

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

Département d'Electronique

Mémoire de projet de fin d'études  
pour l'obtention du diplôme d'ingénieur d'état en électronique

# Implémentation d'une plateforme Smart-City dédiée au Smart-Metering

AITSAID Abderrahman  
KENDI Mohamed Amine

Sous la direction de M. Riad HARTANI Dr.

Présenté et soutenu publiquement le (18/06/2018)

Composition du jury :

Président	M. Adel BELOUHRANI	Prof.	ENP
Promoteur	M. Riad HARTANI	Dr.	Padovani Ventures
Promoteur	M. Rabah SADOON	Dr.	ENP
Examineur	M. Mourad ADNANE	Dr.	ENP

ENP 2018



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

Département d'Electronique

Mémoire de projet de fin d'études  
pour l'obtention du diplôme d'ingénieur d'état en électronique

# Implémentation d'une plateforme Smart-City dédiée au Smart-Metering

AITSAID Abderrahman  
KENDI Mohamed Amine

Sous la direction de M. Riad HARTANI Dr.

Présenté et soutenu publiquement le (18/06/2018)

Composition du jury :

Président	M. Adel BELOUHRANI	Prof.	ENP
Promoteur	M. Riad HARTANI	Dr.	Padovani Ventures
Promoteur	M. Rabah SADOON	Dr.	ENP
Examineur	M. Mourad ADNANE	Dr.	ENP

ENP 2018

## ملخص

مع التقدم التكنولوجي واندلاع إنترنت الأشياء على مدى العقد الماضي ، ظهر مفهوم جديد ، المدن الذكية. المدينة الذكية هي مدينة تستخدم التكنولوجيا لحل مشاكلها.

يتعلق هذا المشروع بدراسة وتنفيذ منصة ذكية للمدن مخصصة للقياس الذكي. ستسمح هذه المنصة بإدارة أفضل للموارد (المياه والغاز والكهرباء) التي تمثل واحدة من المشاكل الرئيسية للمدن الذكية ، وذلك بفضل المراقبة الحقيقية لاستهلاك المشتركين.

**الكلمات الدالة:** أنترنات الأشياء, المدن الذكية, القياس الذكي, منصة.

## Abstract

With the technological advances and the outbreak of the Internet of Things over the past decade, a new concept has emerged, Smart Cities. A smart city is a city that uses technology to solve its problems.

This project concerns the study and implementation of a Smart-city platform dedicated to Smart-metering. This platform will allow better management of resources (Water, Gas, and Electricity) which represents one of the main problems of smart cities, thanks to a real-time monitoring of the client's consumption.

**Key words:** Internet of things, smart-city, smart-metering, platform.

## Résumé

Avec Les avancés technologique et l'explosion de l'Internet des objets au cours la dernière décennie, un nouveau concept est apparue, les villes intelligentes. Une ville intelligente est une ville qui utilise la technologie pour résoudre ses problèmes.

Ce projet porte sur l'étude et l'implémentation d'une plateforme Smart-city dédiée au Smart-metering. Cette plateforme permettra une meilleure gestion des ressources (Eau, Gaz, Electricité) qui représente l'une des principales problématiques des villes intelligentes, grâce à un suivi en temps réel de la consommation des clients.

**Mots clés:** Internet des objets, smart-city, smart-metering, plateforme.

## *Dédicace*

*On dédie ce travail à nos chers parents et grands parents,*

*A nos familles,*

*A nos frères et soeurs,*

*A nos amis*

*Et à tous ceux qui ont fait confiance à Nos Capacités  
pendant notre parcours.*

# *Remerciements*

On remercie Dieu le Tout Puissant de nous avoir donné le courage, la santé, et Nous a accordé son soutien durant les périodes les plus difficiles.

On remercie vivement Notre encadreur, Monsieur **Rabah SADOUN** de nous avoir aidé dans notre travail, et surtout d'avoir cru à nos compétences.

On remercie également Monsieur **Riad HARTANI** de nous avoir donné la chance de travailler sur un projet d'une importance capitale pour notre pays qui est le projet "Alger Smart City".

On remercie aussi la secrétaire du wali d'Alger Madame **Fatiha SLIMANI** et le directeur de la clientèle à SEAAL Monsieur **Sophiane AZZIZI** pour leur encouragement et leur disponibilité.

Nos vifs remerciements vont également aux membres du jury, **M. Adel BELOUHRANI** et **M. Mourad ADNANE** pour l'intérêt qu'ils ont porté à notre travail en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

On remercie chaleureusement nos familles et tous nos amis pour leur soutien et leurs encouragements.

Enfin, nous tenons également à remercier toute personne ayant participé de près ou de loin à la réalisation de ce travail.

# Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

Introduction générale 12

**I État de l’art et concepts 14**

**1 Internet des objets (IoT) 15**

1.1 Introduction . . . . . 16

1.2 Définition et historique . . . . . 16

1.3 Principe et architecture . . . . . 17

1.4 Couche physique : les objets . . . . . 18

1.5 Couche réseau . . . . . 18

1.5.1 Les réseaux LPWA . . . . . 19

1.5.1.1 Sigfox . . . . . 20

1.5.1.2 LoRa . . . . . 22

1.5.1.3 Telensa UNB . . . . . 24

1.5.1.4 Random Phase Multiple Access (RPMA) . . . . . 25

1.5.1.5 Qowisio . . . . . 25

1.5.2 Les réseaux LPWA 3GPP . . . . . 26

1.5.2.1 NB-IoT . . . . . 26

1.5.3 M-Bus . . . . . 28

1.5.4 Autres solutions . . . . . 29

1.6 Couche Applications . . . . . 29

1.7 Conclusion . . . . . 29

**2 Etat de l’art du Smart-Metering 30**

2.1 Introduction . . . . . 31

2.2 Le Smart-Metering dans le monde . . . . . 31

2.3 Smart Metering : AMR/AMI . . . . . 33

2.3.1 AMR - Lecture automatique des compteurs . . . . . 33

2.3.2 AMI - Infrastructure de compteurs avancée . . . . . 33

2.4	Les enjeux du Smart-Metering . . . . .	34
2.5	Conclusion . . . . .	35
<b>3</b>	<b>Plateforme Smart-City</b>	<b>36</b>
3.1	Introduction . . . . .	37
3.2	Architecture générique d'une plateforme . . . . .	37
3.3	Cloud Computing . . . . .	38
3.3.1	Définitions . . . . .	38
3.3.2	Les modèles de services du cloud . . . . .	39
3.3.2.1	SaaS (Software as a Service) . . . . .	39
3.3.2.2	IaaS (Infrastructure as a Service) . . . . .	39
3.3.2.3	PaaS (Platform as as Service) . . . . .	39
3.3.3	Les modèles de déploiement . . . . .	40
3.3.4	OpenStack . . . . .	40
3.3.4.1	Architecture d'openStack . . . . .	41
3.3.4.2	Services d'OpenStack . . . . .	41
3.3.5	Proxmox . . . . .	43
3.3.6	Autres solutions . . . . .	43
3.4	la plateforme Smart-City . . . . .	43
3.5	Big data . . . . .	44
3.5.1	Introduction du big data . . . . .	44
3.5.1.1	Caractéristiques du Big Data . . . . .	45
3.5.1.2	Architecture Big Data . . . . .	46
3.5.2	Les bases de données MPP (Massive Parallel Prcoessing) . . . . .	46
3.6	Les applications . . . . .	47
3.7	Conclusion . . . . .	48
<b>II</b>	<b>Implémentation : Smart Metering pour l'EAU</b>	<b>49</b>
<b>4</b>	<b>Système d'acquisition</b>	<b>50</b>
4.1	Introduction . . . . .	51
4.2	Schéma de notre solution . . . . .	51
4.3	Le compteur d'eau . . . . .	51
4.3.1	Types de compteurs . . . . .	51
4.3.1.1	Compteur à déplacement . . . . .	51
4.3.1.2	Compteur vitesse . . . . .	52
4.3.1.3	Compteur electromagnétiques . . . . .	54
4.3.1.4	Compteurs Ultrasonique . . . . .	54
4.3.2	Compteur SEAAL . . . . .	55
4.3.2.1	Le totalisateur du compteur 420 . . . . .	56
4.4	Capteurs . . . . .	56
4.4.1	Solutions possibles . . . . .	56
4.4.2	Solution choisie . . . . .	57
4.5	Solution de communication . . . . .	58
4.6	Conclusion . . . . .	59
<b>5</b>	<b>Plateforme Smart-City dédiée au Smart-Metering</b>	<b>60</b>

5.1	Introduction . . . . .	61
5.2	Architecture générale . . . . .	61
5.3	Sentilo . . . . .	62
5.3.1	introduction . . . . .	62
5.3.2	Plateforme Pub/Sub . . . . .	62
5.3.3	Stockage temps réel . . . . .	62
5.3.4	API Rest . . . . .	63
5.3.5	Agents . . . . .	63
5.3.5.1	Agent relationnel . . . . .	65
5.3.5.2	Agent d’alarmes . . . . .	65
5.3.5.3	Agent moniteur d’activités . . . . .	65
5.3.5.4	Agent historien . . . . .	66
5.3.6	Bases de données Sentilo . . . . .	66
5.3.6.1	Redis . . . . .	67
5.3.6.2	MongoDB . . . . .	67
5.3.6.3	Le serveur Pub/Sub . . . . .	67
5.3.7	Architecture générale . . . . .	70
5.3.8	Principe . . . . .	70
5.3.9	Fonctionnalités . . . . .	71
5.3.9.1	Utilisateurs . . . . .	71
5.3.9.2	Providers . . . . .	71
5.3.10	Components . . . . .	72
5.3.11	Capteurs/Actionneurs . . . . .	72
5.3.12	Alertes . . . . .	73
5.3.13	Création de règle d’alertes . . . . .	73
5.3.14	Types de capteurs/Actionneurs . . . . .	74
5.3.15	Types de composants . . . . .	74
5.4	GreenPlum . . . . .	74
5.4.1	Définition . . . . .	74
5.4.2	Architecture . . . . .	75
5.4.2.1	Le master . . . . .	75
5.4.2.2	Les segments . . . . .	76
5.4.2.3	L’interconnect . . . . .	77
5.5	Ressources matériels . . . . .	77
5.5.0.1	Plateforme . . . . .	77
5.5.0.2	Réseau de capteurs . . . . .	79
5.6	Architecture réseau . . . . .	80
5.7	Principe de fonctionnement . . . . .	80
5.8	Résultats : Visualisation de données . . . . .	82
5.9	Conclusion . . . . .	85
	<b>Conclusion générale</b>	<b>86</b>
	<b>Bibliographie</b>	<b>88</b>

# Liste des tableaux

1.1	Comparaison Cat-0 vs Cat-m1 . . . . .	26
1.2	Comparaison NB-LTE vs C-IoT vs NB-IoT . . . . .	27

# Table des figures

1.1	Architecture générique IoT . . . . .	17
1.2	Architecture générale d'un réseau LPWA. . . . .	19
1.3	Diversité dans le réseau Sigfox. . . . .	21
1.4	Architecture d'un réseau LoRa. . . . .	22
1.5	Modèle en couche LoRa. . . . .	23
1.6	réseau Telensa. . . . .	24
1.7	Modes de fonctionnement de NB-IoT. . . . .	28
1.8	Principe de M-Bus. . . . .	28
2.1	Architecture d'un Smart-Metering . . . . .	33
3.1	Plateforme générique d'une plateforme IoT - Partie informatique. . . . .	37
3.2	Illustration du principe de cloud. . . . .	38
3.3	Architecture conceptuelle d'OpenStack. . . . .	41
4.1	Système de relevé automatique de la consommation en eau. . . . .	51
4.2	Compteur à turbine . . . . .	53
4.3	Compteur électromagnétiques . . . . .	54
4.4	Principe de mesure du compteur Ultrasonique . . . . .	55
4.5	Courbe de Précision du Modèle 420. . . . .	55
4.6	Schéma électrique pour la lecture du capteur infrarouge. . . . .	57
4.7	cadran du Compteur. . . . .	57
4.8	Communication avec le nRF24L01. . . . .	58
5.1	Architecture générale de la plateforme smart-city. . . . .	61
5.2	Fonctionnement standard des agents. . . . .	64
5.3	Architecture d'interfaçage de sentilo avec ELK. . . . .	66
5.4	Fonctionnement de la couche transport. . . . .	68
5.5	Fonctionnement de la couche service. . . . .	69
5.6	Architecture de la plateforme sentilo. . . . .	70
5.7	Interface de gestion des utilisateurs. . . . .	71
5.8	Interface de gestion de composants. . . . .	71
5.9	Interface de gestion des composants. . . . .	72
5.10	Interface spécifique à un capteur. . . . .	72
5.11	Interface de gestion d'alertes. . . . .	73
5.12	Interface de gestion de règles d'alertes. . . . .	73
5.13	Interface de gestion de types de capteurs et d'actionneurs. . . . .	74

5.14	Interface de gestion des types de composants. . . . .	74
5.15	Architecture globale de Greenplum. . . . .	75
5.16	Monatage du compteur. . . . .	79
5.17	Monatage du capteur sur le compteur. . . . .	79
5.18	Architecture du réseau implémenté sur proxmox . . . . .	80
5.19	Diagramme de fonctionnement du capteur . . . . .	81
5.20	Diagramme de fonctionnement de la passerelle . . . . .	82
5.21	Interface de création de la source de donnée. . . . .	83
5.22	Interface de création de requête SQL. . . . .	83
5.23	Visualisation des données demandées. . . . .	84
5.24	Ajout de Widget au dashboard. . . . .	84

# Liste des abréviations

**AES** : Advanced Encryption Standard  
**CDMA** : Code Division Multiple Access  
**CSS** : chirp Spread Spectrum  
**DBPSK** : Differential Binary Phase-Shift Keying  
**DL** : DownLink  
**DSSS** : Direct-Sequence Spread Spectrum  
**FDMA** : Frequency Division Multiple Access  
**GFSK** : Gaussian Frequency-Shift Keying  
**HTTP** : Hyper Text Transfert Protocol  
**ISM** : Industrielle, Scientifique, Medicale  
**LPWA** : Low Power Wide Area  
**M2M** : Machine To(2) Machine  
**MPP** : Massive Parallel Processing  
**OFDMA** : Orthogonal Frequency Division Multiple Access  
**SNO** : Sigfox Network Operators  
**SQL** : Structured Query Language  
**TDMA** : Time Division Multiple Access  
**UL** : UpLink  
**UNB** : Ultra Narrow Band  
**XML** : Extensible Markup Language

# Introduction générale

Le concept des Villes Intelligentes (Smart Cities) promet un changement radical de notre façon de vivre grâce à une gestion efficace des ressources de la ville. Et ce à travers l'utilisation des réseaux de capteurs, des technologies de l'information et de communication (TIC) ainsi que des outils d'analyse de données.

Parmi les nombreuses problématiques dans la gestion d'une ville, le problème de relevé de la consommation des citoyens est celui qui apparaît le plus souvent. La méthode actuelle classique pour le relevé de la consommation des compteurs consiste au passage de l'agent de l'entreprise pour lire l'indicateur sur le cadran du compteur et remplir un formulaire. Cette méthode nous impose beaucoup d'inconvénients (compteurs installés à l'intérieur de la maison chez certains clients ce qui exige la présence du client lors du relèvement, obligation d'estimer la consommation dans certains cas, fuites non détectée, etc.). Par conséquent, une bonne partie de la quantité des ressources du pays est perdue.

Afin de résoudre ces problèmes, les centres de recherche, les entreprises ainsi que les universités autour du monde ont travaillé pour améliorer ces services et ceci pendant le démarrage de la 4ème révolution industrielle qui concerne l'internet des objets. Le terme "Smart-Metering" est apparu. Le Smart-Metering consiste à rendre le relevé de consommation automatique et centralisé, sans avoir à envoyer un agent semestriellement ou mensuellement. La consommation est lue en temps réel grâce à une nouvelle génération de compteurs communicants ou même en intégrant des capteurs aux compteurs déjà existants.

L'objectif de ce travail consiste à faire une étude et mise en oeuvre d'une chaîne de Smart-Metering pour une ville intelligente, capable de mesurer en temps réel la consommation d'un client (Gaz, Eau, Electricité) et une implémentation dans un cas d'usage qui est la télé-relève de la consommation en eau d'un client.

Ce travail s'inscrit dans le grand projet de la wilaya d'Alger "Alger Smart City" lancé en 2017 et dont la feuille de route a été définie et progresse actuellement.

Ce mémoire est divisé en deux parties. La première composée des trois premiers chapitres ; le premier chapitre expose les principes de base des objets connectés et les réseaux de capteurs ainsi que les technologies existantes. Le deuxième chapitre présente un état de l'art du Smart-Metering, les concepts de la chaîne de travail du Smart-Metering ainsi que les enjeux de cette solution.

Dans le troisième chapitre, nous détaillerons les différentes parties qui constituent une plateforme Smart City avec des exemples de solutions utilisés dans chaque étage.

La deuxième partie montre l'implémentation d'un système complet de Smart Metering appliqué à l'eau. Dans le chapitre 4, nous étudierons les compteurs d'eau possibles dont celui de SEAAL pour faire un choix de la technique de relevé à partir du compteur et construire notre système d'acquisition.

Le dernier chapitre complète celui qui le précède en montrant une implémentation de la partie cloud du système télé-relève allant du capteur jusqu'à l'application finale. Avec comme service l'exposition de la consommation du client sur un tableau de bord.

# Première partie

## État de l'art et concepts

# Chapitre 1

## Internet des objets (IoT)

## 1.1 Introduction

Entre web3.0 et 4<sup>ème</sup> révolution industrielle, l'Internet des objets (IoT) constitue l'un des développements technologiques les plus importants de la dernière décennie. Avec plus de 50 milliards d'objets connectés prévus en 2020, il a le potentiel d'affecter profondément notre style de vie. Cependant, son succès repose en grande partie sur une architecture bien définie qui fournira une plate-forme évolutive, dynamique et sécurisée à son déploiement.

Ce premier chapitre a pour but d'introduire les notions de base concernant l'internet des objets ainsi que les technologies utilisées.

## 1.2 Définition et historique

L'internet des objets est un terme qui réfère à l'interconnexion des objets quotidiens (véhicules, vêtements, instrument de mesure...) entre eux et à l'internet. L'interconnexion permet aux objets qui nous entourent de partager des données, d'interagir et d'agir de manière autonome dans le but de faciliter la tâche aux êtres humains. Cette perspective ouvre de nouvelles portes vers un futur, où le monde réel et virtuel fusionnera de manière transparente à travers le déploiement massif de dispositifs embarqués qui viendront ajouter de nouvelles fonctionnalités aux objets avec des capacités de calcul, de communication et de stockage [1].

Le concept de l'IoT n'est pas nouveau, l'un des premiers exemples d'Internet des objets date du début des années 80 et était un distributeur de boisson située à l'Université Carnegie Mellon. Cette machine était capable de communiquer son état actuel (vide ou remplie) à travers le réseau.

L'Histoire des objets connectés débute en 1999 lorsque Kevin Ashton, pionnier de la technologie RFID (Radio Frequency IDentification – Technologie d'identification automatique), invente l'expression "Internet des objets". Cette même année, le concept

naît aux États-Unis et particulièrement au MIT (Massachusetts Institute of Technology). Ce laboratoire est dédié à la création d'objets connectés à l'aide de l'identification par radiofréquence et les réseaux de capteurs sans fil.

L'Internet des objets (IoT) est apparu à l'issue de la mécanisation et de la standardisation, appliquée à l'automatisation du traitement du document et de l'information sur support matériel puis numérique. Il s'est rapidement diffusé avec la mondialisation. Peu à peu, les objets ont été modifiés (avec des puces RFID par exemple) ou conçus pour « utiliser le protocole IP », devenant des « objets connectés » reliés à des serveurs centralisés et/ou capables de communiquer entre eux.

C'est néanmoins en 2007 que le phénomène des IoT a pris de l'ampleur, avec la démocratisation des Smartphones et la sortie du premier iPhone par Apple. La dématérialisation est en marche [2].

### 1.3 Principe et architecture

L'internet des objets est une structure qui permet de capter l'information de l'environnement et la transférer pour ensuite la traiter afin d'en extraire de la valeur.

L'architecture de l'IoT se divise en 3 couches : une couche physique pour l'interaction avec l'environnement, une couche réseau pour la transmission de l'information et une couche applicative pour son exploitation. La figure suivante est un exemple d'architecture IoT .

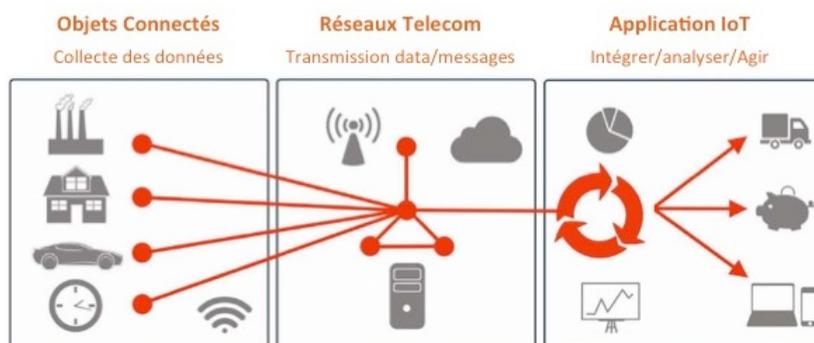


FIGURE 1.1: Architecture générique IoT. [3]

## 1.4 Couche physique : les objets

Les objets sont la couche basse de l'architecture de l'IoT. Cette couche est composée par les différents capteurs et actionneurs déployés dans l'environnement. Elle est responsable de la perception des propriétés physiques de l'environnement et de sa conversion en informations numériques pour la transmettre sur le réseau dans le cas des capteurs, et l'exécution des tâches demandés dans le cas des actionneurs.

