

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière : Électronique
Spécialité : Électronique des Systèmes Embarqués

Présenté par

BELHOUAS Fouad Mourad

&

CHIKHI Amara Yassine

Plateforme de gestion des environnements IoT

Proposé par : Mr. KABIR Yacine

Année Universitaire 2020-2021

Remerciements

Au premier lieu nous remercions d'abord, *DIEU* de nous avoir donné assez de force, de volonté et de patience, malgré les dures périodes et les épreuves que nous avons traversé jusqu'à maintenant, afin d'accomplir ce modeste travail.

Nous tenons à exprimer toute notre reconnaissance à notre promoteur Mr. KABIR Yacine, de nous avoir orienté et conseillé et d'avoir contribué à alimenter notre réflexion au cours de ce travail.

Nous remercions l'ensemble de l'équipe pédagogique de leur disponibilité et de leur patience à notre égard, spécialement notre chef d'option Mme D. NACEUR, qui nous a été d'un grand soutien afin de mener à bien notre deuxième cycle de nos études universitaires.

Nos sincères remerciements à nos chers amis, qui nous ont apporté leur soutien moral et qui avaient répondu par leurs présences aux moments difficiles.

A toutes ces personnes, nous présentons la reconnaissance, le respect et notre gratitude éternelle.

*Ce travail est dédié à notre raison d'être dans ce monde,
et la source de notre persévérance, nos chers parents
que DIEU les protège; qui n'ont jamais cessé de croire en
nous pendant toutes ses années, que DIEU soit témoin,
nous vous serons toujours reconnaissant et présent
pour vous combler et vous apporter la fierté de nous
avoir dans votre vie.*

ملخص:

يمكن لشبكة إنترنت الأشياء أن تجلب فوائد وحلولا مختلفة لمختلف قطاعات المجتمع ، كما أنها تمثل البنية التحتية التي تربط الأجسام بشبكة الإنترنت ، وذلك باستخدام بروتوكولات الاتصال المختلفة. باستخدام هذا الجانب من الاتصال الذي يسمح أيضا للبشر بالتفاعل مع هذه الأجسام عن طريق الإنترنت ، ولتحسين مفهوم الأجسام المترابطة ، نعتزم إدماج مفهوم البيئة ، بحيث لا نتحدث بعد الآن عن الأجسام المترابطة بل عن البيئات المترابطة ، عن طريق إنشاء منصة لإدارة البيئات، بهدف توفير تحكم دقيق عن بعد، أكثر كفاءة وأكثر تنظيما. ونأمل في نهاية المطاف أن نعطي دفعة لعهد جديد من التحكم عن بعد في الوقت الحقيقي في الجزائر.

كلمات المفاتيح: انترنت الأشياء، بروتوكولات الاتصال.

Résumé:

l'**Internet des Objets** peut apporter divers avantages et solutions aux différents secteurs de la société, car elle représente l'infrastructure qui relie les objets avec un réseau internet, et ceci en ayant recours aux différents protocoles de communications. En se servant de cet aspect de communication qui permet aussi aux humains d'interagir avec ces objets via internet, et afin d'améliorer le concept des objets connectés, nous avons l'intention d'intégrer la notion d'environnement, pour que nous ne parlons plus des objets connectés mais plutôt des environnements connectés, en créant une plateforme de gestion des environnements, dans le but d'offrir un contrôle à distance précis, plus performant et plus organisé. Nous espérons donner un coup de pouce à une ère nouvelle de contrôle à distance en temps réel en Algérie.

Mots clés : Internet des objets, protocoles de communications.

Abstract :

The **Internet of Things** can bring different benefits and solutions to different sectors of society, as it represents the infrastructure that connect objects with an internet network, and this by using different communication protocols. Using this aspect of communication that also allows humans to interact with these objects via the internet, and in order to improve the concept of connected objects, we intend to integrate the notion of environment, so that we no longer talk about connected objects but rather connected environments, by creating an environment management platform, with the aim of offering precise, more efficient and more organized remote control. In the end we hope to give a boost to a new era of remote control in real time in Algeria.

Keywords: Internet of things, communication protocols.

Liste des abréviations

6LowPAN : *IPv6 Low power Wireless Personal Area Networks.*

A **Act**: *Actionneurs.*

ANT+: *Advanced and Adaptive Network Technology.*

API: *Application Program Interface.*

ASGI: *Asynchronous Server Gateway Interface.*

B **BD**: *Base de Données.*

BLE: *Bluetooth Low Energy.*

C **Cap**: *Capteurs.*

CISCO: *Computer Information System Company.*

CoAP: *Constrained Application Protocol.*

CSS: *Cascading Style Sheets.*

D

E **EPC**: *Electronic Product Code.*

F

G **GPRS**: *Global Packet Radio Service.*

GPS: *Global Positioning System.*

H **HTML**: *Hyper Text Markup Language.*

HTTP: *Hyper Text Transfer Protocol.*

I **IA**: *Intelligence Artificielle.*

IBM: *International Business Machines.*

ID: *Identification/Identity/Identifier.*

IDE: *Integrated Development Environment.*

IEEE: *Institute of Electrical and Electronics Engineers.*

IoT: *Internet of Things.*

IPv4: *Internet Protocol version 4.*

IPv6: *Internet Protocol version 6.*

ITU: *International Telecommunication Union.*

J

K **kbps:** *Kilo-byte-par-seconde.*

L **LoRa:** *Long Range.*

LPWAN: *Low Power Wide Area Network.*

LTE: *Long Term Evolution.*

M **MHz:** *Mega-Hertz.*

MIT: *Massachusetts Institute of Technology.*

ML: *Machine Learning.*

Mo: *Mega-octets.*

MQTT: *Message Queuing Telemetry Transport.*

MVC: *Models, Views, Control.*

MVT: *Models, Views, Templates.*

M2M: *Machine-to-Machine.*

N **NFC:** *Near Filed Communication.*

O

P **PDA:** *Personal Digital Assistant.*

Pub: *Publish.*

Q

R **RAM:** *Random Access Memory.*

RFID: *Radio Frequency Identification.*

S **Sub:** *Subscribe.*

SDK: *software development kit.*

T **TCP/IP:** *Transmission Control Protocol/Internet Protocole.*

U **UDP:** *User Datagram Protocol.*

URL: *Uniform Resource Locator.*

UUID: *Universally Unique Identifiers.*

UWB: *Ultra-wideband.*

V **VoIP:** *Voice Over Internet Protocol.*

W **WiFi:** *Wireless Fidelity.*

WLAN: *Wireless Local Area Network.*

WSGI: *Web Server Gateway Interface*

WSN: *Wireless Sensor Network.*

X **XML:** *eXtensible Markup Language.*

XMPP: *Extensible Messaging and Presence Protocol.*

Table des matières

Introduction générale	1
CHAPITRE 1: Notions sur l'Internet des Objets	3
1.1 Histoire de l'internet des objets	4
1.2 Définition de l'Internet des Objets	5
1.2.1 Conceptuellement	5
1.2.2 Techniquement	5
1.2.3 Des propositions	5
1.3 Statistiques du marché mondial d'IdO	6
1.4 L'objet connecté	7
1.4.1 Définition	7
1.4.2 Composants des Objets Connectés	7
1.4.3 Caractéristiques communes à tous les objets connectés dans un réseau d'IdO.....	8
1.5 Caractéristiques fondamentales de l'Internet des Objets	9
1.5.1 L'inter-connectivité	9
1.5.2 Les services reliés aux objets	10
1.5.3 L'hétérogénéité	10
1.5.4 Les changements dynamiques	10

1.5.5 Une très grande échelle.....	10
1.6 Les systèmes d'IdO	11
1.6.1 Les fonctionnalités principales d'une architecture IdO.....	11
1.6.2 Les architecture d'IdO	12
1.7 Les technologies de l'IdO.....	15
1.7.1 Radio Frequency Identification (RFID).....	15
1.7.2 Code électronique des produits (EPC).....	16
1.7.3 Wireless Fidelity (Wi-Fi).....	16
1.7.4 Bluetooth.....	16
1.7.5 Zig-Bee.....	16
1.7.6 Le Z-Wave.....	17
1.7.7 Near Filed Communication (NFC).....	17
1.7.8 Wireless Sensor Networks (WSN)	17
1.8 M2M (machine to machine)	18
1.9 Les Protocoles de communication applicative dans l'IdO	18
1.9.1 Protocoles de messagerie	19
1.9.2 Protocoles de transfert web	21
1.9.3 Protocole réseau (Websocket)	22
1.10 Applications de L'IdO.....	22
1.10.1 Le foyer de l'Internet des objets	22

1.10.2 Le contrôle du trafic Internet des objets	23
1.10.3 La santé de l'Internet des objets	24
1.11.4 Le contrôle industriel de l'Internet des objets	24
1. 11 Les enjeux clés du déploiement de L'internet des Objets	25
1. 12 Avantages de l'utilisation des objets connectés	26
1. 13 Les principaux risques liés à l'utilisation des objets connectés	26
Conclusion	27
CHAPITRE 2: la plateforme de gestion	28
2.1 Plateformes IoT	29
2.1.1 Composants d'une plateforme IoT	29
2.1.2 Les niveaux des plateformes	30
2.2 La plateforme générique IoT	31
2.2.1 Présentation	31
2.2.2 Les différents outils utilisés	31
2.2.3 La conception de la plateforme	34
2.2.4 L'environnement de développement de la plateforme	35
2.2.5 La structure de la plateforme IoT	37
2.2.6 Le transfert des données au niveau de la plateforme	41
2.2.7 La Base de Données	42
2.2.8 Points faibles dans l'architecture de la plateforme	44

2.3 Problèmes rencontrés	46
2.3.1 La qualité de code	47
2.3.2 Incompatibilités des modules et packages	48
2.4 Plateforme de gestion des environnements IoT	48
2.4.1 Présentation	48
2.4.2 Conception	49
2.4.3 Normalisation de la base de données.....	50
2.4.4 L'architecture proposée	51
2.4.5 Bootstrap	52
Conclusion	53
CHAPITRE 3 : Réalisation de la plateforme de gestion des environnements IoT	55
3.1 Les interface de la plateforme	56
3.1.1 Page d'accueil	56
3.1.2 Page d'inscription ou de création de compte	56
3.1.3 Page de connexion	57
3.1.4 Page de gestion d'environnement	58
3.1.5 Page de gestion de locaux	58
3.1.6 Page de gestion de Cap/Act	59
3.2 Domaines d'intégrations	60
Conclusion	60

Conclusion générale.....	61
Bibliographie.....	63

Listes des Figures

Figure 1.1 : Nombre d'appareils connectés dans le monde 2015-2025	6
Figure 1.2 : L'étendue des objets connectés	7
Figure 1.3 : Les composants des objets connectés	8
Figure 1.4 : Les fonctionnalités de base d'un système d'IdO	11
Figure 1.5 : les couches de L'architecture proposée par l'ITU-T	12
Figure 1.6 : Les différentes couches de l'IdO selon CISCO	13
Figure 1.7 : La différence entre le M2M et l'IoT	18
Figure 1.8 : Schéma qui explique le fonctionnement du protocole MQTT	19
Figure 1.9 : Schéma qui montre le fonctionnement du protocole XMPP.....	20
Figure 1.10 : Schéma qui montre le fonctionnement du protocole CoAP	21
Figure 1.11 : Schéma qui montre le fonctionnement du protocole WebSocket.....	22
Figure 1.12 : Maison connecté	23
Figure 1.13 : La santé avec l'internet des Objets.....	24
Figure 2.1 : Logo Django Python	32
Figure 2.2 : Architecture MVT	34
Figure 2.3 : Le schéma conceptuelle de la plateforme	35

Figure 2.4 : Création d'un projet Pycharm	36
Figure 2.5 : les applications de la plateforme	37
Figure 2.6 : La page de l'espace personnel	39
Figure 2.7 : la page de création de Cap/Act	40
Figure 2.8 : L'erreur lors du lancement avec la commande runserver	46
Figure 2.9 : l'erreur de sécurité	47
Figure 2.10 : Schéma conceptuelle qui montre les différents étages de l'architecture de la plateforme	49
Figure 2.11 : L'architecture des tables de la BD	52
Figure 2.12 : Logo du Framework Bootstrap	53
Figure 3.1 : La page d'accueil de plateforme	56
Figure 3.2 : Interface d'inscription	57
Figure 3.3 : l'interface de connexion	57
Figure 3.4 : interface de gestion des environnements	58
Figure 3.5 : Interface de gestion de locaux	59
Figure 3.6 : Interface de gestion des Cap/Act	59

Liste des Tableaux

Tableau 1.1 : Les principaux risques liés à l'utilisation des objets connectés.....	26
Table 2.1 : La table de la BD respective à l'identification du personnel.....	43
Table 2.2 : Table de la BD des applications IoT.....	43
Table 2.3 : Table de la BD des Caps/Acts.....	44
Table 2.4 : Table de la BD du client MQTT.....	44

Introduction générale

Le domaine de l'internet des objets dit aussi en anglais « *Internet of Things* »(**IoT**), a envahi le quotidien de l'humain, qui fait référence à une vision technologique large permettant un concept de vie complètement nouveau appelé « vie intelligente ». Ce domaine ne cesse de s'étendre et de s'intégrer dans d'autres domaines. De nos jours chaque personne dans le monde possède au moins un objet connecté, et des statistiques françaises révèlent qu'il y aurait en ce moment entre 30 et 80 milliards d'objets connectés dans le monde (télévisions, montres, électroménagers, Smartphones...).

Le surnombre d'objets connectés qui existent sur marché et qui partagent des informations, ont permis en quelque sorte de récolter une somme énorme de données, ce qui a donné naissance à un outil très utile pour l'analyse de ses données dite technologie des Big Data, dans le but de donner plus d'autonomie aux systèmes à base d'Internet des Objets **IdO**, mais dans cette étude ne nous intéresserons pas à cet aspect de l'**IdO**.

L'internet des objets vise à contrôler et à récolter des données, à partir des objets électroniques connectés à un réseau informatique, grâce aux différents protocoles de connexion. Pour enfin relier les objets physiques à une infrastructure informatique pilotables, assuré le plus souvent par un ordinateur, Smartphones, tablette...

Par exemple sur le plan individuel ou des routines humaines, les objets connectés peuvent nous offrir un plus grand confort dans notre vie quotidienne, en accordant une gestion facile de plusieurs taches d'une manière discipliné, et c'est en répondant à ses besoins des utilisateurs qui recherchent le bon service au bon moment et au bon endroit, que ce domaine sera exploité à toute son ampleur dans l'avenir.

Pour améliorer la qualité des services qu'elle peut nous offrir l'internet des objets, nous allons nous concentrer à perfectionner l'utilisation des plateformes **IdO**, et de rendre sa gestion plus organisée et performante, en mettant en œuvre une plateforme web de gestion des environnements **IdO**, en se servant du Framework basé sur Python **Django**.

Ce mémoire est organisé en trois chapitres essentiels, le premier chapitre est entièrement dédié à l'état de l'art concernant l'internet des objets, présentation, définitions, principe de fonctionnement, avantages et inconvénients d'utilisation, puis nous allons conclure avec quelques applications accréditées par des entreprises de haut rang et des personnes très importantes.

Le deuxième chapitre va présenter brièvement, le travail de M.Daouadji et A.El aichi [38] réalisé au laboratoire de traitement du signal et de l'image LATSI du département d'électronique de l'université Blida1 l'année passée. Ce dernier constitue un point de départ à la conception du système proposé dans ce mémoire.

Au troisième chapitre, nous exposons le travail effectué, les résultats obtenus et les domaines et les champs d'application possibles.

CHAPITRE 1:

Notion sur l'Internet des objets.

Notion sur l'Internet des objets

L'image auquel l'internet des objets fait allusion, est un réseau physique constitué d'objets (capteurs ou actionneurs) reliés tous à internet, afin de donner accès aux humains de n'importe quel endroit. Ce chapitre expose les notions de bases concernant l'internet des objets.

1.1. Histoire de l'internet des objets :

L'idée d'avoir des appareils connectés existe depuis 1970. Le premier objet connecté était un distributeur ou une machine à Coca, à l'Université de Carnegie Mellon, au début des années 1980, mais ce n'était pas vraiment le concept des objets connectés dans le sens où nous l'entendons aujourd'hui.

A la suite de la première implémentation du protocole **TCP/IP** par le fondateur de **FTPSoftware** John Romkey en 1986, Dan Lynch l'a défié de connecter un appareil et de le présenter au salon de l'année suivante. Romkey a accepté le défi ; il réussit à connecter un grille-pain, qu'il pouvait allumer et éteindre via Internet, et qu'il soit ensuite considéré comme le premier appareil de l'histoire de l'**IdO**.

Mais le terme d'internet des objets, a été pour la première fois mentionné par le co-fondateur de l'Auto-ID Center Kevin Ashton, au **MIT** lors d'une présentation faite par Procter & Gamble, en 1999. Dans l'intérêt d'attirer l'attention des directeurs de P&G sur les puces RFID (identification par radiofréquence).

En 2000, LG annonce : le premier réfrigérateur connecté à Internet, malheureusement, c'était un échec commercial. En 2009 Google commence à tester des véhicules autonomes. La première voiture à rouler en autonomie est une Toyota Prius ; elle était équipée de capteurs capables de détecter les piétons, les cyclistes, les travaux routiers et d'autres objets remarquables.

