

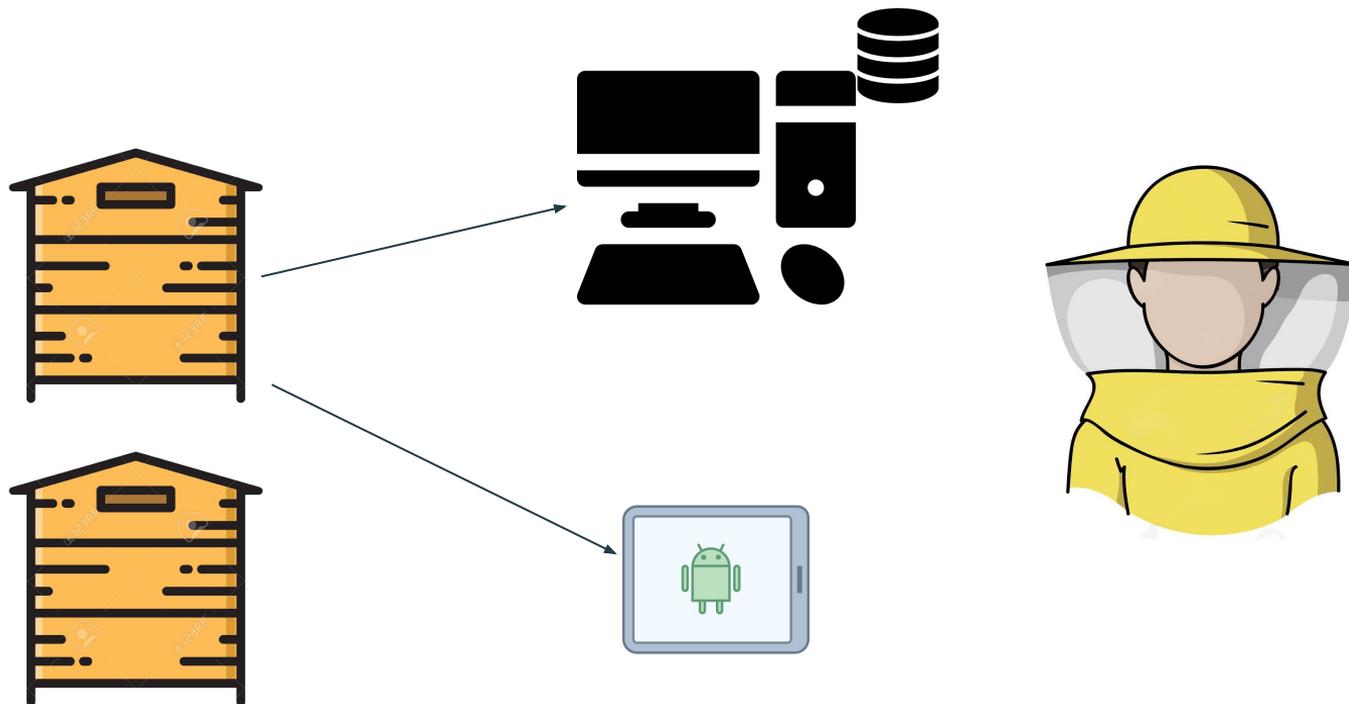
# Projet Bee-Honey't Revue Finale

Enzo Rossi  
BTS SN-IR  
La Salle Avignon

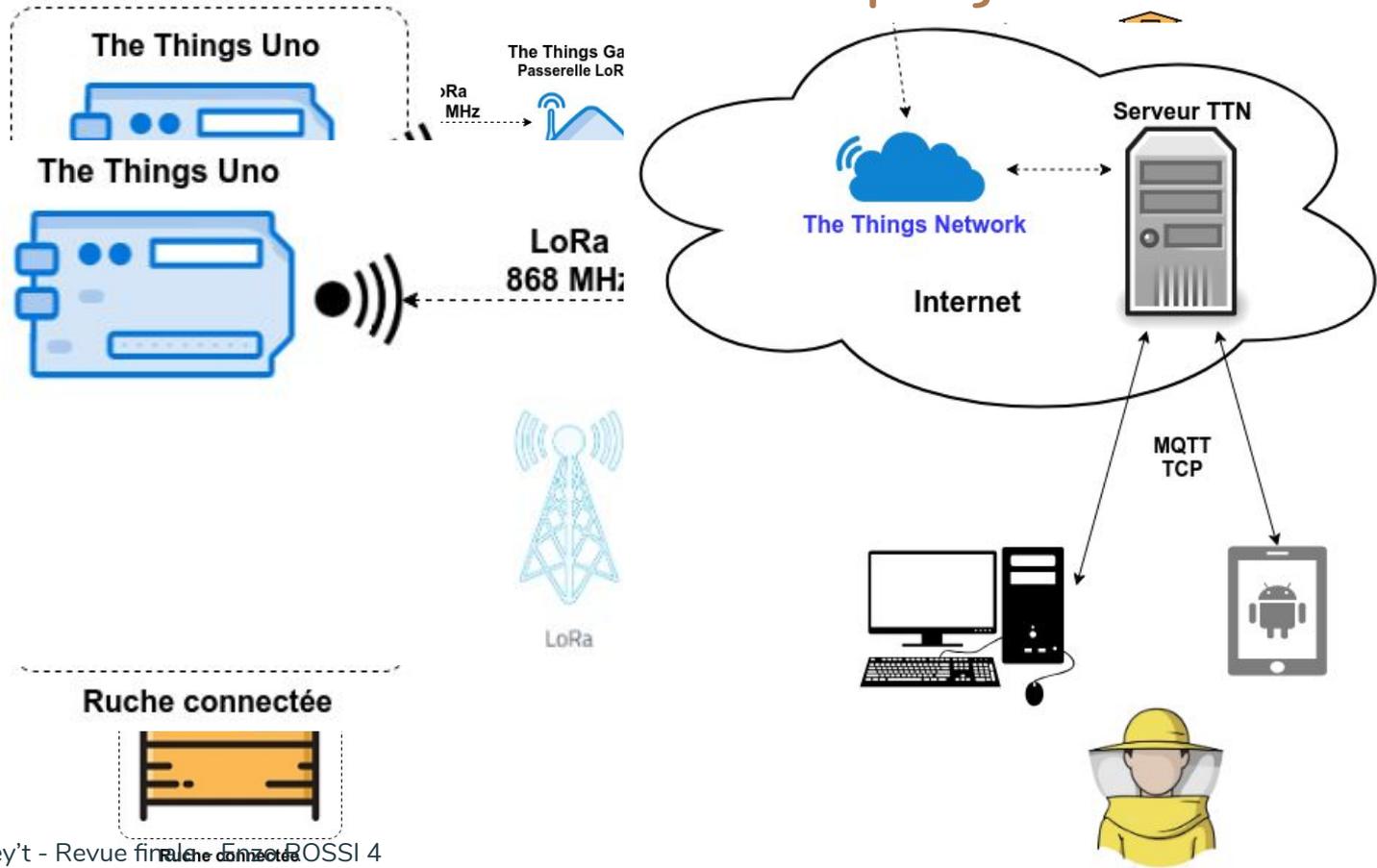
# Sommaire

- Expression du besoin
- Présentation du projet
- Cahier des charges
- Ressources logicielles
- Répartition des itérations
- Planification du projet
- Convention de nommage
- Partie personnelle : Diagramme de cas d'utilisation (Apiculteur)
- IHM
- Spécifications techniques
- Envoie des données
- Récupération des données
- Extraire les données
- Diagramme de classes
- Recevoir et afficher les données de température et d'humidité intérieures de la ruche
- Réception et enregistrement des mesures intérieures
- Enregistrement des mesures intérieures toutes les heures
- Création et suppression d'une ruche
- Tests de validation
- Transmission sans fil
- Conclusion

# Expression du besoin



# Présentation du projet



# Cahier des charges

Étudiant 3  
IR

Enzo ROSSI

- Gérer les ruches : Paramétrer une nouvelle ruche
- Consulter les données d'une ruche (température, humidité, pression atmosphérique, et ensoleillement)
- Recevoir les données des ruches
