

TABLE DES MATIERS

RÉSUMÉ.....	i
REMERCIEMENTS.....	ii
TABLE DES MATIERS.....	iii
TABLE DES FIGURES.....	vi
TABLE DES TABLEAUX.....	vii
TABLE DES ALGORITHMES.....	viii
CHAPITRE 1 : INTRODUCTION	1
1.1 Exemple illustratif	2
1.2 Problématique	2
1.3 Modèle collaboratif de partage des données	5
1.4 Hypothèse	5
1.5 Motivations	6
1.6 Objectifs et Contributions	6
1.7 Méthodologie	7
1.8 Plan de mémoire	8
CHAPITRE 2 : CONCEPTS DE BASE	10
2.1 Services Web	10
2.1.1 Définition des services Web	11
2.1.2 Architecture du service Web	11
2.2 Intergiciel	13
2.2.1 Définition de L'intergiciel	14
2.2.2 Utilité de l'intergiciel	14
2.3 Internet of Things	15
2.3.1 Définition de l'IoT	16
2.3.2 Fonctionnement de l'IoT	15
2.3.3 Domaines d'application de l'IoT	17
2.4. Web of Things (WoT)	19
2.4.1 Définition du WoT	20
2.4.2 Spécifications de WoT	21
2.5 Conclusion	22
CHAPITRE 3 : L'ÉTAT DE L'ART	23

3.1 Découverte de services	23
3.2 Sélection de services	26
3.3 Échange d'informations.....	28
3.4 Internet of Things.....	30
3.5 Web of Things.....	33
3.6 Modèles collaboratifs de partage de données	35
3.6.1 IoT et les modèles collaboratifs.....	35
3.6.2 WoT et les modèles collaboratifs	36
3.6.3 Synthèse.....	32
3.7 Conclusion.....	40
CHAPITRE 4 : CADRE THÉORIQUE	41
4.1 Modèle collaboratif	41
4.2 Phase de découverte	44
4.2.1 Demande de service.....	44
4.2.2 Réponse de la demande.....	52
4.2.3 Contrôle d'accès	53
4.3 Phase de sélection	55
4.3.1 Présélection	55
4.3.2 Sélection	57
4.4 Phase d'échange d'information.....	59
4.5 Conclusion.....	62
CHAPITRE 5 : IMPLÉMENTATION D'UN PROTOTYPE ET TESTS	64
5.1 Domain Specific Language: IoTCollab.....	64
5.2 Expérimentation en laboratoire.....	67
5.2.1 Projet du capteur connecté.....	68
5.2.2 Projet du serveur local.....	77
5.3 Expérimentation virtuel.....	77
5.3.1 Outils de développement	78
5.3.2 Prototype.....	79
5.4 Conclusion et résultats obtenus.....	84
CHAPITRE 6 : CONCLUSION	86
6.1 Contribution	86
6.2 Perspectives.....	88
6.2.1 Limites.....	88

6.2.2 Travaux futurs	88
BIBLIOGRAPHIE	90

TABLE DES FIGURES

Figure 2.1 : Architecture classique des services Web [16]	12
Figure 2.2 : Architecture Middleware [19]	14
Figure 2.3 : Les domaines d'Internet of Things [26]	19
Figure 3.1 : Le processus de contour de la méthode de sélection proposée[39]	28
Figure 4.1 : Comparaison entre un service satisfait directement et indirectement	43
Figure 4.2 : Demande de type actif	45
Figure 4.3 : Demande passive	46
Figure 4.4 : Exemple sur la simulation des trajets	52
Figure 4.5 : Découverte de service par propagation	54
Figure 4.6 : Pair à Pair.....	60
Figure 4.7 L'échange direct.....	60
Figure 4.8 : L'échange indirect.....	61
Figure 4.9 : Déroulement de notre modèle.....	62
Figure 5.1 : Une vue partielle de la grammaire de l'IoTCollab[64]	66
Figure 5.2 : Une vue partielle d'un système de partage de données de météorologie écrit utilisant le langage IoTCollab[64]	67
Figure 5.3 : Image d'un Spark Core tirée du kick starter du projet [67].....	69
Figure 5.4: Interface de CoolTerm sous Windows	71
Figure 5.5 : Identifiant du SparkCore	72
Figure 5.6 : Connexion du SparkCore à internet.....	73
Figure 5.7 : Identification du SparkCore au SparkCloud.....	73
Figure 5.8 : Contrôle des différents pins du Spark Core [32]	74
Figure 5.9 : DEL activée et DEL désactivée.....	75
Figure 5.10 : Montage du capteur et branchement des fils	76
Figure 5.11 : Résultats du capteur DHT22.....	77
Figure 5.12 : Interface principale de notre application	80
Figure 5.13 : Interface de localisation Google Maps.	81
Figure 5.14 : Profile utilisateur	82
Figure 5.15 : Demande de service.....	83
Figure 5.16 : Demandes reçues	84