L'internet des objets avait pour but au début de connecter le monde des humains avec le monde de gestion des objets à distance à partir de l'internet. De nos jours, les ambitions des développeurs de ce domaine ont pour but de faire dialoguer les objets entre eux et avec les individus. [1]

1.2. Définition de l'Internet des Objets :

Plusieurs notions et contextes entrent en jeu lorsqu'il s'agit de l'**IdO**, il faut donc trouver une définition à la fois stable et standard. Certains géants de l'industrie, des institutes et des chercheurs la définissent en insistant sur des aspects techniques et d'autres sur des aspects conceptuels :

1.2.1 Conceptuellement :

L'**IdO** est définie comme étant des « objets ayant des identités et des personnalités virtuelles, opérant dans des espaces intelligents et utilisant des interfaces intelligentes pour se connecter et communiquer au sein de contextes d'usages variés ».

1.2.2 Techniquement :

L'**IdO** est une extension du système de nommage internet où il est possible d'identifier de manière unifiée des éléments d'informations numériques et des éléments physiques.

1.2.3 Des propositions :

Un réseau de réseaux (ex: réseau d'objet de réseau internet) qui permet, via des systèmes d'identification électronique normalisés et unifiés, et des dispositifs mobiles sans fils, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi de pouvoir récupérer, stocker, transférer et traiter, sans discontinuité entre les mondes physiques et virtuels, les données s'y rattachant. [2]

L'Union Internationale des Télécommunications (UIT) définit ainsi en 2012 l'internet des objets comme une « infrastructure mondiale pour la société de l'information ». [3]

Ou bien selon Weil et Souissi [4] « l'Internet des Objets se définit comme l'extension de l'Internet actuel à tous les objets pouvant communiquer, de manière direct ou indirecte, avec des équipements électroniques eux-mêmes connectés à Internet ».

1.3. Statistiques du marché mondial d'IdO :

D'après **IBM**, une maison familiale typique aura environ 500 appareils connectés à Internet d'ici 2022. À échelle mondiale, des estimations donnent un chiffre de 38,5 milliards d'appareils connectés en 2020. D'après une analyse plus récente menée par Juniper Research, le nombre de capteurs et de dispositifs IdO dépasserait les **50 milliards d'ici 2022**. Si nous comparons ces projections aux prévisions qui annoncent une population mondiale de 8 milliards d'habitants, il devient plus facile de percevoir l'immensité du réseau IdO. [5]

Selon les **prévisions mondiales pour 2026**, et le nouveau rapport d'étude du " **Marché de l'IoT industriel** ", la taille du marché devrait passer de 76,7 milliards USD en 2021 à 106,1 milliards USD d'ici 2026, avec un TCAC (Taux de croissance annuel composé) de 6,7% au cours de la période de prévision. [6]

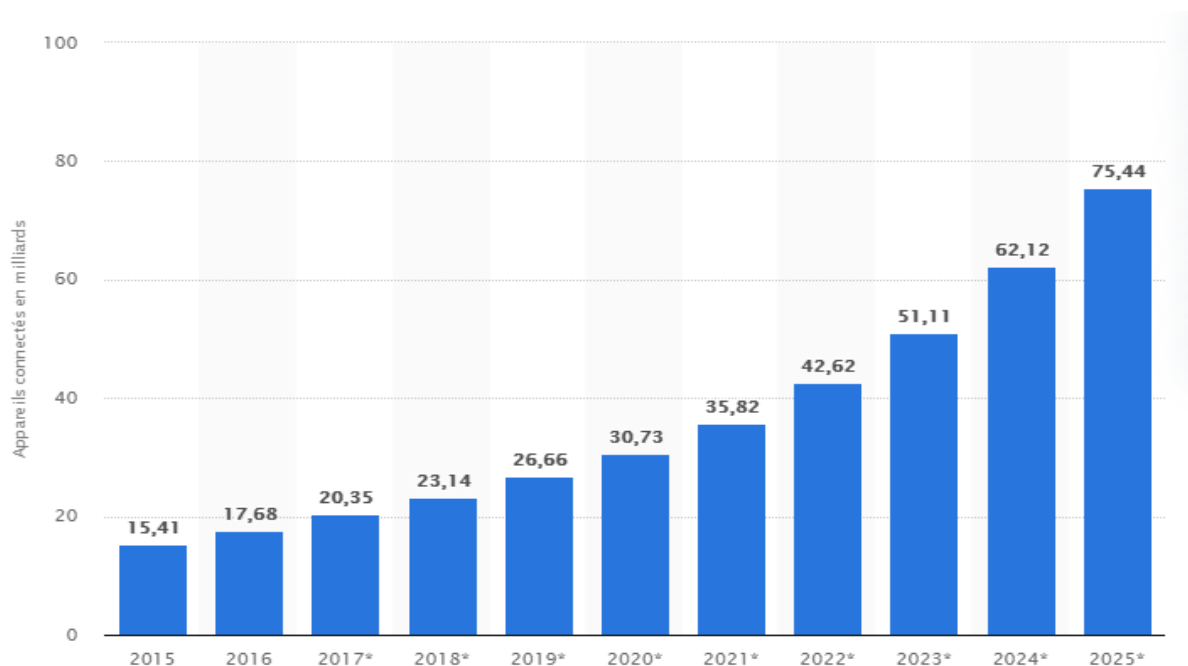


Figure 1.1 : Nombre d'appareils connectés dans le monde 2015-2025. [7]

1.4 L'objet connecté :

1.4.1 Définition :

Un objet connecté est un équipement électronique, doté de capacités très hétérogènes en communication, traitement, mémoire, énergie et collecte de données ambiantes. [8] Par contre l'internet permet seulement de faciliter la connectivité. Pour cela, les objets connectés doivent être vus comme des objets qui possèdent et conservent leurs attributs physiques originels (par exemple un Bracelet Tracker de mouvements), à qui nous avons implémenté des éléments qui les rendent plus intelligents (exemple : un « coach » virtuel permettant de se fixer des objectifs de mobilité). Ensuite, la connexion à Internet par l'objet, lui permet de transmettre les données récoltées, pour les envoyées à des interfaces (par exemple : partage de données sur les réseaux sociaux, sur des plateformes pour les statistiques...). [9]



Figure 1.2 : L'étendue des objets connectés. [11]

1.4.2 Composants des Objets Connectés :

Les objets connectés sont des objets qui intègrent trois éléments :

- Des composants physiques (parties mécaniques, électroniques...).

- Des composants intelligents (comme les capteurs intelligents qui sont dotés d'un processeur et d'une RAM permettant ainsi de s'identifier, mesurer, calculer et comparer des données, et des actionneurs qui vont réaliser des actions en fonction des données captées).
- Des composants de connectivité (un système de transmission de ces données) comme une antenne de communication (RFID, Bluetooth, wifi, ou autre). [10]

Et voici une figure qui représente les différents composants qu'un objet connecté peut contenir :

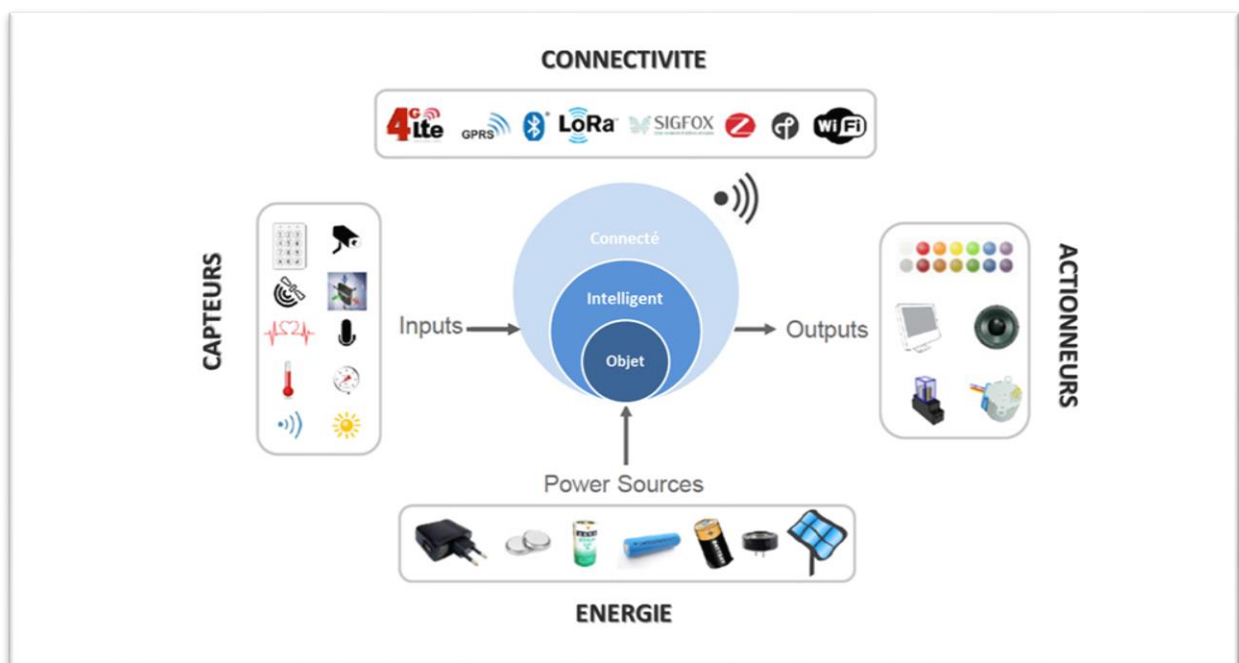


Figure 1.3 : Les composants des objets connectés. [12]

1.4.3 Caractéristiques communes à tous les objets connectés dans un réseau d'IdO:

Généralement, les caractéristiques des objets connectés se résument en :

- **Identité** : pour que les objets soient gérables il est essentiel que chaque objet connecté possède une identité unique qu'il lui est propre, et qui le différencie des autres objets d'un même système, par exemple : une adresse IP dans le cas du protocole de communication TCP/IP.

- **Interactivité** : les progrès technologiques ont permis de connecter un nombre incalculable et varié d'objets et de dispositifs. Et donc quelques objets n'ont pas besoin d'être connecté à un réseau à tout moment, pour des objets dits passifs tels que des livres ou des DVD, des étiquettes RFID doivent seulement être en mesure de signaler leur présence, de temps en temps.
- **Programmable** : l'objet connecté doit être programmé et piloté à distance via un ordinateur, une tablette ou un smartphone.
- **Sensibilité** : un objet a la capacité de percevoir son environnement et peut collecter ou transmettre des informations. Il peut ainsi avoir des capteurs signalant les différentes perturbations dans son environnement.
- **Autonomie** : est la caractéristique la plus importante pour un objet connecté. On désigne par l'autonomie, la capacité d'agir sans l'intervention d'un tiers, donc ils doivent fonctionner indépendamment (chaque objet devient responsable de lui-même) En d'autres termes, les objets doivent pouvoir être traités et surveillés individuellement, généralement depuis un point éloigné. [13]

Et récemment, suite à l'avancement technologique et surtout de l'intégration électronique, des fonctionnalités très particulières ont été rajouté comme :

- **La localisation** : de nos jours, les objets intelligents peuvent être soit au courant de leur position ou peuvent également être localisés, grâce à des technologies de localisation, par exemple : GPS, Réseau mobile, Mesure de temps de vol par ultrasons, Balises radio, UWB...etc. [14]

1.5 Caractéristiques fondamentales de l'Internet des Objets :

Nous allons désormais discuter des caractéristiques fondamentales de l'Internet des Objets qui sont au nombre de 5 ainsi que de leurs exigences :

1.5.1 L'inter-connectivité:

Tous les objets présents dans l'Internet des Objets peuvent être connectés à l'infrastructure mondiale de l'information et de la communication. Parmi les formes de

connectivités les plus connues entre l'IdO et internet, nous retrouvons une variété de mode de connexion et de technologies comme par exemple dans les options sans fils, les technologies : ANT+, Bluetooth, GPRS, LTE, NFC, RFID, Weightless, WLAN, ZigBee, et Z-Wave. [14]

1.5.2 Les services reliés aux objets :

L'IdO est capable de fournir des services qui sont liés aux objets, tout en tenant compte des contraintes, telles que la protection de la vie privée, ainsi que la cohérence sémantique entre les objets physiques et leur(s) objet(s) virtuel(s) associé(s). [14]

1.5.3 L'hétérogénéité :

Étant donné qu'ils sont basés sur des plates-formes matérielles ou des réseaux différents, les appareils reliés à l'IdO sont hétérogènes. Ils peuvent ainsi interagir avec d'autres appareils et dispositifs ou des plates-formes de service via d'autres réseaux. [14]

1.5.4 Les changements dynamiques :

L'état des dispositifs (par exemple, connecté/déconnecté) change de manière dynamique, tout comme le contexte dans lequel ces dispositifs fonctionnent, qu'il soit relié au cadre spatio-temporel, ou également dans le cadre de la vitesse ou encore de la localisation. Il est important de noter que leur nombre est également susceptible d'évoluer lui aussi. [14]

1.5.5 Une très grande échelle :

Dans le futur, en vue de la croissance exponentielle des dispositifs le nombre sera au moins dix fois plus nombreux qu'à l'heure actuelle. Une valeur équivalente à des milliards de zettaoctets (10^{21}) de données feront partie de notre futur, donc les objets doivent être à la fois gérés et capables de communiquer, sans que les données entre en conflit. L'extensibilité est une caractéristique évidente et une exigence de l'IdO, car certaines solutions issues de cette technologie ne révéleront leur potentiel qu'à l'échelle à partir de laquelle elles deviendront exploitables. [14]

1.6 Les systèmes d'IdO :

1.6.1 Les fonctionnalités principales d'une architecture IdO:

Peu importe le domaine, le système ou bien l'application réalisée à base l'IdO, Quatre étapes sont indispensables pour chacun, afin que l'œuvre soit complète, fonctionnelle et optimale.

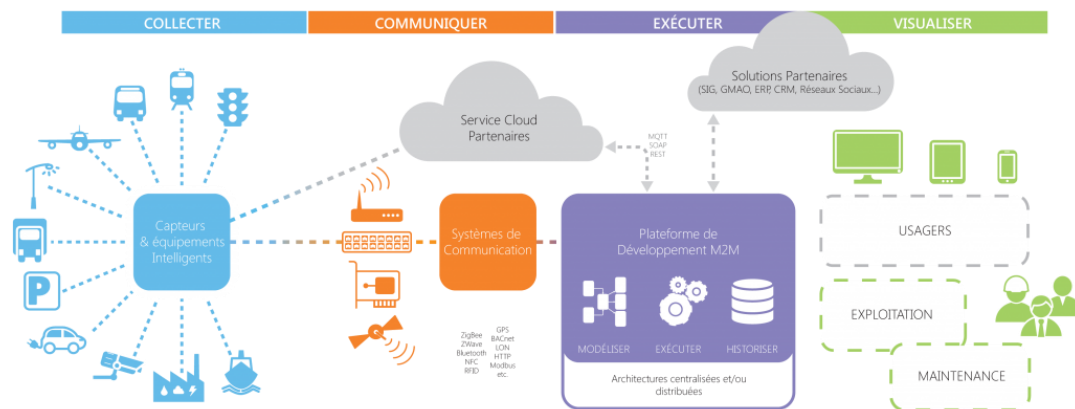


Figure 1.4 : Les fonctionnalités de base d'un système d'IdO. [15]

Explication des différentes étapes :

- a- **Collecter** : c'est la communication entre objets, la première couche au niveau d'un système IdO, qui peut contenir des capteurs de mesures et des actionneurs qui ont le pouvoir d'agir sur l'environnement, pour lesquelles sont implémentés des protocoles de communication, généralement à basse énergie / bas débit et sont classés selon leur portée : courte (ex : BLE, NFC), moyenne (ex : Zigbee) ou longue (ex : Sigfox et LoRa). [16]
- b- **Communiquer** : cette étape est consacrée pour l'envoi des données, du réseau d'objets vers le Cloud. Deux modèles de protocoles sont distingués pour transporter les données : Le modèle Publish / Subscribe avec des protocoles de type MQTT et le modèle REST avec des protocoles comme HTTP. [16]

- c- **Exécuter** : c'est la couche qui s'occupe du stockage, traitement et gestion des données. [16]
- d- **Visualiser** : Cette étape a pour tâche d'afficher et de fournir un service d'interaction avec les objets connectés à travers différentes applications dédiées. Un utilisateur, à travers une application mobile, peut interagir avec ses objets en consultant leurs données ou bien en envoyant des actions vers ses objets. [16]

De même, l'ITU-T a proposé une approche similaire, d'ailleurs elle est considérée comme générale, et qui est composée de quatre couches :

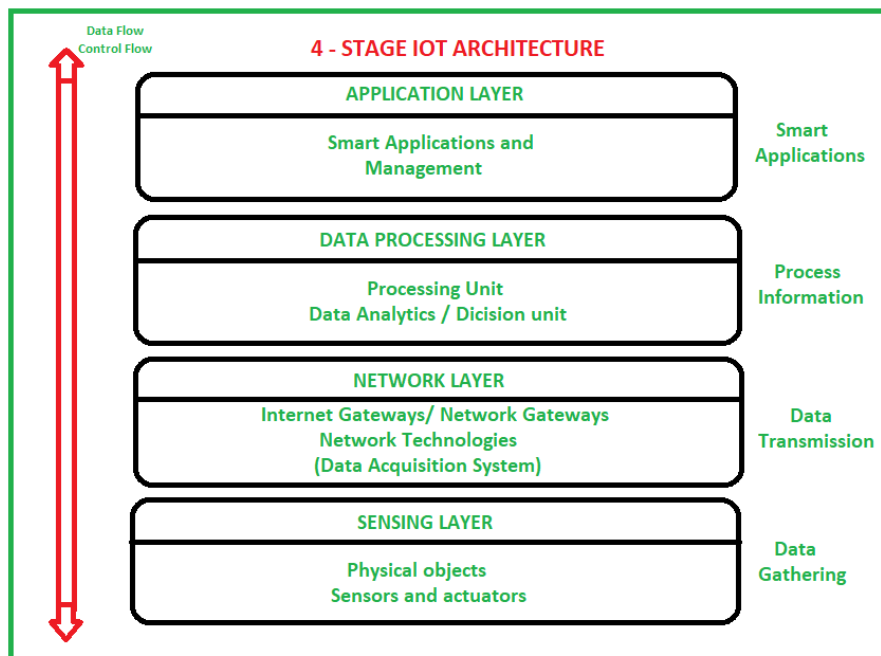


Figure 1.5 : les couches de L'architecture proposée par l'ITU-T. [20]

1.6.2 Les architectures d'IdO :

Même s'il n'existe pas d'architecture IdO unique universellement acceptée, n'empêche que beaucoup d'organismes proposent des architectures qui leur sont favorables, donc les architectures diffèrent d'un organisme à un autre.

a) **Architecture IEEE Standards Association :**

Propose une architecture à trois couches : Perception, Réseau et Application ; ou une autre nomination des couches : Réseaux Capteurs, Cloud Computing et Application.