- Enregistrer les données dans la base de données

# Ressources logicielles du projet

Désignation	Caractéristiques
Système d'exploitation du PC	GNU/Linux Ubuntu 16.04 LTS
Base de données	MySQL
Logiciel de gestion de versions	subversion (RiouxSVN)
Générateurs de documentation	Doxygen version 1.8
Environnement de développement	Qt Creator et Qt Designer
API GUI	Qt 5.5.1

# Répartition des itérations

Tâche	Priorité	Itération
Recevoir les données des ruches	<b>haute</b>	<b>1</b>
Consulter les données d'une ruche (température, humidité, pression atmosphérique, et ensoleillement)	<b>moyenne</b>	<b>1</b>
Gérer les ruches : Paramétrer une nouvelle ruche	<b>haute</b>	<b>2</b>
Enregistrer les données dans la base de données	<b>moyenne</b>	<b>3</b>

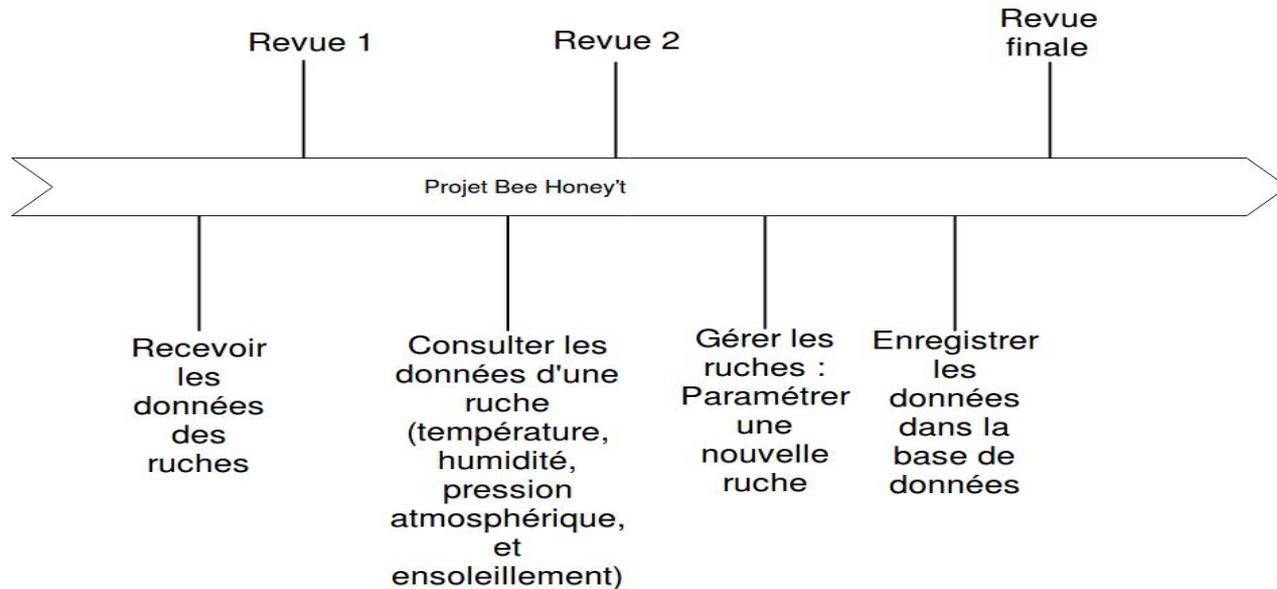
# Planification du projet

Bonjour ! Nous avons apporté une modification à Trello. Vous pouvez toujours créer un nombre illimité de tableaux personnels et utiliser tous vos tableaux actuels sans interruption. Toutefois, nous avons modifié le nombre de tableaux que vous pouvez créer dans une équipe avec la version gratuite.