Pour capturer la donnée, plusieurs technologies sont employées selon l'application : GPS, les capteurs optiques dans les caméras, RFID, etc.

Dans le cas des actionneurs, on utilise souvent des actionneurs mécaniques pour produire l'action : moteurs, relais, etc.

Les caractéristiques principales qu'on cherche dans un objet sont :

- Sa grande sensibilité à la grandeur qu'il mesure.
- Sa capacité à reproduire fidèlement la grandeur physique.
- Un faible encombrement, poids et consommation. L'objet doit être inaperçu.

## 1.5 Couche réseau

Les réseaux de communication classiques tel que les réseaux cellulaires (2G, 3G, 4G), filaires ont été conçus pour permettre un échange de données (voix, vidéo, texte, . . . etc.) de manière constante et rapide avec un débit élevé. Ce type de communication nécessite que l'appareil soit en échange constant avec la station de base et implémente des algorithmes complexes ce qui augmente la consommation d'énergie.

Ceci fait qu'ils sont surdimensionnés et non adaptés pour des applications sur batterie nécessitant un débit d'échange faible et une consommation minimale d'énergie. De plus, les objets connectés sont déployés en masse, ce qui rend le coût de maintenance (changement de batteries principalement) ainsi que les abonnements très élevé.

Pour surmonter ces défis, de nouvelles technologies d'interconnexion de ces objets sont apparues [4]. Dans les paragraphes qui suivent nous exposons quelques technologies qui commencent à avoir de l'ampleur à travers les villes intelligentes autour du monde.

### 1.5.1 Les réseaux LPWA

Les réseaux LPWA sont de nouvelles technologies de communication destinées pour des applications à bas débit permettant une large couverture et une consommation d'énergie très basse avec des durées de vie qui peuvent aller jusqu'à 10 ans. Ce type de réseaux est plus adapté pour les applications d'objets connectés (IoT) par rapport aux technologies classiques.

Ils peuvent être divisés en trois catégories [5] :

- **Les technologies UNB (Ultra Narrow Band)** : ce type de technologie utilise un canal de transmission très étroit (100-1000Hz) et offre un débit de transmission très faible (de l'ordre de quelques bits par heure) afin de se conformer aux réglementations sur les émissions.
- **Les technologies à spectre étalé** : comme leur nom l'indique, ce type étale toute l'énergie transmise sur le canal de transmission, la largeur de la bande peut aller de 125Khz a 1 Mhz, et offrent un débit de l'ordre de quelques dizaines de kbps.
- **Les technologies à bande étroite** : relativement nouvelles par rapport aux précédentes, elles combinent un accès TDMA avec une transmission à spectre étalé à saut de fréquences sur un canal de 12,5 KHz. Ces technologies essaient d'avoir une place sur le marché en exploitant les lacunes des deux autres types.

La figure suivante donne un aperçu de l'architecture générale d'un réseau LPWA :

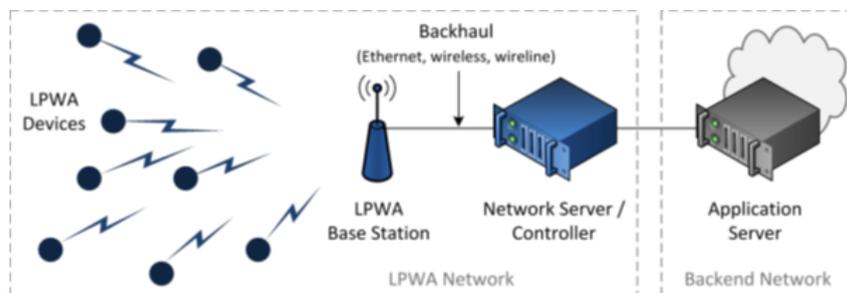


FIGURE 1.2: Architecture générale d'un réseau LPWA.

L'architecture des réseaux LPWA se compose de :

1. LPWA Device : c'est la source des données qui circulent sur le réseau, ce sont les différents capteurs et actionneurs connectés sur le réseau.
2. La station de Base : appelée aussi Gateway, elle assure la communication avec les appareils et la transmission des données vers le contrôleur réseau à travers le Backhaul.
3. Le contrôleur réseau : il est responsable de la gestion des différentes stations de base déployées et du traitement des données envoyées par les objets. il est connecté aux différentes stations de base à travers la Backhaul. plusieurs technologies peuvent être utilisées (celulaire, ethernet, WIFI,...).
4. L'application : c'est l'interface à travers laquelle on peut gérer notre réseau, elle permet aussi l'accès et la visualisation des différentes données collectées.

Dans la suite, nous allons introduire différentes technologies LPWA existants sur le marché.

#### 1.5.1.1 Sigfox

Sigfox est un opérateur de télécommunications français créé en 2009 et spécialisé dans les réseaux IoT (M2M), il offre une solution de connectivité cellulaire bas débit à très basse consommation pour les objets connectés. Pour déployer son réseau, sigfox fait appel à des partenaires (SNO) qui déploient les stations de bases qui sont reliés à des serveurs connectés à internet.

Sigfox est une technologie propriétaire basé sur une transmission à bande ultra étroite (100Hz en Europe et 600Hz aux états unis) opérant dans une bande de fréquences d'environ 192KHz de la bande ISM non licenciée (868-868,2 MHz en Europe et de 902-928 MHz en USA) avec une modulation DBPSK en liaison montante et GFSK en descendante, cette technique permet aux stations de base sigfox de communiquer sur de longues distances sans être affectés par le bruit [5][6].

En transmission, un message en UL peut délivrer jusqu'à 12 octets de charge utile, la transmission est répétée trois fois sur des fréquences différentes (2 seconds pour

chaque transmission) ce qui permet d'avoir une diversité temporelle et fréquentielle. Les messages peuvent être reçus par plusieurs stations de base qui balayent tous les canaux de manière permanente (diversité spatiale), un serveur central filtre les messages redondants et choisit le message le plus approprié, ceci permet une meilleure fiabilité de la transmission.

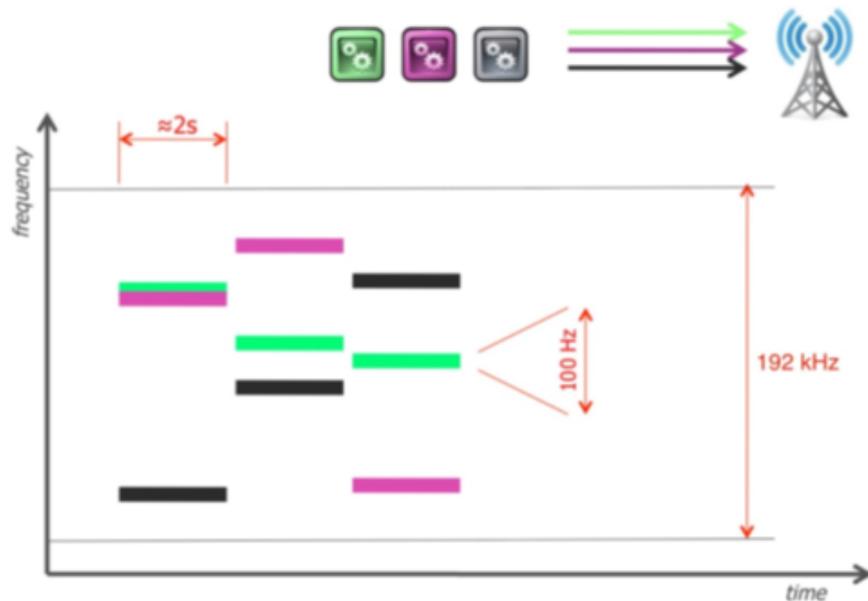


FIGURE 1.3: Diversité dans le réseau Sigfox.

En DownLink, la transmission n'est établie que si l'objet la demande en incluant un DownLink Request Flag (demande d'un retour) dans son message transmis.

Les principaux avantages de sigfox sont l'utilisation efficace du spectre, un niveau de bruit bas ce qui rend la réception plus facile (meilleure sensibilité) et une consommation basse d'énergie (environ 50 mA en transmission à puissance maximale autorisée en Europe (14 dBm)).

Pour les inconvénients, on peut citer le débit faible par rapport aux autres technologies concurrentes, le nombre de messages qui est limité à 140 messages par jour avec une charge de 12 octet en UpLink (UL) (la bande ISM ne peut être utilisée que pendant 1% du temps) et 4 messages de 8 octets en DownLink (DL) (insuffisant pour envoyer des accusés de réception ou des mises à jour) et la sensibilité à l'effet Doppler à cause de la bande étroite qui le rend non adapté pour des applications mobiles.

### 1.5.1.2 LoRa

LoRa (Long Range) est une technologie développée initialement par la startup française Cycleo qui a été rachetée ensuite par l'entreprise américaine Semtech en 2012. Le projet est soutenu par de nombreuses entreprises qui forment "The LoRa Alliance".

La technologie LoRa définit la couche physique (modulation) dans le système de communication, les fréquences utilisées sont dans la bande libre ISM (433 MHz, 868 MHz et 915 MHz selon la zone géographique), elle est basée sur la technique CSS où les données sont réparties sur tout le canal. Le débit est entre 980 bps et 21,9 Kbps en USA, et entre 250 bps et 11 kbps en Europe. La taille maximale des données est fixée à 256 octet.

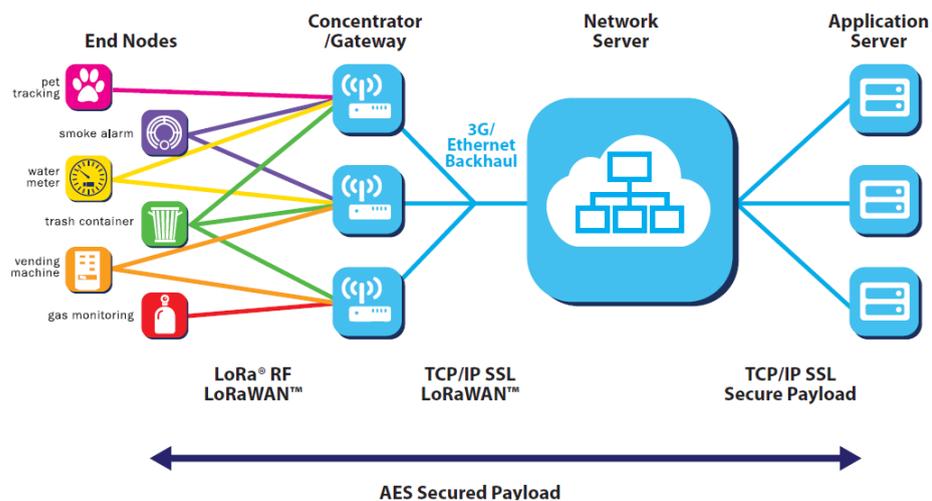


FIGURE 1.4: Architecture d'un réseau LoRa.

Pendant que LoRa définit la couche physique, LoRaWAN définit le protocole de communication et l'architecture du réseau. La topologie utilisée est en Etoile. Trois classes d'appareils peuvent être connectées au réseau selon le mode d'application :

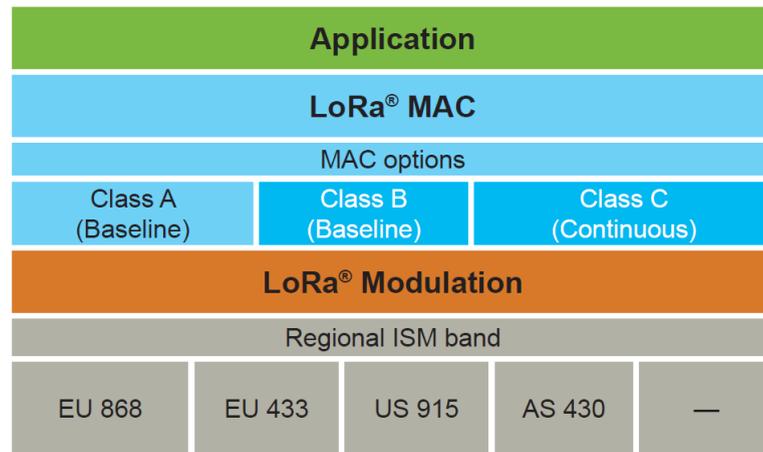


FIGURE 1.5: Modèle en couche LoRa.

- Classe A : la plus économe en énergie. La communication bidirectionnelle, chaque message transmis en UL est suivi par deux fenêtres d'écoute d'environ 4 secondes pour la réception d'un message en DL. Ce mode est adapté pour des applications où le nombre de messages en UL est plus important sans besoin de mise à jour logiciel.
- Classe B : ce mode permet de planifier des transmissions en DL, ce qui permet de faire des mises à jour et de diffuser les messages à plusieurs objets.
- Classe C : ce mode permet une communication continue avec l'objet connecté avec des temps de retard minimal. La consommation en énergie est maximal, ce qui le rend non adapté pour des applications sur batterie ou la durée de vie est un facteur important.

Dans le réseau, la communication est asynchrone et les objets ne sont pas liés à une station de base, Un message transmis par l'objet peut être reçu par plusieurs stations de base, ceci permet une diversité spatiale. Tous les messages sont transmis vers le serveur qui filtre les paquets redondants, effectue les contrôles de sécurité et renvoie l'accusé de réception. Pour sécuriser le réseau, les messages sont cryptés en utilisant l'algorithme AES CCM 128.

L'avantage principal de cette technologie est qu'elle est ouverte, toute entreprise peut construire son propre réseau LoRa [4]. La portée est importante, on peut couvrir de très grandes zones avec une seule Gateway.

LoRa possède une très basse consommation de puissance avec une portée qui peut atteindre 5 Km en zone urbaine et 15 Km en champ libre [4]. Grâce à la largeur du canal, il est possible d'utiliser cette technologie dans des applications mobiles ou de géolocalisation [5].

### 1.5.1.3 Telensa UNB

Telensa UNB est une technologie propriétaire détenue par la société britannique Telensa, elle est utilisée par cette dernière principalement dans son système de gestion d'éclairage.

Telensa UNB est basée sur une transmission à bande ultra étroite (UNB) dans la bande de fréquences libre ISM (868 MHz en Europe, 915 Mhz en USA et 470 Mhz en chine). La topologie implémentée est en étoile où les objets sont connectés à une station de base qui peut supporter jusqu'à 10000 objets connectés.

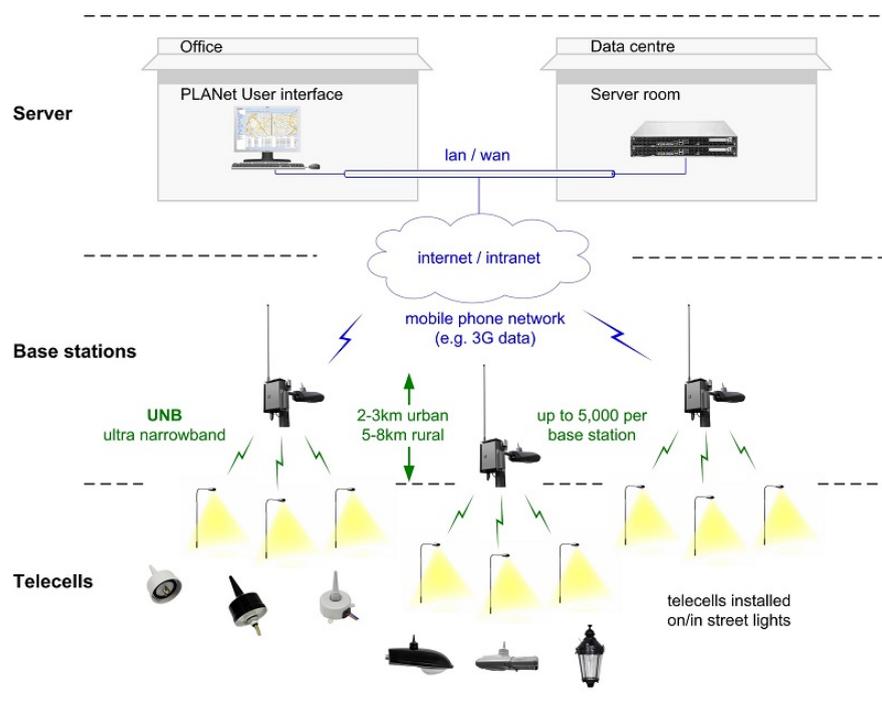


FIGURE 1.6: réseau Telensa.

Pour gérer son réseau d'éclairage connecté, Telensa implémente le Protocol TALQ qui normalise les échanges entre le centre de gestion CMS(!!!!!!!!!!!!!!!!!!!!!!!!!!!!) et le réseau en spécifiant le format de données, le type des messages, les paramètres ainsi

que le comportement au niveau du réseau de capteurs. Les données sont codées en format XML et sont transportés à l'aide de requêtes HTTP.

Le protocole supporte la communication bidirectionnel, ce qui le rend bien adapté pour des applications de gestion et de contrôle, de localisation (supporte l'option paging) ainsi que pour des mises à jour Software.

#### **1.5.1.4 Random Phase Multiple Access (RPMA)**

RPMA est une solution proposée par la société Ingenu qui utilise la bande de fréquences libre ISM de 2,4 GHz disponible mondialement ce qui lui permet d'être déployée dans n'importe quelle région.

La technologie est basée sur la technique d'étalement de spectre DSSS et une modulation D-BPSK, avec une largeur de bande de 1 MHz, et 40 canaux de communication. La communication bidirectionnelle est possible et la vitesse de transmission est de 41 Kbps en USA et 20,5 Kbps en Europe. Contrairement à d'autres technologies concurrentes, il est possible d'envoyer des accusés de réception pour chaque message reçu.

Comme pour le CDMA dans les réseaux cellulaires, l'utilisation du RPMA permet d'avoir plusieurs communications dans un même canal, en ajoutant un déphasage aléatoire à chaque trame, ce qui augmente la capacité des stations de base. Cependant, l'utilisation de la bande de 2,4 GHz peut réduire la portée du réseau considérablement dans les zones urbaines.

#### **1.5.1.5 Qowisio**

Qowisio offre une technologie de communication bidirectionnelle à bande ultra étroite (UNB) dans la bande des 868 MHz (ISM) avec une vitesse de transmission très basse (12 bps), le nombre de messages est limité à 36 par heure. La portée peut aller jusqu'à 50 Km en champ libre et 3 Km en zone urbaine.

Le point fort de cette technologie réside dans le fait que ses stations de base (Qwisio Box) sont compatibles avec les technologies LoRa et Sigfox, ainsi, un équipement communiquant en LoRa ou Sigfox peut être connecté au réseau Qwisio

### 1.5.2 Les réseaux LPWA 3GPP

Le 3GPP, l'organisme responsable de production et la maintenance des spécifications techniques pour les réseaux de 3<sup>ème</sup> et 4<sup>ème</sup> génération, a fait sortir dans la 12<sup>ème</sup> version des spécifications techniques, les premières spécifications orientées aux domaines de l'IoT en définissant une nouvelle catégorie d'appareils (Cat-0) et de fonctionnalités comme le Débit limité à 1 Mbps en UL et DL, l'utilisation d'un seul récepteur au lieu de deux ce qui réduit le coût de fabrication, un mode d'économie d'énergie amélioré en éliminant les signalisations.

La 13<sup>ème</sup> révision des spécifications a apporté une nouvelle catégorie d'appareils (Cat-m1) qui propose une réduction significative des coûts : moins de débit, moins de bande passante, réduction de la puissance de transmission. Le tableau suivant dresse une comparaison entre les catégories Cat-0 et Cat-m1 :

	<b>Cat-0</b>	<b>Cat-m1</b>
<b>Débit max en DL</b>	1 Mbps	~200 kbps
<b>Débit max en UL</b>	1 Mbps	~200 Kbps
<b>Nombre d'antenne de reception</b>	1	1
<b>Modulation (UL/DL)</b>	64/16 QAM	non défini
<b>taille de la donnée (bits)</b>	1000/1000	Non définie
<b>Communication en Duplex</b>	Half duplex	Half duplex
<b>BP de reception</b>	20 Mhz	1,4 Mhz
<b>Puissance de transmission max</b>	23 dBm	20 dBm

TABLE 1.1: Comparaison Cat-0 vs Cat-m1

#### 1.5.2.1 NB-IoT

Le Narrowband IoT est un standard qui est en cours de création par le 3GPP, il a été annoncé dans la 13<sup>ème</sup> révision des spécifications. Le but est d'établir une technologie

compatible aux réseaux cellulaires existant. Deux propositions sont en étude pour converger à un standard commun :

- Narrowband-LTE (NB-LTE) par Ericsson et soutenu par de grandes entreprises comme Nokia et Alcatel.
- Cellular-IoT (C-IoT) de Huawei et Qualcomm

	<b>NB-LTE</b>	<b>C-IoT</b>	<b>NB-IoT</b>
<b>Spectre</b>	non défini	non défini	GSM/LTE (modes in-band/guard band)
<b>version de sortie</b>	13/14	13/14	13/14
<b>Débit max en DL</b>	128 kbps	32 kbps	~180 Kbps
<b>Débit max en UL</b>	64 kbps	48/14.7 kbps	~180 Kbps
<b>Largeur de bande</b>	200 Khz	200 Khz	180 Khz
<b>Gain</b>	164 dB	164 dB	164 dB
<b>M-à-j réseau</b>	oui	oui	oui

TABLE 1.2: Comparaison NB-LTE vs C-IoT vs NB-IoT

Les techniques de modulations utilisées sont le FDMA en UL et l'OFDMA en DL avec des débits de l'ordre de 180 Kbps.