TABLE DES TABLEAUX

Tableau 3.1 : Tableau comparatif des articles.....	38
Tableau 4.1 : Matrice de services de l'objet O1.....	51

TABLE DES ALGORITHMES

Algorithme 4.1 :Processus de la requête du service.....	49
Algorithme 4.2 : Procédure de Réponse à une Demande de Service.....	53

CHAPITRE 1

INTRODUCTION

L'informatisation de plus en plus généralisée des activités humaines représente un effort continu qui a pris naissance avec l'avènement des ordinateurs. L'homme a de plus en plus besoin d'avoir recours aux machines, aux robots et aux objets capables de fournir des services avec peu, voire sans aucune intervention humaine. Les ordinateurs, les appareils électroniques et surtout l'informatique et l'Internet deviennent de plus en plus une nécessité de la vie moderne.

Au fil du temps, on a intégré l'ordinateur dans différents objets de notre vie quotidienne. De plus, avec le Web, ces objets peuvent se connecter et communiquer entre eux. Grâce à cette émergence du Web, les machines liées à l'Internet sont conçues pour exécuter, puis penser, et aujourd'hui, elles apprennent à percevoir, sentir et réagir. C'est dans ce contexte que nous orientons notre recherche sur le concept de lier les objets à l'Internet, connu de nos jours comme « Internet of Things¹ » (IoT).

Nous entendons de plus en plus parler de l'IoT [1] [2] [3]. En effet, en raison de ses applications, l'IoT a retenu beaucoup d'attention autant dans le milieu universitaire que dans l'industrie. Dans notre travail de recherche, nous voulons apporter une contribution à

¹Nous utilisons tout le long de notre mémoire l'abréviation IoT pour « Internet of Things », qui se traduit en français par l'Internet des objets

cet égard en proposant une solution aux problèmes rencontrés lorsque les divers objets dans le monde de l'IoT interconnectent et collaborent.

Ce premier chapitre introduit le contexte et la problématique de la recherche proposée. Vous y trouverez également la motivation et l'objectif de notre contribution. Finalement, nous y décrivons brièvement l'organisation de ce mémoire.

1.1 Exemple illustratif

À titre d'illustration, supposons que nous cherchions à développer une application Web et mobile pour offrir des services météo collaboratifs. Ainsi, il semble pertinent de bénéficier du concept de l'Internet of Things (IoT) pour récolter les données météorologiques dans différents endroits géographiques. Nous nous limitons, dans un premier temps, à la récolte des températures à intervalles réguliers.

Dans notre exemple, l'information recherchée exige à la fois l'interconnexion et la communication entre l'ensemble de thermomètres installés à différents endroits. Chaque thermomètre est considéré comme un objet qui représentera une ville. Ces objets sont connectés entre eux à travers l'Internet, chacun pouvant demander un service de l'autre, et pouvant à son tour lui offrir un service. De ce fait, il y a un échange de données entre les différents objets. Ainsi, un objet qui ne détient pas une information demandée par l'utilisateur peut solliciter les autres objets avec lesquels il est connecté pour demander l'information voulue. Grâce à ce mécanisme, l'utilisateur peut avoir l'information demandée, même si le service interrogé ne détient pas cette information dans sa base de données. C'est ainsi que notre exemple de service météo participatif et collaboratif peut devenir une réalité.