**ruche** | ruche | Free | Visible par les membres d'une équipe | LC E DM F T3 6 | Inviter

A faire IR	En cours IR	Terminé IR	A faire EC	En cours EC	Terminé EC
Archiver dans la base de données les données recueillies 🕒 27 mai	Gérer la planification des tâches 🕒 27 mai	Réalisation diagramme de classes 🕒 8 févr.	Programme: Luminosité ou ensoleillement 🕒 28 mai	Diagramme de bloc ( Général ) 🕒 27 mai	Programme: Temp (DHT22) 🕒 28 févr.
Gérer les ruches : Paramétrer les alertes 🕒 27 mai	Gérer les ruches : Paramétrer une nouvelle ruche 🕒 27 mai	Réalisation d'une maquette IHM PC 🕒 22 mars	Programme: Antivol 🕒 27 mai	Choix de tout les composants 🕒 27 mai	Programme: Pres (BMP280) 🕒 27 mai
Créer les classes alertes et configurer les différentes alertes 🕒 27 mai	Consulter les données d'une ruche (température, humidité, pression atmosphérique, et ensoleillement) 🕒 25 mars	Recevoir les données des ruches 🕒 22 mars	Transmettre les données au PC 🕒 27 mai	Mesurer poids de la ruche 🕒 27 mai	
Consulter les données d'une ruche (poids, niveau de charge, tension et courant de la batterie) 🕒 27 mai	Consulter les données d'une ruche 🕒 27 mai	Création de la convention de nommage 🕒 27 mai	Alerter d'une défaillance de batterie 🕒 27 mai	+ Ajouter une autre carte	+ Ajouter une autre
Enregistrer les données collectées 🕒 27 mai	+ Ajouter une autre carte	Tests de mise en œuvre de MQTT 🕒 3/3	Recharge de la batterie (option) 🕒 27 mai		
Déclencher les alertes 🕒 27 mai		Mettre en oeuvre MQTT : ajout des différents répertoire / mise en place du système de requête et de suscribe 🕒 27 mai	Définition des blocs ( interne ) 🕒 27 mai		
			Acquérir le niveau de charge, tension et courant de la batterie de la ruche. 🕒 27 mai		

# Planification du projet



# Convention de nommage

Nom des classes : **NomDesClasses**

Nom des variables (attributs) : **nomDesVariables** (un nom et un qualificatif si besoin)

Nom des fonctions (méthodes) : **nomDesFonctions** (un verbe à l'infinitif ou à l'indicatif)

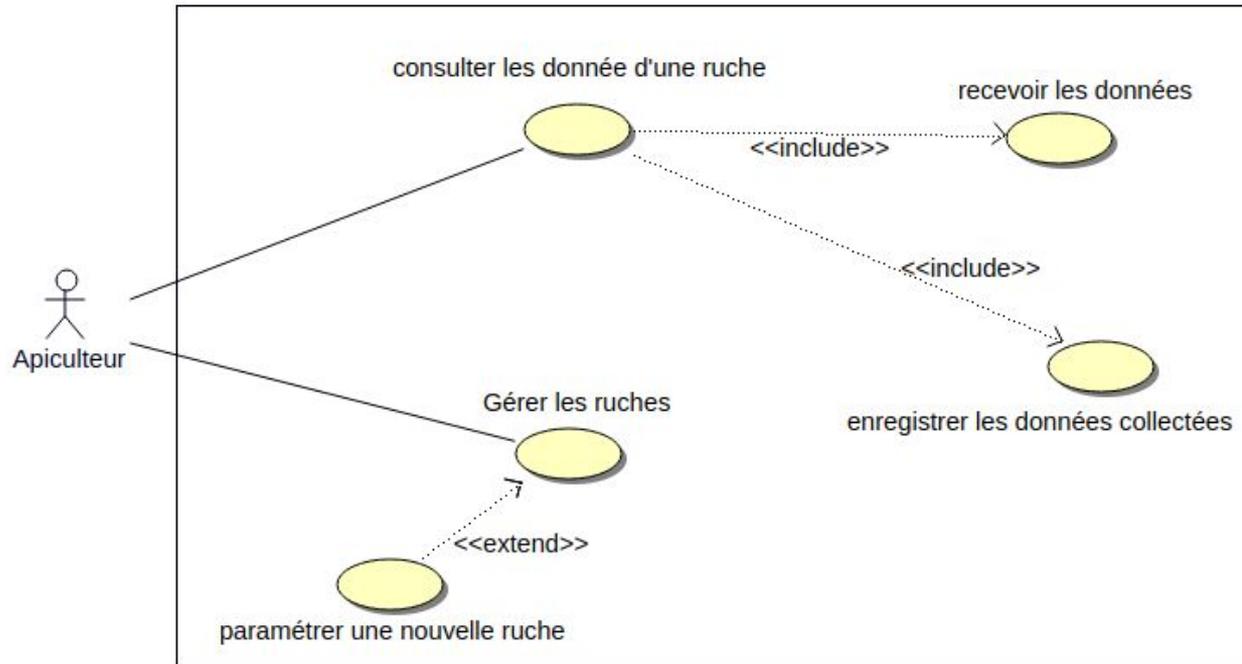
Nom des fichiers : **nomDesFichiers**

Exemple :

