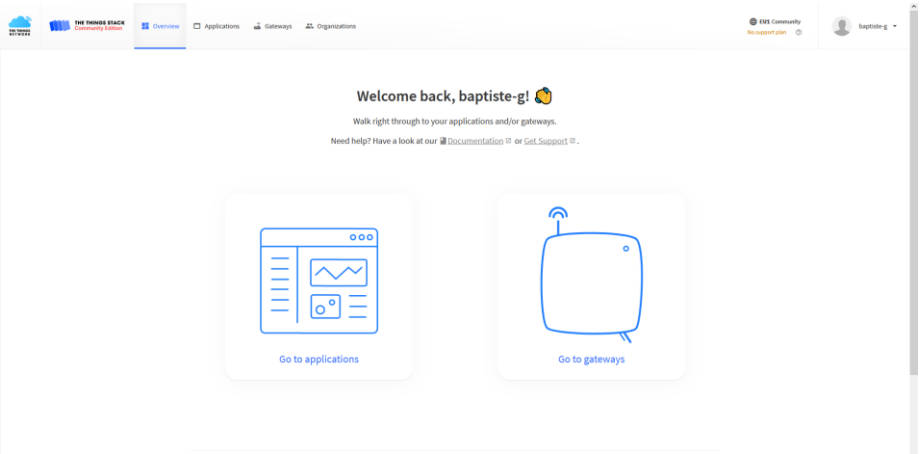
Les zones entourées de vert sont les onglets libres d'accès. C'est-à-dire que l'utilisateur correspondant à le droit de créer, de modifier et de supprimer une partie ou l'entièreté de son contenu.
Les zones en rouge, elles, indiquent un simple accès visuel pour l'utilisateur concerné.

Nous noterons que les « Organization Admin », « Device Admin », « Gateway Admin » et « User » seront limités à leur(s) organisation(s) respective(s).
Nous remarquerons également qu'en fonction du statut de l'utilisateur, même l'accès visuel peut être restreint.

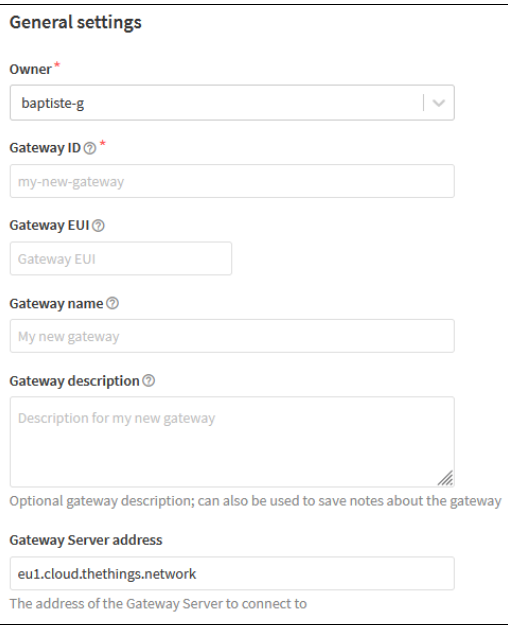
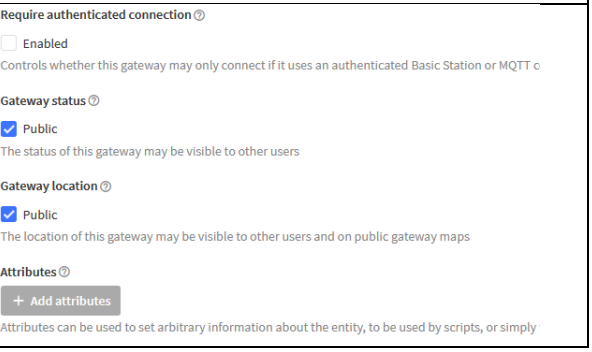
ANNEXE D : paramétrer The Things Network

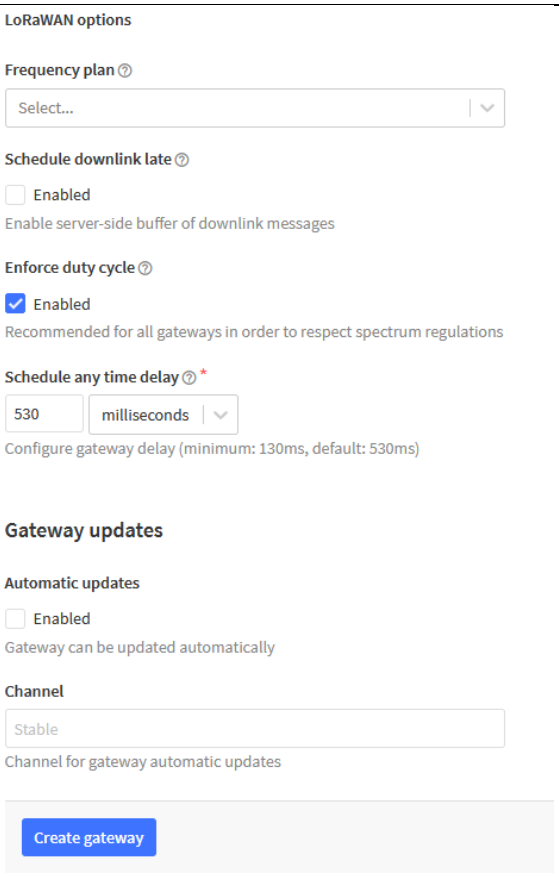
Créer vous un compte sur https://eu1.cloud.thethings.network.
Connecter vous ensuite via le même lien.

Il n'y aura rien à installer sur votre serveur/Raspberry. Tout se fait en ligne.

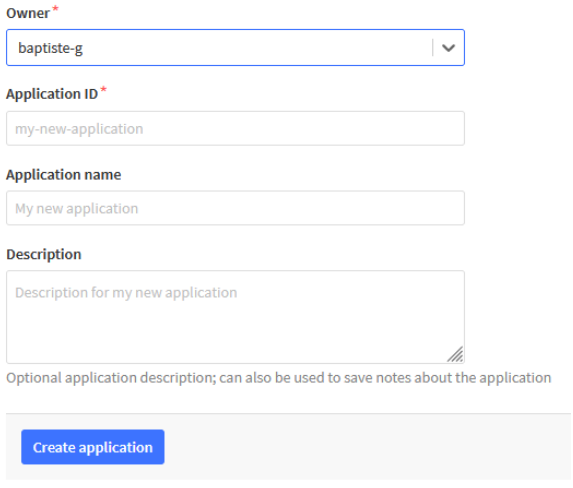


Enregistrer un gateway :