1.2 Problématique

Pour répondre au besoin de l'utilisateur qui cherche à obtenir le bon service au bon moment et au bon endroit, on pense qu'il existe plusieurs défis à la mise en œuvre réussie de l'Internet of Things (IoT). En tant qu'informaticiens, notre souci est d'arriver à combiner le monde physique et le monde virtuel, autrement dit, de travailler sur la standardisation de ce nouveau système en développant de nouvelles applications et de nouveaux environnements ainsi que de nouveaux protocoles de communication. Par exemple, Nawaf *et al.* [4] ont proposé un système qui permet l'identification et la localisation des objets en temps réel en utilisant : • le système RFID (identification par radiofréquence), une technologie d'identification automatique utilisée dans l'enregistrement et la réception de données par l'intermédiaire de dispositifs électroniques, • le protocole ZigBee, une technologie de communication du réseau de capteurs sans fil (Wireless Sensor Network) à courte distance, à faible vitesse et à faible consommation d'énergie, et • le *Cloud Computing*. Mais, la zone de couverture de leur système est très limitée géographiquement, et de plus, leur approche utilise toujours le principe client-serveur qui présente des failles connues comme la saturation, la déconnexion et autres, ce qui met en péril tout le système. Par ailleurs, Angulo Lopez et Jimenez Perez [5] ont proposé un cadre d'applications basé sur un système multi-agents pour simplifier le développement et l'intégration d'applications dans l'IoT, ce qui rend le partage des données et des services possible dans un environnement distribué.

Après l'analyse de ces travaux et de bien d'autres, on observe que les recherches effectuées jusqu'à maintenant sont très modestes face à une présence de plus en plus forte de données et de services qui ne sont pas exploités à leur plein potentiel. Comment peut-

on alors créer un système de collaboration et de partage entre ces données et ces services dans un environnement bien adapté et bien encadré au sein de l'IoT?

Ce travail de recherche vise donc à concevoir et implémenter un modèle collaboratif dans l'IoT. L'idée derrière ce modèle est d'apporter, à l'aide de services Web, une solution liée au partage et à la collaboration des données dans l'IoT. La solution apportée se répartit en différentes phases, à savoir : 1) la découverte de services, 2) la sélection de service, et 3) le mécanisme d'échange de données entre les objets. Dans ce qui suit, nous détaillerons les trois phases.

La première, la découverte de services, permet de déterminer une liste de services qui correspondent aux critères d'une requête envoyée par le demandeur et qui tiennent compte des différentes contraintes de l'IoT (capacités de calcul, de stockage et de communication). En nous basant sur notre exemple illustratif présenté dans la section 1.1, nous supposons qu'un utilisateur à Montréal pourrait demander un service (par ex. : les informations météorologique à Chicoutimi) en lançant une requête à travers le réseau de la province vers tous les utilisateurs connectés, afin de recevoir la liste des offres répondant à cette demande. La question qui se pose alors est la suivante : comment déterminer la liste des services les mieux adaptés à la requête du demandeur, en prenant en compte le cas d'un utilisateur qui ne peut pas offrir le service, mais peut diriger le demandeur vers d'autres utilisateurs avec lesquels il est connecté pour demander le service voulu ?

La deuxième phase consiste à sélectionner un ou plusieurs services qui répondent le mieux aux requêtes de l'utilisateur.

Finalement, la troisième phase sert à établir la communication et l'échange de données entre les objets dans notre nouvelle vision de l'IoT. Ceci nous mène à penser à des solutions plus adaptées au partage et à la collaboration. Pour atteindre cette finalité, nous nous inspirerons des différents modes d'échange de données déjà utilisés sur le Web. Ces derniers vont être plus détaillés dans le chapitre 4.

1.3 Modèle collaboratif de partage de données

La définition que nous pouvons retenir de l'Internet of Things (IoT) consiste principalement à connecter des objets physiques à l'Internet, ce qui permet d'échanger des données entre le monde physique et le monde virtuel via des technologies d'information et de communication.