- **Perception (Réseaux Capteurs) :** C'est la couche physique de l'architecture. C'est là que les capteurs et les appareils connectés entrent en jeu lorsqu'ils collectent diverses quantités de données selon les besoins du projet. Il peut s'agir de dispositifs de périphérie, de capteurs et d'actionneurs qui interagissent avec leur environnement. [17]
- **Réseau (Cloud Computing) :** Les données collectées par tous ces appareils doivent être transmises et traitées. C'est le travail de la couche réseau en connectant ces périphériques à d'autres objets intelligents, serveurs et périphériques réseau. En gérant également la transmission de toutes les données. [17]
- **Application :** La couche d'application est ce avec quoi l'utilisateur interagit. C'est ce qui est responsable de la fourniture de services spécifiques à l'application à l'utilisateur. Il peut s'agir d'une implémentation de maison intelligente, par exemple, où les utilisateurs appuient sur un bouton de l'application pour allumer une cafetière. [17]

b) **Modèle de référence CISCO:**

Ce modèle d'architecture de CISCO est doté de 7 couches principales [Figure 1.5] comme le modèle OSI :

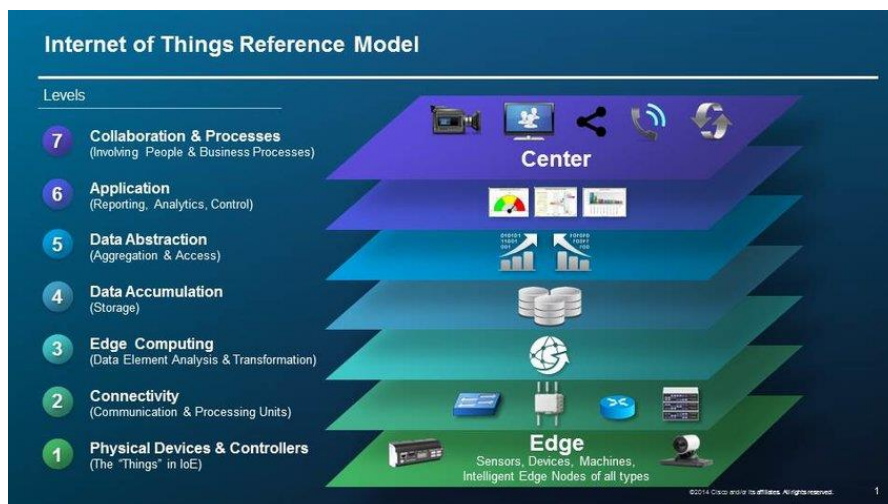


Figure 1.6 : Les différentes couches de l'IdO selon CISCO. [18]

Niveau 1 : Périphériques physiques et contrôleurs

- Il contient les objets, telles que les capteurs, les appareils et les machines, les objets virtuels.

Niveau 2 : Connectivité

- Ce niveau se compose des unités de communication et de traitement.
- Effectue le routage, la commutation et la traduction des protocoles.
- Facilite les communications entre les périphériques de niveau 1 ainsi que la communication entre les réseaux.

Niveau 3 : Calcul Edge (Fog)

- Ce niveau reçoit les paquets de données et génère des données compréhensibles à des niveaux supérieurs.
- Effectue l'analyse et la transformation des éléments de données, le filtrage des données, le nettoyage, l'agrégation et l'inspection du contenu des paquets.

Niveau 4 : Accumulation de données (stockage)

- Ce niveau convertit les données en mouvement en données au repos.
- Le format de données est converti des paquets réseau en tables relationnelles de base de données.
- Les données sont également réduites grâce au filtrage et au stockage sélectif.

Niveau 5 : Abstraction des données (agrégation et accès)

- Il combine des données provenant de sources multiples.
- Il simplifie, filtre, sélectionne, projette et reformate les données pour servir les applications clientes.
- Il réconcilie les différences de forme, de format, de sémantique, de protocole d'accès et de sécurité des données.

Niveau 6 : Application (Reporting, Analytics, Control)

- Ce niveau contrôle les applications et effectue de l'aide à la décision et des analyses.

Niveau 7 : Collaboration et processus

- Ce niveau implique les personnes et les processus d'affaires.

Les niveaux 1 à 3 sont également appelés la « couche côté bord ».

Les niveaux 4 à 6 sont également appelés « couche côté serveur/côté Cloud ».

Le niveau 7 est également appelé la « couche côté utilisateur ». [19]

1.7 Les technologies de l'IdO :

L'Internet des objets a d'abord été inspiré par des membres de la communauté RFID, qui se sont référés à la possibilité de découvrir des informations sur un objet marqué en parcourant une adresse Internet ou une entrée de base de données qui correspond à une technologie particulière de communication RFID ou en champ proche. Les technologies clés inclus dans l'IdO sont nombreuses, car elle touche aussi la technologie des capteurs, l'intelligence intégrée et d'autres.

1.7.1 Radio Frequency Identification (RFID):

L'identification par radiofréquence est un système qui transmet l'identité d'un objet ou d'une personne sans fils, au moyen d'ondes radio sous la forme d'un numéro de série. La technologie RFID est parmi les solutions des problèmes d'identification des objets qui nous entourent de manière rentable.

La technologie est classée en trois catégories en fonction de la méthode de fourniture d'alimentation dans les Tags : RFID active, RFID passive et RFID semi-passive. Les principaux composants de la RFID sont **élément de mémoire, un lecteur, une antenne et un élément de contrôle**. Il est plus fiable, efficace, sécurisé, peu coûteux et précis. La RFID offre une vaste gamme d'applications sans fils. [21]

Parmi les risques de cette technologie :

Les atteintes à la confidentialité, c'est-à-dire le suivi non autorisé des tags et attaques sur la disponibilité. [50]

1.7.2 Code électronique des produits (EPC):

Electronic Product Code (EPC) est un code 64 bits ou 98 bits enregistré électroniquement sur une étiquette RFID et destiné à concevoir une amélioration du système de codes-barres EPC. Le code EPC peut stocker des informations sur le type d'EPC, le numéro de série unique du produit, ses spécifications, les informations du fabricant, etc. EPC a été développé par le centre Auto ID au MIT en 1999. [21]

1.7.3 Wireless Fidelity (Wi-Fi):

La fidélité sans fils (Wi-Fi) est une technologie de réseau qui permet aux ordinateurs et autres appareils de communiquer via un signal sans fils. [21]

Parmi les risques de cette technologie Wi-Fi :

Attaques actives, telles que les attaques de brouillage. [50]

1.7.4 Bluetooth :

La technologie sans fils Bluetooth est une technologie radio de courte portée peu coûteuse qui élimine le besoin de câblage entre les dispositifs tels que les ordinateurs portables, les PDA, les caméras et les imprimantes, permet une portée efficace de 10 - 100 mètres, avec un débit généralement moins de 1 Mbps. Bluetooth utilise la spécification de la norme IEEE 802.15.1. [21]

Parmi les risques de cette technologie Bluetooth :

Attaques Man-In-The-Middle, corruption de données et DOS. [50]

1.7.5 Zig-Bee :

Zig-Bee est l'un des protocoles développés pour améliorer les fonctionnalités des réseaux de capteurs sans fils. La technologie Zig-Bee est créée par l'Alliance Zig-Bee, fondée en 2001. Il s'agit d'un protocole de réseau sans fils de faible puissance basé sur la norme IEEE 802.15.4, Zig-Bee a une portée d'environ 100 mètres et une bande passante de 250 kbps et peut être utilisé avec les topologies étoile, arbre de cluster et maillage. [21]

Parmi les risques de cette technologie Zig Bee :

Traffic sniffing (écoute clandestine), décodage de paquets et manipulation de données. [50]

1.7.6 Le Z-Wave :

C'est un protocole de communication sans fils dédié à la domotique, facile à installer dans la maison. Il possède une portée de 30 mètres. C'est un réseau maillé, c'est-à-dire que chaque appareil connecté au système est émetteur de données mais peut aussi relayer celles qui sont émises par ses voisins, Cela permet d'élargir sa portée. [24]

Parmi les risques de cette technologie Z-wave:

Attaque sur le cryptage codé en dur pour dévoiler la clé de contenu pour dévoiler le contenu. [50]

1.7.7 Near Filed Communication (NFC) :

La communication en champ proche (NFC) est un ensemble de technologies sans fils à courte portée à 13,56 MHz, nécessitant généralement une distance de 4 cm. La technologie NFC rend la vie plus facile et plus pratique pour les consommateurs du monde entier en la rendant plus simple pour effectuer des transactions, échanger du contenu numérique, et connecter les appareils électroniques avec une touche. [21]

Parmi les risques de cette technologie NFC :

Déni de service (DOS), fuite d'informations, détection du trafic (écoute clandestine) en mode actif. [50]

1.7.8 Wireless Sensor Networks (WSN) :

Un WSN est un réseau sans fils, composé de dispositifs autonomes spatialement distribués, en utilisant des capteurs pour surveiller de manière coopérative les conditions physiques ou environnementales, telles que la température, le son, les vibrations, la pression, le mouvement ou les polluants, à différents endroits. Un réseau de capteurs sans fils est un élément important du paradigme IdO. [21]

Parmi les risques de cette technologie WSN :

Attaques « black hole routing », dommages physiques / manipulation non autorisée. [50]

1.8 M2M (machine to machine):

Machine-to-machine, ou M2M, est une étiquette large qui peut être utilisée pour décrire toute technologie qui permet aux appareils en réseau d'échanger des informations et d'effectuer des actions sans l'aide manuelle des humains. L'intelligence artificielle (IA) et l'apprentissage automatique (ML) facilitent la communication entre les systèmes, leur permettant de faire leurs propres choix autonomes. Le M2M est considéré comme la base de l'internet des objets (donc vue comme inspiration ou l'évolution du M2M vers IoT). [22]

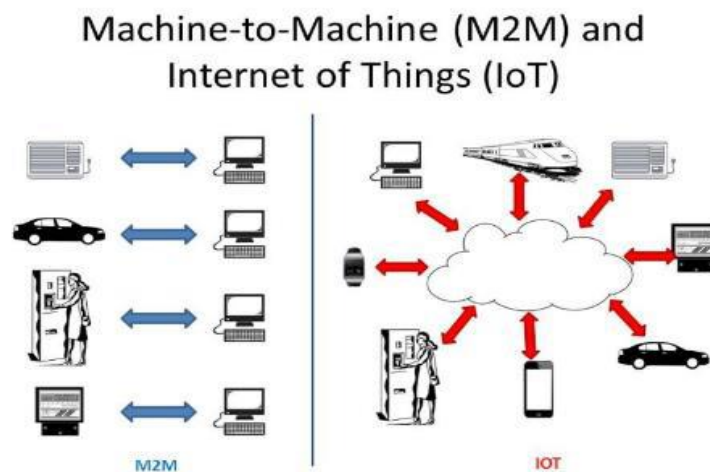


Figure 1.7 : La différence entre le M2M et l'IoT. [24]

1.9 Les Protocoles de communication applicatifs dans l'IdO :

Un protocole applicatif est un ensemble de règles définissant le mode de communication entre deux applications informatiques. Ils se basent sur les protocoles de transport (TCP/UDP) pour établir dans un premier temps des routes et échanger les données selon l'ensemble des règles du protocole sélectionné.

1.9.1 Protocoles de messagerie :

a) MQTT (Standard dans l'IoT depuis 2015) :

MQTT pour « **Message Queuing Telemetry Transport** », est un protocole de messagerie de publication et d'abonnement (**Publish/Subscribe**), basé sur le protocole TCP/IP. [25]

Le protocole MQTT utilise une architecture « *publish/subscribe* » en contraste avec le protocole HTTP et son architecture « *request/response* ». Le point central de la communication est le **Broker** MQTT, en charge de relayer les messages des émetteurs vers les clients. Chaque client s'abonne via un message vers le **Broker** : le « Topic » (sorte de routage pour le **Broker**) qui permettra au **Broker** de réémettre les messages reçus des producteurs de données vers les clients. Les clients et les producteurs n'ont ainsi pas à se connaître, ne communiquant qu'au travers des Topics. Cette architecture permet des solutions multi-échelles. [26]

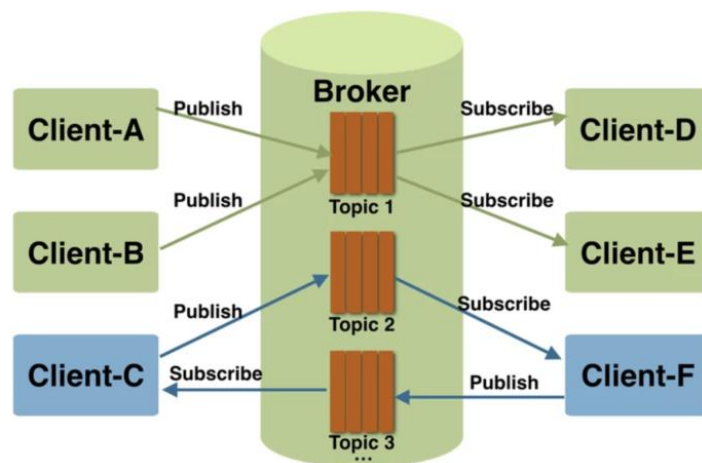


Figure 1.8 : Schéma qui explique le fonctionnement du protocole MQTT.

Un client, appelé **publisher**, établit dans un premier temps une connexion de type **Publish** avec le serveur MQTT, appelé **Broker**. Puis, le **publisher** transmet les messages au **Broker** sur un canal spécifique, appelé **Topic**. Par la suite, ces messages peuvent être lus par des abonnés, appelés **subscribers**, qui au préalable ont établi une connexion de type **Subscribe** avec le **Broker**. [25]

Si la connexion s'arrête, entre le client et le broker alors les messages seront bufférisés et ils seront émis dès que la reconnexion avec le client est rétablie.

Un « Topic MQTT » est une chaîne de caractères qui peut posséder une hiérarchie de niveaux séparés par le caractère « / ». Par exemple, une information de température du salon pourrait être envoyée sur le topic « **maison/salon/temperature** ».

Le signe « + » est un caractère « **wildcard** » qui permet des valeurs arbitraires pour une hiérarchie particulière et le signe « # » pour plus d'un niveau. Les messages envoyés peuvent être de toutes sortes mais ne peuvent excéder une taille de 256 Mo. [26]

b) XMPP (Extensible Messaging and Presence Protocol):

XMPP est à l'origine un protocole de messagerie instantanée utilisé notamment dans les services Jabber et Google Talk. En outre, son extensibilité a permis son utilisation dans d'autres applications telle que la VoIP. Son fonctionnement est basé sur une architecture client/serveur où l'échange de données, au format XML, se fait sur le même principe que les messageries électroniques. [25]

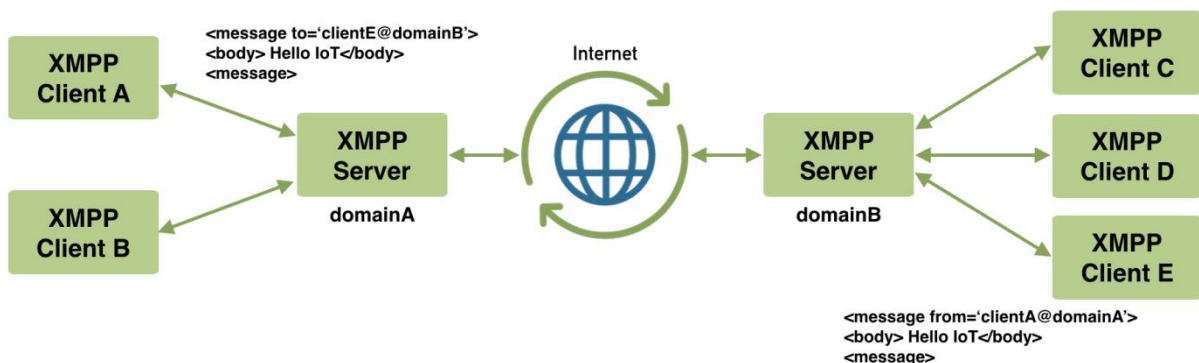


Figure 1.9 : Schéma qui montre le *fonctionnement du protocole XMPP*. [25]

- La communication entre deux clients est asynchrone et est réalisée au travers de serveurs XMPP. Dans un premier temps, un client établit une connexion TCP avec son serveur XMPP, qui communique alors la donnée au serveur XMPP du destinataire. Ce dernier transmet la donnée au destinataire si celui-ci est connecté.
- Le serveur XMPP mémorise la donnée tant que le destinataire en question ne s'est pas connecté.
- Un système XMPP est décentralisé et potentiellement temps réel si l'émetteur et le destinataire sont connectés durant la livraison des messages.