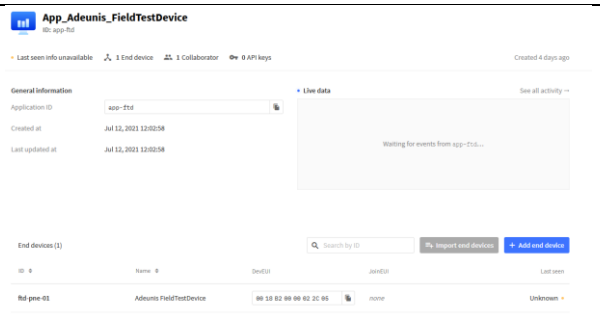
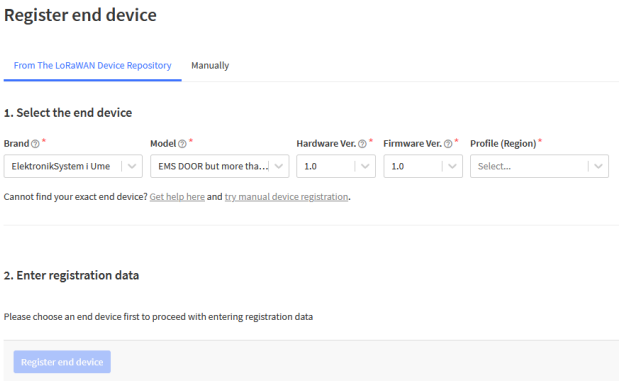
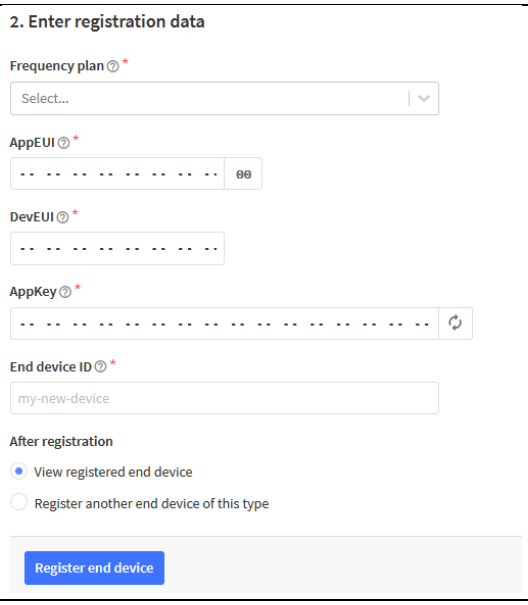
1. Aller dans l'onglet Gateways en haut.	
2. Cliquer ensuite sur « Add gateway »	
3. Entrer les informations de votre passerelle : ➢ Owner : propriétaire ➢ Gateway ID : choisir un identifiant ➢ Gateway EUI : numéro de série de la passerelle ➢ Gateway Name : le nom de votre passerelle ➢ Gateway description : une description de votre passerelle ➢ Gateway Server Address : l'adresse du serveur auquel est connectée votre passerelle.	
4. Les authentification de connexion : ➢ Enabled : contrôle si cette passerelle ne peut se connecter que si elle utilise une station de base authentifiée ou une connexion MQTT ➢ Cocher Public si vous souhaitez rendre publique le statuts de votre passerelle	

➢ Cocher Public si vous souhaitez rendre publique la localisation de votre passerelle	
5. Les options LoRaWAN : ➢ Frequency plan : sélectionner les canaux de fréquences légaux pour votre région ➢ Enabled : activer la mémoire tampon côté serveur des messages de liaison descendante ➢ Enabled : respecter le duty-cycle (recommandé) ➢ Schedule : fortement conseillé pour les passerelles utilisant le satellite ou la 3G. Permet d'envoyer les messages pour les devices classe C en avance. ➢ Enabled : activer les mises à jour automatique ➢ Channel : Appuyer sur « Create gateway ».	

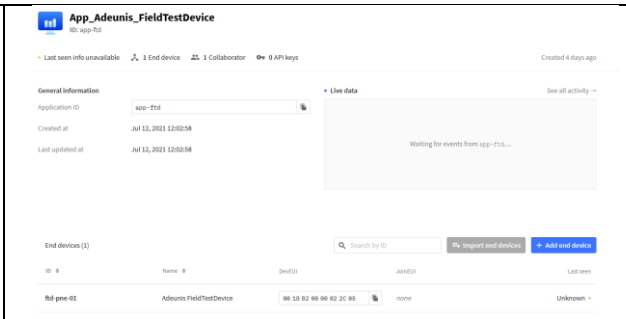
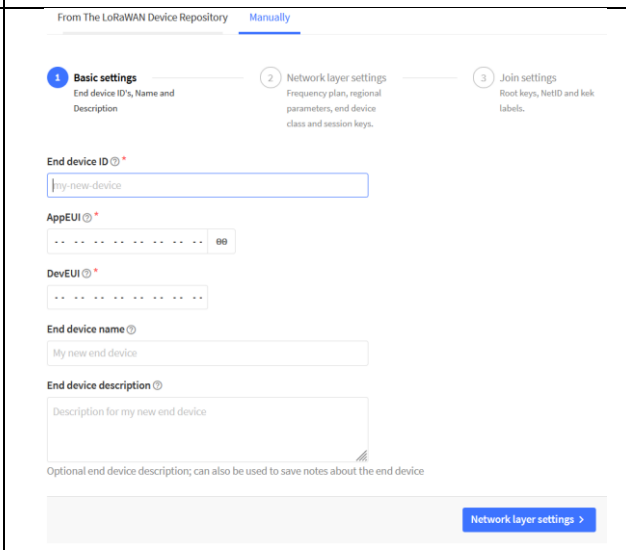
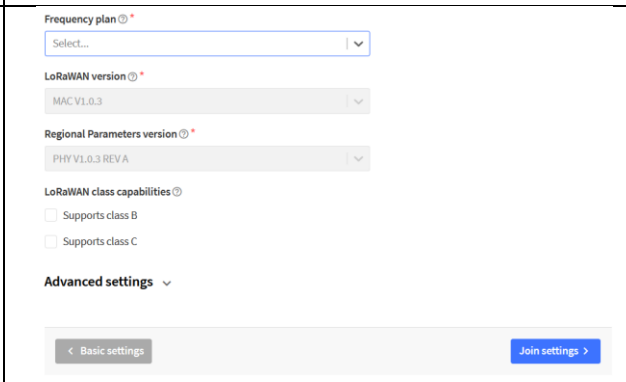
Enregistrer une application (groupement de device) :

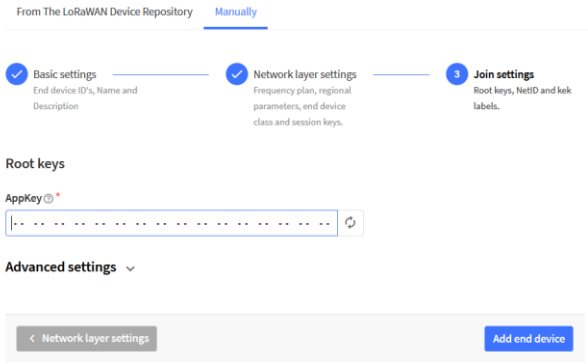
1. Aller dans l'onglet Applications en haut.	
2. Cliquer ensuite sur « Add application »	
3. Informations de l'application : ➢ Owner : propriétaire ➢ Application ID : choisir un identifiant ➢ Application name : le nom de votre passerelle ➢ Description : une description de votre application Appuyer sur « Create application ».	