La naissance du phénomène de l'IoT nous ouvre la voie pour développer des applications et des modèles de collaboration et de partage de données qui n'étaient pas possibles auparavant. La question de collaboration et de partage de données dans le monde de l'IoT a fait l'objet de plusieurs recherches: Guinard et Trifa [6] se sont concentrés sur l'intégration des dispositifs du monde réel sur le Web, ce qui leur permet d'être facilement combinés avec d'autres ressources virtuelles et physiques. De leur côté, Spiess *et al.* [7] se sont dirigés vers l'intégration effective de l'IoT dans les services des entreprises. Enfin, Han et Zhang [8] ont travaillé sur la combinaison IoT et *Cloud Computing* pour faire face à l'augmentation massive des données.

Ainsi, dans le cadre de notre recherche, nous présenterons un modèle qui doit être capable de simplifier le partage et la collaboration dans l'IoT. Dans le chapitre 3, nous

exposerons plus profondément les principaux axes de ce modèle : la découverte, la sélection et l'échange de données.

1.4 Hypothèse

D'une année à l'autre, on remarque l'augmentation du nombre d'appareils connectés à l'Internet. D'après des statistiques récentes, ils sont devenus une fois et demi aussi nombreux que la population mondiale [9]. L'existence d'un lien entre un objet et l'Internet est le fondement principal du fonctionnement de l'Internet of Things (IoT). Ainsi, pour que cette relation soit à la hauteur des besoins de l'utilisateur, il est nécessaire d'adopter un nouveau système de communication basé sur la collaboration et le partage, surtout qu'on se retrouve avec un grand nombre d'appareils connectés à partir de différents domaines.

L'hypothèse principale qui sous-tend la présente recherche repose sur les limites de communication et de circulation des services dans le monde de l'IoT. Nous estimons alors qu'il est nécessaire de proposer une stratégie de collaboration pour partager les données.

1.5 Motivations

Les avantages et les capacités multiples de l'Internet ont permis de donner naissance à l'Internet of Things (IoT) qui peut être considéré comme étant une véritable révolution dans le monde de la technologie, de l'information et de la communication. Selon le géant de la réseautique Cisco, les appareils connectés atteindront le nombre de 50 milliards en 2020 [10] [11], et intégreront notre vie personnelle et professionnelle dans ses différentes facettes. D'où l'importance de mettre l'accent sur la nécessité du partage collaboratif dans l'IoT afin de créer un système à part entière qui répond à nos besoins.

Pour nous assurer du bon fonctionnement de l'IoT, nous avons pensé à des solutions permettant de gérer sa multitude de données. Comme mentionné dans la section précédente, il faudra être capable de réagir efficacement pour établir la collaboration et le partage des données dans l'IoT. Notre préoccupation principale est de nous baser sur les connaissances et technologies déjà existantes pour proposer un modèle collaboratif permettant le partage et l'échange facile des données dans l'IoT.

1.6 Objectifs et Contributions

L'objectif principal de notre travail consiste à présenter un modèle capable de favoriser un partage collaboratif de données dans l'Internet of Things (IoT). Pour réaliser notre projet de recherche et parvenir à proposer notre modèle, nous avons choisi de l'aborder sous l'angle des services Web en passant par trois phases : la découverte, la sélection et l'échange de données.

Dans la présente recherche, nos principales contributions sont les suivantes :

- Proposition d'un modèle collaboratif qui facilite le partage de données et la collaboration dans le monde de l'IoT;
- développement d'un langage DSL appelé IoTCollab pour faciliter la gestion des différents concepts introduits dans le modèle collaboratif.

1.7 Méthodologie

Comme nous l'avons vu dans la section précédente, la finalité de ce travail consiste à faciliter le partage et la collaboration de données qui se transforment ensuite en informations à l'aide d'un modèle collaboratif. Le point de départ de notre stratégie de

recherche était la revue de la littérature. Comme pour toute autre recherche, l'état de l'art est un élément essentiel pour situer la question principale de recherche par rapport aux anciens travaux réalisés sur le même sujet. Ainsi, notre stratégie débute par une recension des travaux sur l'Internet of Things (IoT) qui nous permet : 1) de disposer d'un aperçu sur les problématiques qui ont été débattues sur l'IoT; 2) d'avoir une idée claire sur les techniques et les approches développées pour gérer les données dans l'IoT; 3) de savoir ce qui doit être remis en cause. La revue de la littérature nous permettra alors de positionner notre recherche par rapport à d'autres, ainsi que de valider l'originalité et l'apport du présent travail.