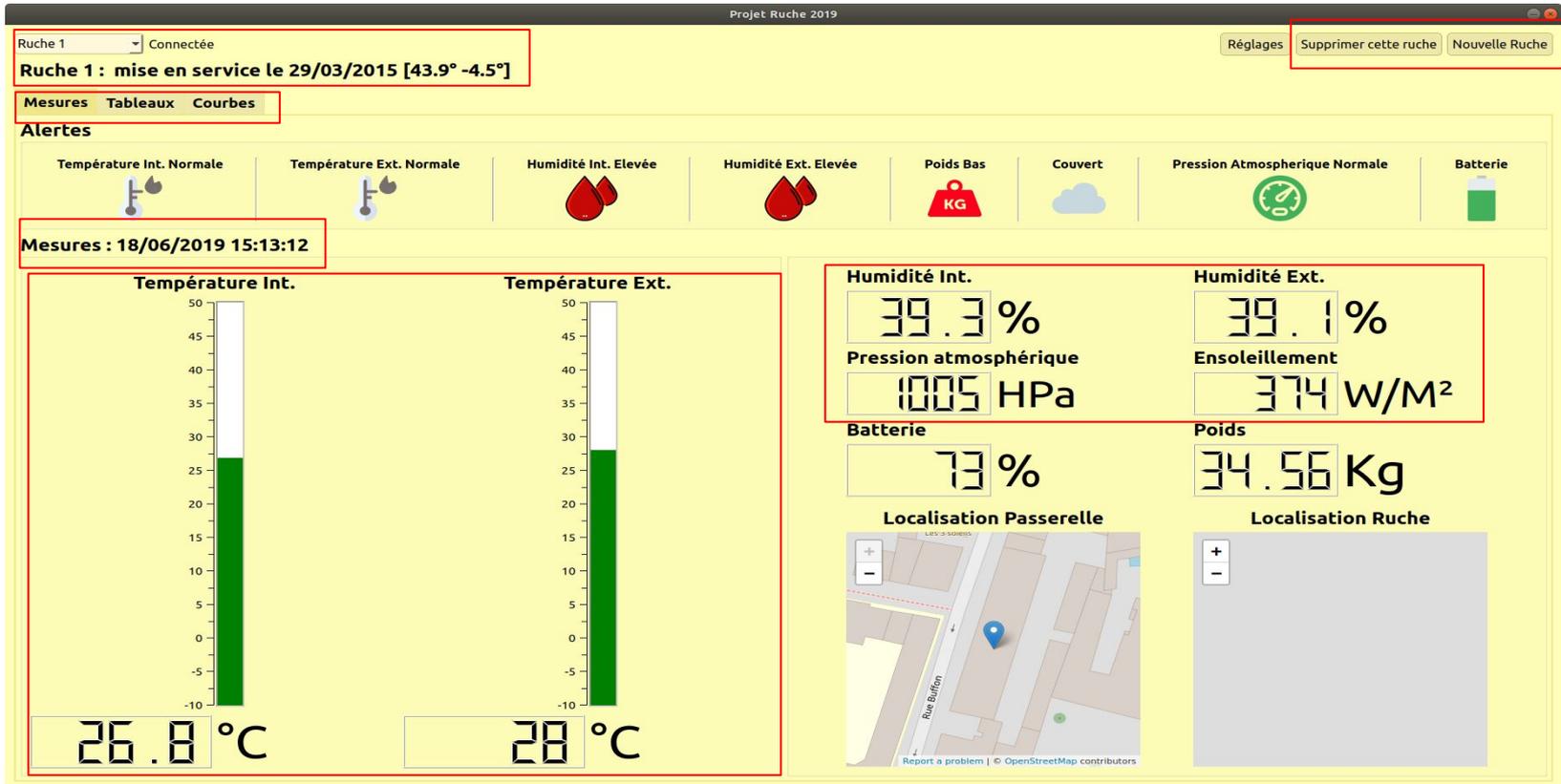
```
void Ruche::extraireDonnees(PortsTTn port, QByteArray messageTTN)
```

# Partie personnelle

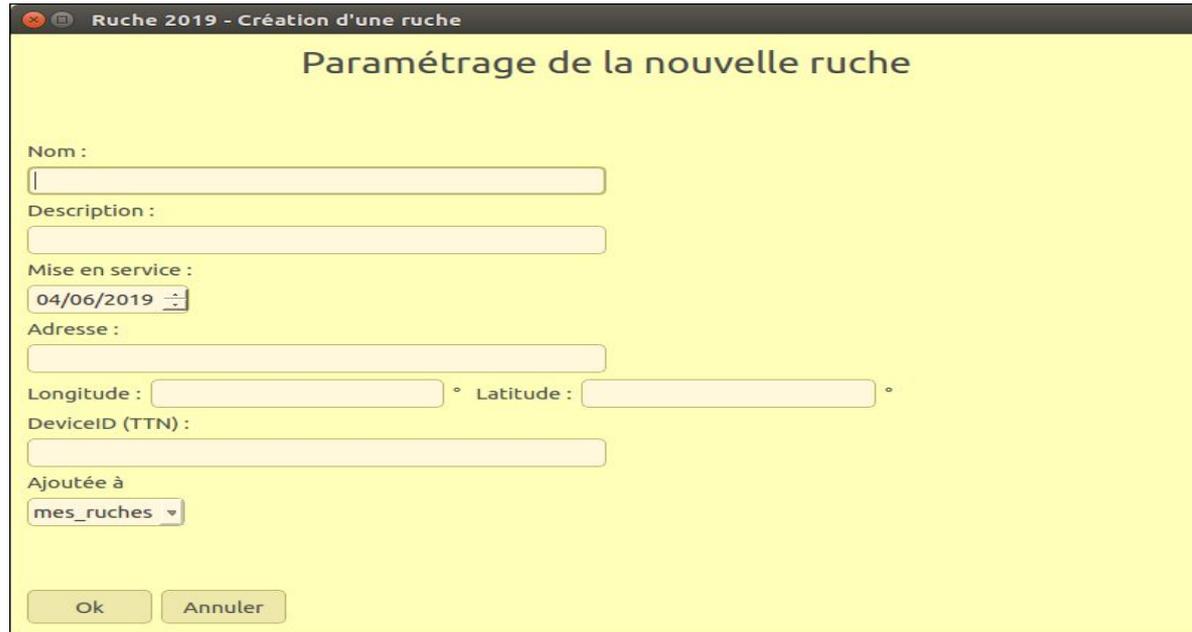
## Diagramme de cas d'utilisation Apiculteur



# IHM version Finale



# IHM version Finale



Ruche 2019 - Création d'une ruche

## Paramétrage de la nouvelle ruche

Nom :

Description :

Mise en service :

Adresse :

Longitude :  ° Latitude :  °

DeviceID (TTN) :