Le NB-IoT peut opérer de 3 manières :

- En occupant la place d'un canal de transmission GSM (mode standalone)
- En s'insérant dans un canal de transmission LTE (mode in-band)
- En utilisant la bande de sécurité situant entre deux canaux LTE (mode guard band)

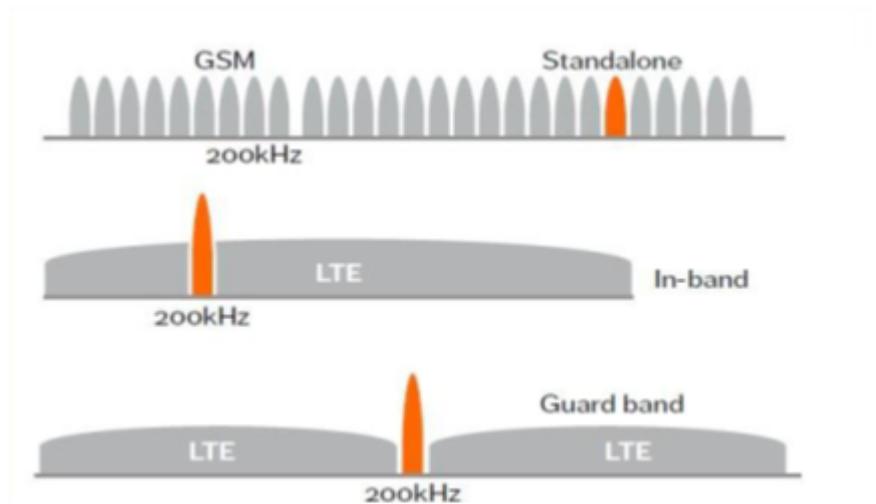


FIGURE 1.7: Modes de fonctionnement de NB-IoT [4].

### 1.5.3 M-Bus

Le Meter-Bus est un standard européen utilisé dans la lecture automatique des compteurs (Gaz, Eau, Électricité). Son utilisation peut être généralisée pour d'autres applications (alarmes, température, ... etc.) [7].

M-Bus utilise une topologie en bus à deux fils. Il est composé d'un maître qui contrôle la communication, et des esclaves (les compteurs) connectés en parallèle [8].

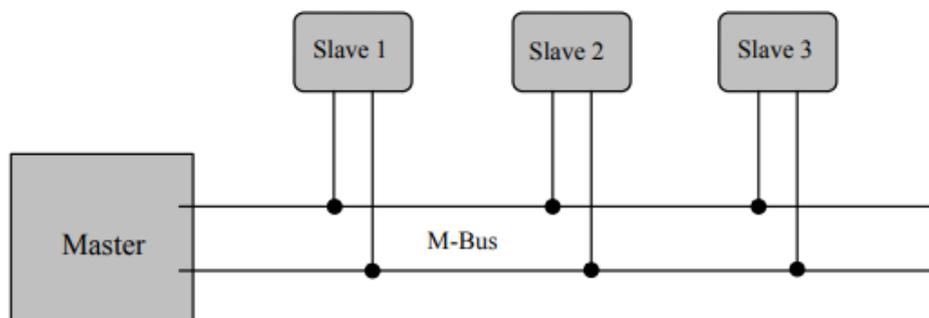


FIGURE 1.8: Principe de M-Bus.

Une autre variante de M-Bus est le WM-Bus, cette version utilise le faisceau hertzien comme support de communication plutôt que le filaire. La transmission se fait dans les bandes de fréquence 169, 433 et 868 MHz.

### 1.5.4 Autres solutions

plusieurs autres solutions existent, on peut citer :

- **PLC (Power Line Communication)** Utilisée dans les réseaux de distribution d'électricité, il consiste à superposer un signal haute fréquence et de faible énergie au courant alternatif 50/60 Hz, ce signal sera celui qui propage la donnée sur le secteur.
- **BLE (Bluetooth Low Energy)** Basé sur le Bluetooth, le BLE permet un même débit ( 1 Mbit/s avec une consommation d'énergie 10 fois moindre.
- **Autres** Zigbee, WIFI, Bluetooth,...

## 1.6 Couche Applications

Cette couche utilise les données collectées par la couche physique et transmise à travers le réseau. C'est la couche par laquelle tout l'intérêt de l'IoT apparait. Elle fournit les outils requis pour exploiter cette masse de données (Big Data, Traitement de données, IA, etc.) [1].

La gamme d'applications possibles est impressionnante : le transport intelligent, la gestion des réseaux de distribution (Smart Grid), la gestion logistique, l'authentification d'identité, les services basés sur la localisation, la sécurité, etc.

## 1.7 Conclusion

Dans ce chapitre, nous avons exposé les concepts de base relatives à l'internet des objets qui nous permettrons de mieux aborder notre thématique. Cette étude nous a permis aussi de faire un état de l'art des technologies qui existent dans le domaine.

Dans le chapitre suivant, nous allons nous intéressés à une des nombreuses applications de l'IoT et celle qui présente le coeur de notre travail, à savoir le smart metering.

## Chapitre 2

### Etat de l'art du Smart-Metering

## 2.1 Introduction

Sans comptage et sans mesure, il ne peut y avoir maîtrise des consommations d'énergie, d'eau ou de gaz. Il faut savoir ce que l'on consomme pour optimiser et pérenniser les ressources d'une ville. Dans ce chapitre, nous allons donner l'état de l'art du Smart-Metering dans le monde et comment les gouvernements des pays développés voient la projection du smart-metering sur leur ville. Nous allons aussi présenter une architecture générique utilisée pour réaliser ce concept.

## 2.2 Le Smart-Metering dans le monde

De nombreux pays dans le monde s'orientent vers le Smart-Metering, chacun a sa perception du futur du smart-metering dans son pays et son plan d'action on en cite quelques exemples dans ce qui suit.

**Les Etats-Unis** Certains états, dont la Californie font des recherches actives sur ces technologies. Le 20 juillet 2006, la Californie a approuvé un programme d'amélioration des compteurs classiques par adjonction d'un processeur de communications électroniques chez 9 millions de ménages clients (gaz et d'électricité). Ces compteurs calculent et communiquent la consommation horaire du ménage, permettant des factures intégrant des coûts horaires différenciés pour une meilleure maîtrise de la demande en électricité.[9]

**Allemagne** Le ministère de l'Économie allemand a demandé au cabinet Ernst & Young une évaluation économique du déploiement des compteurs intelligents. Or, l'audit rendu public en août 2013 indique que les coûts mis en œuvre dépassent nettement les économies réalisées par les consommateurs. Le 9 février 2015, le ministère allemand de l'Économie annonce que les compteurs intelligents ne seront obligatoires que pour les foyers consommant plus de 6 000 kWh par an. Comme la moyenne annuelle de consommation électrique des foyers allemands est évaluée à 3 500 kWh, l'obligation d'équipement ne va concerner qu'une minorité.[10]

**France** La France a lancé plusieurs projet, elle projette d'intégrer les compteurs intelligents tout autour du territoire Français. Dans le contexte de l'eau par exemple, les trois distributeurs majeurs d'eau en France (Veolia, Lyonnaise des Eaux et Saur) installent des compteurs communicants dans le secteur de l'eau soit pour équiper toute une ville, soit pour équiper ponctuellement un site industriel ou un immeuble d'habitation ou de bureaux. Plusieurs fabricants de compteurs proposent des solutions adaptées, différentes de celles utilisées pour les compteurs communicants utilisées pour le gaz et l'électricité (pour ce qui relève des données produites qui sont parfois spécifiques), le système de collecte pouvant lui être commun (collecte mobile piétonne ou par véhicule, ou collecte via un réseau fixe ou mobile).[11]

**Québec** Hydro-Québec avait prévu de remplacer ses 3,8 millions de compteurs électromécaniques et électroniques par des compteurs communicants entre 2012 et 2017 dans le cadre du déploiement d'une «infrastructure de mesurage avancée» et de la lecture à distance des compteurs.[12]

**Algérie** Plusieurs solutions de géants dans la technologie ont été testées ces derniers mois. La branche SDA de Sonelgaz a testé trois solutions différentes pour le gaz et l'électricité dont l'une a été proposée par Huawei, qui opte pour la communication par la ligne électrique vu que tout un réseau est déjà déployé. La SEAAL, premier fournisseur d'eau national, a également testé plusieurs solutions dont une solution proposée par les fabricants des compteurs locaux (AMC-El Eulma) qui est Sensus, un capteur à intégrer directement sur le compteur d'eau mécanique existant, qui lit la consommation en temps réel et transmet l'information via une connection M-BUS. Des contraintes réglementaires bloquent le déploiement des solutions en Algérie en ce qui concerne les fréquences utilisées par les réseaux de communication radio-fréquence.

## 2.3 Smart Metering : AMR/AMI

### 2.3.1 AMR - Lecture automatique des compteurs

Une technologie qui collecte les données de consommation automatiquement et la transfère vers une base de données centrale pour être analysée et pour produire la facturation. La consommation détaillée de l'eau est continuellement collectée à des intervalles réguliers (chaque 30 minutes par exemple) et peut être lue à distance au biais d'un processus automatisé en utilisant le système de gestion et de facturation. AMR peut consister en plusieurs méthodes, allant d'un exemple où un chauffeur passe avec son appareil devant les maisons et la donnée est directement envoyée depuis les capteurs qui se connectent automatiquement à son passage.

### 2.3.2 AMI - Infrastructure de compteurs avancée

C'est une infrastructure qui intègre en premier les capteurs intelligents, et ajoute une communication bi-directionnelle entre le capteur et le consommateur. Ce qui veut dire qu'en plus d'exposer la consommation, le compteur peut aussi recevoir des instructions envoyées par le fournisseur ou par le consommateur lui-même. Comme couper l'électricité, fermer une vanne, étalonner entre autres applications possibles.

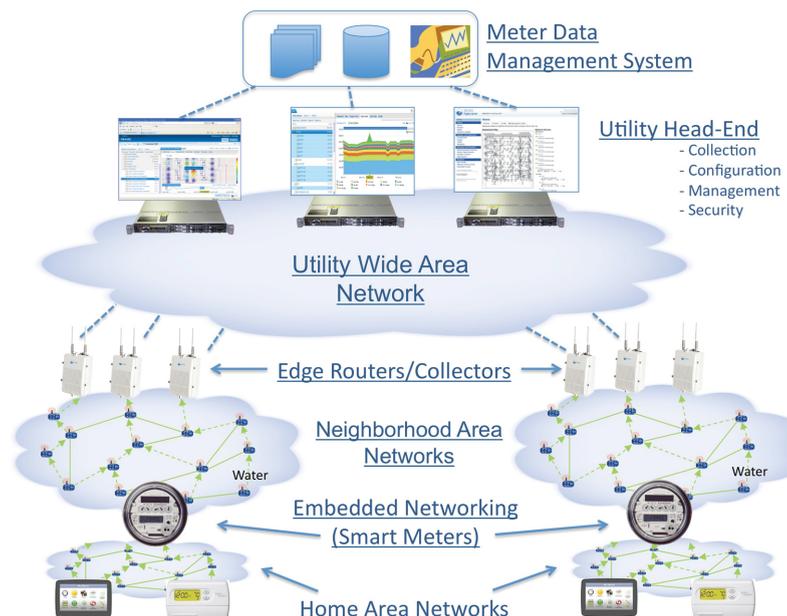


FIGURE 2.1: Architecture d'un Smart-Metering [13].

Les données des compteurs intelligents sont envoyés vers les routeurs qui sont connectés directement au réseau du fournisseur. La donnée est stockée dans ses bases de données pour ensuite être traitée et sortir avec une information utile tel une alerte de haute consommation, une alerte de détection de fuite ainsi que la facturation automatique entre autres applications possibles.[14]

## 2.4 Les enjeux du Smart-Metering

Le Smart-Metering présente des enjeux important que ce soit pour la ville ou pour les fournisseurs :

1. Pour la Ville :

- (a) Amélioration de la qualité de facturation.
- (b) Possibilité de facturation mensuelle
- (c) Permettre aux clients de suivre et piloter leurs consommations et factures.
- (d) Augmenter la satisfaction des citoyens par l'amélioration du service public :  
La réduction de la consommation permet aux opérateurs d'économiser de la ressource et de la redistribuer vers le reste de la population et d'assurer ainsi une meilleure qualité de service.
- (e) en résultat :
  - Moins de délestages.
  - Une meilleure continuité de service.
  - Préservation des ressources pour les générations futures.

2. Pour les Fournisseurs :

- (a) Préservation de la ressource.
- (b) Amélioration de la qualité de la relève et réduction des réclamations.
- (c) Amélioration du service de distribution d'eau potable.
- (d) Détection de fuites et d'anomalies dans le réseau de distribution
- (e) Meilleure information aux clients et meilleure connaissance de la consommation des clients et son évolution.

## 2.5 Conclusion

La majorité des pays développés commencent à converger leurs systèmes de relevé de consommation et de facturation vers le Smart-Metering. Ceci résulte du fait que ce nouveau concept est le fruit de plusieurs années de recherche et développement dans les laboratoires et industries les plus reconnues dans le monde. Et surtout que cela s'intègre dans le contexte de l'IoT qui est maintenant considéré comme la 4<sup>ème</sup> révolution industrielle. Un tel projet présente beaucoup d'intérêt pour les organisations dans leur lutte pour la conservation des ressources pour une économie plus durable pour le pays.

# Chapitre 3

## Plateforme Smart-City

### 3.1 Introduction

Une fois que les consommations sont relevées ou lues par les capteurs, et envoyées vers les concentrateurs, elles doivent être traitées pour en sortir avec des conclusions, des décisions ou des actions à prendre, autrement dit, on doit extraire l'information essentielle des données collectées. Ceci nécessite des traitements et des analyses de données. Vu le nombre croissant du nombre de capteurs, cela nécessite des entrepôts de données assez volumineux et des traitements assez lourds. Les bases de données relationnelles ne seront plus utiles, il a été estimé que pour avoir une donnée d'un des capteurs d'un seul entrepôt de données d'une ville il faudra 20 minutes après avoir envoyé une requête. Pour résoudre ce-là, les données sont stockées dans des bases de données non relationnelles dites Big-Data. Un petit cluster de serveurs ne sera plus suffisant, il sera donc question d'un cloud qui réunit plusieurs clusters de serveurs où emmagasiner les données et les traiter. Dans ce chapitre nous allons exposer les principes du Cloud computing accompagné de quelques exemples de solutions cloud. Nous parlerons aussi du Big data suivi par les plateformes Smart-City étudiées et exploitées dans ce travail et terminer par exposer quelques solutions d'applications qui est l'exposition des données aux clients.

### 3.2 Architecture générique d'une plateforme

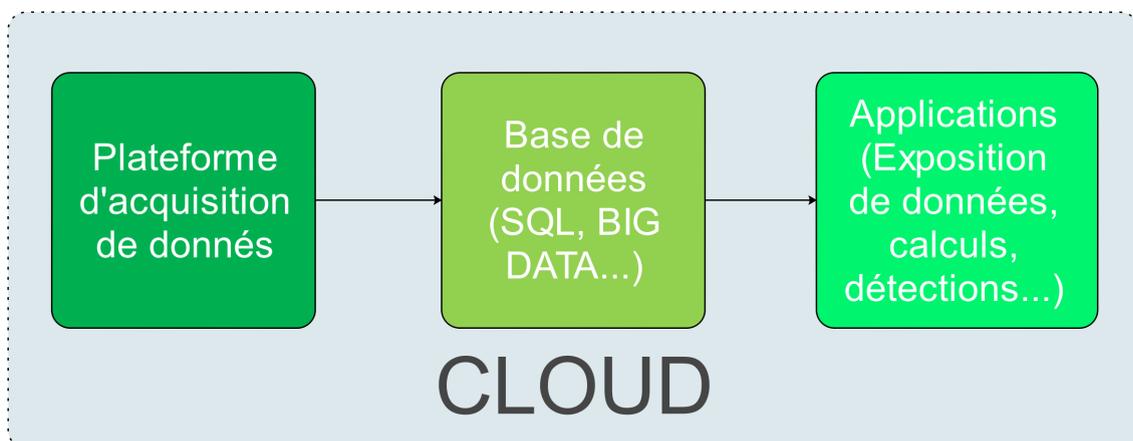


FIGURE 3.1: Plateforme générique d'une plateforme IoT - Partie informatique.

## 3.3 Cloud Computing

### 3.3.1 Définitions

Le Cloud Computing est un terme général employé pour désigner la livraison de ressources et de services à la demande par internet. Il désigne le stockage et l'accès aux données par l'intermédiaire d'internet plutôt que via le disque dur d'un ordinateur. Il s'oppose ainsi à la notion de stockage local, consistant à entreposer des données ou à lancer des programmes depuis le disque dur. La notion de Cloud ne doit pas non plus être confondue avec celle du Network Attached Storage (NAS), utilisée par beaucoup d'entreprises via un serveur en résidence. Ces réseaux locaux n'entrent pas dans la définition du Cloud. Cependant, certains NAS permettent d'accéder aux données à distance depuis Internet.

De manière générale, on parle de Cloud Computing lorsqu'il est possible d'accéder à des données ou à des programmes depuis internet, ou tout du moins lorsque ces données sont synchronisées avec d'autres informations sur internet. Il suffit donc pour y accéder de bénéficier d'une connexion internet.

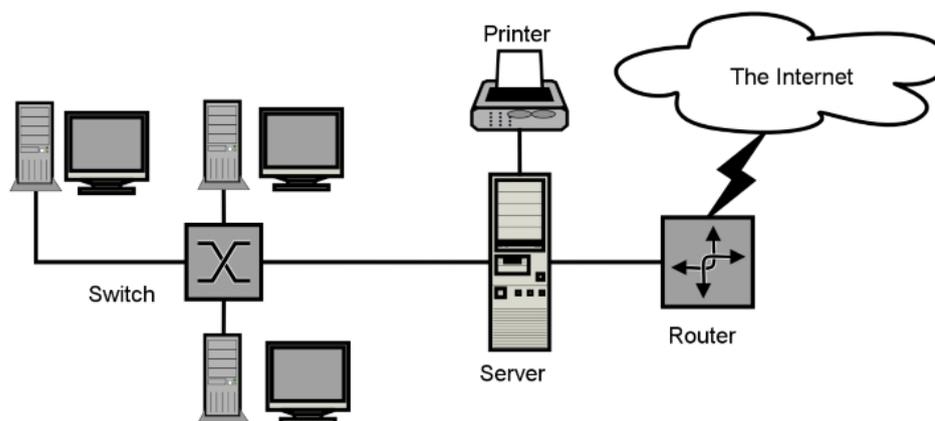


FIGURE 3.2: Illustration du principe de cloud [15].

L'image du Cloud est utilisée de façon métaphorique pour désigner internet. Cette comparaison date de l'époque à laquelle on représentait les infrastructures gigantesques des fermes de serveurs internet sous la forme d'un grand nuage blanc, acceptant les connexions et distribuant des informations tout en flottant.

Cette technologie permet aux entreprises d'acheter des ressources informatiques sous la forme de service, de la même manière que l'on consomme de l'électricité, au lieu d'avoir à construire et entretenir des infrastructures informatiques en interne.

Plusieurs solutions ont été mises en oeuvre par des géants de technologie, parmi ces solutions, certaines sont libre et exploitable. Nous commençons par exposer les deux solutions que nous avons exploré avant de mettre en oeuvre notre système ensuite nous parlerons brièvement des solutions proposées par les teneurs du marché cloud.[15]

### **3.3.2 Les modèles de services du cloud**

#### **3.3.2.1 SaaS (Software as a Service)**

Utilisation d'un logiciel, comme un service. Le logiciel n'est pas installé sur l'ordinateur du client mais il est utilisé à distance sur des serveurs dans des datacenters. La plupart du temps, ces logiciels s'utilisent via un navigateur web.

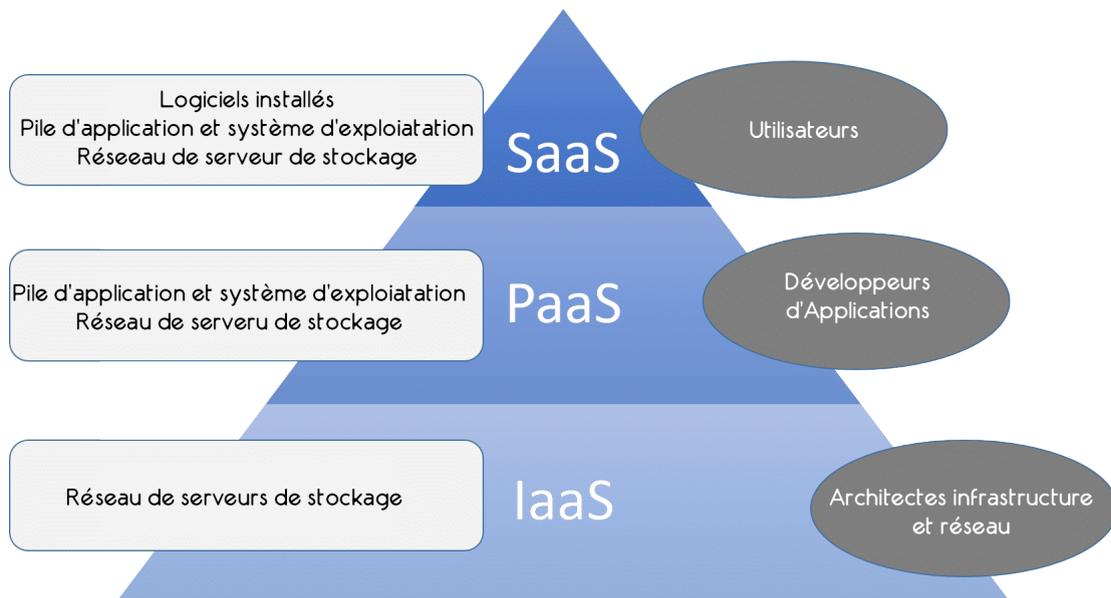
#### **3.3.2.2 IaaS (Infrastructure as a Service)**

L'infrastructure en tant que service (IaaS) contient les blocs de construction fondamentaux de l'informatique dans le cloud et donne habituellement accès à des fonctionnalités de mise en réseau, à des ordinateurs (virtuels ou sur du matériel dédié) et à de l'espace de stockage de données. Le service IaaS offre le niveau le plus élevé de flexibilité et de contrôle de gestion en ce qui concerne les ressources informatiques et est très similaire aux ressources informatiques existantes avec lesquelles les services informatiques et les développeurs sont aujourd'hui familiarisés.