Enregistrer un device (méthode rapide et déconseillée) :

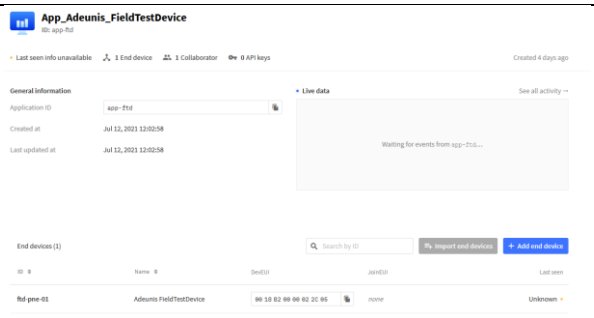
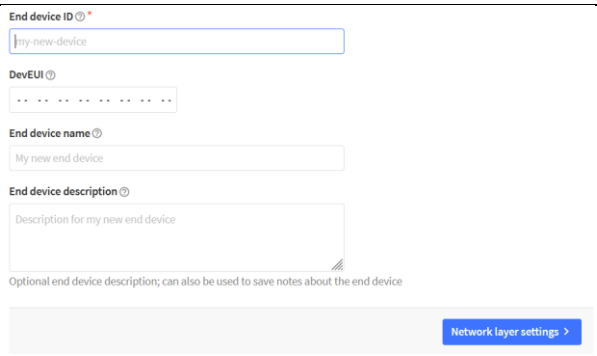
1. Ouvrir une application et cliquer sur « +Add end device » en bas à droite.	
2. Donner la marque, puis le modèle de votre capteur. L'interface ci-contre s'affiche. Remplir les versions Hardware et Firmware puis la fréquence LoRaWAN de votre région.	
3. Une fois ces informations rentrées, la deuxième partie s'affiche. Renseigner votre fréquence LoRaWAN avec plus de précision, l'AppEUI, le DevEUI, l'AppKey et l'identifiant de votre device, à choisir. Cliquer ensuite sur « Register end device ».	

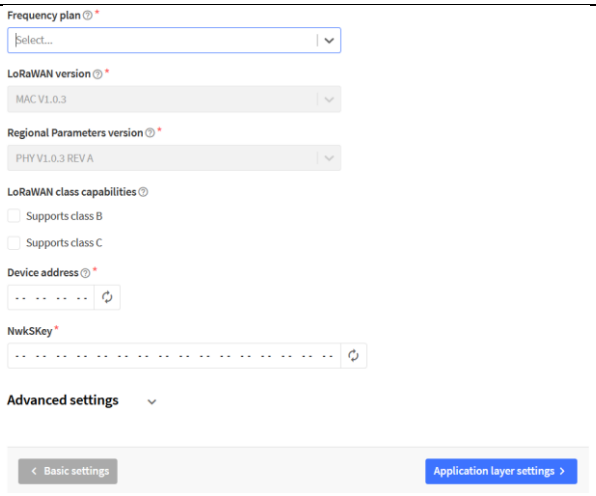
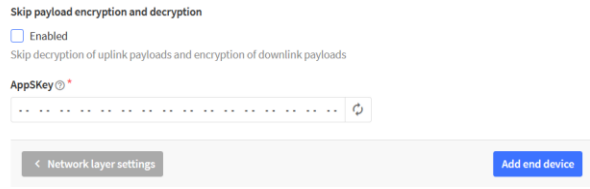
Enregistrer un device (méthode plus longue mais plus détaillée) : **OTAA**

1. Ouvrir une application et cliquer sur « +Add end device » en bas à droite.	
2. Cliquer sur l'onglet « Manually ». Choisir la méthode OTAA. Inscrire la version LoRa supportée par votre device. Laisser le reste par défaut puis cliquer sur « Start ».	
3. Inscrire l'ID, l'AppEUI, le DevEUI et le nom du capteur à enregistrer. Cliquer sur « Network layer settings ». <u>Conseil</u> : essayer de trouver une sorte de code pour vos ID, par exemple : **marque-modele-eui-lieu-n°** Ce qui donnerait (pour un capteur bien précis) : **elt-lite-95a1-chr-01** Il vous servira plus que le nom lui-même donc il faut savoir à quel capteur correspond chacun des ID	
4. Une fois ces informations rentrées, la deuxième partie s'affiche. Renseigner votre fréquence LoRaWAN. Normalement votre version et votre paramètre régional seront bloqués. Dans le cas contraire, donner la version LoRa supportée par votre device et choisir « REV A » pour le paramètre de région. Cliquer ensuite sur « Join settings ».	

5. Enregistrer votre AppKey puis cliquer sur « Add end device ».	

Enregistrer un device (méthode plus longue mais plus détaillée) : **ABP**

1. Ouvrir une application et cliquer sur « +Add end device » en bas à droite.	
2. Cliquer sur l'onglet « Manually ». Choisir la méthode ABP. Inscrire la version LoRa supportée par votre device. Laisser le reste par défaut puis cliquer sur « Start ».	
3. Inscrire l'ID, le DevEUI et le nom du capteur à enregistrer. Cliquer sur « Network layer settings ». <u>Conseil</u> : essayer de trouver une sorte de code pour vos ID, par exemple : **marque-modele-eui-lieu-n°** Ce qui donnerait (pour un capteur bien précis) : **elt-lite-95a1-chr-01** Il vous servira plus que le nom lui-même donc il faut savoir à quel capteur correspond chacun des ID	

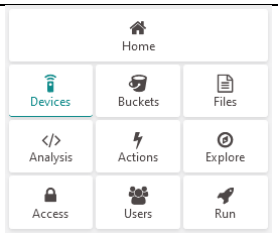
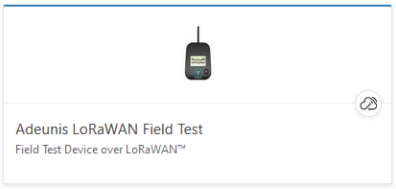
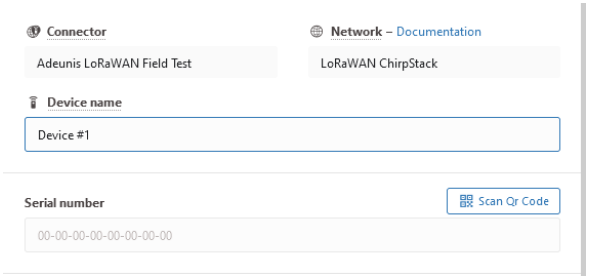
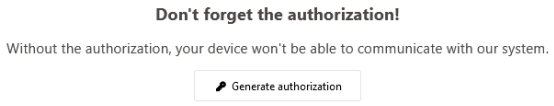
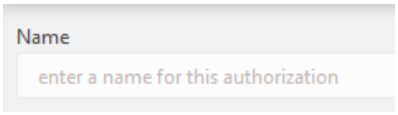
4. Une fois ces informations rentrées, la deuxième partie s'affiche. Renseigner votre fréquence LoRaWAN. Normalement votre version et votre paramètre régional seront bloqués. Dans le cas contraire, donner la version LoRa supportée par votre device et choisir « REV A » pour le paramètre de région. Ecrire le DevAddr et la NwkSKey du device. Cliquer ensuite sur « Application layer settings ».	
5. Cocher « Enabled » si vous souhaitez ignorez le déchiffrement des trames montantes et le chiffrement des trames descendantes. Enregistrer votre AppSKey puis cliquer sur « Add end device ».	

ANNEXE E : Paramétrer Tago.io

1. Paramétrer avec ChirpStack

Pour connecter des devices enregistrés dans votre serveur ChirpStack à Tago.io, vous devez d'abord créer une intégration en http dans ChirpStack. Mais uniquement si vous avez déjà au moins un device-profile, au moins une application et avec au moins un device dans votre Server LoRaWAN.

Créer vous un compte Tago.io puis connectez-vous. Il faut maintenant enregistrer son device dans Tago.io afin de le lier au Network Server.