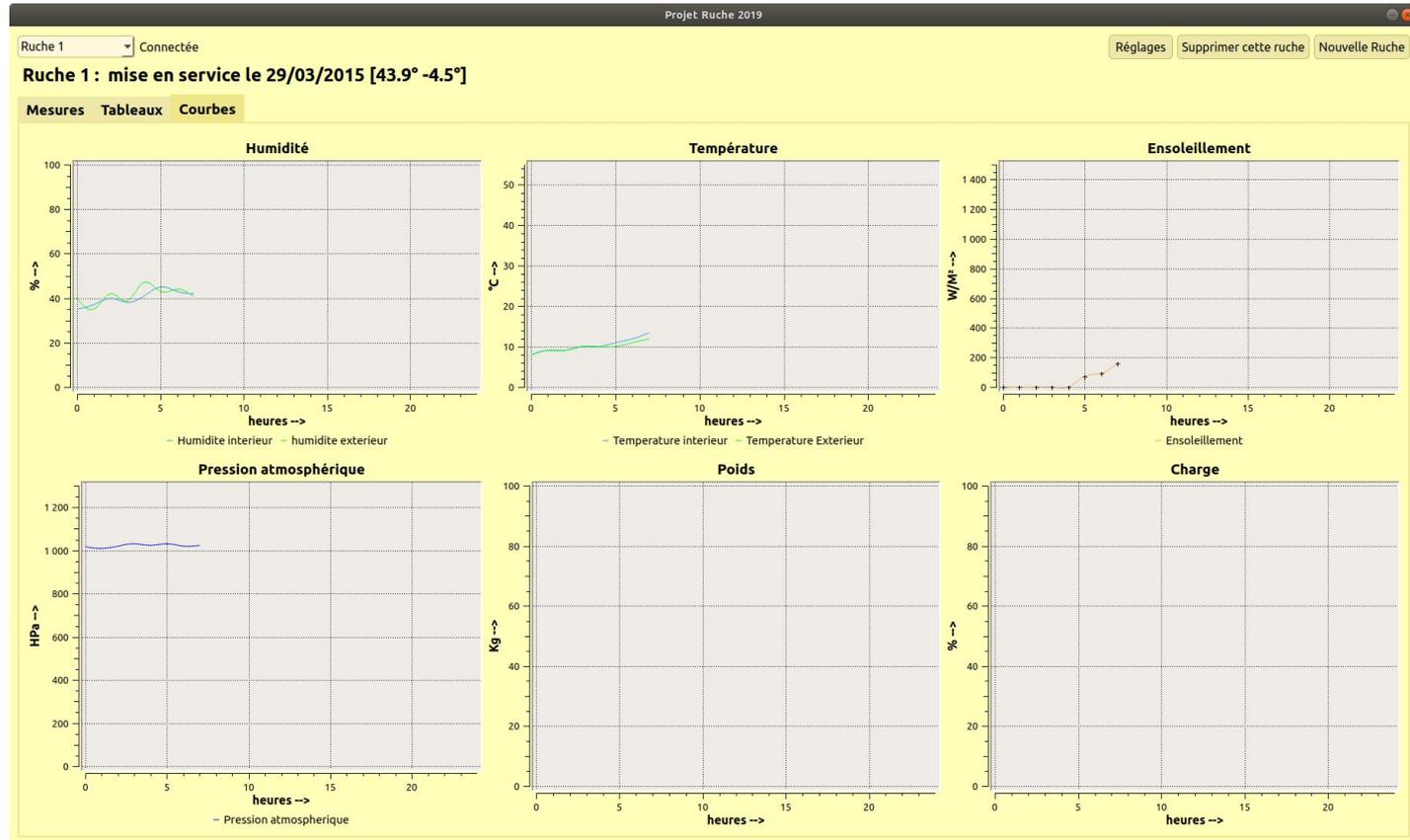
Ajoutée à

# IHM version Finale

The screenshot displays the 'Projet Ruche 2019' application window. At the top, it shows 'Ruche 1' selected in a dropdown menu, followed by 'Connectée'. On the right, there are three buttons: 'Réglages', 'Supprimer cette ruche', and 'Nouvelle Ruche'. Below this, the text 'Ruche 1 : mise en service le 29/03/2015 [43.9° -4.5°]' is displayed. A navigation bar contains 'Mesures', 'Tableaux', and 'Courbes', with 'Mesures' currently selected. Underneath, a dropdown menu is set to 'Mesures intérieures'. The main content area features a table with the following data:

	Temperature	Temperature minimum	Temperature maximum	Humidite	Humidite minimum	Humidite maximum	Heure
1	8	7	9	35	30	40	00:00:00
2	9	8	10	37	35	39	01:00:00
3	9	8	10	40	42	38	02:00:00
4	10	9.5	10.5	38	36	40	03:00:00
5	10	8	12	41	40	39	04:00:00
6	11	10.5	11.5	45	35	40	05:00:00
7	12	11	13	43	40	46	06:00:00
8	13.5	13	14	42	41	43	07:00:00

# IHM version Finale



# Spécifications techniques

## tableau Fports

Fport	Nom	Mesures	IR
1	portMesureEnergie	tension,courant,charge	MELLAH F.
2	portMesurePoids	poids	MELLAH F.
3	<b>portMesureRuche</b>	<b>température,humidité (intérieur)</b>	<b>Enzo ROSSI</b>
4	<b>portMesureEnvironement</b>	<b>température, humidité, pression atmosphérique(extérieur)</b>	<b>Enzo ROSSI</b>
5	<b>portMesureEnsoleillement</b>	<b>ensoleillement</b>	<b>Enzo ROSSI</b>

# Envoie des données



Serveur TTN

Données au format  
JSON



```
{  
  "app_id": "mes_ruches",  
  "dev_id": "ruche_1",  
  "port": 3,  
  ...  
  "metadata": { "..."}  
}
```



Application cliente

# Récupération des données

## Solution n°1 : non retenue

**Température** intérieure  
**Humidité** intérieure  
codées sur 16 bits

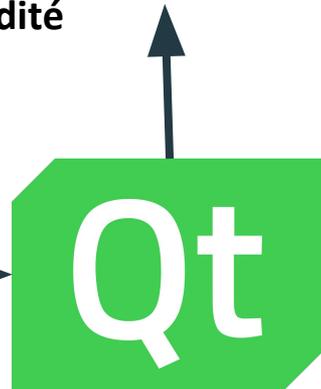
↓  
"port": 3



```
{  Données encodées en Base64
  ...
  "payload_raw": "CAwSwA=="
  ...
}
```

Chaque application cliente :

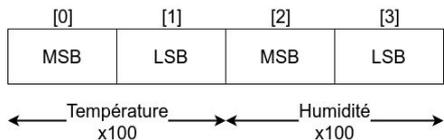
1. extrait le champ "payload\_raw"
2. décode la valeur **Base64**
3. extrait la **température** et l'**humidité**



# Récupération des données

## Solution n°2 : retenue

### Température et Humidité codées sur 16 bits



```
function Decoder(bytes, port) {  
  var decoded = {};  
  switch (port) {  
    case 3: // DHT22  
      decoded.temperature = ((bytes[0]<<8) + bytes[1])/100;  
      decoded.humidite = ((bytes[2]<<8) + bytes[3])/100;  
      break;  
    }  
  return decoded;  
}
```

Chaque application cliente :

1. extrait l'objet "payload\_fields"
2. extrait les champs température et l'humidité



```
{ ...  
  "payload_fields":  
  {  
    "humidite":48,  
    "temperature":20.6  
  }  
  ...  
}
```