Le recours à la revue de littérature comme méthodologie n'est qu'une phase préparatoire que nous allons voir dans plus de détails au chapitre 3, État de l'art. Par la suite, pour présenter notre modèle collaboratif, ce qui est le but principal de cette recherche, nous avons besoin de passer par trois phases importantes définies dans la section 1.2 qui sont : la découverte de services, la sélection de services, et le mécanisme d'échange de données entre les objets.

Notre méthodologie consiste à rechercher les meilleures solutions qu'il faut apporter à ces trois phases. Cela devient notre processus de modélisation de base visant à réaliser un modèle capable de faciliter le partage de données et la collaboration dans l'IoT.

1.8 Plan de ce mémoire

Ce mémoire est organisé en six chapitres. Le premier et présent, l'introduction, présente la problématique traitée et nous permet d'exposer nos objectifs et notre contribution au lecteur.

Le deuxième chapitre sera consacré à la présentation du contexte général de notre étude en définissant l'Internet of Things (IoT) et les autres concepts liés à notre problématique.

Dans le troisième chapitre, nous mettons l'accent sur les travaux de recherche précédents afin de présenter l'état de l'art de l'IoT.

Le quatrième chapitre est le cœur de notre travail, puisqu'il présentera le cadre théorique de notre modèle collaboratif de l'IoT. Dans ce chapitre, chaque phase du modèle est décrite en détail. Nous en expliquerons aussi bien ses fonctionnalités que ses rôles.

Le cinquième chapitre décrira le prototype qui a été développé.

Enfin, le sixième chapitre sera la conclusion générale par laquelle nous terminerons notre travail et qui reprendra les principales contributions et limites de cette recherche.

CHAPITRE 2

CONCEPTS DE BASE

Dans ce chapitre nous présentons les différents concepts de base qui serviront à clarifier les termes utilisés tout au long du document. Cela aidera le lecteur à se familiariser plus rapidement avec les solutions proposées pour la réalisation de notre modèle de partage de données et de collaboration sur l'IoT

Le service Web (1^{er} concept de base) représente la technologie permettant l'échange de données entre les différentes plateformes et leurs applications, bien que ces applications puissent être hétérogènes. L'échange de données nécessite l'utilisation d'un logiciel intermédiaire, l'Intergiciel (2^e concept de base). Ce dernier est un logiciel qui offre une interface d'échange de données entre les objets dans l'Internet. Tout cela se passe dans l'Internet of Things ou IoT (3^e concept de base). Finalement, un objet peut, non seulement être intégré dans l'Internet, mais aussi sur le Web, plus précisément sur une extension de l'IoT nommée le Web of Things ou WoT (4^e concept de base). Ces quatre concepts sont décrits brièvement dans les points suivants (de 2.1 à 2.4).

2.1 Services Web

Étant donné que « les objets connectés ont un talon d'Achille : sans services Web associés, ils meurent ! » [12], dans notre modèle, nous recourons aux services Web pour contourner les problèmes de normes liés à l'interconnexion, ce qui permet un accès souple à des plateformes logicielles et matérielles via Internet.

2.1.1 Définition des services Web

Les services Web sont « des applications qui relient des programmes, des objets, des bases de données ou des processus d'affaires à l'aide de XML et des protocoles Internet standards. Les services Web sont des compléments aux programmes et applications existants, développés à l'aide de langages tel que Visual Basic, C, C++, C# (C sharp), Java ou autre, et servent de pont pour que ces programmes communiquent entre eux » [13].

Plusieurs autres définitions des services Web peuvent être retenues. Nous pouvons citer celle de Abouzaid qui semble la plus simple : « Un service Web est un système logiciel qui permet de soutenir l'interaction entre les machines sur un réseau. Il dispose d'une interface écrite en format exploitable par les machines, spécifiquement le WSDL (Web Services Description Language) » [14].