1. Aller dans l'onglet « device » puis sélectionner « + Add device », puis cliquer sur les trois bandes horizontales au-dessus de « Home » pour fermer cette onglet.	
2. Dans le bandeau de gauche, sélectionner le type de Network Server. Dans ce cas, il s'agira de « LoRaWAN ChirpStack ».	
3. Une liste de capteur apparaît. Sélectionner votre modèle de device. Une barre de recherche est disponible. Par exemple le Field Tester de chez Adeunis.	
4. Rentrer les informations suivantes : ➢ Nom du capteur (ou Device name) ➢ DevEUI (ou serial number) Cliquer sur « Create my device » puis sur « Continue » Vous pouvez aussi scanner le QrCode si vous êtes sur l'application smartphone.	
5. Une demande de création d'autorisation est demandée. Si elle est déjà créée, passez à l'étape 7, sinon cliquez sur « Generate authorization ».	
6. Un nouvel onglet s'ouvre. Donner le nom de l'autorisation que vous souhaitez créer dans le bandeau correspondant. Cliquez sur « Generate » à gauche de votre écran.	
7. Dans la liste des autorisations*, cliquez sur « Copy » pour copier le code de cette dernière.	

*Pour y accéder à volonté, il suffit d'aller l'onglet device (étape 1) puis de cliquer sur le bouton « Authorization » à côté de « +Add device ».	

Aller dans votre serveur ChirpStack puis :

- Aller dans « Application »
- Sélectionner l'application dans laquelle se trouve le ou les devices que vous souhaitez connecter à Tago.io
- Aller dans l'onglet « Integration »
- Trouver la case « HTTP » puis cliquer sur « add ».

Rentrer les informations suivantes en appuyant sur « Add Header » autant que nécessaire :

- Payload marshaler : sélectionner JSON
- Headers :

Content-Type	application/json
Authorization	*Code de votre autorisation**

*Note : il suffit de coller le code d'autorisation que vous avez copié précédemment dans Tago.io

- Endpoints : https://chirpstack.middleware.tago.io/uplink

2. Paramétrer avec TTN

ANNEXE F : Les CODEC*

Les CODEC convertir le payload en texte et en données. Sans ce dernier, le payload ressemble à une suite de lettres et de chiffres aléatoire.
Il faut aller dans les paramètres du device profile, dans l'onglet « CODEC » puis sélectionner parmi les choix possibles l'option : « Custom JavaScript codec functions ».
Supprimer le code déjà en place puis copier-coller le CODEC correspondant à la marque de votre capteur parmi les choix proposés dans un dossier annexe (si le CODEC de votre marque n'y est pas, il faudra aller le chercher sur internet).

C:\Users\b.garcin\Desktop\Estia\2A\Stage 2A\StageParcEcrins\Récap infos\CODEC

ANNEXE G : paramétrer Node-Red

Tout d'abord il faut installer Node-Red sur votre serveur. Pour se faire, il faut suivre les instructions de ce site :
https://nodered.org/docs/getting-started/local#installing-with-npm

Installation de Node-Red

```
sudo npm install -g --unsafe-perm node-red
```

Pour mettre à jour :

```
sudo npm install -g --unsafe-perm node-red
```

Pour redémarrer Node-Red en cas de mise à jour, d'ajout de node etc…

```
systemctl restart node-red
```

Désinstaller nodejs

```
sudo apt remove nodejs
```

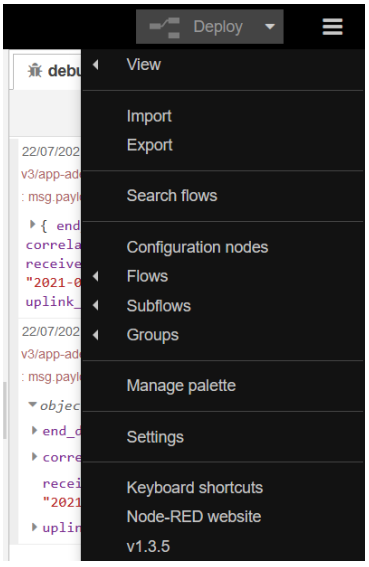
Désinstaller Node-red

```
sudo npm -g remove node-red
```

Désinstaller npm

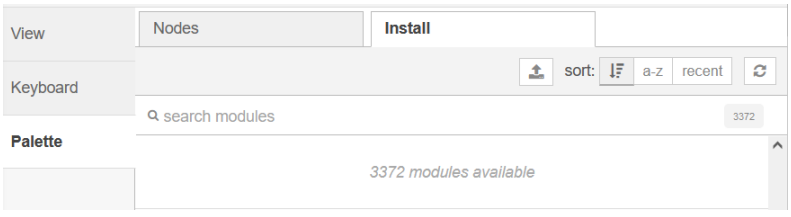
```
sudo apt remove npm
```

Maintenant que Node-Red est installé et à jour, connectez-vous au lien suivant : **@IPlocalhost:1880**
Ce lien permet d'accéder à l'interface de votre Node-Red.



Pour installer une palette de nodes supplémentaires, aller dans le menu en haut à droite et cliquer sur « manage palette » (image ci-contre).

Une fenêtre s'ouvre puis cliquer sur l'onglet « install » et rentrer le nom complet ou partiel de la palette à téléchargée (image ci-dessous).



1. Publier les données (publishers)

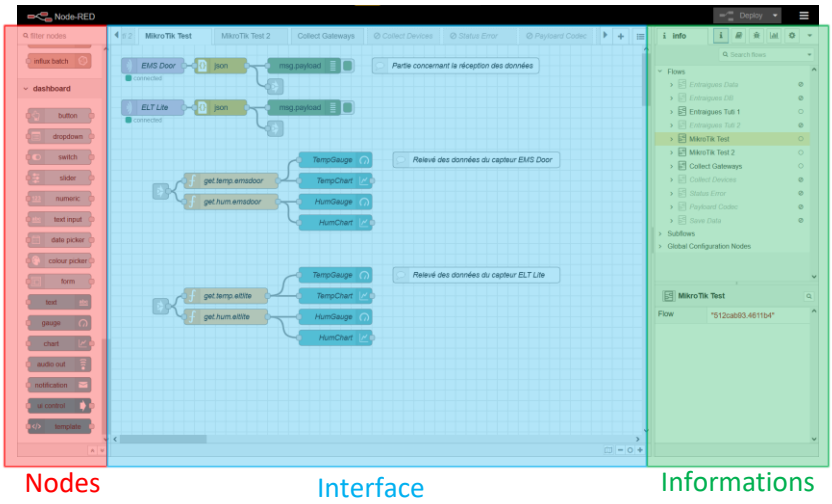
ChirpStack et TTN servent déjà de Broker MQTT, et la publication des données se fait automatiquement via les applications (regroupement de capteurs) de notre AppServer.

2. Récupérer les données (subscribers)

Il faut en plus installer les palettes suivantes :

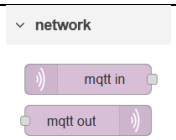
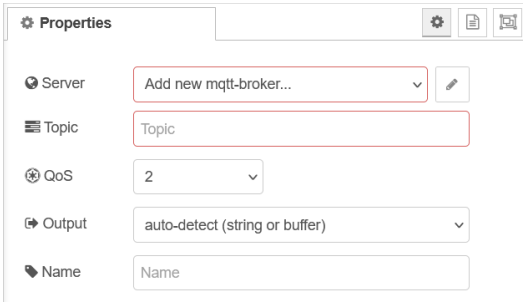
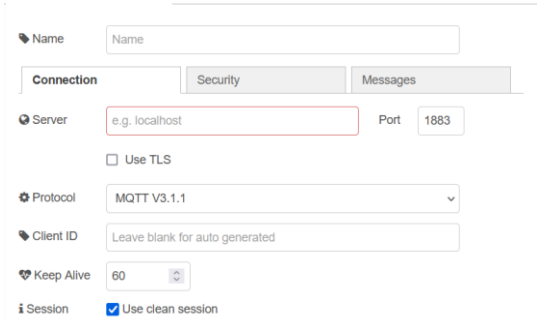
- InfluxDB : *node-red-contrib-influxdb*
- Dashboard : *node-red-dashboard*
- PostgreSQL : *postgrestor*
- E-mail : *node-red-node-email*