- Chaque client est distingué par un identifiant unique construit sur le modèle suivant : `<nom-du-client>@<nom-du-serveur>`. [25]

1.9.2 Protocoles de transfert web :

a) CoAP (Constrained Application Protocol) :

CoAP est un protocole web basé sur une architecture client/serveur. Ce protocole reprend en partie les méthodes et nomenclatures du protocole HTTP. En revanche, contrairement au protocole HTTP, qui se base sur la suite TCP/IP, le protocole CoAP se base sur la suite UDP/IPv6/6LowPAN, dont les mécanismes d'échange de messages définis par le protocole UDP sont nettement allégés.

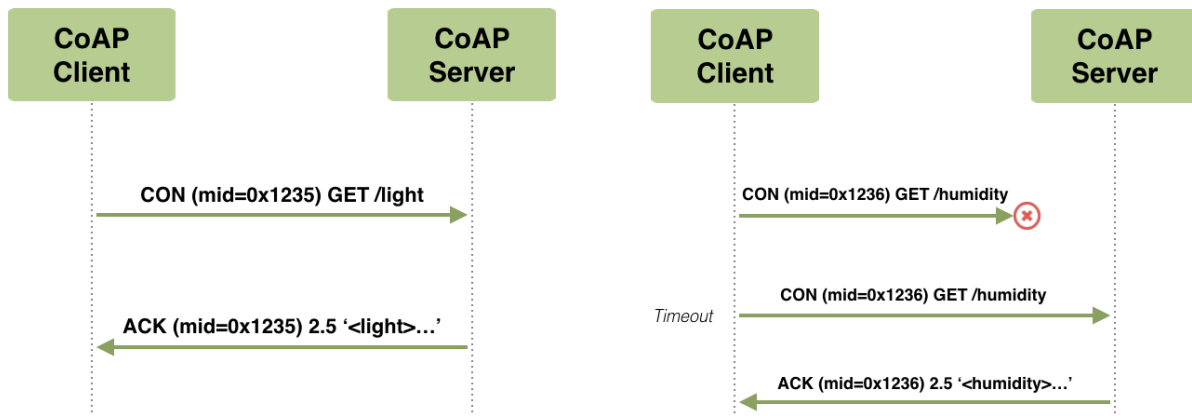


Figure 1.10 : Schéma qui montre le *fonctionnement du protocole CoAP*. [25]

- En-tête d'un message CoAP est fixé à 4 octets, et les types de messages :
- Confirmable : Message envoyé avec une demande d'accusé de réception, noté CON.
- Non-Confirmable : Message envoyé sans demande d'accusé de réception, noté NON.
- Acknowledgment : Accusé de réception du message de type « confirmable », noté ACK.
- Reset : Accusé de réception d'un message qui n'est pas exploitable, noté RST.

Pour transmettre une donnée, un client envoie à un serveur une requête CoAP, dans laquelle se trouve : le type du message (CON ou NON), l'identifiant du message (mid) et une action (GET, POST, PUT ou DELETE) sur une ressource identifiée par une URL. [25]

1.9.3 Protocole réseau (Websocket) :

Le protocole Websocket permet l'établissement d'un canal de communication full-duplex en une seule connexion TCP, entre un client et un serveur. L'image ci-dessous indique les trois principales phases de la vie du canal : [25]

- La phase de connexion appelé « Handshake » initié par le client.
- La phase d'échange bidirectionnel de messages.
- La phase de clôture du canal initié par l'une des deux parties.
- La communication est asynchrone, il exige une requête pour donner une réponse.

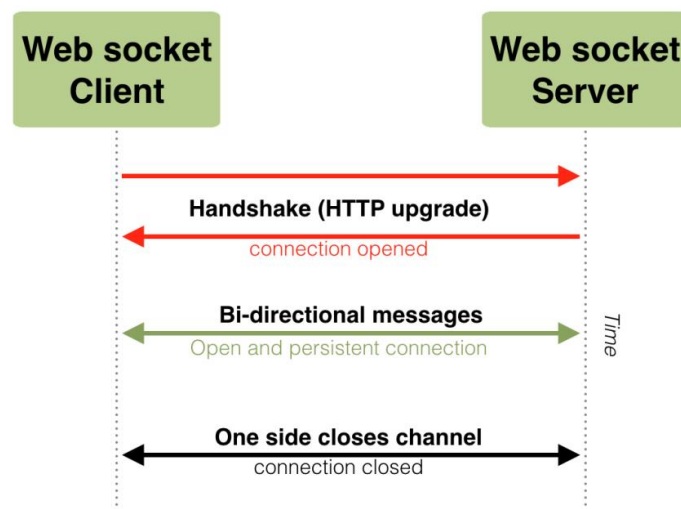


Figure 1.11 : Schéma qui montre le *fonctionnement du protocole Websocket*. [25]

1.10 Applications de L'IdO :

Plusieurs domaines d'applications sont et seront touchés par l'Internet des objets. De nouveaux domaines et environnements où applications IdO amélioreront probablement la qualité de nos vie : la maison, la santé, le travail et l'agriculture et avec de nombreuses façons différentes.

1.10.1 Le foyer de l'Internet des objets :

La vie quotidienne des personnes est considéré comme le premier domaine à être touché, pour le seul et unique objectif de diminuer l'impact des routines sur l'homme, et

donc des capteurs personnels et domestiques peuvent être utilisés pour contrôler les réfrigérateurs, les machines à laver, le climatiseur, le système de surveillance, etc. La consommation des services publics est mesurée toutes les heures, ensuite les données sont transmises sans fils au centre de services publics plusieurs fois par jour. Le service public et les clients peuvent suivre l'utilisation. Cependant, seuls quelques pour cent des clients des pays développés sont équipés de tels dispositifs intelligents et la mise en œuvre globale est lente. [27]

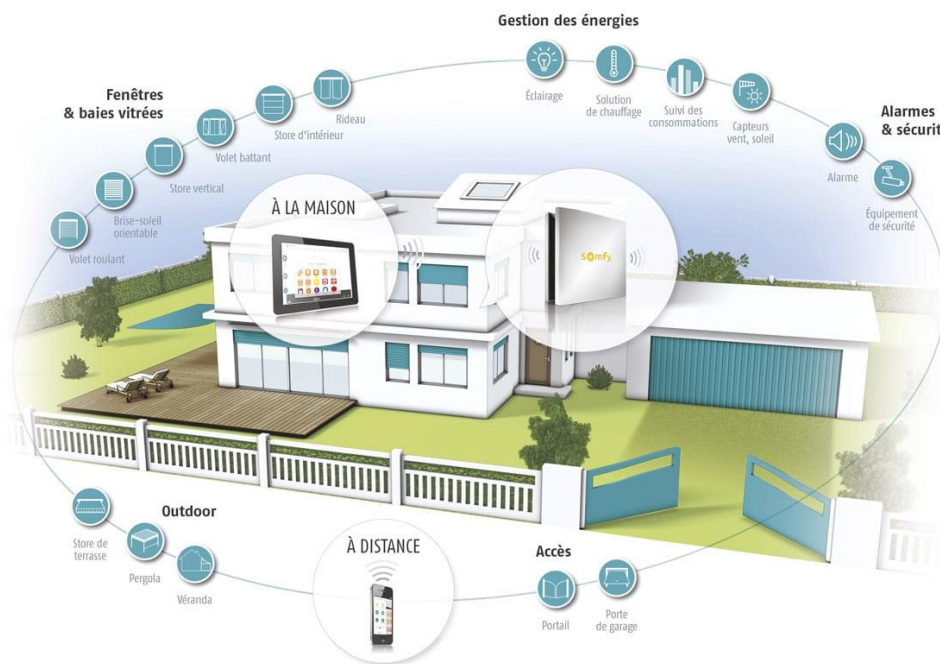


Figure 1.12 : Maison connectée. [29]

1.10.2 Le contrôle du trafic Internet des objets :

L'internet d'objets trouve des applications dans d'autres domaines, tels que le contrôle du trafic routier. Les données relatives à la circulation et à l'état des routes seraient recueillies à l'aide de capteurs, puis communiquées aux centres de contrôle de la circulation puis aux conducteurs sous forme d'informations et de conseils sur la circulation. Par exemple, les informations recueillis sur les routes permettent de contrôler intelligemment les feux de circulation, aussi, les conducteurs sont informés en temps réel afin d'éviter les routes encombrés suite à des accidents de circulation. L'information sur le trafic peut être fournie

par le réseau de capteurs pour déterminer le meilleur itinéraire. L'utilisation des IoT permet donc de réduire les accidents, les embouteillages et les victimes de la circulation. La pollution générée par les voitures peut être réduite, ainsi que la consommation de carburant des voitures. [27]

1.10.3 La santé et l'Internet des objets :

Aujourd'hui, la société moderne est responsable de changer le modèle de soins de santé de l'hôpital orienté vers la maison. En incluant les capacités de l'Internet des objets dans ce segment, de nombreuses solutions efficaces peuvent être mises en œuvre. Certains des plus importants sont dans le domaine du suivi et de la surveillance de l'état du patient à l'aide de la technologie WSN, dans le domaine du service à distance où le diagnostic peut être fourni par Internet, la gestion des informations patient où toutes les données sur le patient sont stockées à un endroit central et peuvent être atteintes via Internet à tout moment, n'importe où. [27]



Figure 1.13 : La santé avec l'internet des Objets. [30]

1.10.4 Le contrôle industriel et l'Internet des objets :

Les applications suivantes sont dans le contrôle industriel. Par exemple, la surveillance des employés, le travail des ascenseurs dans un immeuble, l'utilisation de l'électricité, le chauffage et d'autres systèmes industriels ou de construction. Grâce aux capteurs sans fils de la technologie ZigBee et aux étiquettes passives, divers emplacements intérieurs et

employés peuvent être surveillés. Un capteur peut également être utilisé pour surveiller le niveau de gaz toxiques et d'oxygène à l'intérieur de salles fermées afin d'assurer la sécurité des travailleurs. [27]

1. 11 Les enjeux clés du déploiement de L'internet des Objets :

Lors d'une conférence entre les grands du marché de la technologie, 5 enjeux sont structurés et identifiés :

a) Assurer une connectivité multiple, mobile, fiable et à coût réduit : Pour répondre à une palette de besoins variés, l'internet des objets nécessitent une connectivité étendue qui repose sur une diversité de technologies. De nouveaux réseaux dédiés à l'internet des objets (LPWAN, etc.) complètent les réseaux existants fixes, cellulaires, satellitaires ou de courtes portées (Wi-Fi, Bluetooth, etc.). [31]

b) Veiller à la disponibilité des ressources rares :

Fréquences : Les fréquences sont soumises à un régime d'autorisation individuelle ou générale. Dans ce dernier cas, elles sont dites « libres » et sont utilisées par des acteurs de plus en plus nombreux aux usages non répertoriés. Si, à court terme, une saturation de ces bandes ne semble pas devoir se produire, à moyen terme, cette situation pose la question de la quantité de spectre à même de répondre aux besoins croissants de l'internet des objets. [31]

Identifiants : En ce qui concerne l'adressage, la croissance exponentielle du nombre d'objets connectés, interrogent sur l'unicité ou l'éventuelle pénurie des identifiants et, en particulier, sur la transition de l'IPv4 vers l'IPv6. [31]

c) Garder un jeu ouvert à tous : Le secteur de l'internet des objets est en cours de structuration. Plusieurs technologies existent, sans qu'aucun standard ne s'impose à ce stade, il convient de ne pas entraver l'innovation, en laissant avant tout les utilisateurs arbitrer entre les options. [31]

d) Contribuer à bâtir la confiance : La confiance des utilisateurs pour les solutions de l'internet des objets repose sur deux piliers : la bonne gestion des données, qu'elles soient personnelles ou qu'elles appartiennent aux entreprises, et la sécurité des réseaux, qui permet de garantir les usages. [31]

e) Accompagner les acteurs, favoriser l'écosystème : Les solutions offertes par l'internet des objets sont en pleine multiplication et bouleversent déjà le quotidien des particuliers, des territoires et des entreprises. Cette mutation appelle à la poursuite des travaux pour réunir régulièrement l'écosystème de l'internet des objets autour de thématiques spécifiques. [31]

1. 12 Avantages de l'utilisation des objets connectés :

Les avantages des IoT sont nombreux, en plus de ces caractéristiques nous pouvons ajouter les avantages suivants :

- Améliorer les services quotidiens généraux comme le transport et les parkings.
- La surveillance et maintenance des lieux publics.
- Suivi du personnel.
- Réduire le temps perdu dans les transactions administratives dans la ville.
- Offre un contrôle nouveau, organisé en améliorant la qualité des activités.
- Economiser la consommation de l'énergie dans la ville l'éclairage intelligent pour la ville ...etc. [28]

1. 13 Les principaux risques liés à l'utilisation des objets connectés :

Voici un Tableau qui résume les différents risques et inconvénients :

Risques	Descriptions
-Risque de sécurité.	Il est lié à la sécurisation de la communication entre les objets connectés. Il fait référence au piratage des données et à la prise de contrôle de l'objet par des tiers non autorisés.
-Risque de violation de la vie privée.	La vie privée (collecte d'informations personnelles sensibles par les objets connectés à l'insu du consommateur lors de leur stockage,

	accès, modification, utilisation ou commercialisation non autorisés).
-Risque de performance.	C'est le risque fonctionnel associé à : <ol style="list-style-type: none"> 1- La fiabilité des objets connectés (bugs, dysfonctionnements...). 2- La fiabilité des informations communiquées par ces objets (mesures non fiables ou erronées). 3- La performance attendue de ces objets (valeur ajoutée non significative).
-Risque psychologique.	Il peut faire référence à deux types de risques : le risque de dépendance technologique et le risque d'isolement engendré par la substitution de la communication avec les humains par la communication avec des objets connectés ou la communication entre objets connectés.
-Risque physique.	Il s'agit du risque lié à la santé du consommateur au regard du volume conséquent d'ondes émises au quotidien par ces objets (qui seront omniprésents et ubiquitaires) d'autant plus que certains objets connectés sont en contact direct avec le corps : lentille connectée, bracelet connecté...
-Risque financier.	Il fait référence aux pertes financières du consommateur en cas de mauvais choix (non correspondance de l'objet connecté aux attentes ou aux besoins du consommateur).

Tableau 1.1 : Les principaux risques liés à l'utilisation des objets connectés. [10]

Conclusion :

Dans ce premier chapitre, nous avons présenté les notions générales de l'IoT, définitions, caractéristiques, composantes, architectures et l'impact sur l'homme. Nous pouvons constater que L'IoT est sur le point de changer le monde et la vie des humains sur le plan technologique, économique et social.

CHAPITRE 2:

La plateforme de gestion des environnements IOT.

La plateforme de gestion des environnements IOT

Comme nous l'avons déjà vu, dans le chapitre précédent, les fonctionnalités principales d'une architecture IdO, quatre étapes sont essentielles et nécessaire pour le bien du fonctionnement du projet IdO à réaliser, parmi ces étapes nous avons cité la visualisation, qui exprime l'interactivité entre le personnel et son réseau d'objets dédié, pour cela une plateforme pour la gestion est le meilleur moyen permettant d'assurer aux utilisateurs ce contrôle. La réalisation d'une telle plateforme de gestion a déjà été mise en œuvre [38]. Nous souhaitons rajouter l'opportunité d'offrir un service encore plus vaste, de tel façon à ce que les utilisateurs auront l'avantage de contrôler non seulement, un seul réseau dédié à un seul environnement, mais plusieurs réseaux qui forme des environnements, par exemple : une voiture peut servir d'environnement avec son ensemble de capteurs, une maison, une serre... Tout implémenté dans une, et une seule plateforme connectée et organisée en un ensemble d'environnement.

Dans ce chapitre nous allons d'abord présenter l'ancienne plateforme, avec quelques commentaires et critiques, puis nous discuterons les modifications et ajouts que nous proposons pour améliorer cette plateforme.

2.1 Plateformes IoT :

Une plateforme IoT est un ensemble de services permettant de collecter, stocker, corréler, analyser et exploiter les données. [50]

2.1.1 Composants d'une plateforme IoT :

a) **Connectivité et normalisation** : Elle a pour fonction d'apporter différents protocoles et différents formats de données dans une seule interface, standardisée une interface de communication avec plateforme. [50]

b) Stockage des données : La gestion des données issues de différents dispositifs IoT, apporte aux exigences des bases de données un nouveau niveau: Le volume, la Variété, la Rapidité, la Véracité. [50]

c) Gestion des actions et traitement : Les données capturées par le module connectivité et normalisation sont stockées dans la base de données, prennent vie dans cette partie de la plateforme, suite à un événement, une action intelligente basée sur les données récoltées est déclencher automatiquement. [50]

d) Analytique : De nombreux cas d'utilisation de l'IoT vont au-delà de la gestion des actions, et utilise un moteur d'analyse de données, qui par la suite après un apprentissage peut fournir des algorithmes de gestion en choisissant la meilleure combinaison à effectuer selon quelques facteurs (ciel nuageux augmentation d'éclairage dans la maison). [50]

e) La visualisation : permet aux utilisateurs de voir les modèles et observer les tendances. Ses informations se présentent dans la plateforme sous la forme de lignes, empilées ou camemberts, des graphes... [50]

2.1.2 Les niveaux des plateformes :

Nous pouvons identifier 3 types de plateformes de niveau différents :

Niveau 1: Plateforme de connectivité (ou middleware), qui collecte des données et possède un bus de messagerie.