De ce qui précède, on peut conclure que le service Web est un service électronique, offert par des technologies universelles développées en fonction des protocoles réseautiques mondiaux (essentiellement d'Internet) qui agit comme infrastructure de communication.

2.1.2 Architecture du service Web

Les services Web permettent, à travers un espace de communication, d'échanger différentes informations. Dans cette section, on tente de comprendre comment cet échange de données se fait en présentant un modèle d'architecture classique pour les services Web appelée SOA (Architecture orientée service).

La SOA « est une architecture d'application distribuée, de plateforme indépendante et à couplage faible, dans laquelle les différents services communiquent les uns avec les autres par des interfaces à définition simple et précise, et ne tient pas compte de l'interface de programmation sous-jacente et des modèles de communication. » [15]. Comme le montre la Figure 2.1 ci-dessous, le modèle SOA met en interaction trois acteurs : un client, un fournisseur et un intermédiaire.

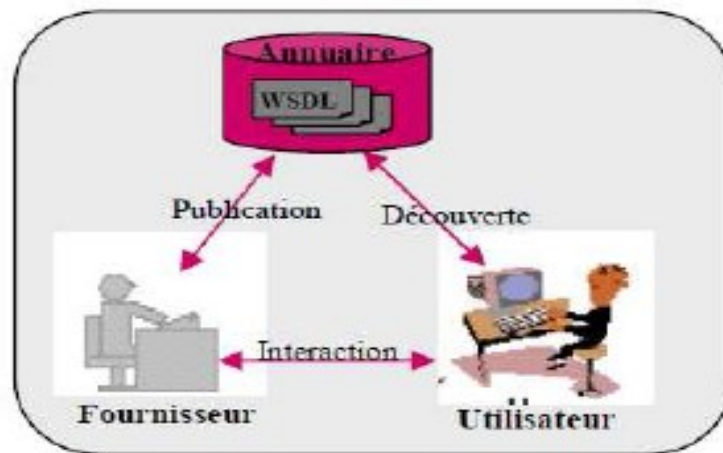


Figure 2.1 :Architecture classique des services Web [16]

Le modèle SOA fait intervenir les trois acteurs suivants : 1) le client, qui est le demandeur du service représenté par l'application, qui va chercher et consommer le service; 2) le fournisseur, qui est le propriétaire du service offert dans la plateforme d'accueil; 3) l'annuaire des services, un intermédiaire entre le client et le fournisseur, qui offre au fournisseur la possibilité de publier ses services, et au client la capacité de localiser les services répondant à ses besoins. Le fonctionnement des services Web se base sur un modèle composé de trois phases fondamentales : la publication, la découverte et l'interaction [16].

2.1.2.1 Publication

Après avoir structuré sémantiquement son service, le fournisseur passe ensuite à la publication de ce service afin d'être découvert puis utilisé par d'autres utilisateurs ou machines. La publication du service se fait par l'UDDI (Universal Description, Discovery, and Integration), un annuaire de services fondé sur XML. Toutefois, avant qu'il soit publié, le service doit être décrit, cette description se faisant dans le langage WSDL (Web Service Description Language) [16].

2.1.2.2 Découverte

Une fois la publication du service faite, le client peut alors interroger l'annuaire (UDDI) en utilisant des mots clés permettant l'obtention d'un ensemble de descriptions (WSDL) qui contiennent toutes les informations nécessaires à l'invocation du service [14].

2.1.2.3 Interaction

Aussitôt que la description du service publiée dans l'annuaire correspond à la requête lancée par le client, ce dernier aura une interaction directe avec le service par l'intermédiaire de SOAP (Simple Object Access Protocol) qui invoque les descriptions du service Web [16].

2.2 Intergiciel

Nombreux sont les travaux de recherche qui s'intéressent à l'intergiciel. Dans cette section, nous mettons l'accent sur les éléments qui caractérisent ce concept et aussi sur l'intérêt de l'utiliser dans une application informatique.