# Extraire les données

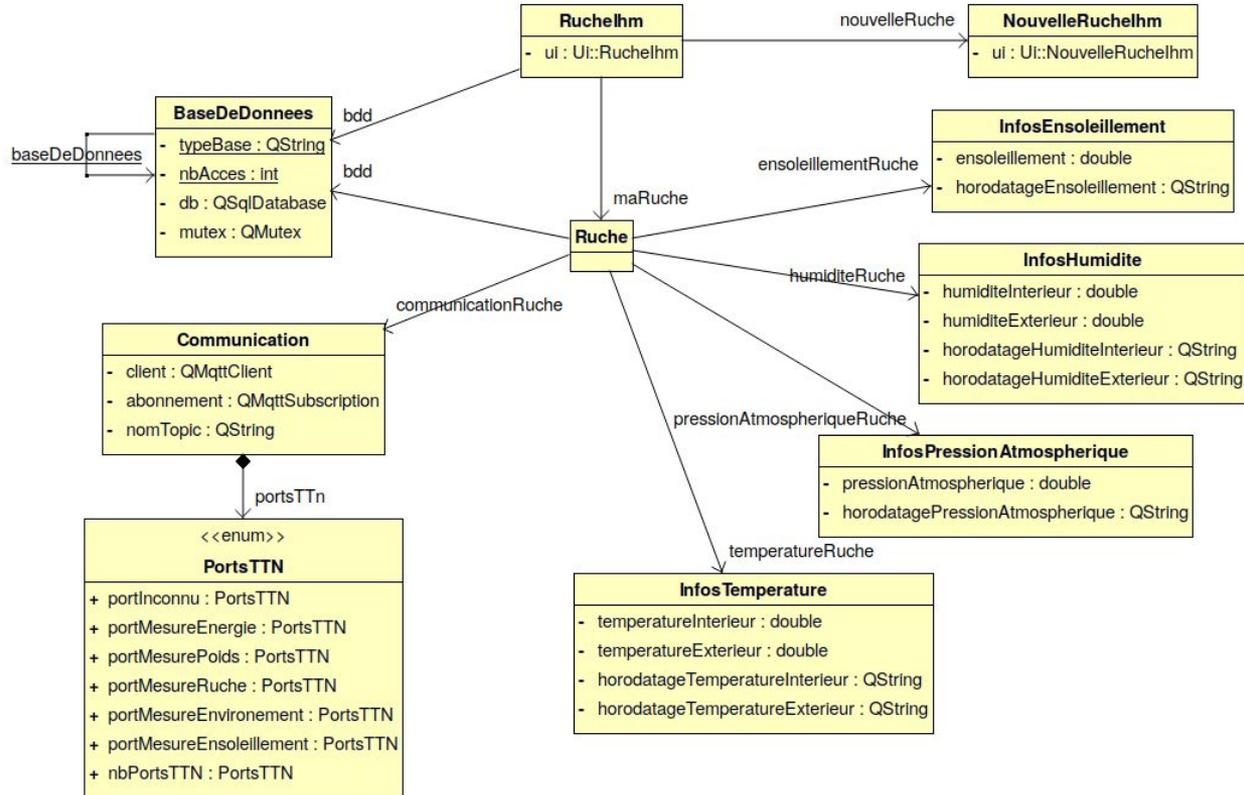


```
{ ...  
  "payload_fields":  
  {  
    "humidite":48,  
    "temperature":20.6  
  }  
  ...  
}
```

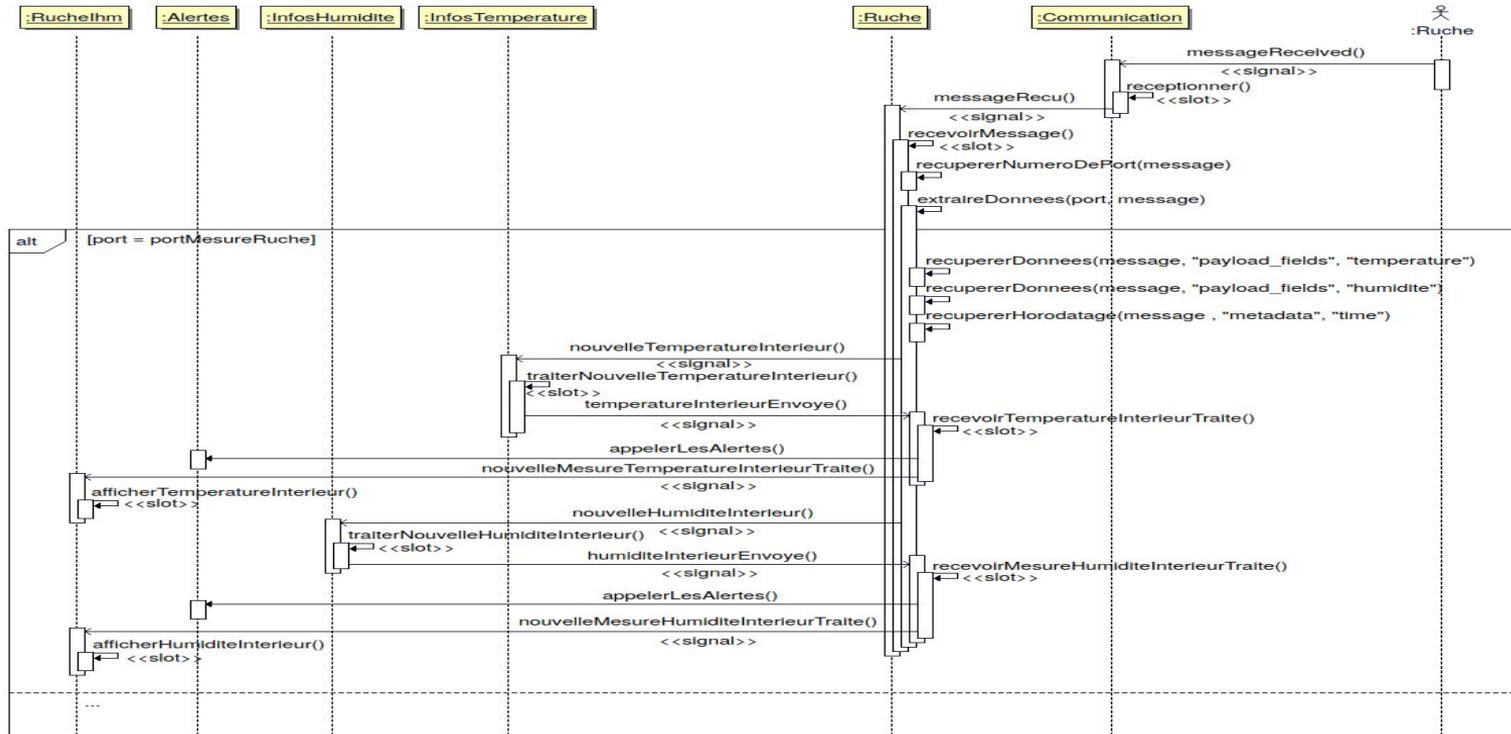


```
QJsonDocument documentJSON = QJsonDocument::fromJson(message);  
  
QJsonObject objetJSON = documentJSON.object();  
  
objetJSON.value("payload_fields").toObject().value("temperature").toDouble();
```