Niveau 2 : Plateforme d'actions, elle fait le traitement des données.

– Gestion des événements-actions, un évènement engendre automatiquement une action sur l'environnement.

Niveau 3 : Plateforme de grande envergure qui disposent de :

– Dispositifs, protocoles, les normes indépendantes.

– Back-end visuels multiformes.

– Interfaces externes sophistiquées (par exemple, API, SDK).

- Solutions de base de données avancées, conçues pour le Big Data.
- Extensibilité de la plateforme: pour gérer un grand nombre d'appareils. [50]

2.2 La plateforme générique IoT :

2.2.1 Présentation :

Cette plateforme qui a été réalisée déjà par M.Daouadji et El A.aichi [38], dans le seul et unique but, de fournir une plateforme standard dédiée aux systèmes IoT. Le travail réalisé est divisé comme suit :

- Partie Software : elle englobe tout ce qui concerne l'aspect programmation des services que la plateforme offre
- Partie Hardware : c'est le réseau d'objets connectés qui vont être reliés à la plateforme.
- Partie communication et protocoles : exprime le lien qui relie la partie Software au Hardware.

Le plus grand travail est basé sur la partie logicielle, en effet, il sera question de mettre en œuvre une plateforme Web améliorée, avec laquelle nous aurons la possibilité, d'effectuer des décisions à distance sur un environnement (un réseau), constitué d'un ensemble d'objets connectés.

2.2.2 Les différents outils utilisés :

La plateforme est conçue à l'aide de plusieurs langages et protocoles mais les principaux outils sont :

a) Python :

Python est un langage de programmation portable, dynamique, extensible, gratuit, qui permet (sans l'imposer) une approche modulaire et orientée objet de la programmation, créer et développer depuis 1989 par Guido van Rossum et de nombreux contributeurs bénévoles. [32]

Ce langage est dédié pour :

- Développement web (côté serveur).
- Développement de logiciels.
- Mathématiques.
- Script système.

Pourquoi utiliser Python ?

- Python fonctionne sur différentes plates-formes (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python a une syntaxe simple similaire à la langue anglaise un langage de haut niveau.
- Python a une syntaxe qui permet aux développeurs d'écrire des programmes avec moins de lignes que certains autres langages de programmation multi-bibliothèques.
- Python s'exécute sur un système d'interpréteur, ce qui signifie que le code peut être exécuté dès qu'il est écrit. Cela signifie que le prototypage peut être très rapide.
- Python peut être traité de manière procédurale, orientée objet ou fonctionnelle.[33]

Un langage aussi ancien implique sûrement plusieurs versions, pour notre cas nous utilisons la version 3.9.5 et **Pycharm** de Jet Brains comme environnement de développement.

b) le Framework Django :

Un **Framework** (**infrastructure logicielle ou cadre de travail** en français) désigne en programmation informatique, un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application. Considéré comme une structure logicielle ou un canevas, qui établit les fondations d'un logiciel ou son squelette applicatif. Tous les développeurs qui l'utilisent peuvent l'enrichir pour améliorer son utilisation.

Une infrastructure logicielle est associée spécifiquement à un langage de script ou de programmation comme le cas de **Django** sur Python.

L'objectif de l'utilisation d'un **Framework** est de simplifier et d'uniformiser le travail des développeurs web. [36]



Figure 2.1 : Logo Django Python. [34]

Django possède une structure en étage, conçue selon la structure **MVT** (Models, Views, Template), ceci séparera le travail du développeur et favorise le travail en équipe, de telle façon à ce que les données (Models) sont séparées des traitements (Templates) qui eux même sont séparés des vues (Views).

- ➔ **Models** : sont responsables des données de l'application ; ils définissent, manipulent et stockent les données de votre application, en gros les modèles fournissent un mécanisme de gestion.
- ➔ **View** : est responsable de la représentation des données de l'application, dans le cadre d'une application web, la vue retourne une requête HTTP puis se présente généralement sous la forme d'une partie statique en HTML et CSS.
- ➔ **Template** : c'est la partie qui se charge d'**appeler nos modèles**. Un template est simplement un code HTML qui contient des champs, des données qui doivent être remplies par la vue. Ce code HTML peut contenir des variables, des énoncés conditionnels et des boucles qui assureront le remplissage du template avec toutes les données nécessaires qui doivent être affichées. [37]

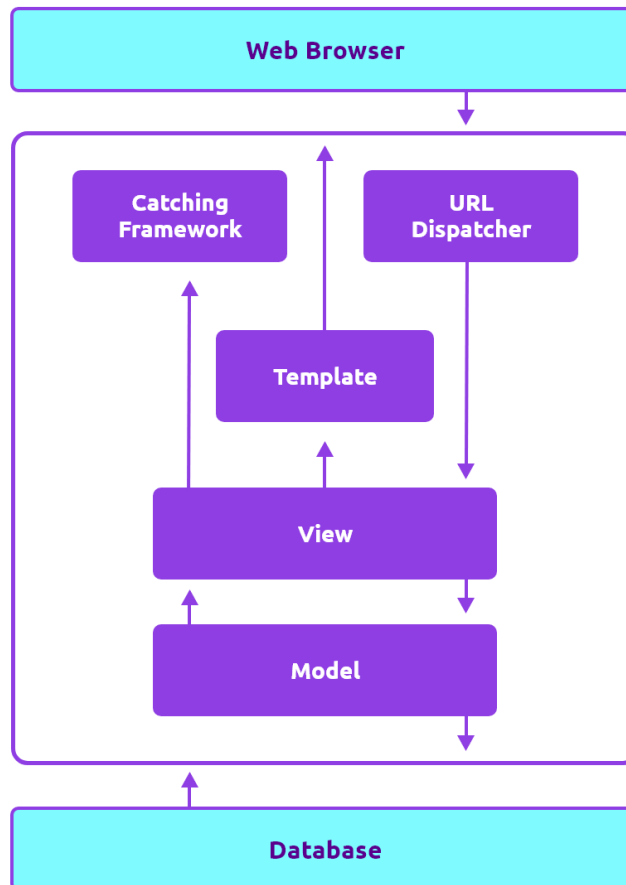


Figure 2.2 : Architecture MVT.

Imposer une structure, obligera les développeurs à coder selon une certaine organisation. Tout commence à partir du navigateur, qui est l'utilisateur final qui tente d'accéder à une page Web d'un site Web, de sorte que la première partie de Django qui impliquerait et interagirait avec le navigateur est le répartiteur d'URL, qui prend l'URL que le navigateur tente d'accéder et de le mapper à sa vue appropriée puis charger cette page spécifique. Cette dernière nécessite des modèles et des Templates prédéfinis afin de charger l'affichage de cette page spécifique avec les données requises.

2.2.3 La conception de la plateforme :

Le principe du fonctionnement de la plateforme se résume dans l'image ci-dessous :

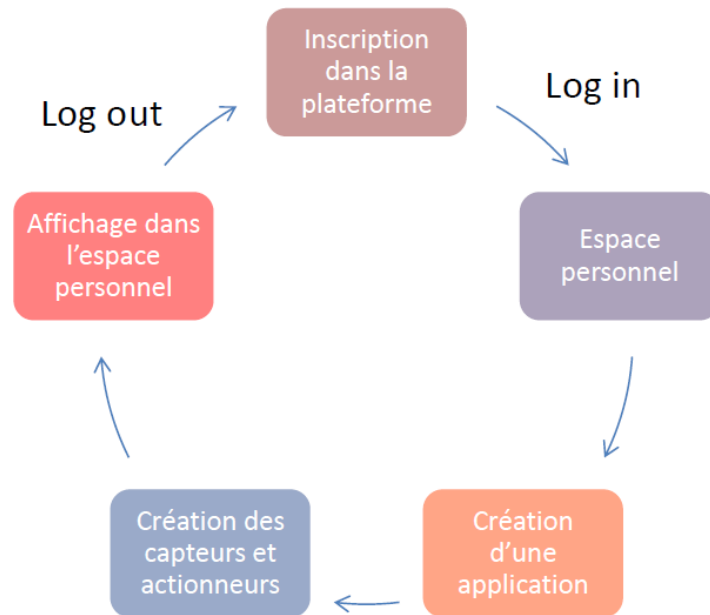


Figure 2.3 : Le schéma conceptuelle de la plateforme. [38]

Dans un premier temps, le personnel intéressé accède d’abord à la plateforme, ce dernier sont dits « user », un compte ou profile doit être crée, afin d’avoir l’accès illimité aux avantages et la sécurité que la plateforme peut offrir, en se souscrivent avec un nom d’utilisateur et un mot de passe.

Une fois le formulaire de l’inscription est validé, nous serons directement redirigées vers un espace personnel, ou nous pourrons ajouter des applications, surveiller nos capteurs, agir sur nos actionneurs ou plutôt visualiser l’historique des changements, éditer ou supprimer des applications. Puis toutes ses actions seront stocker automatiquement à chaque modification effectuer dans la base de données, afin d’y avoir accès à chaque connexion du user correspondant.

2.2.4 L’environnement de développement de la plateforme :

Afin de réaliser cette structure, avoir recours à **Django** offre des avantages. Parmi ses avantages et comme l’exige les formalités, le travail a été conçu selon des applications, créés avec **Django**, pour que le code soit lisible, organisé et facile à modifier.

1^{ère} Étape :

Ce genre de projet exige l'installation d'un environnement virtuel, dans lequel seront installés les différents packages et bibliothèques, que le projet utilisera comme **Django**, et donc nous utilisons **Pycharm**, un IDE dédié pour **Python**, ce dernier nous donne l'avantage de créer un venv (virtual environment) [Figure 2.4]. Alors choisissez le venv puis ajoutez à partir de cet environnement votre « projet IoT », ou bien commencez directement un nouveau projet.

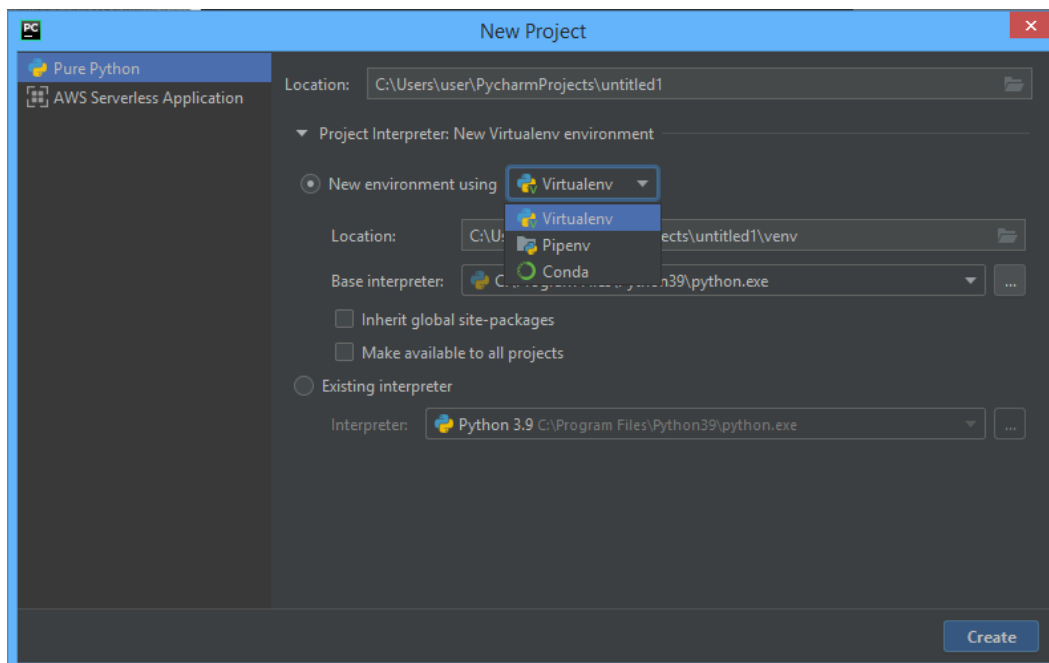


Figure 2.4 : Création d'un projet **Pycharm**.

Un outil **Python** des plus banals, mais très efficace ; préinstallé lors de l'installation du langage **Python**, « le **pip** » cette commande qui s'exécute au terminal de votre IDE vous offre l'avantage d'installer différents packages Python en exécutant seulement la commande **pip install "nom de package == version"**. Exemple :

Pip install django==3.0

2^{ème} Étape :

Après avoir ouvert le projet respectif est la création du venv, procédez à l'installation de **Django** via la commande **pip**, afin de pouvoir exécuter le projet avec les multiples

commandes **Django** à partir du terminal. 3 commandes sont essentielles pour faire marcher la plateforme :

Py manage.py makemigrations

Grace à l'exécution de cette commande **Django** examine tous les modèles disponibles et crée de nouvelles migrations, pour les tables de base de données qui n'existent pas encore, en fonction des modifications détectées dans vos modèles.

Py manage.py migrate

Synchronise l'état de la base de données avec l'ensemble actuel de modèles et de migrations. C'est le moyen pour Django d'enregistrer les modifications apportées.

Py manage.py runserver

Démarre un serveur Web de développement léger sur l'ordinateur local. Par défaut, le serveur s'exécute sur le port 8000 sur l'adresse IP. [39]

2.2.5 La structure de la plateforme IoT :

En ouvrant le « projet IoT », nous distinguons 2 fichiers essentiels, un est dédié pour la plateforme web conçus avec le Framework **Python Django**, et le deuxième attribué au protocole **MQTT** ou plus précisément pour le partage de données. Nous commençons par la première partie qui concerne le développement web.

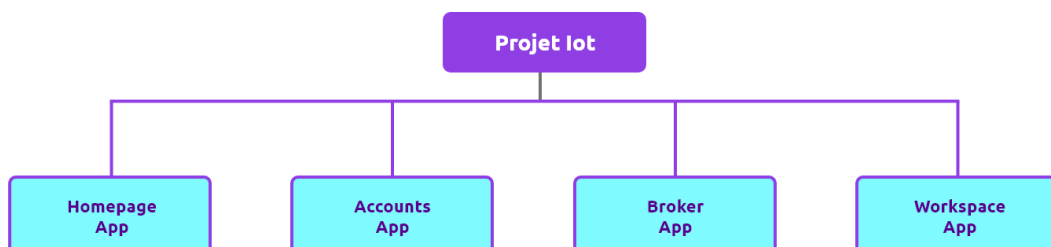


Figure 2.5 : les applications de la plateforme. [38]

Nous pouvons constater que l'utilisation d'un Framework signifie que nous allons suivre une certaine démarche propre. Afin de réaliser cette conception sous le Framework **Django**, nous procédons par la création d'un certain nombre d'applications, qui vont garantir le fonctionnement de la plateforme [Figure 2.6], voici les applications créées pour la plateforme :

a) Application Home page :

C'est la page d'accueil de notre plateforme, affichant à l'utilisateur, le nom, le slogan, des informations générales qui concerne l'utilisation de l'interface, nous proposons aux visiteurs l'éventualité de s'inscrire avec un **SIGN-UP**, ou de se connecter directement, en cas de possession d'un compte, avec **LOG-IN**. Cette application est conçue à partir d'une vue seulement, ce qui implique l'appel directe des Templates (code HTML et CSS prédéfini) avec l'adresse IP du site qui est directement relié à l'URL de la vue.

b) Application accounts :

Cette application a pour but de nous permettre d'inscrire et de confirmer l'identité des utilisateurs, afin d'accéder à l'espace personnel. Nous utilisons un système d'authentification du Framework **Django** dit **USER**, par la suite nous pouvons distinguer deux types de démarches possibles :

- **Inscription** : cette démarche se résume en la sauvegarde des informations utilisateurs dans la base de données, en remplissant le formulaire d'inscription.
- **Connexion** : pour se connecter, l'utilisateur doit saisir les informations relatives à son compte, et permettre l'authentification de son identité.

c) Application workspace:

C'est dans cette application que réside l'intérêt de la plateforme. Prédestiné à créer des applications IoT selon le besoin de l'utilisateur, en capteurs et actionneurs, qu'ils souhaitent surveiller ou contrôler. Ses applications sont identifiées avec un nom et un ID unique de 11 caractères générés automatiquement par la bibliothèque de **Django** UUID puis ses applications IoT sont affichées dans l'accueil de l'utilisateur concerné, et classés selon la date

de création. Grâce à ce dernier, chaque application IoT est stockée dans la BD selon ses deux champs « Name » et « ID » [Figure 2.6].

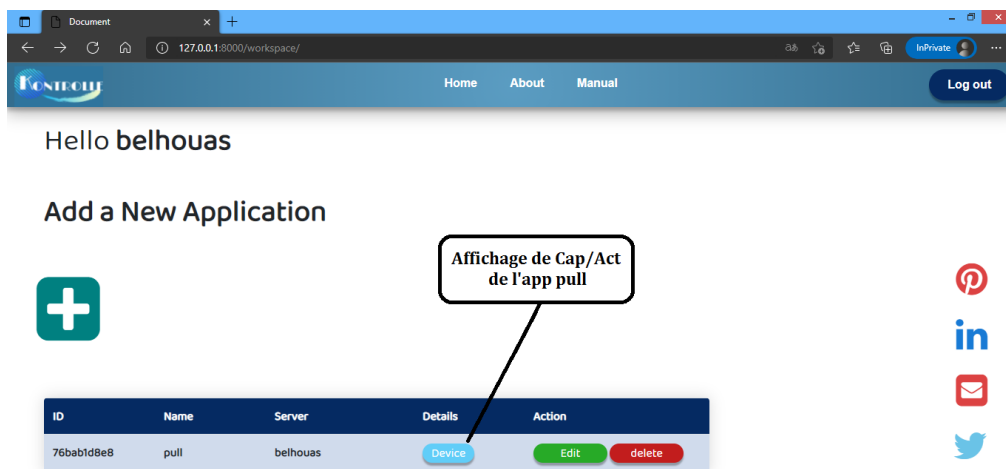


Figure 2.6 : La page de l'espace personnel.