L'interface de Node-Red se compose comme suit :

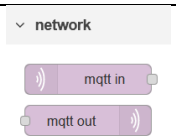
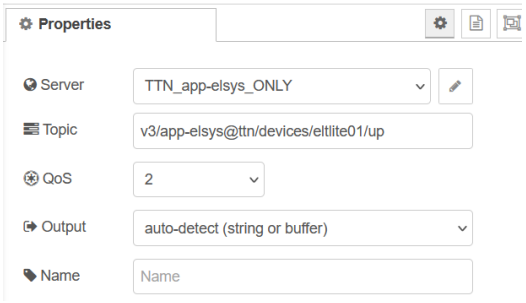
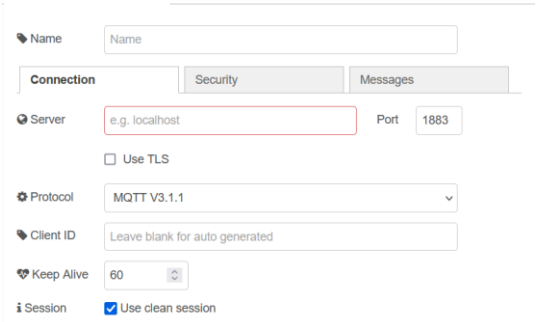
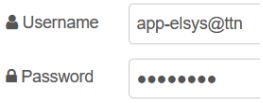


Les bibliothèques des nodes se trouve dans la partie rouge. La partie bleu correspond à la fenêtre d'interface et d'atelier, c'est là que la conception de la récupération des données se fera. Chaque « feuille de travail » est appelé « Flow ». Quant à la partie en vert, elle est associée à différents menus qui apportent des informations comme la réception des trames, la hiérarchie des dashboards etc…
Pour placer une node dans l'interface, il suffit de faire un cliquer-déposer et de paramétrer la node.
Dans un premier temps nous chercherons d'abord à récupérer les données qui ont été publiées comme vu précédemment.

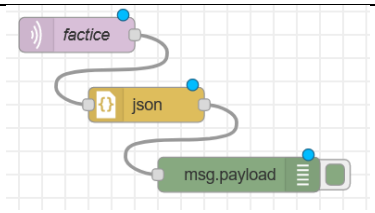
Créer un subscriber **pour CHIRPSTACK** :

1. Dans le dossier « Network » du menu des Nodes, sélectionner la node « mqtt in ».	
2. Double cliquer dessus. Une fenêtre s'ouvre. Pour connecter la Node à un serveur existant, dérouler le menu adjacent et choisir le serveur déjà enregistré. Sinon, il faut enregistrer son serveur (étape 3). Le topic est l'adresse du subscriber : application/**{ID}**/device/**{devEUI}**/event/up Choisissez votre QoS. Laisser Output par défaut et donner un nom à votre Node (optionnel).	
3. Pour enregistrer son serveur, il faut cliquer sur l'icône crayon à côté du champs attribué au serveur. La fenêtre ci-contre apparaît. Vous pouvez nommer cette accès avec le champ « name » en haut. Mettre le lien ou l'adresse IP de votre Broker dans le champ « server » (dans ce cas ce sera l'adresse IP de votre serveur ChirpStack). Laisser le reste par défaut. *Un triangle rouge/orange au-dessus de la Node indique que le chemin d'accès vers le Broker MQTTet/ou le lien vers le Broker n'y ait pas inscrit.*	

Créer un subscriber **pour THE THINGS NETWORK** :

1. Dans le dossier « Network » du menu des Nodes, sélectionner la node « mqtt in ».	
2. Double cliquer dessus. Une fenêtre s'ouvre. Pour connecter la Node à un serveur existant, dérouler le menu adjacent et choisir le serveur déjà enregistré. Sinon, il faut enregistrer son serveur (étape 3). Le topic est l'adresse du subscriber : TTN v2 **{AppID}**/devices/**{DeviceID}**/up TTN v3 v3/**{AppID}**@ttn/devices/**{DeviceID}**/up Choisissez votre QoS. Laisser Output par défaut et donner un nom à votre Node (optionnel).	
3. Pour enregistrer son serveur, il faut cliquer sur l'icône crayon à côté du champs attribué au serveur. La fenêtre ci-contre apparaît. Vous pouvez nommer cette accès avec le champ « name » en haut. Mettre le lien ou l'adresse IP de votre Broker dans le champ « server » : TTN v2 eu.thethings.network TTN v3 eu1.cloud.thethings.network Laisser le reste par défaut. *Un triangle rouge/orange au-dessus de la Node indique que le chemin d'accès vers le Broker MQTTet/ou le lien vers le Broker n'y ait pas inscrit.*	
4. Dans l'onglet « Security » : username : **{AppID}**@ttn password : **{APIKey de votre application}** Pour générer une API Key, aller dans TTN, onglet « application », sélectionner votre application puis dans le bandeau de gauche « Integration/MQTT » puis cliquer sur Generate new API key et copier-la.	

Recevoir les données :

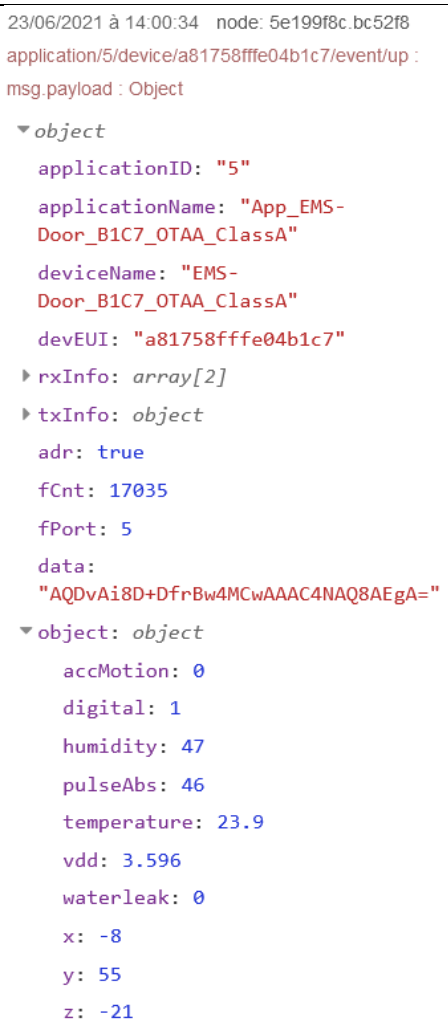
1. Dans l'onglet « Parser », séletionner la Node « json » et déposer là. Aucune modification de paramètre n'ait à faire, laisser tout par défaut. *Cette Node sert à convertir la trame en javascript.*	
2. Dans l'onglet « Common », sélectionner la Node « debug » et placer là. Laisser les paramètres par défaut. *Cette Node sert à afficher dans le menu debug à droite ( ) les trames reçues.*	
3. Relier les entre elles (voir ci-contre).	