#### **3.3.2.3 PaaS (Platform as as Service)**

Utilisation d'une plateforme, comme d'un service. On peut louer une machine avec un OS pré-installé et on l'utilise directement. Grâce au service PaaS, les entreprises n'ont plus besoin de gérer l'infrastructure sous-jacente (en règle générale, le matériel et les systèmes d'exploitation). On se concentre sur le déploiement et la gestion des

applications. On ne se soucie plus de l'approvisionnement des ressources, de la planification des capacités, de la maintenance logicielle, de l'application de correctifs ou de toute autre charge indifférenciée liée à l'exécution de l'application.[16]



### 3.3.3 Les modèles de déploiement

1. Cloud Privé : L'ensemble des ressources n'est affecté qu'à une seule organisation ou entreprise.
2. Cloud Communautaire : L'ensemble des ressources provient de plusieurs organisations ou entreprises qui se les partagent.
3. Cloud Public : Les ressources sont mises à disposition de tout le monde.
4. Cloud Hybride : Il s'agit d'un mixte entre le privé et le public.

### 3.3.4 OpenStack

OpenStack est un système d'exploitation cloud qui contrôle un grand nombre de ressources dédiées aux calculs, stockage ainsi qu'à la gestion du réseau à travers un datacenter, toutes gérés à travers une interface web qui donne contrôle aux administrateurs sur les utilisateurs pour leur fournir les ressources demandées. OpenStack fournit une solution Infrastructure-as-a-Service(IaaS) à travers des services reliés entre eux.

### 3.3.4.1 Architecture d'openStack

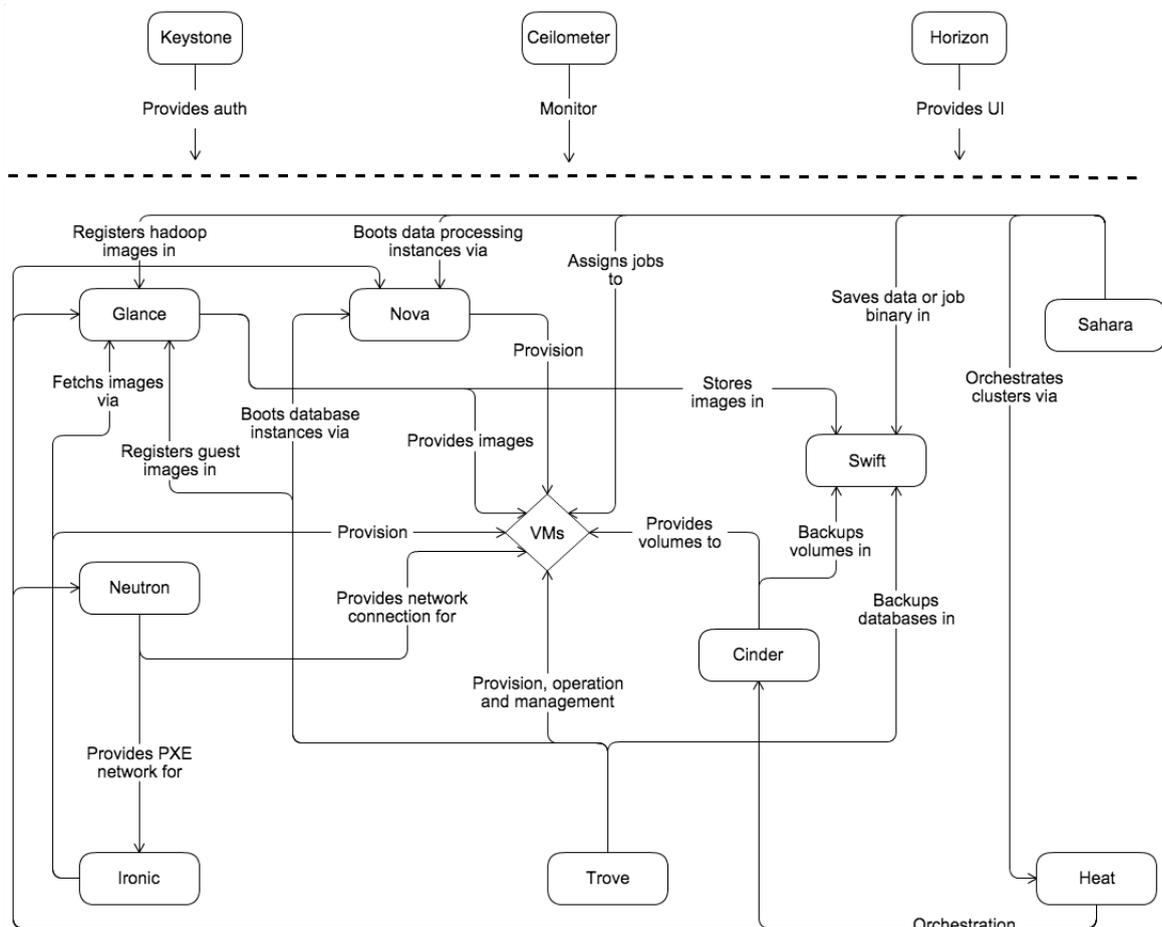


FIGURE 3.3: Architecture conceptuelle d'OpenStack [17].

Dans la section qui suit nous expliquons le rôle des services exposés dans cette architecture [18].

#### 3.3.4.2 Services d'OpenStack

**Horizon (Tableau de bord)** Horizon fournit un portail web pour interagir avec les services d'openstack, comme lancer une instance, attribuer une adresse IP ou configurer le contrôle d'accès pour les utilisateurs.

**Nova(Service Calcul)** Nova gère le cycle de vie des instances compute. Ce service aider à générer, planifier et désaffecter des machines virtuelles à la demande.

**Neutron(Service Réseau)** Neutron active Network-Connectivity-as-a-Service pour les autres services comme le service Compute. Il fournit une API pour les utilisateurs afin de configurer leurs réseaux. Il dispose d'une architecture qui supporte plusieurs technologies et vendeurs réseau.

**Swift(Service de stockage objet)** Stock et peut retrouver des objets de données non structurés via une API-REST. Swift est tolérant en terme d'erreur de replication et du dimensionnement de l'architecture. Swift écrit les objets et fichiers dans plusieurs disques, assurant ainsi que la donnée est répliquée à travers un cluster de serveurs.

**Cinder(Service de stockage bloc)** Cinder fournit un stockage bloc persistant aux instances en exécution. Il dispose d'une architecture de driver qui facilite la création et la gestion des dispositifs de stockage bloc.

**Keystone(Service d'identification)** Fournit un service d'authentification et d'autorisation pour les autres services.

**Glance(Service d'image)** Stock et retrouve les images disque de machines virtuelles. Le service calcul (Compute Service) utilise glance pendant l'approvisionnement d'une instance.

**Ceilometer(Service de télémétrie)** Surveille et mesure le cloud pour le benchmarking, la scalabilité ainsi que pour faire des statistiques sur la consommation de ressources...

**Heat(Service d'orchestration)** Orchestre plusieurs applications cloud en utilisant le format de template natif HOT à travers une API Rest.

**Trove(Service de base de données)** Fournit une fonctionnalité scalable et fiable de Cloud Database-as-a-Service pour les bases de données relationnelles ou non relationnelles.

**Sahara(Service de traitement de données)** Permet l’approvisionnement et la mise à l’échelle des clusters Hadoop dans OpenStack en spécifiant les paramètres comme la version d’Hadoop, la topologie des clusters et les détails sur les noeuds hardware.

### 3.3.5 Proxmox

Proxmox VE est une solution open source complète de management de serveurs virtualisés. Proxmox est basé sur la virtualisation KVM et la virtualisation de type conteneur et permet le management de machines virtuelles, du stockage, de réseau virtualisé et de cluster haute disponibilité.

Contrairement à OpenStack, Proxmox ne nécessite qu’un seul serveur pour être prêt à l’emploi. Il est installé au dessus d’un noyau linux. Il est ensuite possible d’étendre à plusieurs noeuds (serveurs) pour augmenter la capacité en RAM, en disque et en nombre de processeurs.

### 3.3.6 Autres solutions

Les solutions connues de géants technologiques sont généralement payantes et proposent tout type de services, SaaS, IaaS et PaaS.

- Microsoft AZURE.
- Amazon Web Service.
- IBM Cloud.

## 3.4 la plateforme Smart-City

Une plateforme Smart-City peut être vue comme une couche qui isole la partie applicative composée des différentes applications développées pour exploiter les informations «générées par la ville» et la couche de capteurs déployés à travers la ville pour collecter et diffuser ces informations.

Le plus grand intérêt de ces plateformes réside dans le fait qu’elles permettent une interopérabilité facile entre les deux couches en normalisant les échanges à travers

l'introduction du concept des API. Les API sont un ensemble normalisé de fonctions et méthodes qui permettent l'interfaçage et l'accès au service d'un logiciel.

L'exemple le plus connu de plateforme est celui de Sentilo. Cette plateforme est utilisée par la ville de Barcelone principalement et plusieurs autres villes pour gérer les réseaux de capteurs déployés.

## 3.5 Big data

### 3.5.1 Introduction du big data

De grandes quantités d'informations sont mises en ligne sur le web par des milliers d'entreprises, d'organisations et d'individus, la charge ainsi que le volume de données à gérer ont crû de façon exponentielle, pour cela les sociétés ont recouru au Datawarehouse ou entrepôt de données pour l'analyse et le stockage de données. Généralement le Datawarehouse est centralisé dans un serveur connecté à une baie de stockage, cette solution est difficilement scalable (ajout de puissance à la demande) en plus du fait qu'elle ne gère que les données structurées. Pour faire face à l'explosion du volume des données, on parle actuellement de pétaoctet (billiard d'octets) voir de zettaoctet (trilliard d'octets) et aussi face à la grande variété des données (image, texte, web, etc.) un nouveau domaine technologique a vu le jour : le Big Data inventé par les géants du web, au premier rang comme Yahoo, Google et Facebook, qui ont été les tous premiers à déployer ce type de technologie. Ce concept apporte une architecture distribuée et scalable pour le traitement et le stockage de données. Ce nouveau paradigme a pour principal objectif l'amélioration des performances et l'augmentation de la vitesse d'exécution des requêtes et des traitements.[19]

Le terme de Big Data a été évoqué la première fois par le cabinet d'études Gartner en 2008 mais la naissance de ce terme effective remonte à 2001 et a été évoqué par le cabinet Meta Group. Il fait référence à l'explosion du volume des données (de par leur nombre, la vitesse à laquelle elles sont produites et leur variété) et aux nouvelles solutions proposées pour gérer cette volumétrie tant par la capacité à stocker et explorer, et récemment par la capacité à analyser et exploiter ces données dans une

approche temps réel. Big data, littérairement les grandes données, est une expression anglophone utilisée pour désigner des ensembles de données qui deviennent tellement volumineux qu'ils en deviennent difficiles à travailler avec des outils classiques de gestion de base de données. Il s'agit donc d'un ensemble de technologies, d'architecture, d'outils et de procédures permettant à une organisation très rapidement de capter, traiter et analyser de larges quantités et contenus hétérogènes et changeants, et d'en extraire les informations pertinentes à un coût accessible. [19]

### 3.5.1.1 Caractéristiques du Big Data

Le Big Data (en français "Grandes données") regroupe une famille d'outils qui répondent à une triple problématique : C'est la règle dite des 3V.

**Volume** Le Big Data est associé à un volume de données vertigineux, se situant actuellement entre quelques dizaines de téraoctets et plusieurs péta-octets en un seul jeu de données. Les entreprises et tous les secteurs d'activités confondus, devront trouver des moyens pour gérer le volume de données en constante augmentation qui est créé quotidiennement. Les catalogues de plus de 10 millions de produits sont devenus la règle plutôt que l'exception.

**Vélocité** La vélocité ou la vitesse décrit la fréquence à laquelle les données sont générées, capturées et partagées. Les entreprises doivent appréhender la vitesse non seulement en termes de création de données, mais aussi sur le plan de leur traitement, de leur analyse et de leur restitution à l'utilisateur en respectant les exigences des applications en temps réel.

**Variété** La croissance de la variété des données est la conséquence des nouvelles données multi structurelles et de l'expansion des types de données provenant de différentes sources hétérogènes. Aujourd'hui, on trouve des capteurs d'informations aussi bien dans les appareils électroménagers, les trains, les automobiles ou les avions, qui produisent des informations très variées. Ces nouvelles données dites non-structurées sont variées : photos, e-mails avec analyse de leur contenu, les données issues des réseaux sociaux, etc.[19]

### 3.5.1.2 Architecture Big Data

On distingue principalement les couches suivantes :

1. Couche matériel (infrastructure Layer) : peut-être des serveurs virtuels VMware, ou des serveurs bare métal.
2. Couche stockage (Storage layer) : les données seront stockées soit dans une base NoSQL, ou bien directement dans le système de fichier distribué ou les Datawarehouse.
3. Couche management et traitement : on trouve dans cette couche les outils de traitement et analyse des données comme MapReduce ou Pig.
4. Couche visualisation : pour la visualisation du résultat du traitement.

### 3.5.2 Les bases de données MPP (Massive Parallel Processing)

Ce sont des bases de données distribuées, tant au niveau du stockage que du processing. Le principe est de partager automatiquement les données entre une collection de segments, une grappe de segments étant gérée de façon autonome par un noeud de calcul et tous les noeuds communiquent via un réseau dédié.

Un processeur SQL ordonne l'exécution des requêtes entre les différents noeuds qui traitent localement chaque requête, puis les résultats de chaque noeud sont concaténés, soit au niveau du processeur, soit directement par la couche réseau.

La distribution des données entre les différents segments/noeuds est critique pour les bonnes performances du système. A la différence d'un système de base de données classique qui utilise également des techniques de partitionnement des données pour maintenir de bonnes performances, le secret de la performance d'un système MPP réside dans l'utilisation d'une clé de distribution indépendante de la logique de données. Là où un système transactionnel utilisera par exemple un partitionnement par mois de la transaction pour pouvoir intelligemment filtrer les segments utiles dans une requête et ainsi diviser les besoins de chargement, l'approche MPP privilégie l'utilisation de clé

de hachage opaque qui permet une distribution des données entre un nombre variable de segments.

L'avantage premier est une grande évolutivité du système lorsque les volumes de données augmentent : s'il y a  $n$  segments, chaque segment ne stocke que  $1/n$  pourcent des données.

Le second avantage est de grandement simplifier l'administration du système puisqu'il n'est généralement pas nécessaire de recourir à un partitionnement explicite (la plupart des systèmes MPP actuels supportent également des fonctions de partitionnement classiques au dessus du mécanisme natif de distribution, les partitions étant elles-mêmes distribuées).

L'ajout de segments est généralement transparent pour l'utilisateur et le système est capable de redistribuer l'ensemble des données automatiquement. Ce mécanisme de redistribution des données peut également intervenir lors d'une jointure si les clés de distribution des tables concernées ne sont pas compatibles entre elles – de fait le choix de la clé de distribution est un des leviers majeurs dans l'optimisation d'un système MPP. [20]

## 3.6 Les applications

Les applications représentent la dernière couche du système. Dans le cas du Smart Metering, ils exploitent la masse de données collectées des mesures de consommation des clients en appliquant des algorithmes d'analyse de données et de l'IA pour tirer les informations qui peuvent être utile pour les prestataires de service dans le suivi et la gestion de leur réseaux de distribution.

Parmi les applications, on peut citer :

- **suivi de consommation** avec la disponibilité des données, les entreprises proposent au client de visualiser et suivre l'évolution de sa consommation en temps réel. Ce service permet aux citoyens de mieux gérer leur consommation

- et d'économiser de l'argent. Ils auront même la possibilité de payer leur factures mensuellement au lieu d'un paiement semestriel pour éviter les grosses sommes.
- **Détection d'anomalies** Sachant que les fournisseurs de ressources ont des compteurs individuels installés chez les clients et des compteurs de régions. Chaque compteur de région englobe un territoire d'un rayon de 25Km. On peut en cas de fuite, exploiter cette information en procédant par élimination. SEAAL n'aura pas à balayer dix régions alors que le plus grand nombre de fuites est centré en une seule région.

### 3.7 Conclusion

Une plateforme de Smart metering est un assemblage de plusieurs technologies logiciels. En premier, on trouve le Cloud qui héberge toutes les composantes de la plateforme. Ensuite, la plateforme d'acquisition de données qui est la couche d'isolation entre les applications et les capteurs. Les outils de Big Data sont une partie importante de l'architecture vu le volume de données énorme collecté à partir des capteurs. Enfin, les applications qui exploitent les mesures pour améliorer la gestion des ressources (Eau, Gaz ou électricité).

La partie suivante propose une implémentation de ces principes dans un cas d'usage qui est le Smart-Metering appliqué à l'eau.

## Deuxième partie

# Implémentation : Smart Metering pour l'EAU

# Chapitre 4

## Systeme d'acquisition

## 4.1 Introduction

Il est important de bien voir le problème posé et étudier les compteurs d'eau installés pour pouvoir interagir avec et choisir le bon capteur pour relever la consommation en eau. Dans ce chapitre nous allons commencer par définir l'architecture de notre système d'acquisition suivi de l'étude des compteurs d'eau et des capteurs possibles pour la lecture, ensuite nous parlerons des solutions de communication pour bien éclaircir notre système d'acquisition de données et son mode de fonctionnement.

## 4.2 Schéma de notre solution

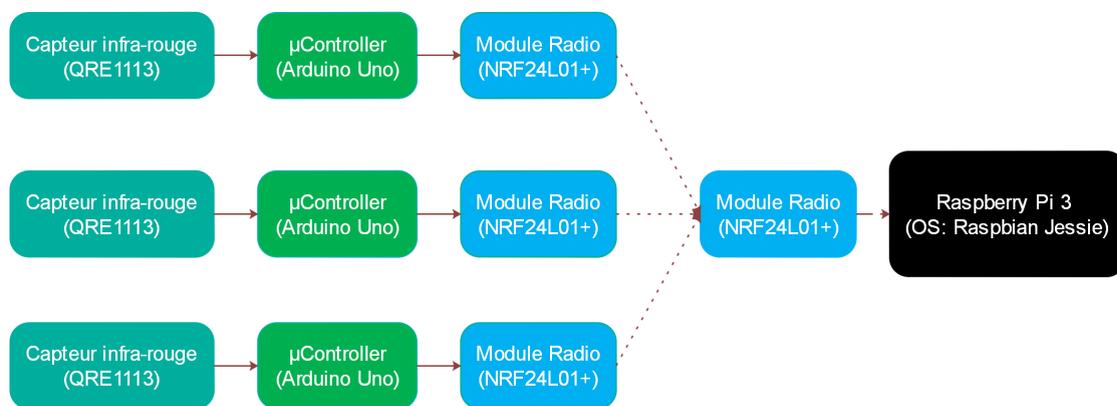


FIGURE 4.1: Système de relevé automatique de la consommation en eau.]

## 4.3 Le compteur d'eau

Un compteur d'eau est un appareil qui mesure le volume d'eau utilisée par un consommateur. Ils peuvent également être utilisés dans un puits, une source d'eau ou un système d'eau pour déterminer le débit d'eau.

### 4.3.1 Types de compteurs

#### 4.3.1.1 Compteur à déplacement

Ces compteurs mesurent l'espace occupé par l'eau pendant un temps déterminé. Le flux d'eau déplace le dispositif de mesure selon le volume d'eau en  $m^3$  qui passe à

travers le compteur. Un piston ou un disque contrôle le mouvement d'un aimant qui active l'enregistrement du volume d'eau.

**Compteur à déplacement positif** C'est un compteur qui exige que le fluide déplace mécaniquement ses composants. Il mesure le débit volumétrique d'un liquide ou d'un gaz en mouvement en divisant le milieu en volumes fixes mesurés (incrémentations ou volumes finis du fluide). Ces dispositifs consistent en une ou plusieurs chambres qui obstruent le flux et un mécanisme de rotation ou de va-et-vient qui permet le passage de quantités de volume fixe. Le nombre de "parcelles" qui traversent la chambre détermine le volume du support. La vitesse de rotation ou de réciprocité détermine le débit. Il existe deux types de débitmètres à déplacement positif. Les systèmes à capteurs uniques ou les transducteurs sont des dispositifs analogues à des commutateurs qui fournissent des sorties électroniques pour les processeurs, les contrôleurs ou les systèmes d'acquisition de données.

#### 4.3.1.2 Compteur vitesse

Les compteurs de vitesse suivent généralement une relation linéaire par rapport au débit volumique. Leur rangeabilité est beaucoup mieux comparé à d'autres débitmètres. Ils sont aussi moins sensibles aux changements de viscosités. Presque tous les boîtiers sont équipés de brides ou de raccords. Les principaux types de compteurs vitesse comprennent les modèles à turbine entre autres.

**Compteurs de vitesse à jets multiples** Le flot frappe par injection en plusieurs points la turbine. Les compteurs vitesse sont peu sensibles à la présence d'éventuelles impuretés dans l'eau et s'avèrent économiques à l'achat comme à la location. Problème : Ils ne détectent pas toujours les fuites légères et doivent obligatoirement être posé horizontalement. Par ailleurs, dans le cas de retours d'eau, le totaliseur tourne à l'envers, ce qui fausse les relevés.

**Les débitmètres à trubines** Les débitmètres à turbine sont constitués d'un rotor à lames multiples installé avec un tuyau dans une direction perpendiculaire à l'écoulement du fluide. Lorsque le liquide s'écoule à travers les lames, le rotor tourne.