Après la création de l'application, nous sommes redirigés vers une autre vue où l'utilisateur va pouvoir nommer et sélectionner le type (Act/Cap) et l'application correspondante.

La dernière étape consiste à sélectionner le Topic que l'application va utiliser sur le serveur **MQTT** [Figure 2.7].

Figure 2.7 : la page de création de Cap/Act.

d) Application broker :

Cette application sert de lien entre les objets connectés et la plateforme, en se servant du serveur **MQTT** comme intermédiaire de cette communication. Les données reçus du serveur sont enregistrées dans la base de données [Id, auto_Id, Created_date, Topic, pub_info, palyoad, State].

e) Channels :

C'est un outil qui offre la possibilité de travailler avec des codes synchrones et asynchrones. Par default, **Django** utilise des codes synchrones via la configuration WSGI. Grâce à Channels, **Django** peut étendre ces capacités au-delà de HTTP pour gérer les Websocket, les protocoles de chat, de l'IoT et établir des connexions de longue durée ceci en utilisant une spécification python dite ASGI. [40]

- **Websocket** : Suite à un conflit entre Channels et les clients **MQTT**, empêchant la récupération des données du Broker, nous avons adopté une autre solution un peu fastidieuse, en passant par le websocket et JavaScript. Le Websocket est une technologie évoluée qui permet d'ouvrir un canal de communication bidirectionnelle, entre un navigateur (côté client) et un serveur. Avec cette API, il est possible d'envoyer des messages à un serveur et recevoir ses réponses de manière événementielle, sans avoir à aller consulter le serveur pour obtenir une réponse. [41]
- **Daphne** : est un serveur de protocole HTTP, HTTP2 et Websocket pour ASGI et ASGI-HTTP, développé pour alimenter **Django** Channels. Il soutient la négociation automatique de protocoles ; il n'est pas nécessaire d'avoir un préfixe d'URL, une fois le canal est établi le reste n'a pas d'importance, comme les points de terminaison Websocket par rapport aux points de terminaison HTTP. [42]

Les deux spécifications de base sont destinées à fixer en place l'ensemble des API, par lesquelles ces serveurs interagissent et exécutent le code de l'application. Chaque protocole pris en charge (tel que HTTP), a une sous-spécification qui décrit comment encoder et décoder ce protocole en messages. Les spécifications se limitent à :

- **WSGI**: Web Server Gateway Interface, une spécification destinée à fournir une interface normalisée entre les serveurs Web compatibles synchrone, les Framework et les applications Python. [43]
- **ASGI** : Asynchronous Server Gateway Interface, là où WSGI fournissait une norme synchrone, ASGI tente de préserver une interface d'application simple, tout en fournissant une abstraction qui permet d'envoyer et de recevoir des données à tout moment, et à partir de différents threads ou processus d'application d'une manière asynchrone. [44]

2.2.6 Le transfert des données au niveau de la plateforme :

Le MQTT est un protocole de communication dédié aux applications IoT qui fonctionne selon une notion (Pub/Sub), **Django** utilise le protocole HTTP par default, pour assurer le lien entre le client et les objets, alors une application **Django** a été écrite

pour garantir cette communication plateforme/réseaux d'objets avec le serveur Mosquitto.

Afin de mieux expliquer la relation du broker Mosquitto avec la plateforme. Tout d'abord nous avons les données issues des différents capteurs, qui sont publiés sur le broker Mosquitto selon un Topic défini (dans notre cas, le projet réalisé a été testé seulement sur le réseau local, et à cet effet, deux capteurs virtuel ont été créés pour la simulation T_data,W_data), puis d'un notre côté nous avons les utilisateurs, qui naviguent sur la plateforme, une fois que le client s'inscrit sur un Topic, ce dernier reçoit automatiquement les données qui sont publiés sur ce même Topic, puis ces données sont stockées dans la base de données. Autre un programme réalise des graphes statistiques pour voir l'activité des capteurs.

2.2.7 La Base de Données (BD):

Une base de données est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes et par des utilisateurs différents. Afin d'offrir aux utilisateurs une large manœuvre de création de fichier, modifier, supprimer, éditer..., les différentes données stocker. [45]

Dans notre cas le Framework **Django** utilise par default **sqlite3**, alors nous l'avons adopté comme BD pour notre plateforme, dans chaque projet Django lancé par la commande :

Django-admin startproject « nom du projet »

Deux fichiers **db.sqlite3** et **manage.py** sont créés automatiquement l'un représente la base de données et l'autre est un utilitaire en ligne de commande qui vous permet d'interagir avec le projet Django de différentes façons. [39]

a) SQLite :

Est une bibliothèque logicielle qui implémente un moteur de base de données SQL (SQL est le langage qui nous permet de manipuler les BD) transactionnel autonome, sans serveur,

sans configuration. SQLite3 peut être intégré à Python, à l'aide du module sqlite3, et comme python préfère la simplicité alors c'était la BD intégré par default.

Parmi les avantages de SQLite :

- Pas besoin de la configurer dans notre système.
- Une base de données SQLite complète est stockée dans un seul fichier disque multiplateforme.
- SQLite est autonome, ce qui signifie qu'il n'y a pas de dépendances externes.
- SQLite ne nécessite pas de processus ou de système serveur distinct pour fonctionner.

Si nous venons à explorer les codes écrits pour la plateforme, **Django** définit sa table de données grâce aux fichiers **Models**, les lignes de codes écrites dans ces fichiers définissent la structure des données. Nous trouverons 4 tables définies sous forme de code **Python**, qui donne ce résultat :

Application accounts : grâce à l'utilisation de l'UserCreationForm pour l'inscription vers la plateforme :

ID	User name	Password
01	User01	Yyrdx87xch6QH

Table 2.1 : La table de la BD respective à l'identification du personnel.

Application workspace :

ID	Application name	UUID
01	App01	Po87ckjehm0

Table 2.2 : Table de la BD des applications IoT.

ID	Application	Sensorname	Sensor type	Acutatorname	Acutator type	Sub_topic
----	-------------	------------	-------------	--------------	---------------	-----------

01	App01	Cap164987	humidite	Act149820	Motor	Labo/refroidissem
----	-------	-----------	----------	-----------	-------	-------------------

Table 2.3 : Table de la BD des Caps/Acts.

Application broker :

Device_id	Auto_id	Topic	Pub_info	Payload	Created_date	State
2000	06	Labo/ refroidissem	26	Topic : Labo/refroidissem, Info : 26	29/09/2021	1

Table 2.4 : Table de la BD du client MQTT.

2.2.8 Points faibles dans l'architecture de la plateforme :

La plateforme actuelle présente un certain nombre de points défailants qui n'empêchent pas le fonctionnement général, mais qui pourraient s'avérer fatal par la suite, lors de l'implémentation des modifications.

Critiques :

- Si nous avons des ressources supplémentaires dans l'environnement physique, nous ne sommes pas en mesure d'ajouter des capteurs ou des actionneurs séparément.
- Si un problème survient au niveau d'un capteur ou d'un actionneur, ce dernier affectera les deux, car ils appartiennent à la même table, valable aussi si une redondance de données se produit à l'un ou l'autre, les deux seront également affectés.
- Le coût de la requête est plus élevé, car la sélection faite par lignes, et non pas par colonnes, donc à chaque fois que nous aurons besoin de données sur un capteur, nous chargerons les données de l'actionneur en même temps. Alors que nous n'avons peut-être besoin que d'un seul.
- De plus, la requête est plus lourde, ce qui affecte les performances car elle prend plus de temps à être chargée et traitée.

- Au cas où nous aurions besoin de supprimer un actionneur ou un capteur, nous aurons des lignes avec des colonnes nulles car elles se trouvent dans la même ligne.
- Si par malheur nous voudrions changer le local physique d'un capteur ou actionneur, nous devons le supprimer de la base de données, et le soumettre à nouveau dans le nouvel environnement/local.
- Si les tables étaient normalisées, il sera facile de mettre à jour, d'ajouter tout type d'information ou d'attribut à un capteur / actionneur, de modifier à tout moment sans aucune difficulté ni perte de temps. (Normalisation de la base de données).
- Lorsque les options disponibles (environnement, type local, capteur et actionneur) sont définies dans le code source, cela nous oblige à revenir au code source à chaque fois que nous voulons ajouter plus d'options donc une intervention du développeur est nécessaire. Ce n'est pas optimal car toute erreur de frappe peut facilement déstabiliser toute la plate-forme, en plus de cela l'administrateur de la base de données ne pourra pas le faire lui-même car nous n'avons pas accès au code source.
- L'ajout ou la suppression d'une nouvelle option existante peut être effectué, si la plate-forme fonctionne sans aucun temps d'arrêt, car il n'y a pas de mise à jour du code (pas d'édition de modèles donc aucune migration n'est requise, donc pas de temps d'arrêt).
- Le capteur et l'actionneur qui se trouvent dans la même rangée ont le même ID, cela peut entraîner une redondance.

2.3 Problèmes rencontrés :

La plateforme ne se lance pas avec la commande ordinaire de Django :

```
python manage.py runserver
```

Et l'erreur 500 du serveur Daphne est lié à l'application channels. Seulement en exécutant la commande du serveur Daphne devient accessible:(iotframework est le nom du projet)

```
Daphneiotframework.asgi:application
```

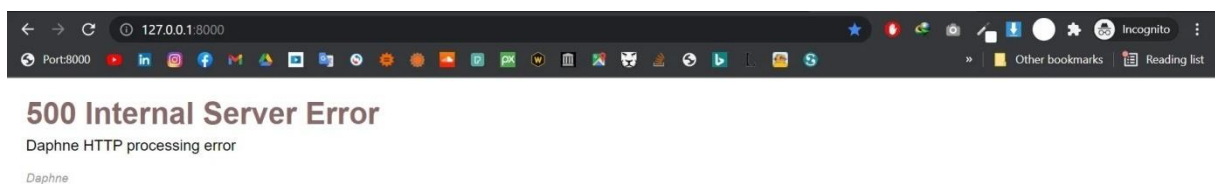


Figure 2.8 : L'erreur lors du lancement avec la commande runserver.

- ➔ Le système de sécurité proposé dans le mémoire en se servant de la configuration du fichier d'installation du broker MQTT, n'est pas fonctionnel.

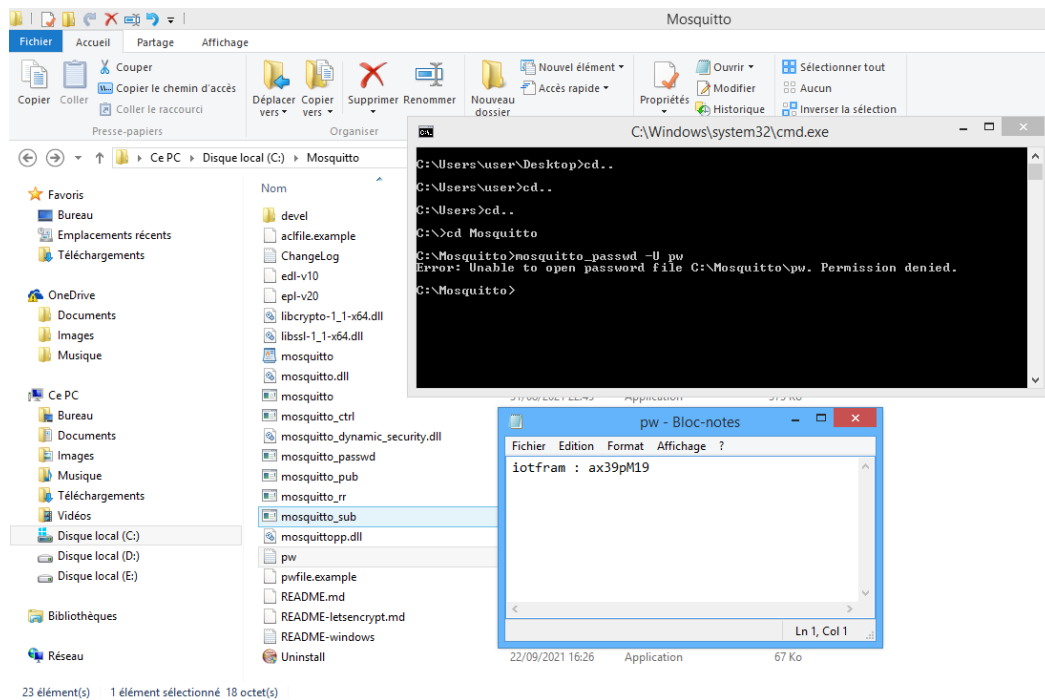


Figure 2.9 : l'erreur de sécurité.

➔ En essayant de faire marcher la plateforme dans le local, le navigateur n'autorise pas cette démarche à un certain point comme l'affichage des graphes liés aux fichiers `temperature_data.php` et `water_data.php`, malgré la disponibilité des fichiers requis, et l'importation du fichier recommande dans le `www` ; du `wamp` server.

2.3.1 La qualité de code :

le Framework Django nous permet d'écrire un code plus ou moins organisé, mais nous devons être méthodiques afin d'obtenir un bon code, parmi les grand problèmes que nous avons rencontré :

- La mauvaise appellation des variables, modèles, views, URLs... etc. Certaines fonctions et variables n'ont pas des noms significatifs, cela nous oblige à revoir les classes et les modèles pères à chaque fois, lorsque nous manipulons ces variables et classes. Cette dernière affecte aussi la compréhension de la structure du code, car il nous est presque impossible de dire, si c'est un modèle principal ou secondaire... etc.

- Django offre la liberté de mettre les URLs et les Views dans n'importe quelle répertoire à l'intérieur du projet, et les réutiliser et les importer dans des scripts de d'autres répertoires à

l'aide des fonction « Include() » pour les URLs et « import » pour les Views et les modèles , mais l'utilisation de cette dernière de façon pas organisée et hiérarchique peut causer des difficultés lors de la manipulation et modifications de ses URLs et Views.

2.3.2 Incompatibilités des modules et packages :

Nous pouvons remarquer directement que nous avons trop des modules et packages mis en jeu dans ce projet, cette diversité peut provoquer des problèmes de compatibilités, si un module ou plus ont une mise à jour surtout, ou si nous utilisons des langages différents (Python, JavaScript...).

Les projets doivent donc inclure un fichier d'exigence (dit requirements.txt par conventions), Il s'agit d'un fichier .txt qui contient une liste de tous les modules avec leurs versions utilisées dans ce projet particulier.

Cela peut garantir le fonctionnement dans d'autres machines et éviter certains problèmes de compatibilité inter-modules, et aider à reproduire un environnement de développement similaire.

Tous ces problèmes et critiques potentiels rendent l'administration de la base de données plus difficile.

Notre nouvelle architecture aidera à prévenir la plupart de ces problèmes, et à réduire la probabilité de survenue d'autres problèmes, cette architecture est facilement évolutive et adaptable.

2.4 Plateforme de gestion des environnements IoT :

2.4.1 Présentation :

La plateforme à laquelle nous aspirons, offre à la fois la possibilité de créer et de contrôler plusieurs environnements en parallèle, mais aussi de mettre à disposition de l'utilisateur, un moyen de personnaliser son environnement, par exemple : ajouter des locaux, des Cap/Act, ... ; contrairement à l'ancienne qui est destiné au contrôler seulement l'application.

C'est pour ces raisons qu'il est nécessaire de modifier et d'optimiser la BD, et permettre à la nouvelle plateforme d'être adaptée au concept des environnements IoT.

Mise à part la nouvelle notion que nous allons intégrer, nous avons au premier lieu inspecté et corrigé quelques parties, qui posait des problèmes au bon fonctionnement de la plateforme, et garder quelques applications que nous avons jugé utiles à l'utilisateur comme l'interprétation des données captées à partir de l'environnement sous forme de graphes.

2.4.2 Conception :

Nous avons établi un schéma conceptuel pour définir une hiérarchie d'étape et de démarche à effectuer dans la plateforme, afin de réaliser la meilleure conception possible, en localisant les étapes à modifier et à rajouter, exemple : nous avons rajouté les deux notions intitulées « Environnement et Local » ; dans la conception occurrente, par contre nous avons gardé quelques étapes que nous avons estimées nécessaires, comme la page de connexion/inscription à la plateforme.

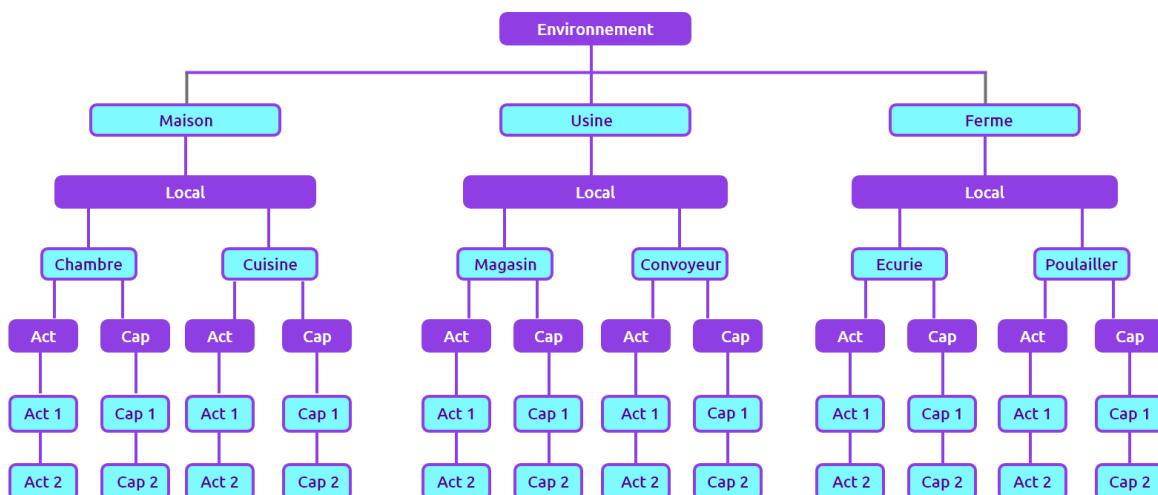


Figure 2.10 : Schéma conceptuelle qui montre les différents étages de l'architecture de la plateforme.