Trier les données :

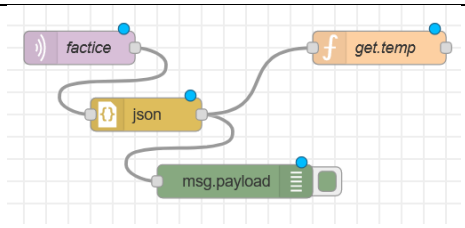
1. Dans le dossier « function », placer la node « function ». Double clique dessus et aller dans l'onglet « Message ».
   Recopier le programme ci-dessous :

```
var temp=msg.payload.object.temperature;
msg.payload=temp;
msg.topic="température";
return msg;
```

   Cette Node avec ce programme permet de récupérer la donnée de température de notre device.
   Le « temp » en vert est le nom de la variable propre à ce programme. Vous pouvez les remplacer par ce que vous voulez dans qu'ils sont rigoureusement identique et que la casse est respectée.
   Le « température » en rouge est le nom que vous souhaitez attribuer à la donnée.
   Le « object.temperature » est le chemin d'accès vers la donnée, voir ci-contre :
   msg>payload>object>temperature
   Si par exemple vous voulez le devEUI, vous n'auriez qu'à remplacer « object.temperature » par « devEUI » car il ne se situe pas dans un dossier.



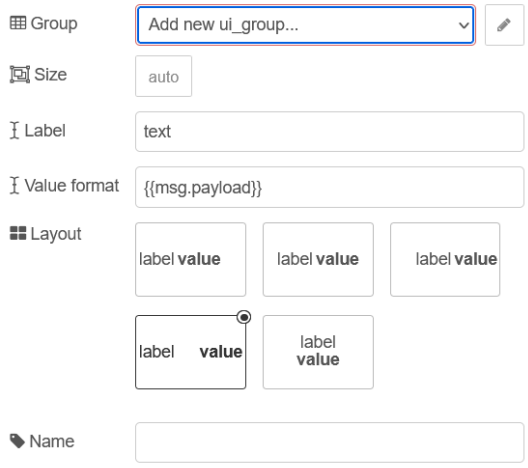
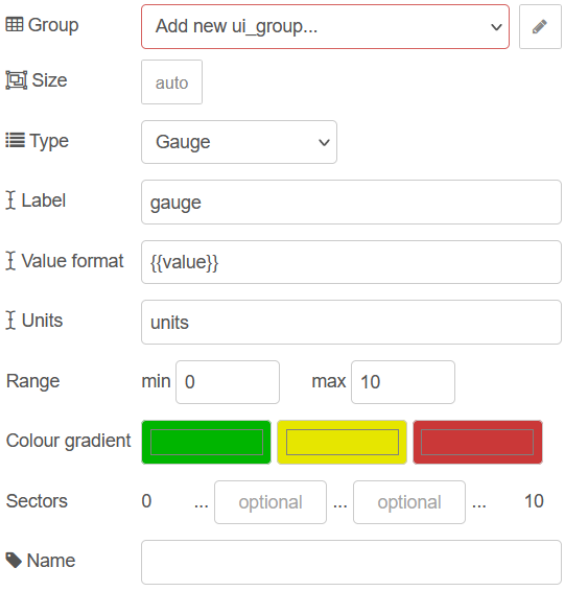
2. Relier cette Node au json.
   Vous pouvez connecter autant de Node « function » que vous voulez à la Node « json ».



Maintenant que nous avons réussi à récupérer les données et à les trier, il faut les afficher.

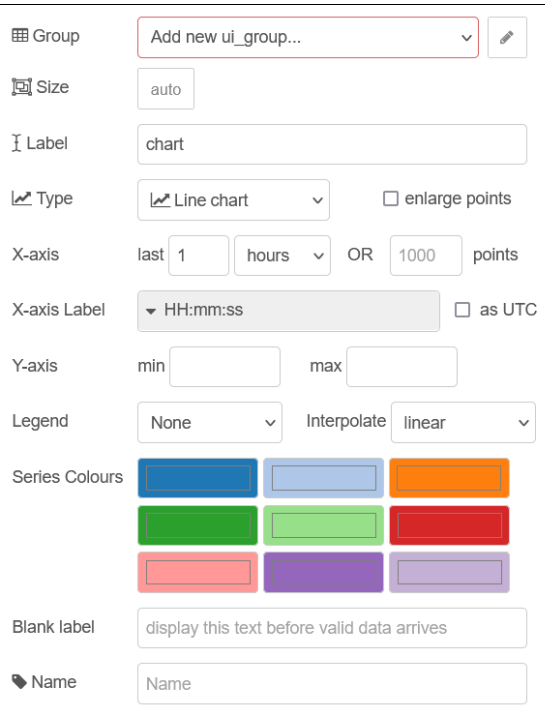
3. Afficher les données (dashboard)

Afficher les données :

1. Vous trouverez toutes les Nodes permettant d'afficher dans le dossier « dashboard » dans le menu à gauche. La première partie en bleu clair constitue des Nodes qui envoie des données vers le Serveur/device. Ce n'est pas ce nous souhaitons. Nous allons donc nous intéresser à la seconde partie. Principalement aux Nodes « text » « gauge » et « chart ».	
2. Node « text » : ➢ Label : donner un nom correspondant au type de la valeur de la donnée recueillie (température, humidité etc…) ➢ Layout : sélectionner le format d'affichage parmi la liste proposée. ➢ Name : donner un nom à votre Node. Nous verrons ce qu'est le « group » à l'étape 5.	
3. Node « gauge » : ➢ Type : choisir le format d'affichage ➢ Label : donner un nom correspondant au type de la valeur de la donnée recueillie (température, humidité etc…) ➢ Units : donner l'unité de la valeur ➢ Range : donner la valeur minimale et maximale du graph. ➢ Sectors : définir des valeurs intermédiaires si besoin. ➢ Colour gradient : définir une couleur pour chaque tranche de valeur ➢ Name : donner un nom à votre Node. Nous verrons ce qu'est le « group » à l'étape 5.	

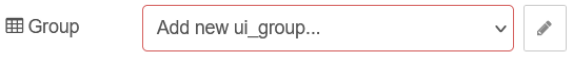
4. Node « chart » :
   - Label : donner un nom correspondant au type de la valeur de la données recueilli (température, humidité etc…)
   - Type : choisir le format d'affichage
     Cocher « enlarge point » pour voir les points sur le graph
   - X-axis : paramétrer l'échelle de temps
   - X-axis label : donner le format d'affichage
   - Y-axis : donner la valeur minimale et maximale de vos données
   - Series colour :
   - Blank label
   - Name : donner un nom à votre Node.

Nous verrons ce qu'est le « group » à l'étape 5.

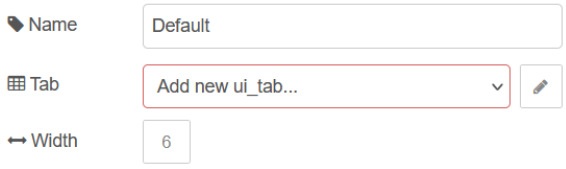


5. Les « groups » : ils sont composés de ce que nous allons appeler une « page » et d'une « colonne ». Le lien pour un dashboard contient plusieurs pages, et chaque page contient plusieurs colonnes.
   L'icône en haut à gauche du dashboard permet de sélectionner la page à afficher.
   Les graphiques et autre systèmes d'affichages vont se placer automatiquement les uns en dessous des autres dans une ou plusieurs colonnes.
   Ci-dessous la méthode pour créer votre « group ».