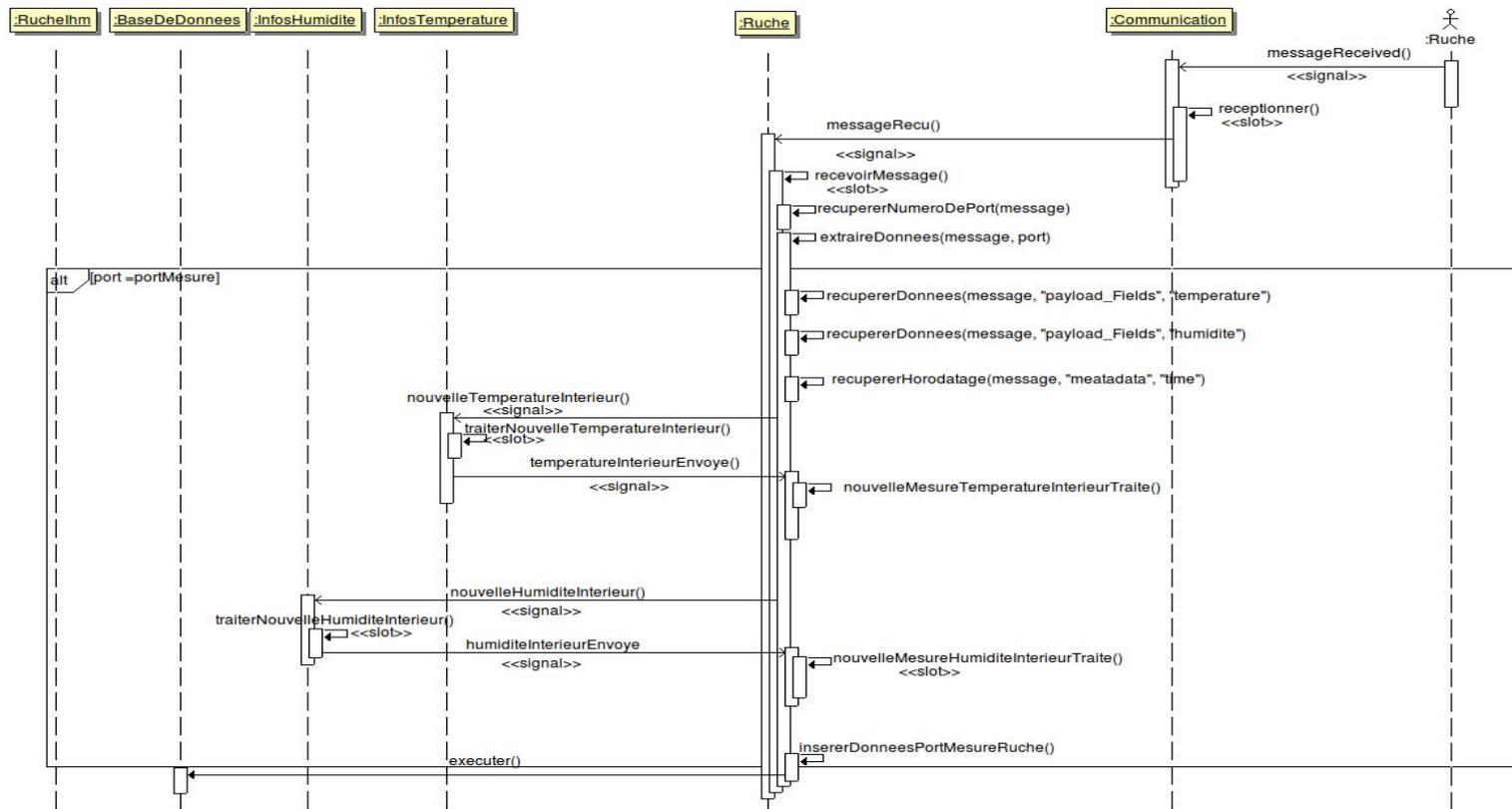
# Diagramme de classes



# Recevoir et afficher les données de température et d'humidité intérieures de la ruche



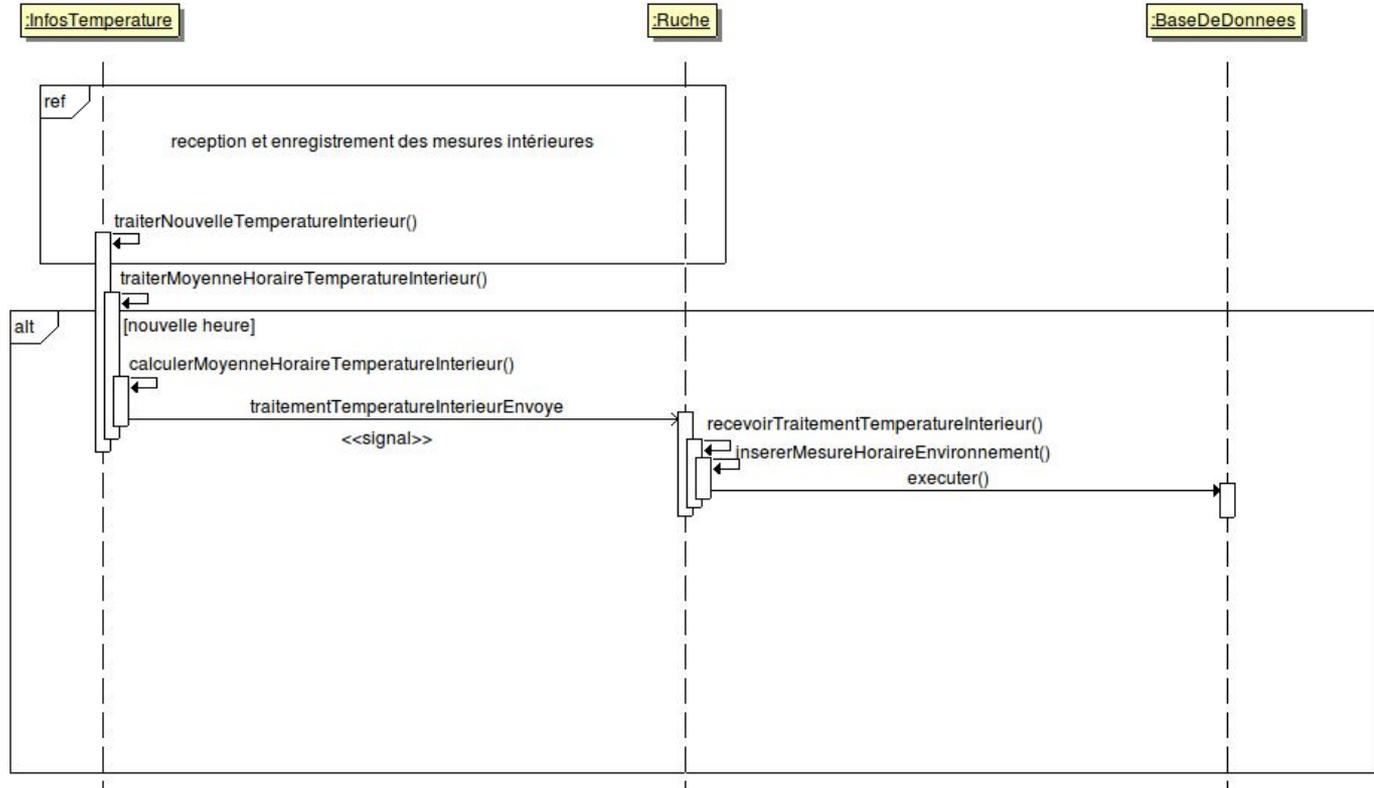
# Réception et enregistrement des mesures intérieures



# Requête Sql

```
INSERT INTO MesuresRuche (idRuche, Temperature, Humidite, Horodatage)  
VALUES (...)
```

# Enregistrement des mesures intérieures toutes les heures



# Requêtes Sql

Enregistrement :

```
INSERT INTO MesuresJournalieresRuche (MesuresJournalieresRuche.idRuche,  
MesuresJournalieresRuche.Temperature, MesuresJournalieresRuche.TemperatureMin,  
MesuresJournalieresRuche.TemperatureMax, MesuresJournalieresRuche.Humidite,  
MesuresJournalieresRuche.HumiditeMin, MesuresJournalieresRuche.HumiditeMax,  
MesuresJournalieresRuche.DateMesure, MesuresJournalieresRuche.HeureMesure) VALUES (...)
```

Affichage:

```
SELECT Temperature, Humidite, HeureMesure FROM MesuresJournalieresRuche WHERE DateMesure = '' +  
dateCourante.toString("yyyy-MM-dd") + '' AND idRuche = '' + mesRuches[positionDeLaRuche].at(0)  
+ '' ORDER BY HeureMesure ASC";
```

# Création et suppression d'une ruche

```
INSERT INTO Ruche INSERT (idTTN, Nom, Description, DateMiseEnService, Adresse, Longitude, Latitude, DeviceID) VALUES (...)
```

```
INSERT INTO Seuils (idRuche, TemperatureIntMin, TemperatureIntMax, HumiditeIntMin, HumiditeIntMax, TemperatureExtMin, TemperatureExtMax, HumiditeExtMin, HumiditeExtMax, PressionMin, PressionMax, PoidsMin, PoidsMax, EnsoleillementMin, EnsoleillementMax, Charge) VALUES (...)
```

```