D'abord nous devons définir et introduire quelques concepts liés à la conception des bases de données.

2.4.3 Normalisation de la base de données :

La normalisation est le processus d'organisation efficace des données dans une base de données afin d'éliminer les données redondantes. Ce processus garanti que toutes les données de l'application ayant besoin d'une BD, se présentent et se lisent de la même manière dans tous les enregistrements. En mettant en œuvre la normalisation des données. [46]

La normalisation des données implique l'organisation des champs de données (les colonnes et les tables) d'une base pour s'assurer que leurs dépendances soient correctement appliquées. La « forme normale » fait référence à l'ensemble de règles ou de données de normalisation, et une base de données est dite « normalisée » si elle est exempte ou dégagée de suppression, de mise à jour et d'insertion d'anomalies. [46]

a) Les formes de normalisations :

Première forme normale (1NF) :

Cette forme permet de supprimer toutes les colonnes en double de la table, en créant des tableaux pour les données associées et identification des colonnes de façon unique.

Deuxième forme normale (2NF) :

Satisfaire toutes les exigences de la première forme normale, en plaçant des sous-ensembles de données dans des tables séparées et créer des relations entre les tables à l'aide de clés primaires.

Troisième forme normale (3NF) :

Cela devrait répondre à toutes les exigences de la 2NF. En procédant à la suppression des colonnes qui ne dépendent pas des contraintes d'une clé primaire.

Quatrième forme normale (4NF) :

Répondre à toutes les exigences de la troisième forme normale et de s'assurer de ne pas avoir de dépendances à valeurs multiples.

B) Évolutivité et élasticité :

L'évolutivité : fait référence à la capacité d'un système, un réseau, un processus ou d'une base de données à gérer une quantité croissante de travail, ou son potentiel à effectuer un travail plus complet tout en augmentant la puissance de traitement pour s'adapter à la croissance. Un système est dit évolutif s'il peut augmenter sa charge de travail et son débit lorsque des ressources supplémentaires sont ajoutées. [47]

L'élasticité : est le degré auquel un système peut s'adapter aux changements de charge de travail, en allouant et en supprimant des ressources à la demande, de sorte qu'à tout moment les ressources disponibles correspondent le plus possible à la demande actuelle. [47]

2.4.4 L'architecture proposée :

Pour la nouvelle architecture nous avons ajouté quelques tables, certaines dans le but d'intégrer la nouvelle notion d'environnement comme les « Environnements, Locaux », et a d'autre à qui nous avons seulement changé la configuration pour des raisons d'optimisation comme la séparation des types des tables principales, en leur attribuant de nouveaux modèles. Voici la figure qui montre les changements effectués :

UTILISATEUR

ID user	Username	Password
---------	----------	----------

ENVIRONNEMENT

ID env	Name	Creation Date	FK. ID User	FK. ID Setup Env
--------	------	---------------	-------------	------------------

TYPES DES ENVIRONNEMENTS

ID	Name	Creation Date
----	------	---------------

LOCAL

ID local	Name	Creation Date	FK. ID Env	FK. ID Setup Local
----------	------	---------------	------------	--------------------

TYPES DES LOCAUX

ID	Name	Creation Date
----	------	---------------

CAPTEUR

ID sen	Name	Type	FK. ID Local	Sub Topic
--------	------	------	--------------	-----------

TYPES DES CAPTEURS ET ACTIONNEURS

ID	Name
----	------

ACTIONNEUR

ID act	Name	Type	FK. ID Local	Sub Topic
--------	------	------	--------------	-----------

TABLE TERNAIRE

ID Sen / Act	ID Local
--------------	----------

MQTT CLIENT

Device ID	Auto ID	Topic	Pub info	Payload	State
-----------	---------	-------	----------	---------	-------

Figure 2.11 : L'architecture des tables de la BD.

2.4.5 Bootstrap :

a) Description :

Bootstrap est un Framework cote client (front-end, design) open source, basé sur du CSS, qui peut aussi inclure du HTML et du JavaScript. Destiné pour construire les interfaces de site Web compatible avec tous les appareils.

Ce Framework est défini par un ensemble de fichiers CSS et JavaScript qui intègrent des règles prédéfinies, auxquelles nous pouvons faire appel sous forme de classes dans notre code afin d'appliquer un ensemble de styles sur certains éléments HTML. [48]



Figure 2.12 : Logo du Framework Bootstrap. [49]

b) les avantages du Bootstrap :

Ce Framework offre 3 avantages essentiels pour ces utilisateurs :

- Construit sur la base de HTML, CSS et JavaScript, ce qui facilite son utilisation.
- Compatible avec tous les différents dispositifs, une fois maîtrisé, permet d'économiser pas mal de temps.
- Compatibles avec tous les navigateurs (Chrome, Firefox, Opera, Safari...).

En résumé, avec Bootstrap nous allons pouvoir créer des designs complexes de manière relativement simple, et donner l'avantage d'accéder à la plateforme de n'importe quels appareils et de l'endroit que nous souhaitons. [48]

Conclusion :

Dans ce chapitre, nous avons présenté l'architecture de la plate-forme précédente, sa conception et les outils utilisés pour la construire et nous avons expliqué son fonctionnement.

Ensuite, nous avons mentionné certains problèmes survenus lorsque de l'exécution de la plateforme.

Nous avons exposé les nouvelles fonctionnalités que nous avons ajoutées et présenté la nouvelle architecture de la base de données qui aidera à intégrer facilement les nouvelles fonctionnalités et peut être évolutive et stable en cas d'améliorations et de modifications futures.

Sans oublier la nouvelle interface utilisateur qui a l'air plus propre et uniforme et construite à l'aide d'un Framework et de technologies récentes.

CHAPITRE 3:

Réalisation de la plateforme de gestion des environnements

IoT.

Réalisation de la plateforme de gestion des environnements IoT

Après avoir modifier, ajouter, créer, manipuler, coder, tester et simuler, nous sommes enfin arriver à apporter du nouveau à l'ancienne plateforme, même si ça ne représente qu'une infime partie de ce que nous voulons réaliser, et apporter à notre communauté.

3.1 Les interfaces de la plateforme :

3.1.1 Page d'accueil :

C'est la première page que nous apercevons lorsque nous plaçons l'URL équivalant au site de la plateforme, ou nous aurons 3 boutons de redirection essentiels vers, l'inscription, la connexion et démarrage direct.

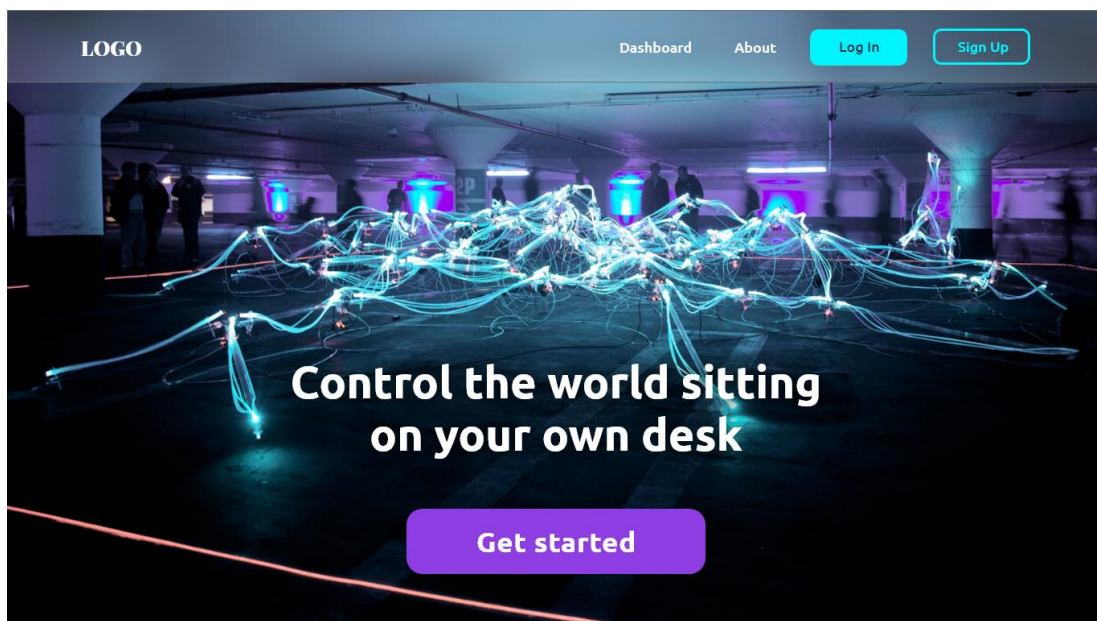


Figure 3.1 : La page d'accueil de plateforme.

3.1.2 Page d'inscription ou de création de compte :

Cette figure montre l'interface permettant l'inscription afin d'avoir l'accès à toutes les fonctionnalités offertes par la plateforme.

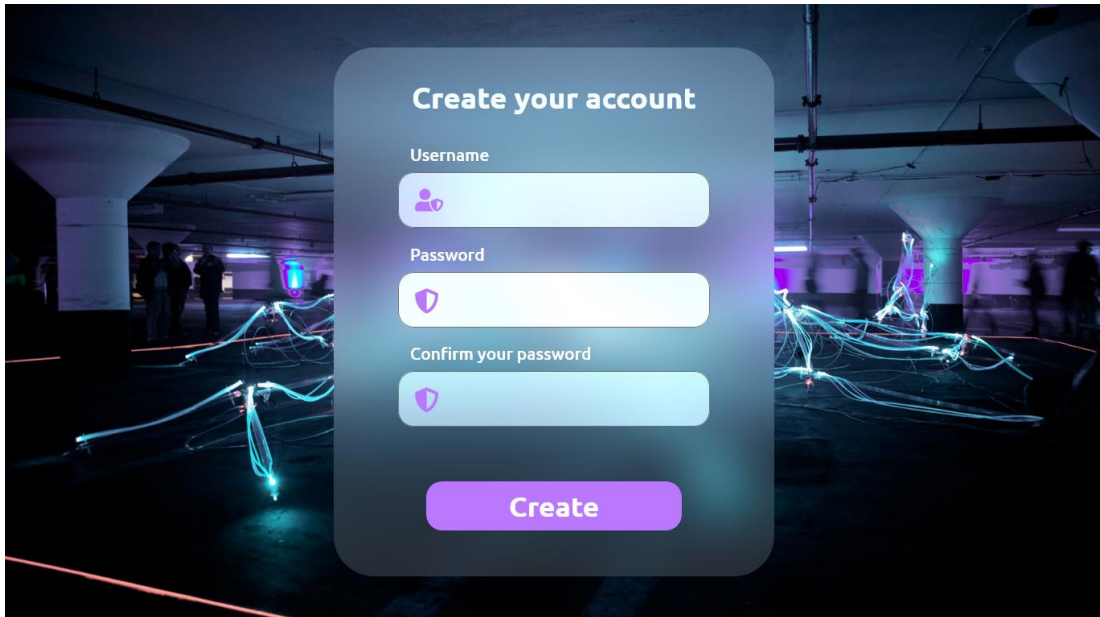


Figure 3.2 : Interface d'inscription.

3.1.3 Page de connexion :

Une fois inscrit, cette page permet à l'utilisateur d'être authentifié permet l'accès à son environnement de gestion.

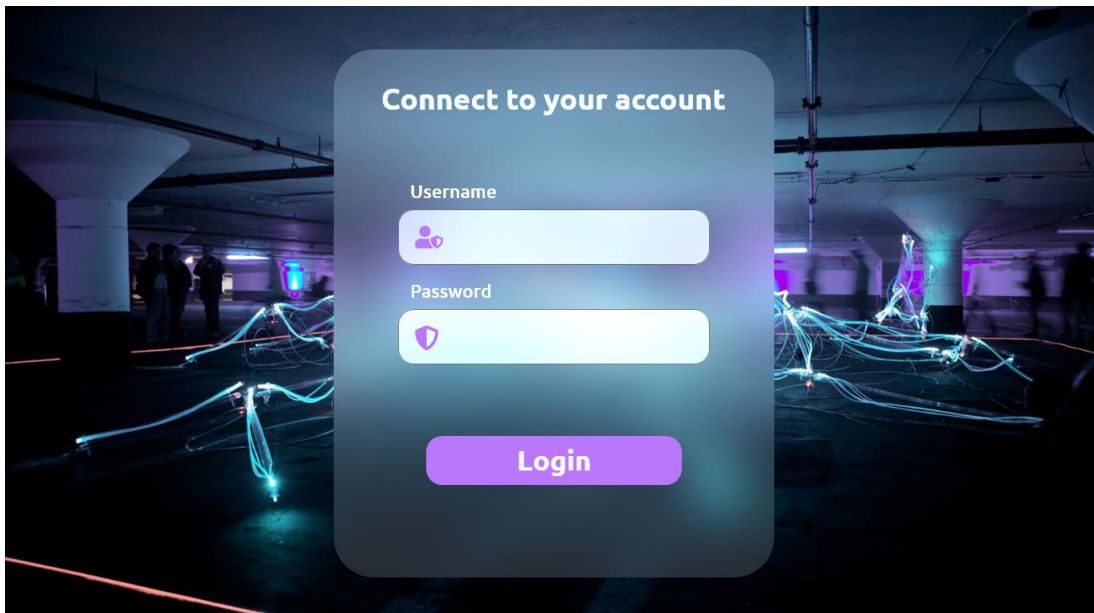


Figure 3.3 : l'interface de connexion.

3.1.4 Page de gestion des environnements :

A ce stade nous sommes dans l'espace personnel de l'utilisateur et tous les éléments qui sont présent ici, sont là afin de d'aider de donner à l'utilisateur le contrôle qu'il souhaite sur ces environnements crée, éditer ou ouvrir.

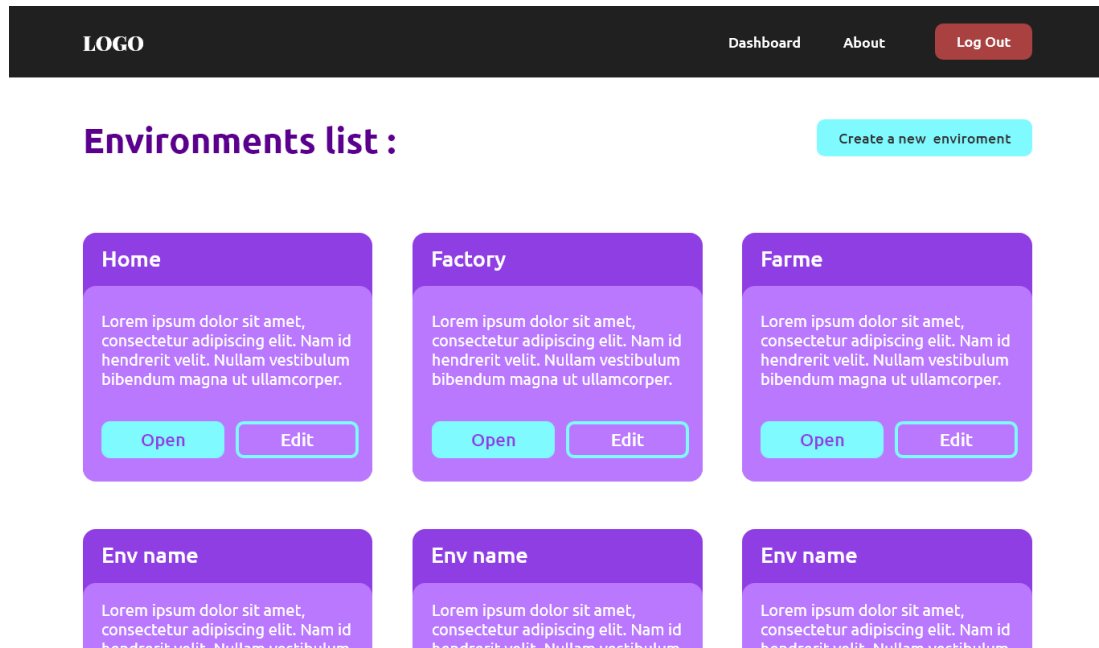


Figure 3.4 : interface de gestion des environnements.

3.1.5 Page de gestion des locaux :

Une fois que nous avons créé un environnement ou plus, nous avons par la suite accès à cette interface afin d'ajouter la constitution de notre environnement en locaux selon ses besoins, puis nous avons le choix de créer, d'éditer, supprimer ou même consulter les détails des locaux déjà existant.