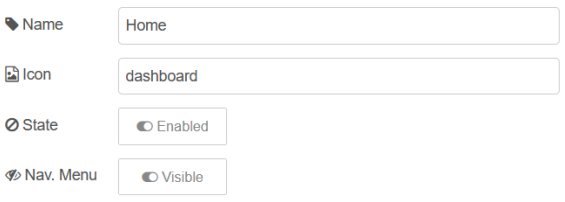
6. Sélectionner un ensemble « [page] Colonne » déjà existant dans le menu déroulant ou créer le vôtre en cliquant sur le crayon (voir ci-après).

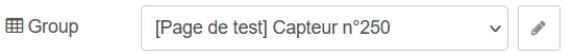
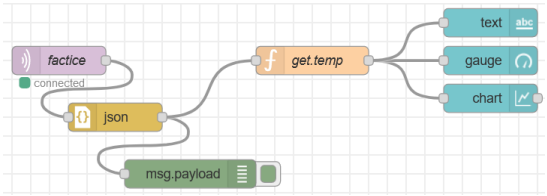
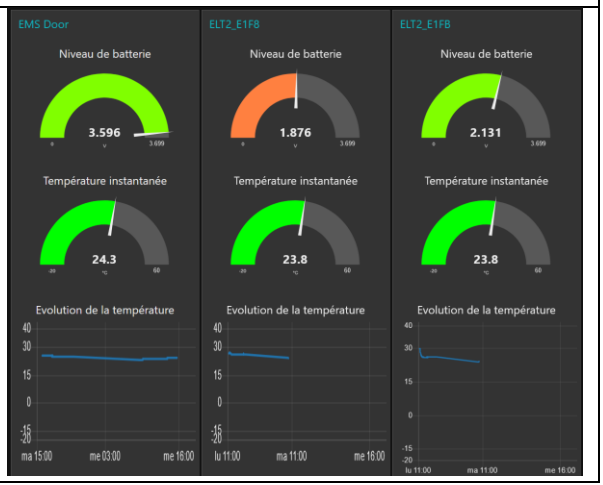


7. Définir le nom de votre colonne. Par exemple « Capteur n°250 ».
   Sélectionner dans « Tab » une page déjà existante ou créer la vôtre en cliquant sur le crayon (ci-après).
   Sélectionner la largeur de votre colonne.



8. Définir le nom de votre page. Par exemple « Page de tests ».
   Sélectionner l'icône que vous voulez (laisser par défaut est une bonne option).
   Cliquer sur « enabled » pour désactiver la page.



Cliquer sur « visible » pour rendre cachée la page.	
9. Cliquer sur « Add », puis de nouveau sur « Add ». Vous retrouvez la fenêtre principale avec le nom de votre dashboard.	
10. Vous pouvez maintenant relier ces Nodes à votre Node « function » puis appuyer sur « Deploy » en haut à droite pour sauvegarder. Les indicateurs bleus disparaissent et votre subscriber tente de se connecter au MQTT Broker. Une fois connecté, le voyant vert avec la mention « connected » s'affiche. Pour voir l'affichage, ouvrer un nouvel onglet et rentrer le lien suivant : @IPlocalhost :1800/ui	
11. Pour voir l'affichage, ouvrer un nouvel onglet et rentrer le lien suivant : @IPlocalhost :1800/ui Ci-contre un exemple de dashboard	

ANNEXE H : paramétrer InfluxDB

Tout d'abord il faut installer InfluxDB sur votre serveur. Pour se faire, il faut suivre les instructions de ce site :
https://docs.influxdata.com/influxdb/v1.8/introduction/install/

```
wget https://dl.influxdata.com/influxdb/releases/influxdb2-2.0.7-amd64.deb
```

```
sudo dpkg -i influxdb2-2.0.7-amd64.deb
```

Pour Ubuntu/Debian, ajouter le dossier InfluxData avec les commandes suivantes :

```
wget -qO- https://repos.influxdata.com/influxdb.key | gpg --dearmor > /etc/apt/trusted.gpg.d/influxdb.gpg

export DISTRIB_ID=$(lsb_release -si); export DISTRIB_CODENAME=$(lsb_release -sc)

echo "deb [signed-by=/etc/apt/trusted.gpg.d/influxdb.gpg] https://repos.influxdata.com/${DISTRIB_ID,,}
${DISTRIB_CODENAME} stable" > /etc/apt/sources.list.d/influxdb.list
```

Ensuite, installer et lancer InfluxDB :

```
sudo apt-get update && sudo apt-get install influxdb

sudo service influxdb start
```

Ou si vous êtes en système Linux (Ubuntu 15.04+, Debian 8+) :

```
sudo apt-get update && sudo apt-get install influxdb

sudo systemctl unmask influxdb.service

sudo systemctl start influxdb
```

InlfuxDB ne possède pas d'interface graphique. Pour utiliser cette application il faut nécessairement passer par des lignes de commandes qui seront détaillées ci-dessous.

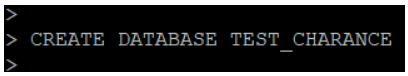
1. Créer une base de données

Pour démarrer l'application :

```
influx
```

Pour créer votre base de données :

```
CREATE DATABASE <database name>
```



Pour visualiser les différentes base de données enregistrées :

```
show databases
```

Pour définir l'utilisation d'une base de données :

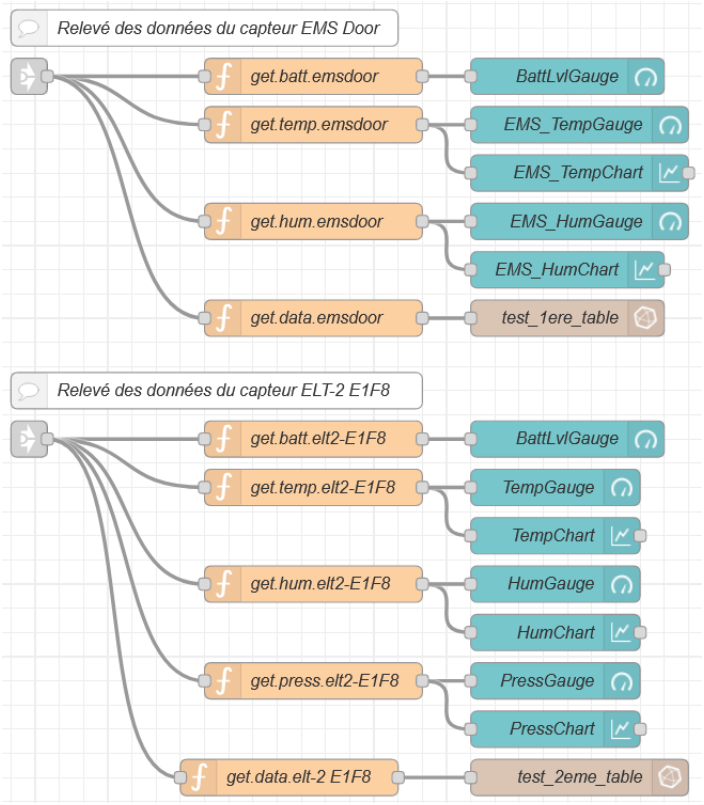
```
USE <database name>
```

```
> USE TEST_CHARANCE
Using database TEST_CHARANCE
>
```

2. Conception et remplissage de la base de données

Maintenant que la base de données est créée, il faut créer les tables : ce sont les « measurements ». Elles seront générées automatiquement via Node-Red.

Nous allons dans un premier temps paramétrer Node-Red pour qu'il puisse communiquer avec InfluxDB, puis nous verrons comment afficher les données dans InfluxDB.



Voici notre flow Node-Red pour les relevés des données que nous prendrons comme exemple.

Les deux Nodes grises à gauche sont des liens vers une autre flow qui gère la récupération des données (mqtt out, json, debug).