La vitesse de rotation du rotor est corrélée au débit du liquide. La vitesse du rotor peut être détectée par divers mécanismes tels qu'un capteur magnétique, une cellule photoélectrique ou des engrenages. Un tachymètre peut également être fixé à la turbine pour mesurer sa vitesse de rotation, ce qui à son tour aide à déterminer le débit du liquide. Les compteurs à turbine offrent une bonne précision, principalement avec des liquides à faible viscosité. Par conséquent, les compteurs à turbine sont largement utilisés dans des applications où des mesures précises sont nécessaires. Lorsqu'un débitmètre à turbine est calibré et utilisé pour un fluide unique, des conditions constantes, son rapport de réduction peut aller au-delà de 100 :1. De plus, la précision d'un débitmètre à turbine peut atteindre  $\pm 0,1\%$ . Le principal problème associé à l'utilisation de ces compteurs est l'usure des roulements. Pour éviter ce problème, une conception de compteur à turbine sans palier a été introduite récemment.

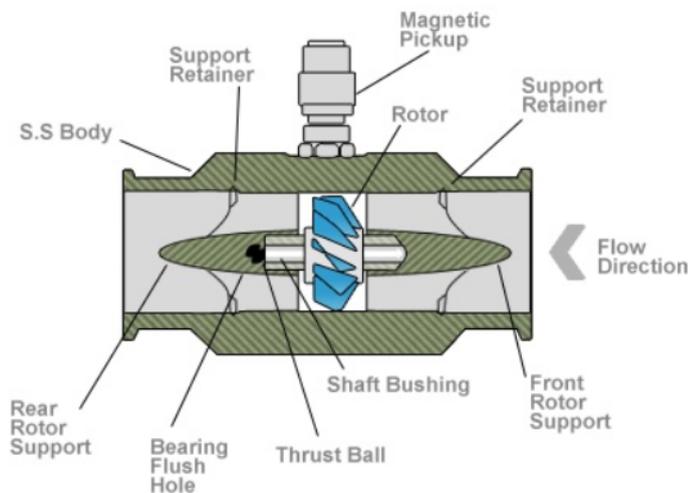


FIGURE 4.2: Compteur à turbine

**Compteurs d'eau composés** Dans les cas où le débit d'eau peut varier d'une vitesse élevée à faible, un compteur composé peut mesurer avec précision l'utilisation de l'eau. Ceux-ci ont deux composants de mesure et une valve pour contrôler le débit à chacun. Le compteur à turbine gère des débits élevés et un compteur à jets multiples mesure le débit le plus bas. Le lecteur utilitaire ajoute les totaux indiqués sur les deux compteurs pour déterminer la consommation totale d'eau.

### 4.3.1.3 Compteur électromagnétique

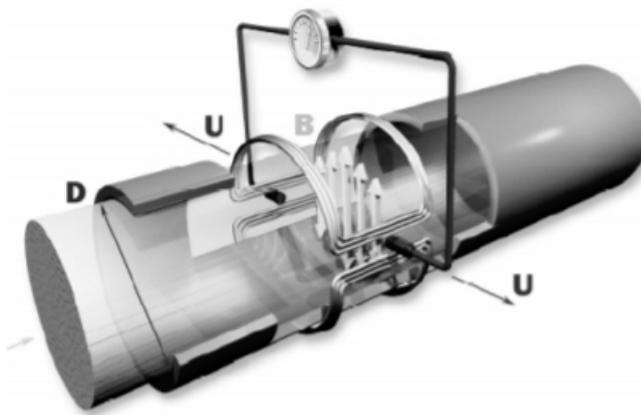


FIGURE 4.3: Compteur électromagnétique

Il fonctionne suivant le principe de Faraday. Quand un liquide conducteur s'écoule perpendiculairement à travers un champ magnétique, une différence de potentiel électrique est créée au sein du liquide.

### 4.3.1.4 Compteurs Ultrasonique

Le compteur à ultrasons est un instrument utilisant les ultrasons pour mesurer la vitesse moyenne d'un fluide.

**Principe de mesure** Une corde de mesure est constituée de deux sondes : la première est placée en amont (sonde A) tandis que la seconde est placée en aval (sonde B). Chaque sonde d'une corde est alternativement émettrice et réceptrice d'une onde ultrasonore impulsionnelle dirigée vers l'autre sonde. On mesure la différence du temps de propagation amont-aval et aval-amont. Connaissant la distance entre les deux sondes, on peut mesurer la vitesse d'écoulement moyenne et en déduire le débit.

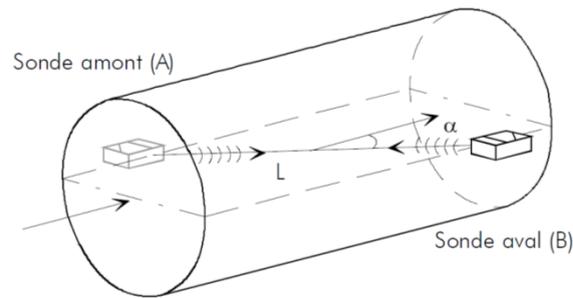


FIGURE 4.4: Principe de mesure du compteur Ultrasonique

Différentes méthodes sont possibles :

1. Compteur à effet Doppler : en analysant le changement de l'onde réfléchie par une particule du fluide, on peut déduire la vitesse des particule et donc du fluide.
2. Compteur par mesure de différence des temps de transit : on mesure le temps de parcours de l'onde ultrasonore d'amont/aval à aval/amont. Cette différence de temps est en fait l'image de la vitesse moyenne du fluide.

### 4.3.2 Compteur SEAAL

Les compteurs SEAAL sont des compteurs à vélocité à jet multiple 420. Fiables, résistant aux eaux chargées et silencieux. La transmission à travers les mécanismes des compteurs vers le cadran permet d'obtenir une bonne sensibilité aux faibles débits. Ceci est illustré dans le prochain graph :

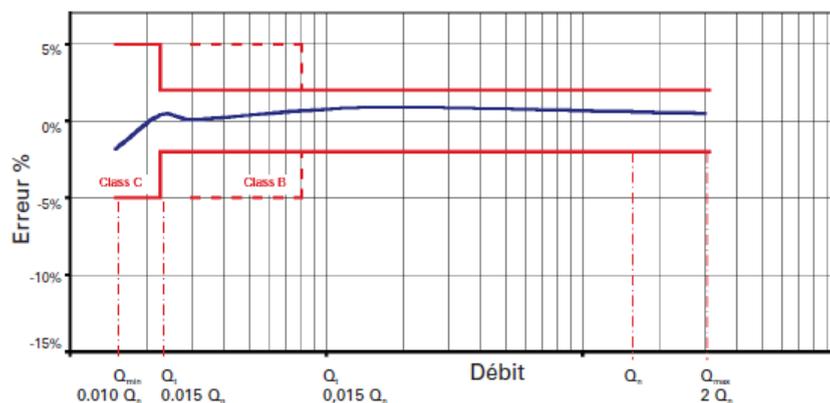


FIGURE 4.5: Courbe de Précision du Modèle 420[21].

On peut voir que l'erreur ne dépasse pas les 5% que ce soit en ajout ou en soustraction de la valeur exacte. En faible débit, l'erreur va jusqu'à -2%, ce qui est minime puisque le débit est petit. En forts débits, l'erreur ne dépasse pas les 1.5%.

#### 4.3.2.1 Le totalisateur du compteur 420

Il dispose de 5 rouleaux avec un marquage noir sur fond blanc. L'index peut être lu à une distance de plus d'un mètre. Les aiguilles indiquent les sous-multiples du m<sup>3</sup>. Le corps porte deux flèches, indiquant le sens de l'écoulement, le débit nominal, la classe métrologique, le numéro d'approbation de modèle CEE, l'année de fabrication et le numéro de série du compteur figurent sur la plaque d'identification au-dessus du cadran. Les compteurs 420 sont insensibles aux champs magnétiques appliqués à proximité, ce qui protège contre toute tentative de fraude. [21]

## 4.4 Capteurs

### 4.4.1 Solutions possibles

Beaucoup de solutions sont possibles selon la forme du cadran, son affichage, la forme des aiguilles, leur composition (métal, plastique...) :

- **Détection magnétique** Une bonne partie des compteurs mécaniques existants contient un aimant superposé à l'engrenage. D'autres compteurs sont caractérisés par un métal collé sur l'aiguille principale du cadran (l'aiguille qui compte les litres) tel le 420 de SEAAL. Il serait donc intéressant d'utiliser un capteur magnétique pour éventuellement trouver une courbe de détection en fonction de la consommation. Nous avons donc pensé au capteur à effet Hall.
- **Détection Infrarouge** Le principe d'un capteur par réflexion est d'éclairer l'obstacle avec une LED infrarouge, et de mesurer la lumière réfléchie avec une photodiode ou un phototransistor. Cette technique peut être utilisée pour détecter le nombre de tours d'une des aiguilles.
- **Capture photographique** On peut poser une caméra au-dessus du cadran du compteur. Ceci permet de capter le mouvement des différents indicateurs

présents. Un traitement d'image permet ensuite d'extraire la consommation. Cette technique demande plus de ressources en calcul et elle est peu utilisée.

#### 4.4.2 Solution choisie

Les compteurs de SEAAL sont des compteurs construits à base de plastique, il n'est donc pas possible d'utiliser des détecteurs magnétiques. Nous avons opté pour un capteur émetteur/récepteur Infrarouge, le QRE1113. Ce capteur est composé d'une LED émettrice et d'un photo-transistor pour recevoir la lumière réfléchie.

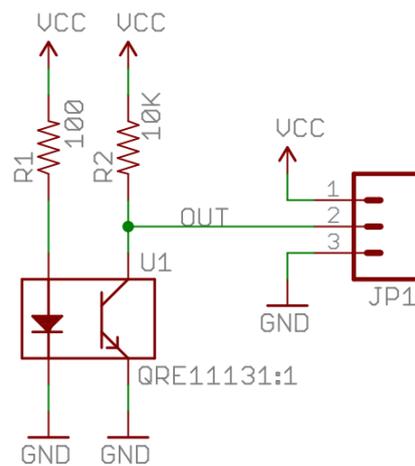


FIGURE 4.6: Schéma électrique pour la lecture du capteur infrarouge[22].

Le capteur infrarouge est placé sur l'aiguille des litres du cadran du compteur d'eau au point 0. La figure suivante montre l'aiguille :

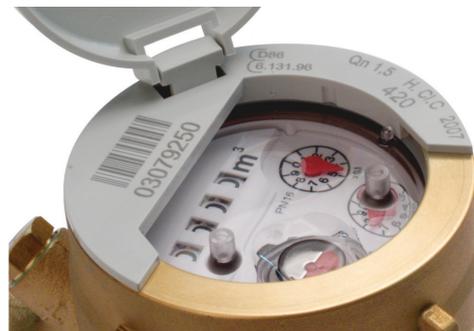


FIGURE 4.7: cadran du Compteur.

## 4.5 Solution de communication

Pour la partie communication, notre choix est allé vers le module nRF24L01 par contraintes de disponibilité de module et d'utilisation des fréquences (l'utilisation des fréquences des réseaux IoT est interdite par la loi Algérienne).

Le module radio nRF24L01 est un module radio tout intégré du fabricant Nordic Semiconductor. C'est un module ayant les fonctionnalités nécessaires pour émettre et recevoir des données sur la gamme de fréquences de 2.4GHz en utilisant le protocole de communication propriétaire de Nordic nommé "ShockBurst".

Ce protocole de communication permet au nRF24L01 d'être considéré comme un modem complet, avec adressage, gestion des erreurs de transmission et retransmission automatique en cas de non-réponse du destinataire.

Le nRF24L01 configuré comme receveur est capable de communiquer avec 6 émetteurs en même temps.

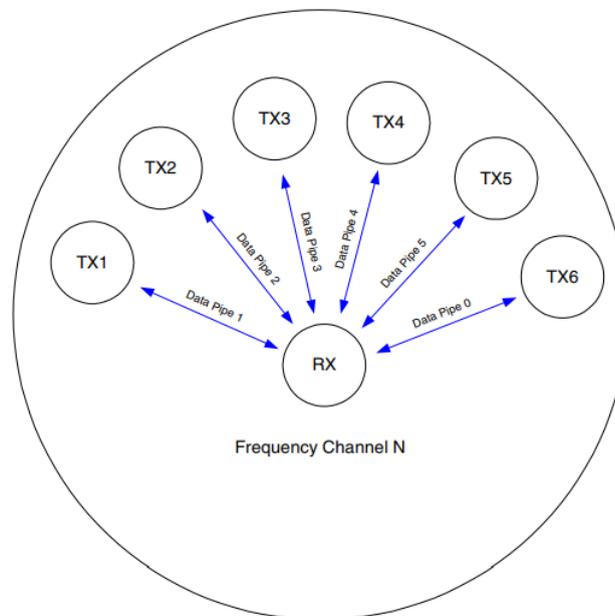


FIGURE 4.8: Communication avec le nRF24L01.

## 4.6 Conclusion

Dans ce chapitre d'implémentation. Nous nous sommes intéressés à la couche basse d'un système de Smart-Metering pour l'eau. Nous avons étudié les différents types de compteurs possibles en plus de celui fourni par SEAAL pour faire un choix sur la technologie à utiliser pour relever la donnée du cadran. Nous avons ensuite présenté notre choix de technologie de communication et les raisons de notre choix.

Le chapitre suivant présentera notre implémentation de la chaîne complète du capteur à la plate-forme Smart-City dédiée au Smart-Metering et nous terminerons par présenter les résultats du travail.

## Chapitre 5

# Plateforme Smart-City dédiée au Smart-Metering

## 5.1 Introduction

Dans ce chapitre, nous proposons une implémentation de l'architecture décrite précédemment. Nous commencerons par l'architecture globale de notre implémentation suivie par une description des différentes solutions choisies dans chaque étage de l'architecture, les ressources matérielles utilisées ainsi que l'architecture réseau. Nous aborderons ensuite le principe de fonctionnement de tout le système allant du capteur jusqu'à la visualisation de la consommation en eau des clients. Enfin, les résultats de cette implémentation seront présentés.

## 5.2 Architecture générale

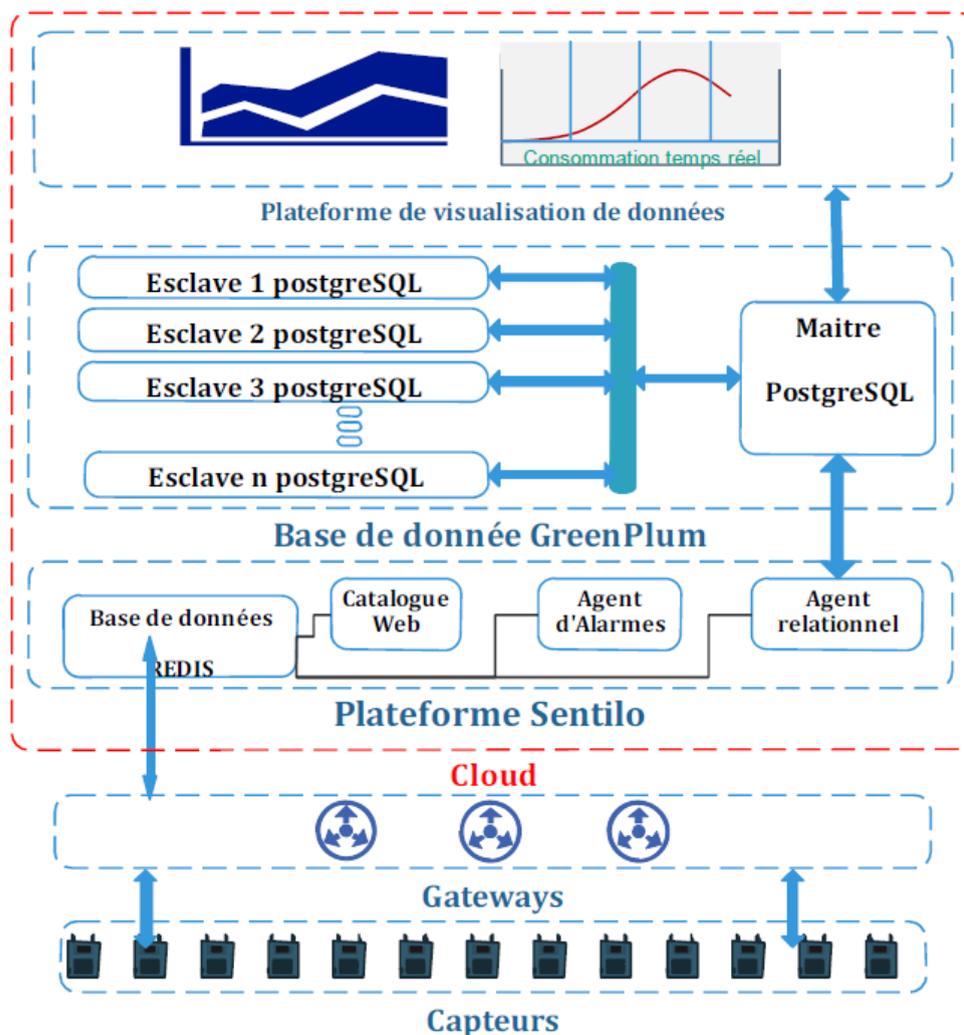


FIGURE 5.1: Architecture générale de la plateforme smart-city.

## 5.3 Sentilo

### 5.3.1 introduction

Sentilo est une plateforme smart-city open-source conçue par une équipe espagnole. Le but de cette plateforme est de réunir l'ensemble des capteurs d'une ville, récupérer les données collectées par ces derniers, les afficher et permettre aux clients de voir ces données. [23]

Les principaux modules de sentilo sont :

- API Rest.
- Une application web qui fournit une plateforme d'administration.
- Un système de publication et de souscription de données.
- Une base de donnée à mémoire pour l'enregistrement de données en temps réel.
- Des bases de données non-SQL pour enregistrer les données du catalogue de capteurs.
- Des agents qui peuvent étendre les fonctionnalités de la plateforme.

### 5.3.2 Plateforme Pub/Sub

Sentilo permet aux clients de publier et de demander des informations ainsi que de souscrire aux évènements du système. C'est un processus Java qui utilise Redis comme un mécanisme Publication/Souscription.

Les types d'informations considérées sont :

- Les observations.
- Les alarmes.
- Les ordres.

### 5.3.3 Stockage temps réel

C'est le moteur Publication/Souscription où la plateforme emmagazine toutes les informations reçues. Elle est configuré pour faire des sauvegardes periodique dans le fichier système.

### 5.3.4 API Rest

Le mécanisme de communication Publication/Souscription entre le client et la plateforme est réalisé au biais de l'API Rest de Sentilo.

Les services offerts par l'API de sentilo peuvent être classifiés en cinq groupes principaux :

- Donnée : Permet des opérations de publication, acquisition, et suppression de de la donnée.
- Ordre : Permet des opérations de publication, acquisition et suppression des ordres.
- Alarme : Permet des opérations de publication, acquisition et suppression des alarmes.
- Souscription : Permet des opérations de publication, acquisition et annulation des souscriptions.
- Catalogue : Permet les opérations d'insertion, mise à jour, requête et de suppression des ressources du catalogue (capteurs, composants et alertes).

### 5.3.5 Agents

Les agents sont des processus Java qui étendent les fonctionnalités de la plateforme à travers un système Plug and Play en utilisant le mécanisme Publication/Souscription de Redis. [24]

Chaque agent représente un processus qui agit comme un souscripteur pour la plateforme Pub/Subs. Ces processus vont souscrire directement vers Redis comme clients indépendants. Cette souscription va fournir la donnée en entrée (base de donnée relationnelle, alarmes, alertes, génération de statistiques...). La figure suivante illustre le fonctionnement des agents :

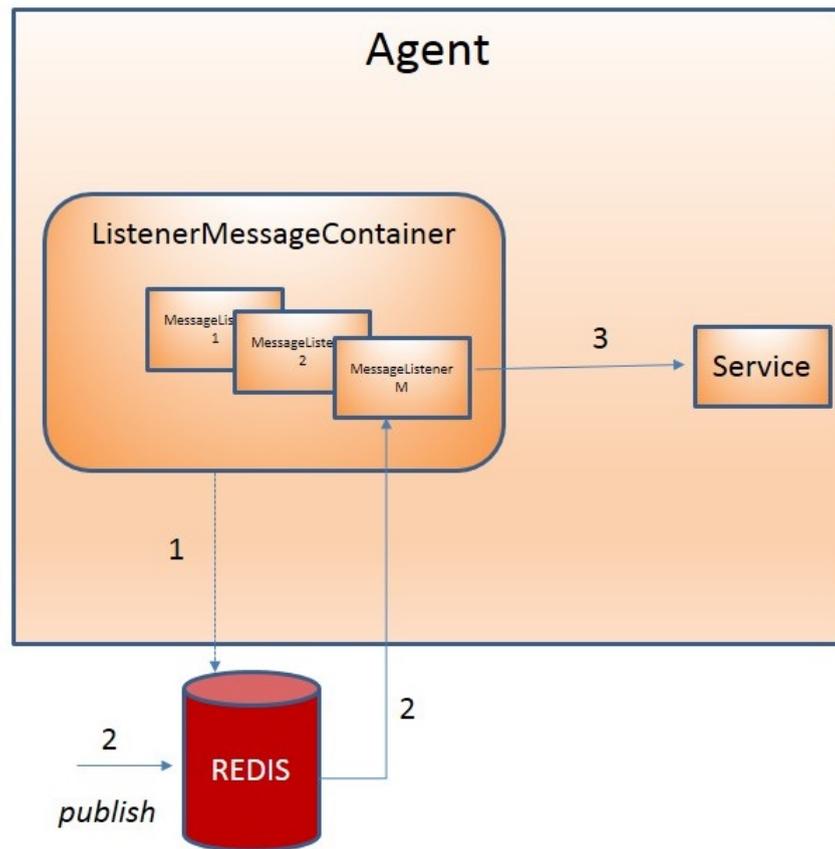


FIGURE 5.2: Fonctionnement standard des agents.