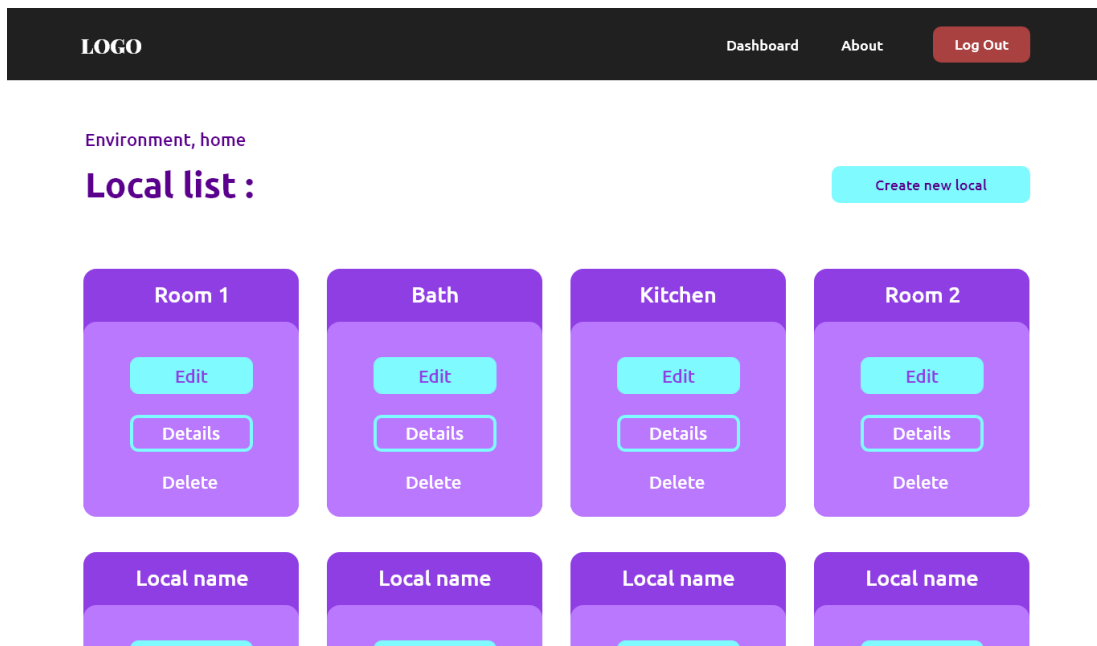


Figure 3.5 : Interface de gestion de locaux.

3.1.6 Page de gestion de Cap/Act :

Cette page représente le cœur de la hiérarchie de gestion de notre plateforme. C'est à ce niveau que nous pouvons choisir les capteurs et actionneurs spécifiques dont nous avons besoin, et procéder à l'affectation aux différents Topics, supprimer et ainsi éditer.

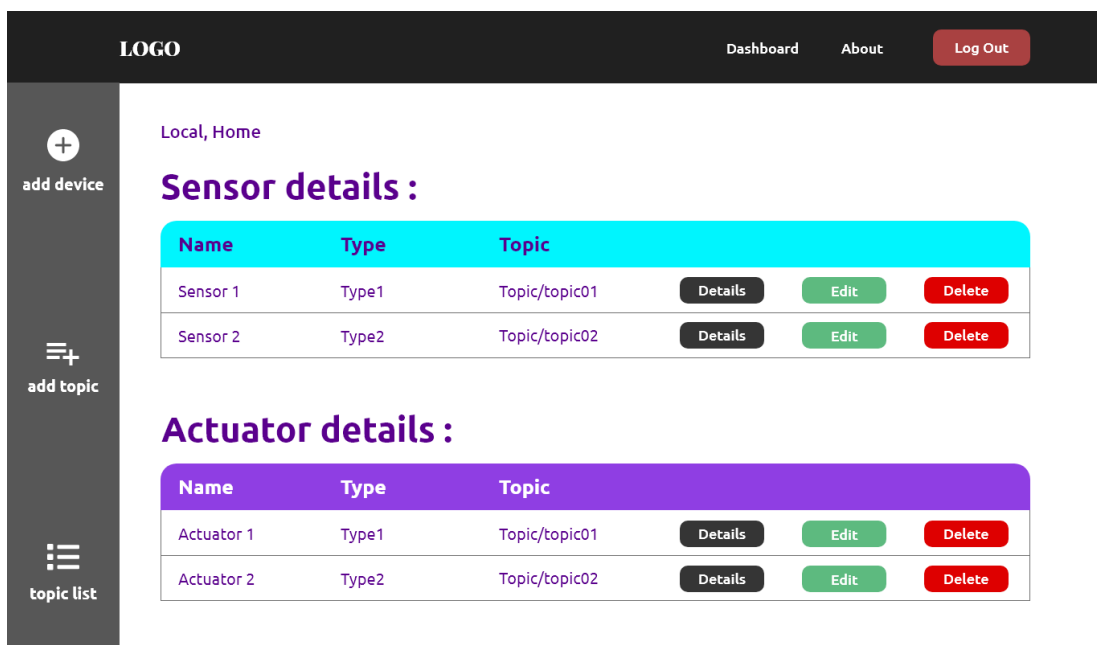


Figure 3.6 : Interface de gestion des Cap/Act.

3.2 Domaines d'intégration :

L'IoT est définie selon le mode d'emploi et la technologie adopté, l'IoT peut être intégrée partout (Santé, Industrie, Agriculture...). Pour mettre en œuvre un tel projet d'IoT en Algérie, le premier pas qui doit être effectué d'abord, est l'installation des outils pour les communications nécessaires entre les réseaux d'objets et les plateformes (ex : mettre un serveur dédié seulement pour les connexions des objets), assurer une sécurité plus performante, puis l'implémentation des réseaux des objets. La plateforme que nous avons conçus peut être utilisé avec tous types d'objets, tant qu'ils sont capables de transmettre des données.

Conclusion :

Nous avons présenté les différentes interfaces réalisées au sein de notre plateforme, en montrant leur utilisation.

Conclusion générale

L'internet des objets est un domaine très vaste qui ne cesse de s'agrandir, une définition propre et stable reste à trouver, pour cela nous voulons aussi participer à ce défi en donnant notre propre vision sur ce domaine, l'IoT est un domaine qui se définit selon le domaine d'application, les technologies utilisées et l'intérêt de son utilisation.

La problématique traitée dans ce mémoire, est de montrer comment donner naissance à une application qui permettra aux gens d'avoir un accès illimité vers leur environnement, à distance et en temps réel.

Pour réaliser ce travail nous nous sommes inspirées d'une plateforme déjà réalisée par un groupe d'étudiants [38], qui avaient mis en œuvre une plateforme générique pour la gestion des applications. Nous avons voulu améliorer ce concept en le modifiant pour donner naissance à une gestion des environnements IoT, une gestion multiple et en parallèle.

Pour réaliser ce travail, nous avons utilisé deux Framework, Django basé sur le langage Python pour le BACK-END et Bootstrap basé sur CSS et HTML pour le FRONT-END.

Mais, suite à quelques empêchements liés surtout à la pandémie du COVID19, et au temps qui nous est impartie, nous n'avons pas pu finaliser ce projet.

Malgré cela, le travail réalisé nous a servi de grandes leçons théoriques, pratiques et morales, car ça nous a permis de faire un pas dans le milieu professionnel et aider à affronter les différents obstacles que nous avons rencontrés, et de trouver des solutions pour bien agir face à ces défis.

Nous avons aussi appris à manipuler les bases de données, comment régler les différentes redondances, comment détecter les erreurs, comment organiser nos projets afin qu'ils aboutissent à une réussite certaine.

Nous avons surtout acquis des connaissances dans un domaine très prometteur, avec une expérience enrichissante pleine de suspense et d'acharnement, et nous allons continuer ce travail jusqu'à ce qu'il soit en fin prêt à être exploitable dans la pratique.

Bibliographie

- [1] <https://byothe.fr/2018/10/histoire-internet-des-objets-iot/>. Consulté le 02/07/2021.
- [2] Pierre-Jean Benghozi, Sylvain Bureau, Françoise Massit-Folléa: L'Internet des Objets Quels enjeux pour les Européens?, Octobre 2008.
- [3] LIVRE BLANC, PREPARER LA REVOLUTION DE L'INTERNET DES OBJETS, DOCUMENT N°1 – UNE CARTOGRAPHIE DES ENJEUX
- [4] Nasreddine Bouhaï, Imad Saleh: Internet des objets: Evolutions et innovations, London ISTE editions, 2017.
https://books.google.com/books/about/Internet_des_objets.html?id=E71XDwAAQB_AJ&printsec=frontcover&source=kp_read_button&hl=fr, consulté le 01/09/2021.
- [5] NicaLatto, INTERNET DES OBJETS: Qu'est-ce que l'Internet des objets (IoT) ?, 13 nov 2019, <https://www.avast.com/fr-fr/c-what-is-the-internet-of-things>, consulté le 15/07/2021.
- [6] M. Aashish Mehra, Marché de l'IoT industriel Date de publication : juil. 2021 | Code de rapport : SE 2909, 214 pages.
<https://www.marketsandmarkets.com/PressReleases/industrial-internet-of-things.asp>, consulté le 14/09/2021.
- [7] Statista Research Department, Internet des objets : nombre d'appareils connectés dans le monde 2015-2025, 27 nov 2016.
- [8] Cédric Gossart, « Les enjeux sociétaux des objets connectés », Terminal [En ligne], 120 | 2017, <http://journals.openedition.org/terminal/1500> ; DOI : <https://doi.org/10.4000/terminal.1500>, mis en ligne le 03 mai 2017, consulté le 10/09/2021.

- [9] Pauline Folcher, Sarah Mussol, Anne-Sophie Cases. Loft Study : que pensent les Millennials des objets connectés?, 16ème colloque sur le Marketing digital, Sep 2017, Paris, France.
- [10] Inès Chouk, Zied Mani, LES OBJETS CONNECTÉS PEUVENT-ILS SUSCITER UNE RÉSISTANCE DE LA PART DES CONSOMMATEURS ? UNE ÉTUDE NETNOGRAPHIQUE, EMS Editions, 2016/4 N° 84 | pages 19 à 41
- [11] Philippe Rioux, La revolution des objetsconnectés, Technomedia, 5/1/2014.
<http://www.technomedia.org/2014/01/la-revolution-des-objets-connectes.html>,
consulté le 10/09/2021.
- [12] <https://www.connectwave.fr/techno-appli-iot/iot/reseaux-et-infrastructures-iot/>,
consulté le 10/09/2021.
- [13] HIDOUCI FARID, Réalisation et implémentation d'une application a base de protocole MQTT dans IoT, Mémoire de master académique, Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie Département d'informatique, Université Mohamed Khider – BISKRA, 07 juillet 2019, 144 pages.
- [14] Maxime DECHANY, L'impact de l'internet des objets sur le futur de la logistique et du transport: cas du transport routier, Memoire de Master, HEC-Ecole de gestion de l'Université de Liège, Belgique, 2018, 129 pages.
- [15] Sameh Ben Fredj, L'Internet des Objets : 101, article,
<https://blog.engineering.publicissapient.fr/2015/12/02/linternet-des-objets-101/>,
publié le 02/12/2015, consulté 15/09/2021.
- [16] MEKRIOU RYMA, MAZARI WALID, Introduction à l'internet de l'objet et réalisation d'un système domotique, Memoire de Master en télécommunication, Faculté de Technologie Département ATE, Université A/Mira de Bejaïa, 2016, 53pages.
- [17] IAN J.H REYNOLDS, INTERNET DES OBJETS IOT Architecture,
<https://www.zibtek.com/blog/iot-architecture/>, article, publié le 23 juillet 2020,
consulté le 2 sept 2021.

- [18] https://www.researchgate.net/figure/Illustration-of-the-IoT-Reference-Model-by-Cisco-1_fig1_336564357, Consulté le 10/09/2021.
- [19] White Paper, The Internet of Things Reference Model, 4 june 2014 Cisco and/or its affiliates, 12 pages.
http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf, consulté le 11/09/2021.
- [20] Satyabrata Jena, Architecture de l'Internet des objets (IoT), article, 25 juin 2020, <https://www.geeksforgeeks.org/architecture-of-internet-of-things-iot/?ref=rp>, consulté le 17/09/2021.
- [21] SomayyaMadakam, R. Ramaswamy, Siddharth Tripathi, Internet of Things (IoT): A Literature Review, Journal of Computer and Communications, IT Applications Group, National Institute of Industrial Engineering (NITIE), Vihar Lake, Mumbai, India, May 2015, 3, pages 164-173.
- [22] Sharon Shea, definition machine to machine M2M, article, août 2019.
<https://internetofthingsagenda.techtarget.com/definition/machine-to-machine-M2M>, consulté le 17/09/2021.
- [24] TIZZAOUI YOUVA, Internet des Objets « IoT » Application : Industrie 4.0, Mémoire de Master, Faculté de Technologie Département Génie Électrique Option Télécommunication, Université Abderrahmane Mira de Bejaïa, 2017, 73 pages.
- [25] Yassir Sennoun, Internet des Objets : Quels protocoles applicatifs utiliser ?, Article, publié le 16/04/2018,
<https://blog.engineering.publicissapient.fr/2018/04/16/internet-des-objets-quels-protocoles-applicatifs-utiliser-1-2/>, consulte 18/09/2021.
- [26] Philippe Truillet, MQTT : MQ Telemetry/Transport Un protocole pour l'IoT, article, Université de Toulouse III PAUL SABATIER, Juillet 2018, 3 pages.

- [27] Indira Muhic, Migdat I. Hodzic, Internet of Things: Current Technological Review Faculty of Engineering and Natural Sciences, International University of Sarajevo, Bosnie-Herzégovine, 2014, 8 pages.
- [28] HAMDJ Wissam, Développement d'un système de gestion d'objets connectés, Memoire de Master, Faculté : sciences et sciences appliquées Département Génie Electrique, Université Larbi Ben M'hidi -Oum El Bouaghi, juin 2018, 75 pages.
- [29] <https://www.lexpertfenetre.fr/wp-content/uploads/2015/11/shema-maison-connectee.jpg> , consulté le 20/09/2021.
- [30] <https://www.swissdigitalhealth.com/events/uberisation-de-la-sante-fantasme-ou-realite/>, consulté le 20/09/2021.
- [31] Livre blanc ARCEP, PRÉPARER LA RÉVOLUTION DE L'INTERNET DES OBJETS, Document n°2 – Inventer une régulation pro-innovation, 19 pages, 7 novembre 2016. Le site dédié à la démarche IoT de l'Arcep : <http://www.arcep.fr/iot/>. arcep: Autorite de Regulation des Communications Electroniques et des Postes.
- [32] Gérard Swinnen, Apprendre à programmer avec python 3, édition Eyrolles(Paris) 2012, 473 pages.
- [33] <https://ams-training.com/formation/programmation-objet-langage-python/>, consulté le 29/09/2021.
- [34] <https://www.skysilk.com/blog/2017/how-to-make-a-blog-with-django/>, consulté le 29/09/2021
- [35] W3Schools, Python Tutorial. https://www.w3schools.com/python/python_intro.asp, consulté le 28/09/2021.
- [36] La rédaction JDN, Framework ou infrastructure logicielle : définition et traduction, article, 10/12/2017,. <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203355-framework/> , consulté le 02/10/2021.

- [37] Mickaël Andrieu, AymerikDiebold, Structurez le traitement de votre transaction avec l'architecture MVC, <https://openclassrooms.com/fr/courses/6031956-creez-une-application-web-en-php-de-qualite-professionnelle/6107761-structurez-le-traitement-de-votre-transaction-avec-larchitecture-mvc> . Mise à jour le 20/01/2021, consulté le 02/10/2021.
- [38] Daouadji Manel , El aichi Asma, Plateforme IOT générique adaptable, mémoire de Maste, Université SAAD DAHLAB de BLIDA, Faculté de Technologie, Département d'Électronique, Filière Télécommunication Spécialité Réseaux & Télécoms, 2020, 102 pages.
- [39] documentation django site officiel. [Django documentation | Django documentation | Django \(djangoproject.com\)](#) , consulté le 02/10/2021.
- [40] <https://channels.readthedocs.io/en/stable/introduction.html#turtles-all-the-way-down>, consulté le 02/10/2021.
- [41] MDN Web Docs, WebSockets, Référence Web API. https://developer.mozilla.org/fr/docs/Web/API/WebSockets_API , consulté le 02/10/2021.
- [42] <https://pypi.org/project/daphne/> , consulté le 02/10/2021.
- [43] GILBERT MOISIO, Python WSGI ou ASGI, 27 JANVIER 2020, <https://moisio.fr/2020/01/27/python-wsgi-ou-asgi/>, consulté le 04/10/2021.
- [44] <https://asgi.readthedocs.io/en/latest/specs/main.html>, consulté le 04/10/2021.
- [45] Jean-François Pillou, Bases de données - Introduction, article, jeudi 12 avril 2018. <https://www.commentcamarche.net/contents/104-bases-de-donnees-introduction#:~:text=Une%20base%20de%20donn%C3%A9es%20%28son%20abr%C3%A9viation%20est%20BD%2C,utilis%C3%A9es%20par%20des%20programmes%2C%20par%20des%20utilisateurs%20diff%C3%A9rents>, consulté le 04/10/2021.

- [46] <https://www.talend.com/resources/what-is-data-redundancy/>, consulté le 04/10/2021.
- [47] Craig S. Mullins, Zone de base de données, Qu'entendons-nous par évolutivité des bases de données ?, févr. 15, 2018, <https://dzone.com/articles/what-do-we-mean-by-database-scalability>, consulté le 05/10/2021.
- [48] RIAGH HAJJI, Presentation de bootstrap, article, 27 fev 2019, <https://apcpedagogie.com/presentation-de-bootstrap/>, consulté le 04 oct 2021.
- [49] <https://comidoc.net/udemy/bootstrap-5-the-complete-guide>, consulté le 05/10/2021.
- [50] Hend Ben Hadji, Les fondamentaux de l'IoT, Présentation, IT Spécialiste, , Directrice au Centre d'Etudes et de Recherche des Télécommunications (CERT), Tunisie, 24/08/2020. 586 pages.