SELECT Ruche.idRuche, Ruche.Nom, Ruche.DeviceID, TTN.idTTN, TTN.Hostname, TTN.Port, TTN.Username, TTN.Password, TTN.ApplicationID, Ruche.Adresse, Ruche.DateMiseEnService, Ruche.Longitude, Ruche.Latitude FROM Ruche INNER JOIN TTN ON Ruche.idTTN = TTN.idTTN;
```

```
DELETE FROM Ruche WHERE Ruche.DeviceId = "id de la ruche sélectionnée"
```

# Tests de validation

Désignation	Résultat attendu	Oui / Non
Recevoir les données des ruches	Recevoir les données des ruches	Oui
Consulter les données d'une ruche (température, humidité, pression atmosphérique, et ensoleillement)	Visualiser les mesures avec leur unité sur l'IHM	Oui
Gérer les ruches : Paramétrer une nouvelle ruche	Créer ou supprimer une ruche	Oui
Enregistrer les données de (température, humidité, pression atmosphérique, et ensoleillement)	Enregistrer les données dans la base de données	Oui

# Transmission sans fil

	Courte portée			Moyenne portée			Longue portée	
Technologie	NFC	Bluetooth	Zigbee	Z-Wave	Wi-Fi	BLE	SigFox	LoRa
Portée moyenne (en intérieur)	<10 cm	10 m	10 m	50 m	50 m	50 m	>2km	>2km
Débit (Mbit/s)	$1.10^{-3}$	$1.10^{-3}$	$1.10^{-2}$	$1.10^{-2}$	$1.10^2$	$1.10^{-3}$	$1.10^{-3}$	$1.10^{-3}$
Autonomie	Mois	Jours	Années	Années	Jours	Mois	Années	Années
Fréquence	2,4 GHz	2,4 GHz	2,4 GHz 868 MHz	868 MHz	2,4 GHz 5 GHz	2,4 GHz	868 MHz	868 MHz
Usages	Téléphonie Cartes de paiement	Périphériques informatiques et multimédia	Domotique		Navigation Internet Transferts conséquents de données	Périphériques informatiques et multimédia	Prévention d'incidents Collecte de données Gestion de réseaux	

# Conclusion

## **Tâches : prochaine itération 1.1**

*Intégrer les seuils des alertes aux différentes courbes*

Je vous remercie de votre écoute