Cette interface est divisée en deux parties. La première en partant du haut concerne la récupération des données du capteur EMS Door, la seconde celles du capteur ELT-2.

Les codes dans les fonctions *get.data.emsdoor* et *get.data.elt-2 E1F8* sont différents des autres fonctions vu précédemment mais se basent sur le même principe :

get.data.emsdoor	**get.data.elt-2 E1F8**

var obj={
NomCapteur: msg.payload.deviceName,
Temperature: msg.payload.object.temperature,
Humidite: msg.payload.object.humidity,
Batterie: msg.payload.object.vdd,
 	
}
msg.payload=obj;
msg.topic="Filtre1";
return msg;
	var obj={
	NomCapteur: msg.payload.deviceName,
	Temperature: msg.payload.object.temperature,
	Humidite: msg.payload.object.humidity,
	Batterie: msg.payload.object.vdd,
	Pression: msg.payload.object.pressure,
	}
	msg.payload=obj;
	msg.topic="Filtre1";
	return msg;

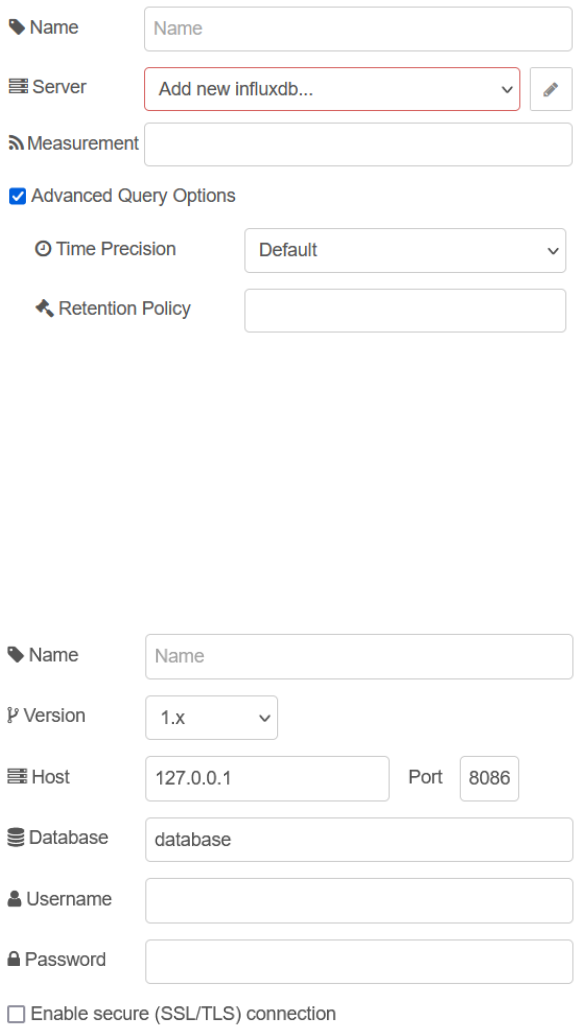
Ces deux fonctions sont connectées à leur sortie à des Nodes « InfluxDB out » qui sont rangées dans le dossier « storage ».

Voici la configurations de ces Nodes :

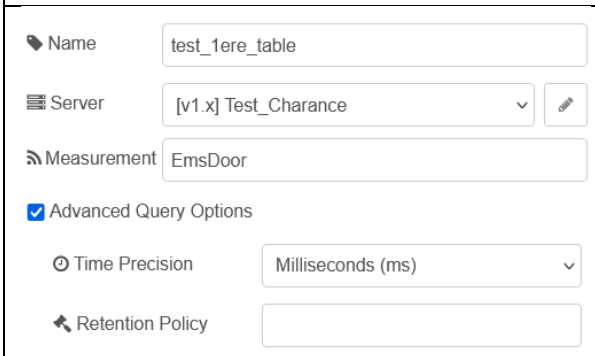
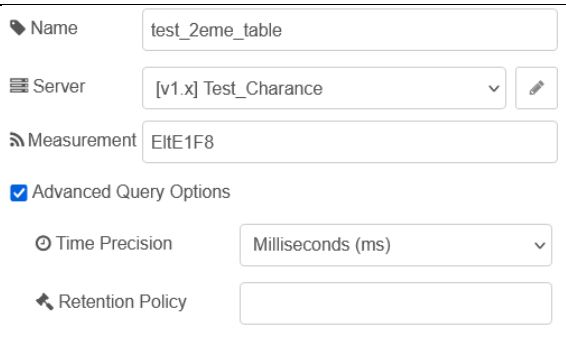
1. Double cliquer sur la Node pour ouvrir son menu de configuration.
   - Cocher « advanced query option »
   - Name : nom de la Node
   - Server : sélectionner une base de données déjà existante ou en créer une autre (cf. étape suivante)
   - Measurement : nom de la table dans InfluxDB dans laquelle vous voulez envoyer les données (NE PAS METTRE D'ACCENTS OU DE CARACTERES SPECIAUX) => *exemple : EmsDoor*
   - Time precision : configuration de la précision temporel du relevé de donnée
   - Retention policy : laisser vide

2. Créer un base de données en cliquant sur le crayon :
   - Name : nom de la bdd dans l'affichage de Node Red
   - Version : sélectionner la version d'InfluxDB
   - Host & Port : laisser par défaut
   - Database : le nom de la bdd dans InfluxDB (NE PAS METTRE D'ACCENTS OU DE CARACTERES SPECIAUX) => *exemple : TEST_CHARANCE*
   - Username & Password : identifiant et mot de passe pour se connecter à la base de données (optionnel)
   - Cocher « enable secure » pour ???

3. Cliquer sur « Add » puis sur « Done ».



Avec l'exemple vu plus haut, nous avons deux tables pour une seule base de données.
La première s'appelle « EmsDoor » et la seconde « EltE1F8 ». Les Nodes correspondantes sont paramétrées comme suit :

EmsDoor (node : test_1ere_table)	EltE1F8 (node : test_2eme_table)
	

Effacer les trames reçues dans Node-Red (menu debug à droite) puis laisser assez de temps pour que de nouvelles trames soient reçues et traitées au moins une fois. Une fois que les trames s'affichent, revenez dans vos ligne de commande et passer aux étapes suivantes.

3. Affichages des valeurs

Afficher les tables :

```
show measurements
```

Afficher les types de données de chacune des tables :

```
show field keys
```

Nous remarquerons que les tables et les variables se sont générées automatiquement.

Afficher le contenu intégrale d'un table :

```
select * from <measurements name> limit 10
```

Le « * » signifie que l'on prendra tous les types de valeur (température, humidité etc…). Si vous souhaitez avoir uniquement un champ en particulier, il faut remplacer le * par le champ en question.
Le « limit 10 » est optionnel est permet de limiter l'affichage à 10 valeurs. Ne pas le mettre si vous souhaitez afficher absolument toutes les valeurs.

Supprimer une table :

```
drop measurement <measurements name>
```

Si le flow Node-Red correspondant n'est pas désactivé, dès réception de la trame suivante, la table sera de nouveau créée.

Supprimer une base de données :

```
drop database <database name>
```

ANNEXE I : paramétrer Grafana

Tout d'abord il faut installer Grafana sur votre serveur. Pour se faire, il faut suivre les instructions de ce site :
https://grafana.com/docs/grafana/latest/installation/debian/
https://grafana.com/docs/grafana/latest/installation/upgrading/

ANNEXE J : paramétrer PostgreSQL

BIBLIOGRAPHIE