2.2.1 Définition de L'intergiciel

Zeng définit l'intergiciel comme étant « une couche de logiciel ou un ensemble de sous-couches interposées entre la technologie et les niveaux d'application. Sa fonction, qui consiste à cacher les détails des différentes technologies, est fondamentale pour éliminer les publications non pertinentes. L'intergiciel gagne de plus en plus d'importance en raison de son rôle majeur dans la simplification du développement de nouveaux services » [17].

« La définition de l'intergiciel généralement admise est la suivante : un intergiciel est un logiciel de communication qui permet aux processus exécutés sur une ou plusieurs machines d'interagir à travers un réseau. » [18].

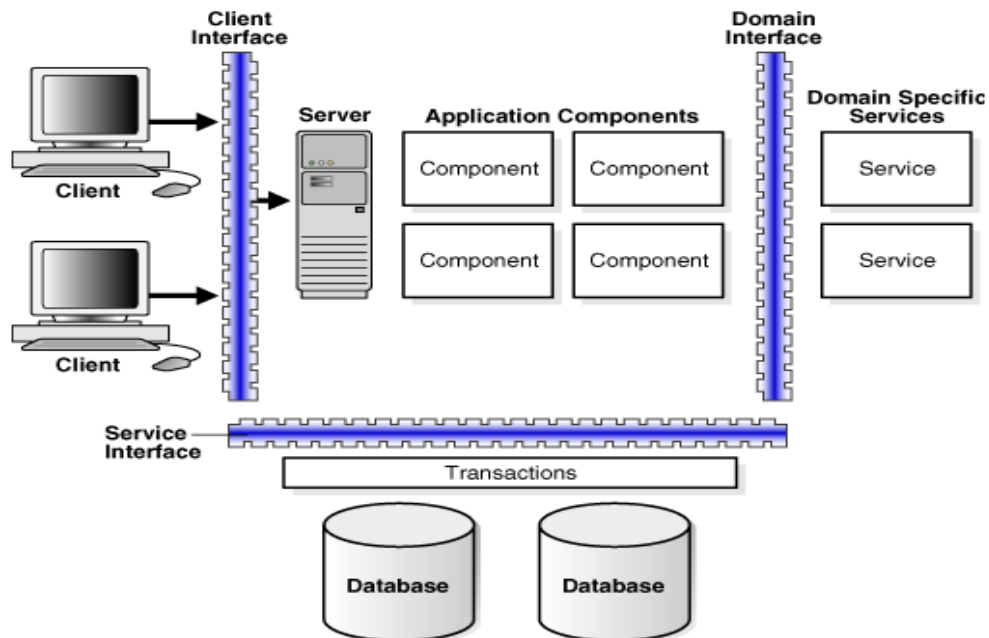


Figure 2.2 : Architecture Middleware [19]

Comme le montre la Figure 2.2, l'intergiciel (la partie en bleu) est « le logiciel qui connecte les composants logiciels ou les applications d'entreprise. C'est la couche

logicielle qui se trouve entre le système d'exploitation et les applications de chaque côté d'un réseau d'ordinateurs distribué » [19].

2.2.2 Utilité de l'intergiciel

L'intergiciel est « une classe de logiciels permettant la communication entre des applications qui n'étaient pas conçues pour dialoguer entre elles. Il homogénéise l'accès à l'ensemble des services proposés par le réseau. Les intergiciels sont utilisés pour supporter des applications complexes et distribuées. Ils permettent aussi de lier deux ou plusieurs logiciels entre eux afin que les applications correspondantes puissent échanger des données. »[20].

« L'intergiciel rend le développement d'une application plus facile en fournissant une abstraction de programmation commune, en masquant l'hétérogénéité et la distribution du matériel et du système d'exploitation, et en masquant les détails de programmation de haut niveau. » [19].

2.3 Internet of Things

Pour définir le concept de l'Internet of Things (IoT) évoqué tout au long de ce document, nous exposons, pour commencer, les différentes définitions de l'IoT existant dans la littérature pour ensuite mettre l'accent sur son fonctionnement et enfin, terminer par ses domaines d'applications.