- Lorsqu'un agent est lancé il souscrit comme client à Redis pour recevoir des notifications.
- Lorsque Redis reçoit la donnée, l'agent est automatiquement notifié avec un nouveau message.
- Le message est traité et transféré au service responsable de l'agent correspondant pour mener à bien la logique commerciale sous-jacente.

Sentilo fournit plusieurs agents :

- Agent de base de donnée relationnelle : Utilisé pour exporter les données historiques vers une base de donnée relationnelle.
- Agent d'alertes : Utilisé pour traiter chaque donnée reçue par la plateforme et la valider.
- Agent moniteur d'activités : Utilisé pour exporter les événements vers Elastic-Search.
- Agent historien : Utilisé pour remonter les événements vers OpenTSDB.

### 5.3.5.1 Agent relationnel

l'agent enregistre l'information reçue de la plateforme PubSub dans un ensemble de bases de données relationnelles.

### 5.3.5.2 Agent d'alarmes

Cet agent traite chaque alerte interne définie dans le catalogue et publie une notification quand une des règles n'est pas respectée. Deux processus sont mis en place à cause des types de règles disponibles sur la plateforme :

- Un processus interne qui s'exécute chaque minute, évalue le statut de chaque compteur associé aux alertes. Chaque fois qu'une notification Redis est reçue, les alertes associées à la donnée reçue sont évaluées.
- Un processus interne synchronise régulièrement la liste d'alertes pour synchroniser l'information enregistré dans la mémoire du catalogue.

### 5.3.5.3 Agent moniteur d'activités

Sentilo est une plateforme de Publish-Subscribe. La quantité de données emmagasinée dans le système est proportionnelle au déploiement de Redis et dépend directement de la RAM disponible sur le serveur Redis. En d'autres termes, les données doivent être effacées après un certain temps pour libérer la RAM. Par exemple, dans le déploiement de Barcelone, les données sont supprimées après environ une semaine. En plus de l'expiration des données, Sentilo ne fournit pas beaucoup de tableaux de bord et ces tableaux de bord ne sont pas personnalisables.

Pour combler le manque d'historisation et de tableaux de bord, sentilo utilise Elasticsearch et Kibana. Elasticsearch est une puissante base de données de recherche fulltext basée sur Java avec l'API REST. Elle est fréquemment utilisée avec ses modules, Kibana pour les tableaux de bord et Logstash pour la collecte des logs. La combinaison de Elasticsearch, Logstash et Kibana est souvent appelée la pile ELK. ELK fournit un moyen confortable de stocker et d'exploiter des informations historiques, ainsi qu'un suivi en temps quasi réel de la plate-forme.

Les événements Sentilo sont envoyés à Elasticsearch via un l’agent appelé Activity Monitor Agent. La figure suivante illustre une configuration possible de Sentilo avec la pile ELK. Logstash est facultatif et peut être utilisé pour la surveillance des journaux Sentilo (Les erreurs de connexion, les messages invalides, etc.), ainsi que la surveillance des ressources du système.

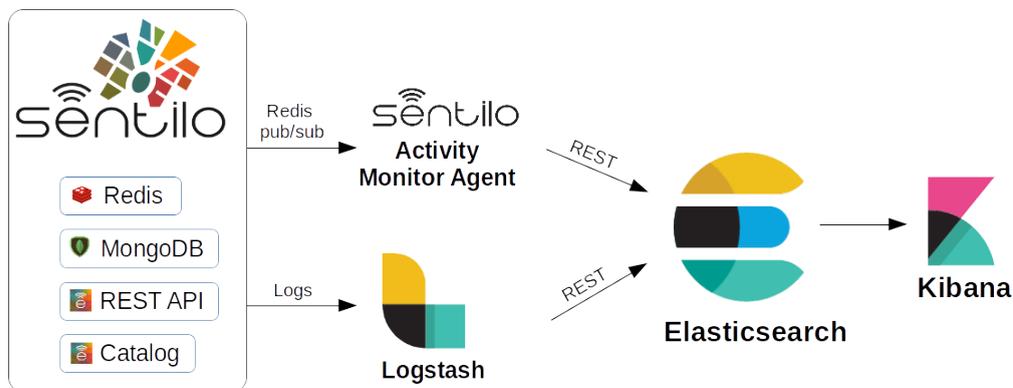


FIGURE 5.3: Architecture d’interface de sentilo avec ELK.

#### 5.3.5.4 Agent historien

Comme dit au dernier paragraphe, Sentilo ne conserve pas les données pour toujours en raison de ressources système limitées. La configuration couramment utilisée d’une instance Sentilo consiste à utiliser l’un des agents pour envoyer les données vers une base de données ou un stockage externe. Etant donné que les volumes de données peuvent être assez importants et que les données sont principalement structurées (sauf lorsque les observations sont du texte), il est pratique d’utiliser une solution évolutive pour des séries chronologiques (Time-Series) telles que OpenTSDB. OpenTSDB s’installe sur HBase et HDFS. Expose une API HTTP REST et peut être utilisé à partir de Grafana comme l’une de ses sources de données.

#### 5.3.6 Bases de données Sentilo

Sentilo nécessite une base de donnée REDIS pour emmagasiner les observations (données transmises depuis les capteurs), le temps de l’observation, l’identifiant du capteur.

### 5.3.6.1 Redis

Redis est un système de gestion de base de données clef-valeur scalable. Il permet de manipuler les chaînes de caractères, tableaux, listes, ensembles et ensembles ordonnés. Redis conserve l'intégralité des données en RAM, ce qui permet d'augmenter la rapidité d'accès aux données et éviter l'accès aux disques.

Sentilo utilise Redis comme base de données tampon.

### 5.3.6.2 MongoDB

MongoDB est un système de gestion de base de données orientés documents, répartitionnable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini de données. Il permet de manipuler des objets structurés au format BSON (JSON binaire), sans schéma prédéterminé. Des clés peuvent être ajoutées à tout moment sans reconfiguration de la base.

Sentilo utilise MongoDB pour emmagasiner les utilisateurs, les providers, les composants, les types de capteurs, d'alertes, d'alarmes entre autres.

### 5.3.6.3 Le serveur Pub/Sub

C'est une application Java divisée en deux couches[23] :

**Couche transport** Structurée suivant le modèle "Thread Pool" et implémenté avec la librairie Apache HttpCore. La figure suivante illustre le fonctionnement d'une requête dans cette couche :

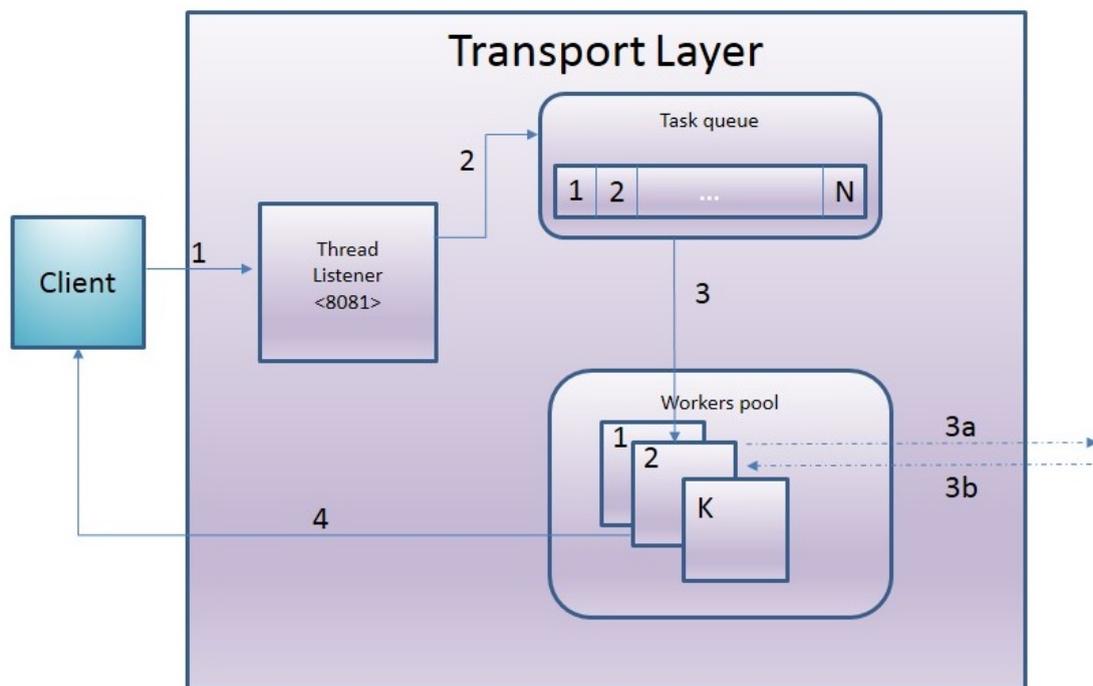


FIGURE 5.4: Fonctionnement de la couche transport.

Le client envoie une requête HTTP à l'API REST, le serveur accepte la requête et la met dans la liste des requêtes. Quand un worker est disponible, une tâche en attente lui est attribuée. Il délègue la requête à un élément de la couche service et construit la réponse HTTP à partir de l'information reçue. Il envoie ensuite la réponse au client.

**Couche service** La conception de cette couche a pour mission principale de minimiser le temps de traitement des demandes, de sorte que tout le travail principal est conservé en mémoire (Redis) qui stocke les données dans une base de données dans la RAM, mais aussi sur le disque pour assurer la durabilité de ces données.

La prochaine figure illustre le fonctionnement de la requête à travers cette couche :

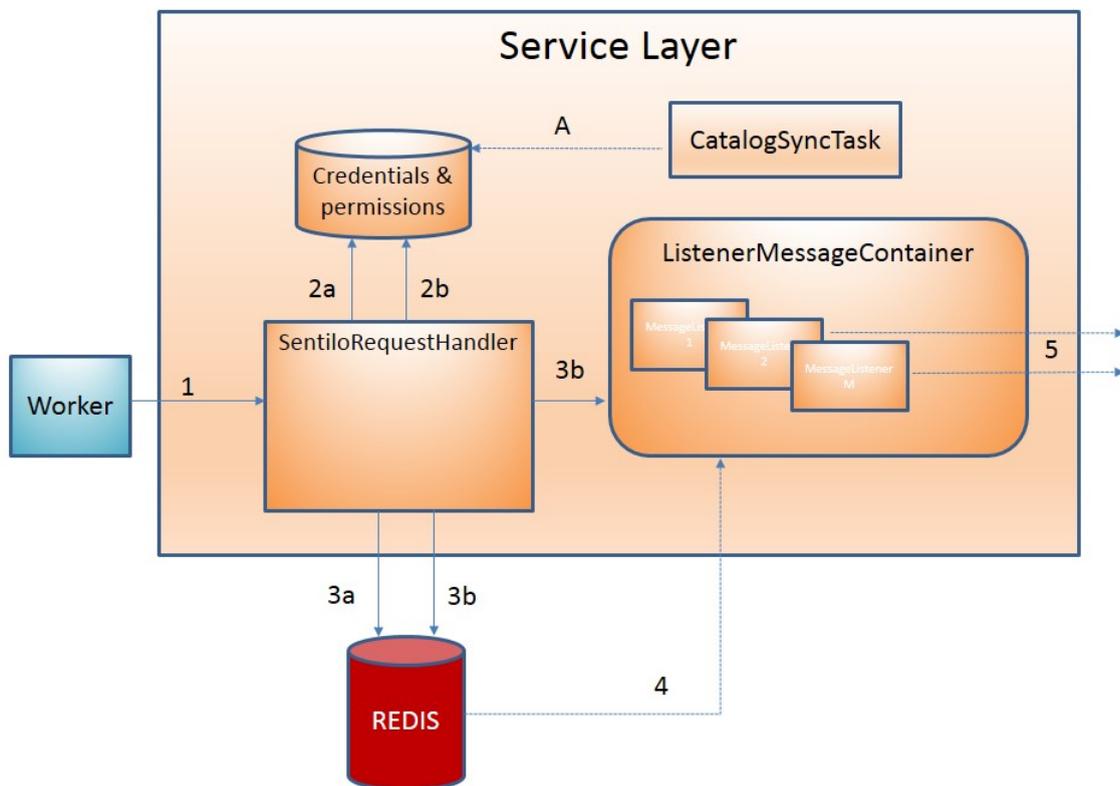


FIGURE 5.5: Fonctionnement de la couche service.

Le travailleur délègue la requête au handler associé selon le type de la requête (donnée, ordre, alarme...). Les validations suivantes sont exécutées sur chaque requête :

- (2a) Intégrité des informations d'identification : vérifie le jeton reçu envoyé dans l'en-tête en utilisant la base de données interne en mémoire contenant toutes les informations d'identification actives dans le système.
- (2b) Autorisation d'exécution de la requête : valider que l'action demandée peut être effectuée conformément à la base de données d'autorisation.
- La validation des paramètres de la requête : principalement, la structure et la typologie.

Vient par la suite l'enregistrement de la donnée sur Redis et selon le type de donnée :

- (3a) Publier la donnée à travers le mécanisme de publication.
- (3b) Enregistrer la souscription dans le ListenerMessageContainer.

Redis prend en charge l'envoi de l'information à l'événement publiée au "ListenerMessageContainer" qui lui prend en charge la gestion de la souscription à Redis comme client pour n'importe quel type d'évènement. Le container notifie l'évènement à la souscription associée en envoyant une requête via HttpCallback.

### 5.3.7 Architecture générale

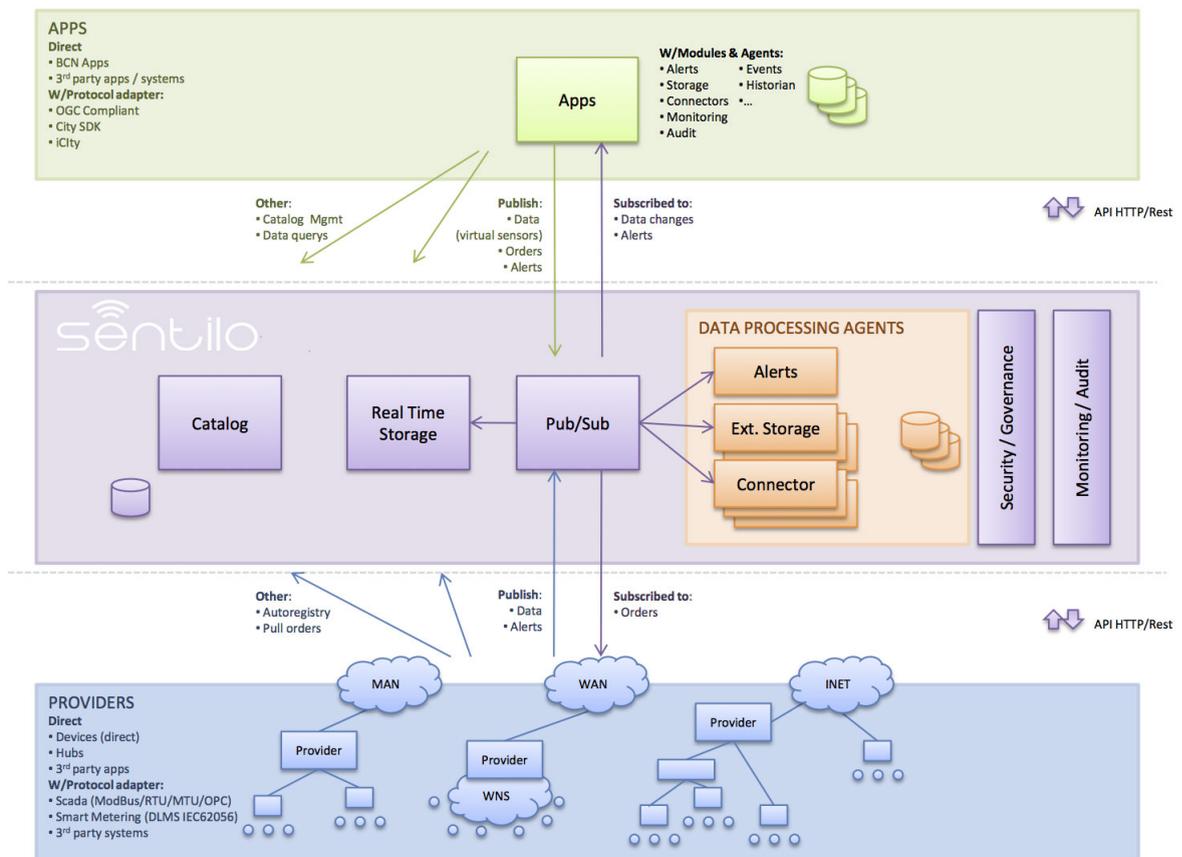


FIGURE 5.6: Architecture de la plateforme sentilo.

### 5.3.8 Principe

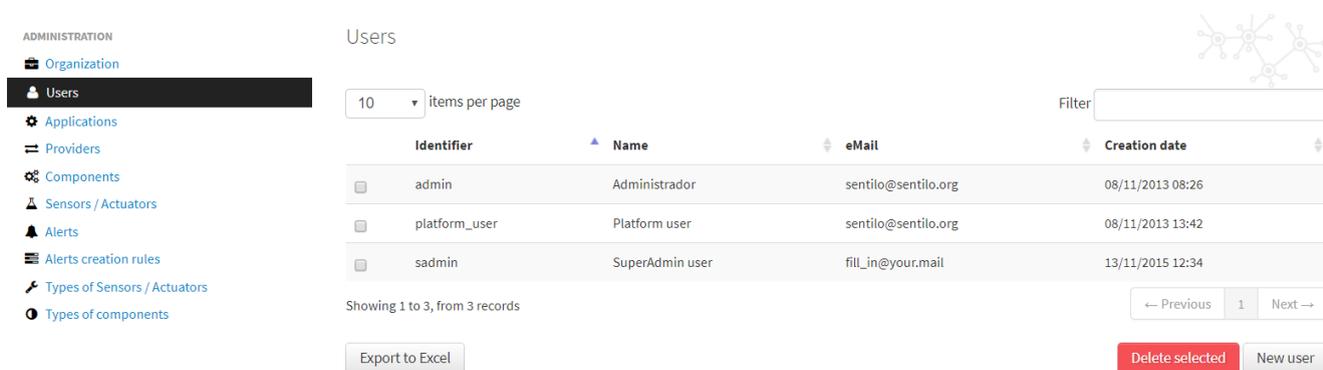
L'organisation de Sentilo se fait comme suit. Un ensemble de capteurs est connecté à un concentrateur ou une passerelle qui est représentée sur Sentilo comme "composant". La passerelle est connectée à internet, elle envoie des requêtes HTTP (PUT) à l'API Pub/Sub de Sentilo. Les données sont tout de suite enregistrées dans la base de données REDIS qui est considérée comme base de données tampon, ce qui permet l'affichage de ces données sur le catalog qui est relié à REDIS. Sentilo dispose d'agents

qui déclenchent des alertes, agents d'alarmes, elle est aussi dotée de plusieurs agents qui facilitent la connection avec diverses bases de données (MySQL, ElasticSearch, OpenTSDB...). Ceci permet aux développeurs de créer des applications de visualisation et de traitement de données personnalisés externes à sentilo.

## 5.3.9 Fonctionnalités

### 5.3.9.1 Utilisateurs

La section utilisateurs permet de créer des comptes pour les clients afin qu'ils puissent accéder à leurs données et éventuellement ajouter des composants, capteurs, alarmes, ou alertes selon les permissions que l'administrateur leur donne.



The screenshot shows the 'Users' management interface. The sidebar on the left lists various administration options, with 'Users' currently selected. The main content area displays a table of users with the following data:

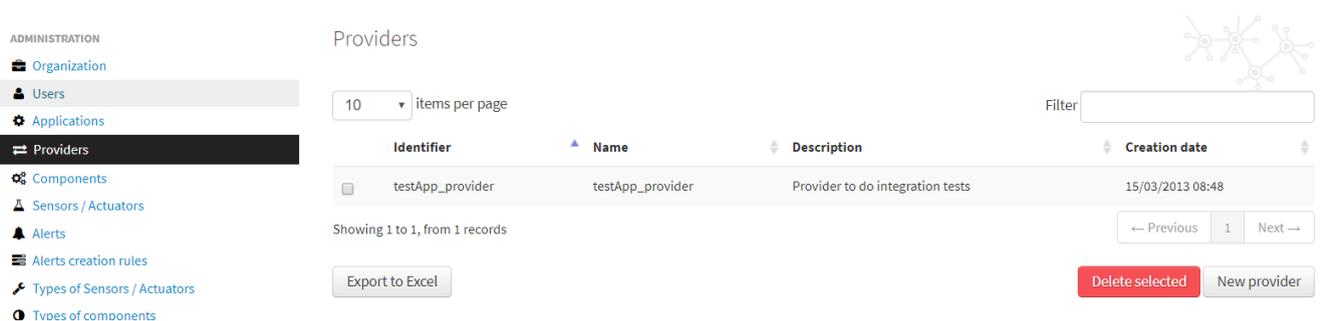
Identifier	Name	eMail	Creation date
admin	Administrador	sentilo@sentilo.org	08/11/2013 08:26
platform_user	Platform user	sentilo@sentilo.org	08/11/2013 13:42
sadmin	SuperAdmin user	fill_in@your.mail	13/11/2015 12:34

Below the table, it indicates 'Showing 1 to 3, from 3 records'. There are buttons for 'Export to Excel', 'Delete selected', and 'New user'. A navigation bar at the bottom shows 'Previous', '1', and 'Next'.

FIGURE 5.7: Interface de gestion des utilisateurs.

### 5.3.9.2 Providers

La section Providers permet de créer des Providers, et d'exporter la liste dans un fichier excel pour une éventuelle gestion locale plus tard.



The screenshot shows the 'Providers' management interface. The sidebar on the left lists various administration options, with 'Providers' currently selected. The main content area displays a table of providers with the following data:

Identifier	Name	Description	Creation date
testApp_provider	testApp_provider	Provider to do integration tests	15/03/2013 08:48

Below the table, it indicates 'Showing 1 to 1, from 1 records'. There are buttons for 'Export to Excel', 'Delete selected', and 'New provider'. A navigation bar at the bottom shows 'Previous', '1', and 'Next'.

FIGURE 5.8: Interface de gestion de composants.

### 5.3.10 Components

La section Components permet de créer des composants, voir les capteurs associés, les composants associés ainsi que d'autres détails techniques (adresse MAC, type d'alimentation...). Elle permet aussi de gérer les composants et les mettre en accès publique ou privé.

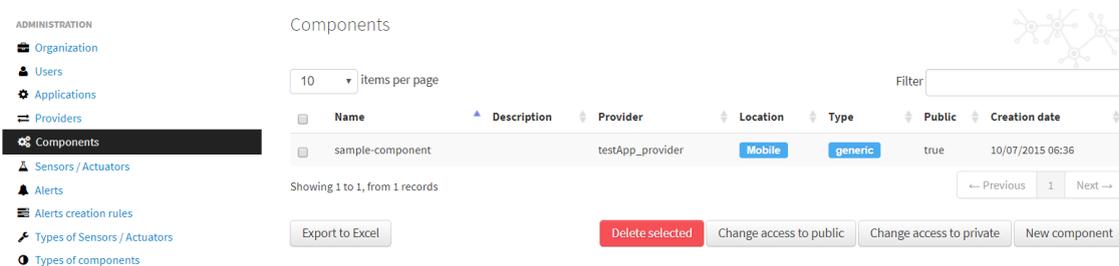


FIGURE 5.9: Interface de gestion des composants.

### 5.3.11 Capteurs/Actionneurs

La section capteurs et actionneurs permet de créer des capteurs/actionneurs, les associer à un composant, changer leur état, ainsi que voir les dernières données reçues par le capteur.

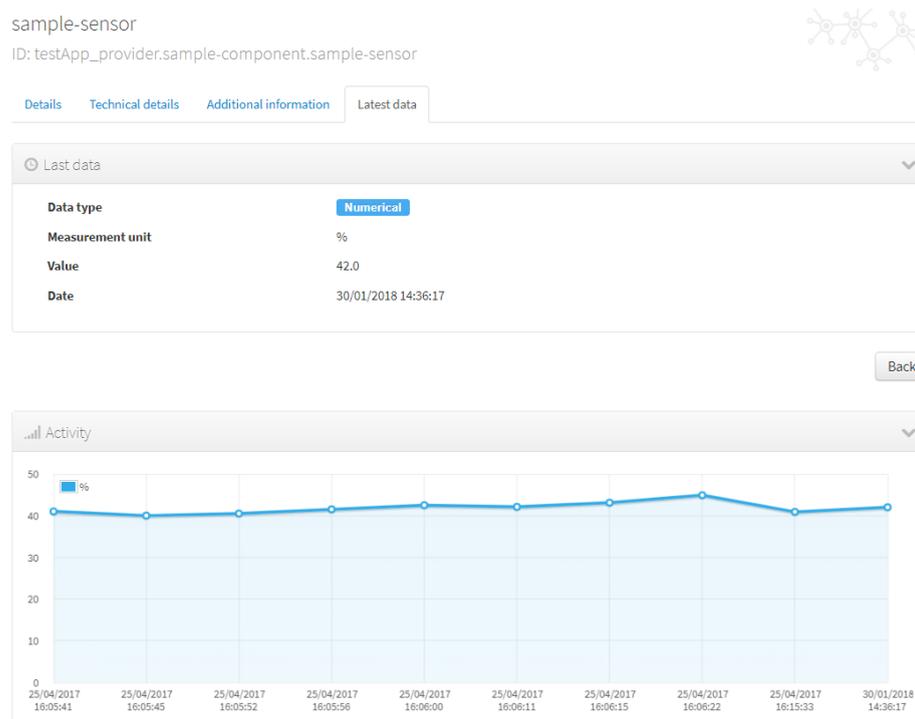


FIGURE 5.10: Interface spécifique à un capteur.

### 5.3.12 Alertes

La section Alertes permet la création et la gestion des alertes.

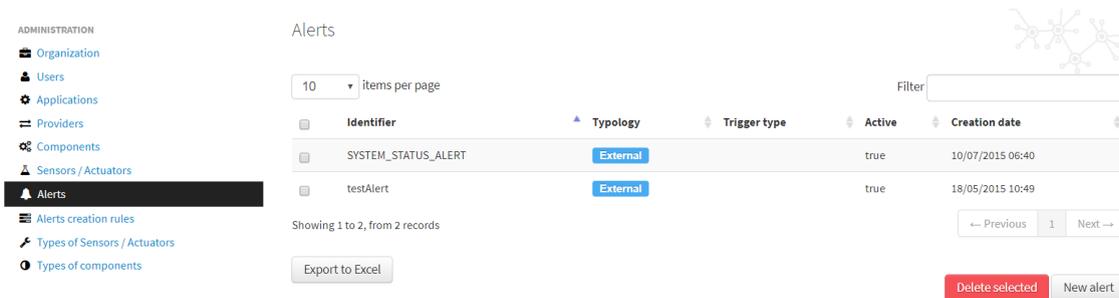


FIGURE 5.11: Interface de gestion d'alertes

### 5.3.13 Création de règle d'alertes

La section de création de règle d'alertes permet de définir des alertes avec des règles tel que "plus grand que, plus petit que ...".

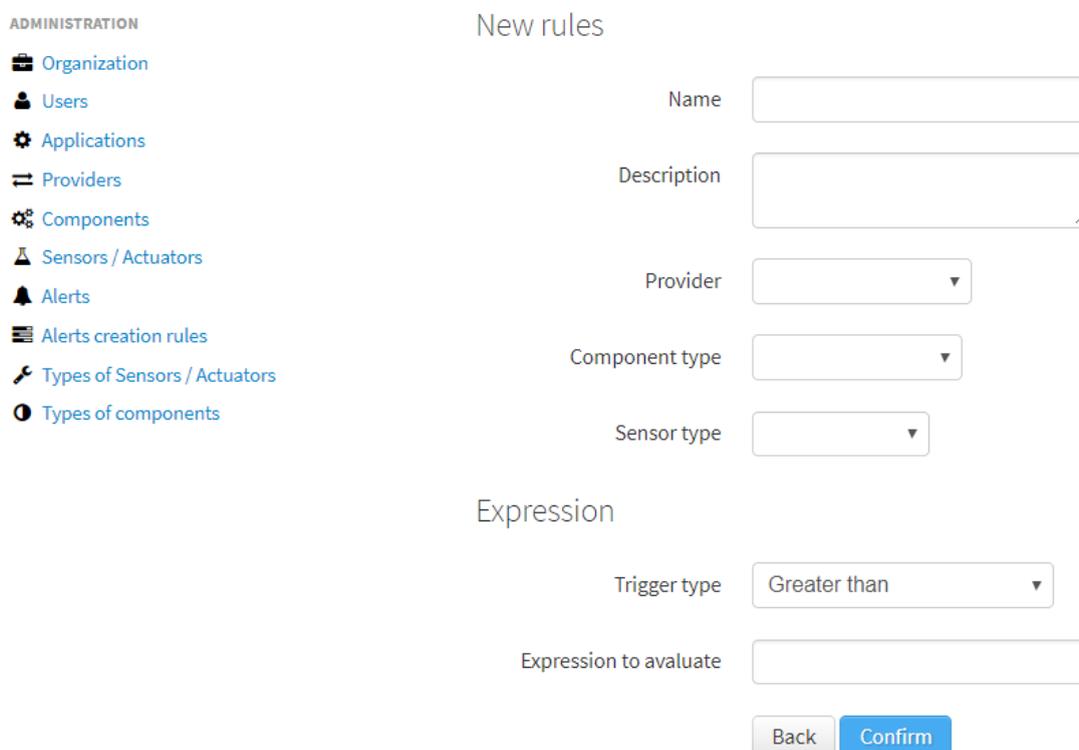


FIGURE 5.12: Interface de gestion de règles d'alertes.

### 5.3.14 Types de capteurs/Actionneurs

Dans cette section on crée le type de capteurs pour les associer aux capteurs réels.

Types of Sensors / Actuators

10 items per page Filter

Identifier	Name	Description	Creation date
anemometer	Anenometer		08/11/2013 10:27
humidity	Humidity		08/11/2013 10:27
noise	Noisemeter		08/11/2013 10:27
pluviometer	Pluviometer		08/11/2013 10:27
rain	Rain		08/11/2013 10:27
status	status	status	10/07/2015 06:44
temperature	Temperature		08/11/2013 10:27
wind	Wind		08/11/2013 10:27

Showing 1 to 8, from 8 records

Export to Excel Delete selected New typology

FIGURE 5.13: Interface de gestion de types de capteurs et d'actionneurs.

### 5.3.15 Types de composants

Dans cette section, on crée le type de composants pour les associer aux composants existants.

Component's typology

10 items per page Filter

Identifier	Name	Description	Creation date
electricity_meter	Electricity meter		08/11/2013 10:28
generic	Generic	Generic component type	08/11/2013 10:28
meteo	Meteo		08/11/2013 10:28
noise	Noise meter		08/11/2013 10:28

Showing 1 to 4, from 4 records

Export to Excel Delete selected New typology

FIGURE 5.14: Interface de gestion des types de composants.

## 5.4 GreenPlum

### 5.4.1 Définition

Greenplum est une base de donnée qui utilise le MPP (Massively parallel processing) avec une architecture faite pour faire de l'analyse de données sur un grand nombre de données. Elle est basée sur la technologie opensource PostgreSQL. Greenplum est

constituée de plusieurs instances de PostgreSQL qui travaillent ensemble pour former un seul système de gestion de base de donnée. [20]

## 5.4.2 Architecture

l'architecture de GreenPlum haut niveau est illustrée dans la figure suivante :

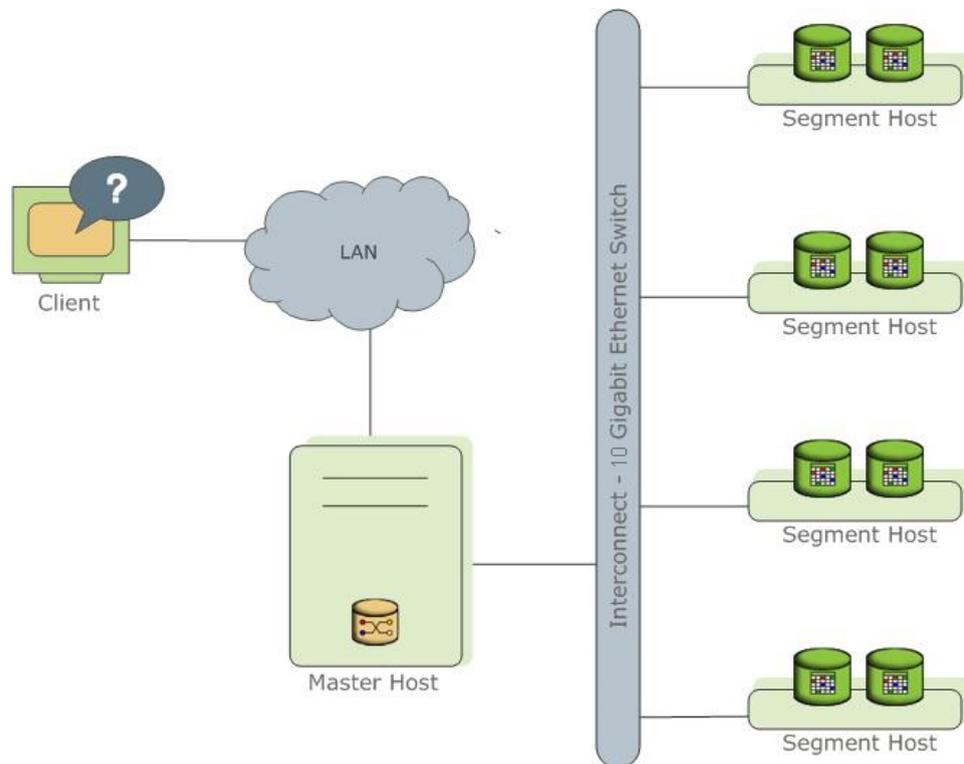


FIGURE 5.15: L'architecture globale de Greenplum[20].

Le cluster est constitué d'un Master qui lui est relié à plusieurs esclaves interconnectés entre eux. Le client est connecté via un réseau LAN au cluster via le Master. Seul le master est visible pour le client.

### 5.4.2.1 Le master

Le master est l'entrée du système de base de données Greenplum, il accepte les connexions client et les requêtes SQL, et distribue le travail aux segments du cluster.

Les clients interagissent avec la base de données Greenplum (via le maître) en tant que base de données PostgreSQL. Ils se connectent à la base de données en utilisant des

programmes clients tels que PsqL ou des interfaces de programmation d'applications (API) telles que JDBC, ODBC ou Libpq (l'API C de PostgreSQL).

Le maître est où réside le catalogue système global. Le catalogue système global est l'ensemble des tables système contenant des métadonnées sur le système de base de données Greenplum lui-même. Le maître ne contient aucune donnée utilisateur. Les données résident uniquement sur les segments. Le maître authentifie les connexions client, traite les commandes SQL entrantes, répartit les charges de travail entre les segments, coordonne les résultats renvoyés par chaque segment et présente les résultats finaux au programme client.

La base de données Greenplum utilise la journalisation à écriture immédiate (WAL, Write-Ahead Logging) pour la mise en miroir maître / veille. Dans la journalisation basée sur WAL, toutes les modifications sont écrites dans le journal avant d'être appliquées pour garantir l'intégrité des données pour toutes les opérations en cours.

#### **5.4.2.2 Les segments**

Les instances de segment de base de données Greenplum sont des bases de données PostgreSQL indépendantes qui stockent chacune une partie des données et effectuent la majorité du traitement des requêtes.

Lorsqu'un utilisateur se connecte à la base de données via le maître Greenplum et émet une requête, des processus sont créés dans chaque segment pour gérer le travail de cette requête.

Les tables définies par l'utilisateur et leurs index sont répartis entre les segments disponibles dans un système de base de données Greenplum ; chaque segment contient une partie distincte des données.

Les segments sont exécutés sur des serveurs appelés hôtes de segment. Un hôte de segment exécute généralement de deux à huit segments Greenplum, en fonction des coeurs de processeur, de la RAM, du stockage, des interfaces réseau et des charges de travail. Les hôtes de segment doivent être configurés de manière identique. La clé pour

obtenir les meilleures performances de Greenplum Database est de répartir les données et les charges de travail de manière homogène sur un grand nombre de segments de même capacité, afin que tous les segments commencent à travailler simultanément sur une tâche et achèvent leur travail simultanément.

### 5.4.2.3 L'interconnect

L'interconnect est la couche réseau de l'architecture de la base de données Greenplum. Il fait référence à la communication inter-processus entre les segments et l'infrastructure du réseau sur laquelle repose cette communication. Elle utilise une matrice de commutation Ethernet standard.

Par défaut, l'interconnect utilise le protocole de datagramme utilisateur avec contrôle de flux (UDPIFC) pour que le trafic envoie des messages sur le réseau. Le logiciel Greenplum effectue la vérification des paquets au-delà de ce qui est transmis par UDP. Cela signifie que la fiabilité est équivalente au protocole TCP (Transmission Control Protocol) et que les performances et l'évolutivité dépassent TCP. Si l'interconnect est modifiée en TCP, la base de données Greenplum a une limite d'évolutivité de 1000 instances de segment. Avec UDPIFC comme protocole par défaut, cette limite est vite supprimée.

## 5.5 Ressources matériels

### 5.5.0.1 Plateforme

Pour implémenter notre plateforme, nous avons utilisé un serveur qui possède les caractéristiques suivantes :

- Processeur : intel XEON 24 (à compléter) coeurs.
- RAM : 24 GigaOctets.
- Disque dur : 1 TeraBytes en RAID.
- 4 interfaces réseau dont la principale est connectée au réseau 172.28.0.x avec l'adresse 172.28.0.103.

1. 4 machines CentOS pour greenplum

- (a) 1 master avec les caractéristiques :
  - i. 16 Go de RAM.
  - ii. 40 Go de disque dur.
  - iii. 8 coeurs.
  - iv. une carte réseau (192.168.20.14).
- (b) 3 esclaves avec les caractéristiques :
  - i. 16 Go de RAM.
  - ii. 40 Go de disque dur.
  - iii. 8 coeurs.
  - iv. une carte réseau (192.168.20.28 192.168.20.33 et 192.168.20.34).
- 2. Une machine Ubuntu pour Sentilo avec les caractéristiques :
  - (a) 6 Go de RAM.
  - (b) 20 Go de disque dur.
  - (c) 4 coeurs.
  - (d) une carte réseau (192.168.20.18).
- 3. Une machine Ubuntu pour la plateforme de visualisation avec les caractéristiques :
  - (a) 4 Go de RAM.
  - (b) 20 Go de disque dur.
  - (c) 4 coeurs.
  - (d) une carte réseau (192.168.20.17).
- 4. Une machine FreeBSD pour la passerelle/parefeu pfsense :
  - (a) 2 Go de RAM.
  - (b) 10 Go de disque dur.
  - (c) 2 coeurs.
  - (d) deux cartes réseau (192.168.20.254 comme gateway du réseau interne et 172.28.0.5 comme machine appartenant au réseau 172.28.0.5).

### 5.5.0.2 Réseau de capteurs

Pour implémenter notre réseau de compteurs, nous avons utilisé :

- Une Raspberry PI 3 pour la Gateway.
- Deux Arduino pour la partie capteur (2 compteurs).
- Trois modules nRf24L01 pour la communication entre la Passerelle et les compteurs (capteurs).

le compteur a été monté avec une pompe pour faire circuler l'eau :



FIGURE 5.16: Montage du compteur.



FIGURE 5.17: Montage du capteur sur le compteur.

## 5.6 Architecture réseau

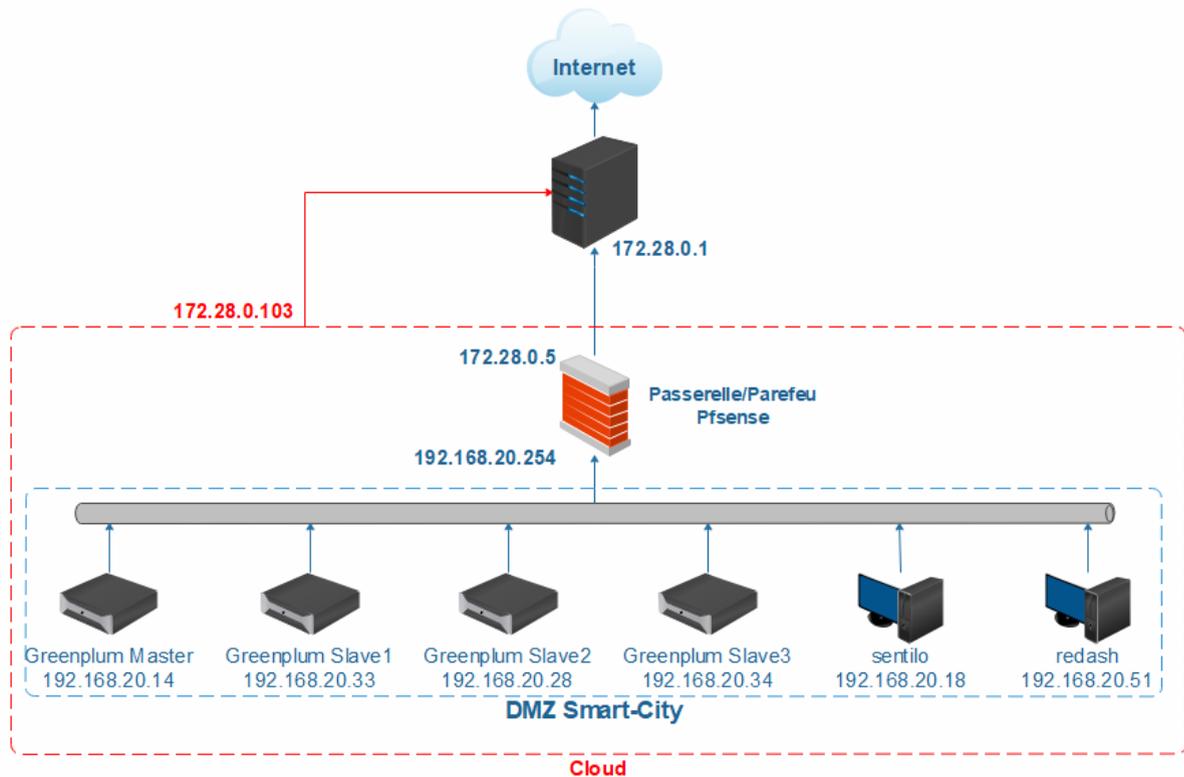


FIGURE 5.18: Architecture du réseau implémenté sur proxmox.

Nous avons créé 4 machines virtuelles dédiées à greenplum, une machine virtuelle pour la plateforme de visualisation et une autre machine virtuelle dédiée à la plateforme Sentilo, nous avons aussi créé une machine qui fait le travail d'un pare-feu pour isoler toute la plateforme de l'extérieur. Des règles ont été établies dessus pour ne permettre l'accès qu'aux ports utilisés (8081 pour l'API Rest, 8080 pour apache de Sentilo, et 5000 pour l'accès graphique à redash). Tout ce système est installé sur le système de virtualisation proxmox qui est monté sur un serveur du département d'électronique qui lui est connecté à un réseau connecté à internet.

## 5.7 Principe de fonctionnement

La donnée est lue à partir du compteur d'eau au biais du capteur infrarouge, on calcule le cumule de nombre de tours que fait l'aiguille des litres de manière continue. En

même temps, il reste en attente d'une requête venant de la Gateway à travers le module nRF24L01. Dès qu'un message arrive, le capteur répond en renvoyant le cumule de consommation enregistré. La figure suivante illustre le principe de fonctionnement du capteur :

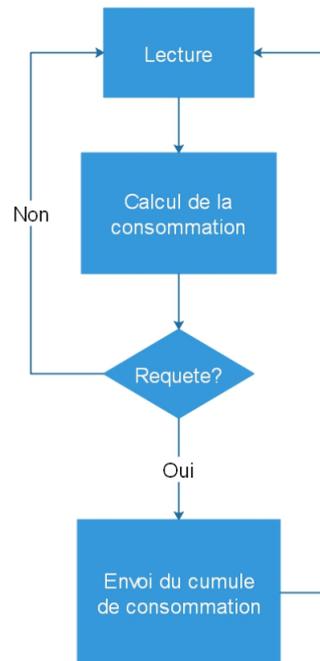


FIGURE 5.19: Diagramme de fonctionnement du capteur.

Au niveau de la passerelle, chaque 15 minutes, la raspberry envoie une requête aux différents compteurs/capteurs un par un et reste en écoute d'une réponse de leur part contenant la consommation du client. L'envoi de la requête est répété trois fois Si un compteur ne répond pas. La passerelle étant connecte internet envoie la donne vers la base de données REDIS de sentilo à travers le port 8081 dédié a l'API Rest du serveur Pub/Sub en requête HTTP PUT. La figure suivante illustre le principe de fonctionnement de la passerelle :

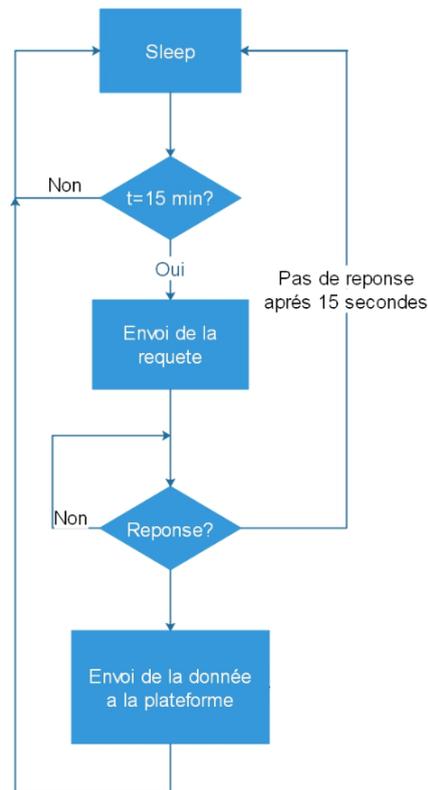


FIGURE 5.20: Diagramme de fonctionnement de la passerelle.

Arrivée au niveau de la plateforme sentilo, la donnée est ensuite envoyée par l'agent relationnel vers le master de greenplum qui s'occupe de sa distribution sur les 3 escalives qui sont interconnectés à travers la couche réseau de greenplum. La donnée est ensuite demandée par les requêtes de redash pour être affichés sur le dashboard personnalisé.

## 5.8 Résultats : Visualisation de données

Pour implémenter notre service qui est l'exposition des données de consommation, nous avons opté pour la plateforme redash comme plateforme de visualisation de données, elle est simple à utiliser et dispose de fonctionnalités qui nous intéressent. La fonctionnalité qui nous facilite la tâche est l'ajout facile de requêtes et l'affichage des données acquises. De plus, redash est interfaçable avec un grand nombre de bases de données. Dans les figures qui suivent nous montrons la procédure de création d'un graphe pour un capteur précis.

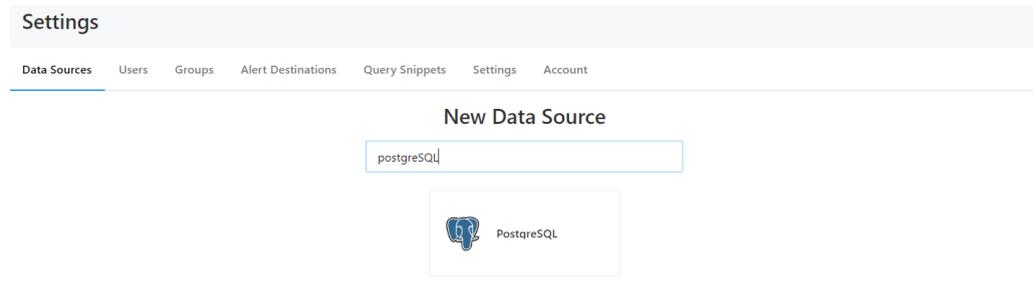


FIGURE 5.21: Interface de création de la source de donnée.

On ajoute d'abord notre source de données qui est PostgreSQL (greenplum).

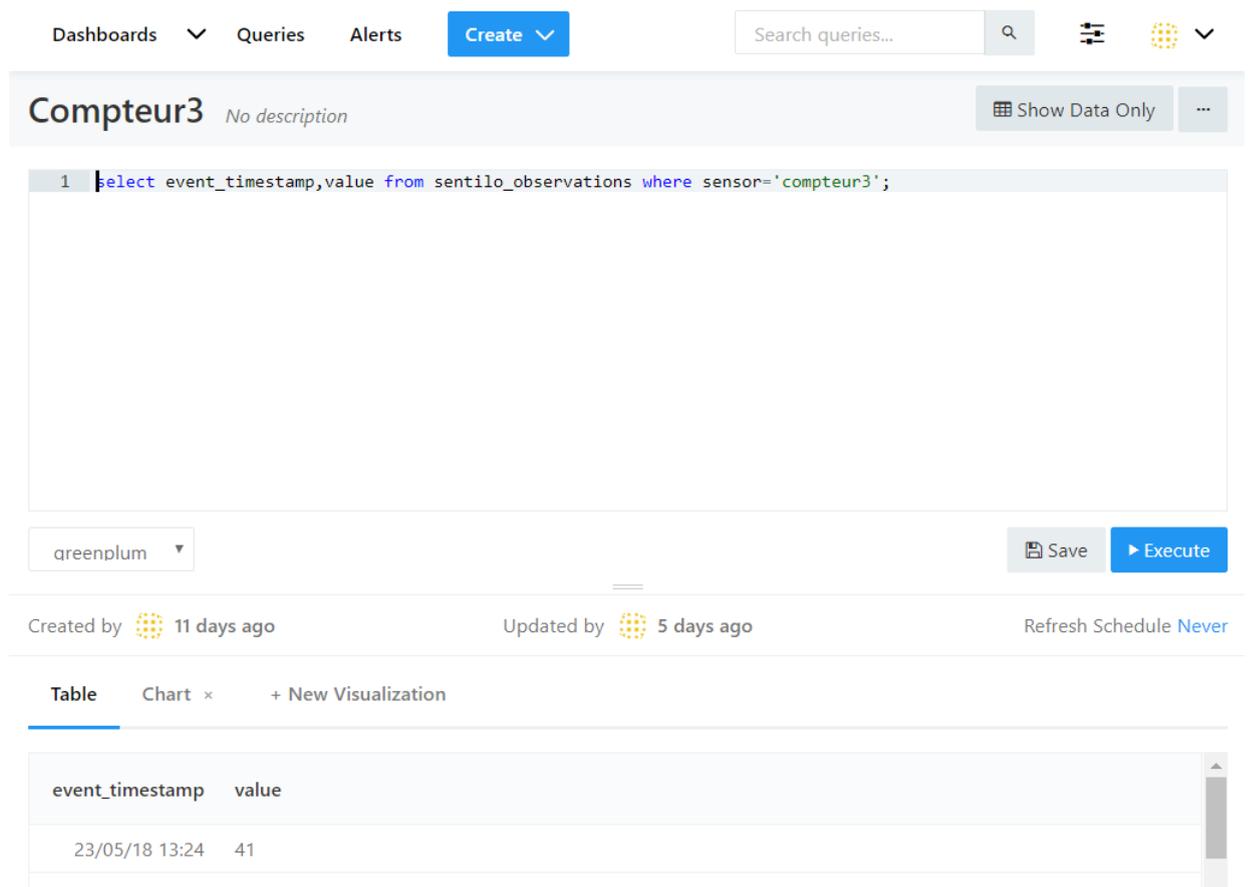


FIGURE 5.22: Interface de création de requête SQL.

On écrit la requête SQL et on attribue un nom à la requête. Ensuite on personnalise notre visualisation, on prend event\_timestamp en abscisse et value en ordonnée pour avoir le résultat suivant :

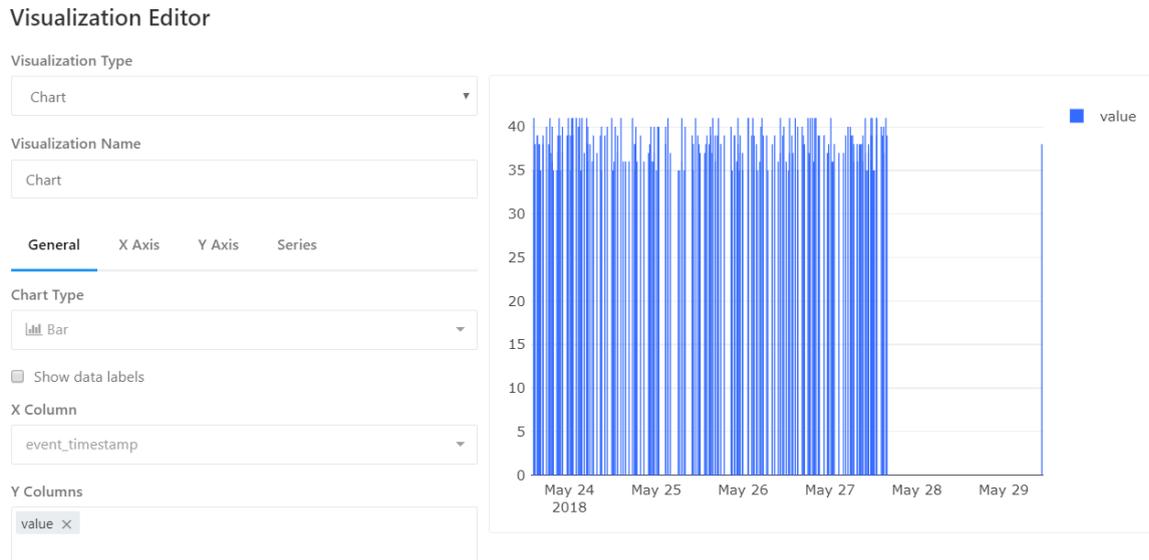


FIGURE 5.23: Visualisation des données demandées.

On crée ensuite un dashboard et on y ajoute un widget qui permet l’affichage des requêtes qu’on a créé auparavant.

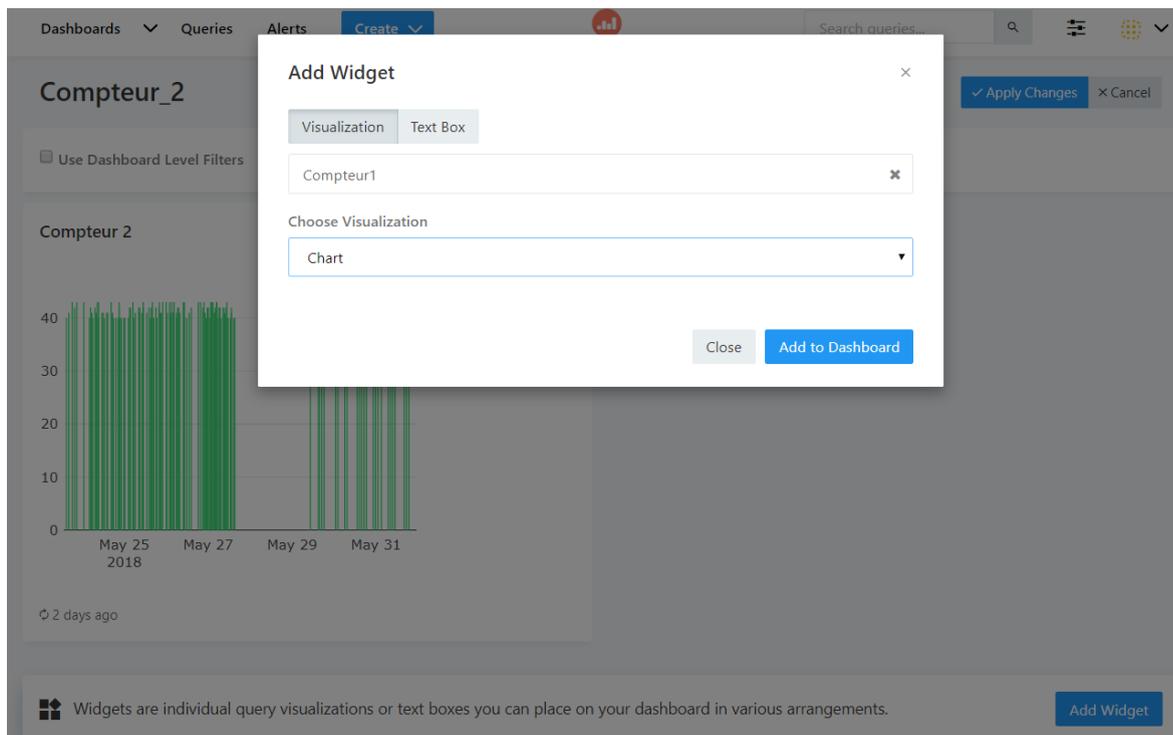


FIGURE 5.24: Ajout de Widget au dashboard.

## 5.9 Conclusion

Le système que nous avons implémenté a permis le relevé automatique de la consommation en eau et ceci en respectant la chaîne complète de l'Internet des Objets. La lecture de la consommation est réalisée en utilisant un capteur infrarouge lié à un micro-contrôleur qui fait des calculs en local de la consommation en exploitant le nombre de tours que fait l'aiguille des litres du compteur de SEAAL. Il envoie ensuite la donnée vers un concentrateur en Radio-Fréquence (2.578GHz) qui est connecté à une plateforme Smart-City en lançant une requête HTTP PUT à l'API Rest de Sentilo. La donnée est ensuite transmise vers une base de donnée Big Data de type MPP pour ensuite être exposée au client comme consommation.

# Conclusion générale

L'internet des objets a fait explosion ces dernières années, plusieurs applications ont été développés, plusieurs systèmes ont été mis en place afin d'améliorer le style de vie des humains dans le monde. Un des styles de projets les plus intéressants depuis la montée de l'IoT, la Smart-City. Ce projet de fin d'étude s'intègre dans un grand projet intitulé "Algiers Smart City" qui a pour but de rendre la ville d'Alger une ville intelligente. La partie traitée concerne le relevé automatique de consommation en eau, gaz et électricité et de centraliser toutes ces données sur une seule plateforme afin que la facturation soit mise en ligne et que le client voit sa consommation en temps réel au lieu d'avoir un agent qui passe chaque trois mois pour relever la consommation et sortir avec une facture semestrielle. Ce travail est spécifique à la consommation en eau.

Dans la première partie trois chapitres ont été présentés. Le premier est un état de l'art sur l'Internet des objets et les différents réseaux de communications utilisés pour connecter les capteurs à une seule plateforme. Le deuxième chapitre donne un petit aperçu sur l'adoption du Smart-Metering par les villes de pays développés, il présente aussi les deux manières de faire : AMR et AMI, et est terminé par les enjeux du Smart-Metering. Le troisième chapitre expose une architecture générique d'une plateforme Smart-City et un état de l'art sur les différentes technologies et solutions (Cloud, Big Data) utilisées afin de composer cette dernière.

La deuxième partie concerne l'implémentation du système entier, du capteur à l'application. Le quatrième chapitre présente le système d'acquisition. Suite à plusieurs études menées sur les types de compteurs d'eau et spécifiquement celui de la SEAAL qui est installé dans les maisons algériennes, un système à base d'un capteur optique a

été réfléchi et conçu afin d'arriver à une solution non invasive au compteur, c'est-à-dire un capteur à intégrer afin de récupérer la consommation en eau. Le protocole de communication choisi a été imposé par la réglementation Algérienne imposée par l'agence nationale de fréquence qui ne permet pas d'utiliser certaines fréquences (celles des réseaux IoT). Le cinquième chapitre expose l'architecture générale de notre système. La solution cloud pour laquelle nous avons opté a été imposée par la contrainte de temps, PROXMOX nécessite peu de temps pour l'installation et pour le déploiement. Ce chapitre met en valeur l'implémentation de toute la chaîne, il montre les fonctionnalités de la plateforme smart-city, sa connection avec la base de données Big Data MPP et de l'application d'exposition de données.

La chaîne de Smart-Metering conçue a donné de bons résultats, la consommation en eau chez les individus peut être désormais être automatique et centralisée sur une même plateforme. La SEAAL n'aura plus à envoyer un agent pour relever la consommation.

**Perspective** La partie capteur a été conçue pour les compteurs d'eau uniquement, néanmoins, elle peut s'étendre aux compteurs de gaz et ceux de l'électricité. Ce travail est susceptible d'avoir des perspectives notamment les suivantes :

1. Intégrer une batterie d'une durée de vie assez importante pour que le capteur soit indépendant d'une source de tension par secteur.
2. Opter pour la mise en place d'OpenStack qui a plus de fonctionnalités tel que la gestion des réseaux internes, le cryptage des accès entre autres.
3. L'amélioration de la plate-forme d'exposition de données en intégrant d'autres fonctionnalités tel que la possibilité de faire des demandes d'intervention, et de l'intelligence tel que la détection de fuites, les alertes en cas de hausse importante de consommation ou en cas d'anomalies, l'état de la batterie entre autres applications possibles.
4. Réfléchir à une solution d'intégration des trois cas eau, gaz et électricité sur un seul système d'acquisition.

# Bibliographie

- [1] Imed Romdhani Mohammed Riyadh Abdmeziem, Djamel Tandjaoui. Architecting the internet of things : State of the art. 2015.
- [2] Le tendanceur. Les objets connectés : histoire et définitions. [en ligne] <http://www.letendanceur.bzh/les-objets-connectes-histoire-et-definitions-de-cette-revolution-technologique/>.
- [3] Département communication. Alfa Safety. Architecture d'une solution iot [en ligne] <https://actu.alfa-safety.fr/iot-internet-des-objets-connectes/la-bonne-architecture-pour-votre-application-iot/>.
- [4] France Centre National RFID. Etat de l'art lpwa.
- [5] XONA Partners. Mobile and wide-area iot : Lpwa and lte connectivity. 2016.
- [6] Communication department. LoRa Alliance. About lorawan. [en ligne] <https://lora-alliance.org/about-lorawan>.
- [7] Communication department. M-Bus. M-bus, an overview [en ligne] <http://www.m-bus.com/info/mbuse.php>.
- [8] Communication department. M-Bus.
- [9] Pacific gas, Electric-Gas, and power company for california. Smart metering au usa [en ligne] [https://web.archive.org/web/\\*/http://www.pge.com/news/news\\_releases/q32006/060720a.htm](https://web.archive.org/web/*/http://www.pge.com/news/news_releases/q32006/060720a.htm).
- [10] France Le moniteur. Smart metering en allemagne [en ligne] <https://www.lemoniteur.fr/article/l-allemagne-renonce-a-la-generalisation-du-compteur-intelligent.1920289>.

- 
- [11] département communication. Veolia. Smart metering en france [en ligne]  
<https://www.veolia.com/fr/newsroom/communiqués-de-presse>.
- [12] Radio Canada. Smart metering en quebec [en ligne] <https://ici.radio-canada.ca/nouvelle/796240/compteurs-intelligents-hydro-quebec-deux-options>.
- [13] Richard Farley. Amr ami architecture [en ligne]  
<http://highscalability.com/blog/2015/9/7/want-iot-heres-how-a-major-us-utility-collects-power-data-fr.html>.
- [14] Ph.D Lon W. House. Presentation to california water association [en ligne].  
<http://www.waterandenergyconsulting.com/amrami.pdf>.
- [15] Département communication. Le Big Data. Définition du cloud computing [en ligne]  
<https://www.lebigdata.fr/definition-cloud-computing>.
- [16] département communication Amazon. Types de services cloud [en ligne]  
<https://aws.amazon.com/fr/types-of-cloud-computing/>.
- [17] département communication. OpenStack. Openstack : Architecture d'openstack [en ligne] <https://docs.openstack.org/install-guide/get-started-conceptual-architecture.html>.
- [18] Département communication OpenStack. Services d'openstack [en ligne]  
<https://www.aneu.eu/2018/02/06/introduction-a-lopenstack/>.
- [19] Université de tlemcen. Implantation du modele mapreduce dans l'environnement distribue hadoop distribution cloudera.
- [20] Documentation Pivotal. Architecture overview mpp greenplum [en ligne]  
[https://gpdb.docs.pivotal.io/520/admin\\_guide/intro/arch\\_overview.html](https://gpdb.docs.pivotal.io/520/admin_guide/intro/arch_overview.html).
- [21] Sensus. Datasheet du compteur 420.
- [22] Sparkfun. Datasheet du capteur qre1113 [en ligne]  
<https://cdn.sparkfun.com/datasheets/sensors/proximity/qre1113.pdf>.
- [23] Sentilo. Sentilo's architecture [en ligne]  
<http://www.sentilo.io/xwiki/bin/view/sentilo.community.documentation/architecture>.

[24] Sentilo. Sentilo agents [en ligne] <http://www.sentilo.io/xwiki/bin/view/sentilo.community.documentation/extensions>.