2.3.1 Définition de l'IoT

L'Internet of Things (IoT) est « un réseau qui relie et combine les objets avec l'Internet, en suivant les protocoles qui assurent leurs communication et échange d'informations à travers une variété de dispositifs. » [8].

L'IoT peut se définir aussi comme étant « un réseau de réseaux qui permet, via des systèmes d'identification électroniques normalisés et unifiés, et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi, de pouvoir récupérer, stocker, transférer et traiter les données sans discontinuité entre les mondes physiques et virtuels. » [21].

Il existe plusieurs définitions sur le concept de l'IoT, mais la définition la plus pertinente à notre travail de recherche est celle proposée par Weill et Souissi qui ont défini l'IoT comme « une extension de l'Internet actuel envers tout objet pouvant communiquer de manière directe ou indirecte avec des équipements électroniques eux-mêmes connectés à l'Internet. Cette nouvelle dimension de l'Internet s'accompagne avec de forts enjeux technologiques, économiques et sociaux, notamment avec les économies majeures qui pourraient être réalisées par l'ajout de technologies qui favorisent la standardisation de ce nouveau domaine, surtout en matière de communication, tout en assurant la protection des droits et des libertés individuelles » [22].

2.3.2 Fonctionnement de l'IoT

L'Internet of Things (IoT) permet l'interconnexion des différents objets intelligents via l'Internet. Ainsi, pour son fonctionnement, plusieurs systèmes technologiques sont nécessaires. Citons quelques exemples de ces technologies.

« L'IoT désigne diverses solutions techniques (RFID, TCP/IP, technologies mobiles, etc.) qui permettent d'identifier des objets, de capter, stocker, traiter, et transférer des données dans les environnements physiques, mais aussi entre des contextes physiques et des univers virtuels. » [21].

En effet, bien qu'il existe plusieurs technologies utilisées dans le fonctionnement de l'IoT, nous mettons l'accent seulement sur quelques-unes qui sont, selon Han et Zhanghang, les technologies clés de l'IoT. Ces technologies sont les suivantes : RFID, WSN et M2M, et sont définies ci-dessous.

- RFID (Radio Frequency Identification) : le terme RFID englobe toutes les technologies qui utilisent les ondes radio pour identifier automatiquement des objets ou des personnes. C'est une technologie qui permet de mémoriser et de récupérer des informations à distance grâce à une étiquette qui émet des ondes radio [23]. Il s'agit d'une méthode utilisée pour transférer les données des étiquettes à des objets, ou pour identifier les objets à distance. L'étiquette contient des informations stockées électroniquement pouvant être lues à distance [8].
- WSN (Wireless Sensor Network) : c'est un ensemble de nœuds qui communiquent sans fil et qui sont organisés en un réseau coopératif. Chaque nœud possède une capacité de traitement et peut contenir différents types de mémoires,

un émetteur-récepteur RF et une source d'alimentation, comme il peut aussi tenir compte des divers capteurs et des actionneurs [24]. Comme son nom l'indique, le WSN constitue alors un réseau de capteurs sans fil qui peut être une technologie nécessaire au fonctionnement de l'IoT.

- M2M (Machine to Machine) : c'est « l'association des technologies de l'information et de la communication avec des objets intelligents dans le but de donner à ces derniers les moyens d'interagir sans intervention humaine avec le système d'information d'une organisation ou d'une entreprise » [25].

2.3.3 Domaines d'application de l'IoT

Nous constatons que le concept de l'Internet of Things (IoT) est en pleine explosion vu que nous avons de plus en plus besoin dans la vie quotidienne d'objets intelligents capables de rendre l'atteinte de nos objectifs plus facile. Ainsi, les domaines d'applications de l'IoT peuvent être variés.

Plusieurs domaines d'application sont touchés par l'IoT. Dans leur article, Gubbi *et al.* [26] ont classé les applications en quatre domaines : 1) le domaine personnel, 2) le domaine du transport, 3) l'environnement et 4) l'infrastructure et les services publics. Comme le schéma ci-dessous le montre, on trouve alors l'IoT dans notre vie personnelle quotidienne et également dans les services publics offerts par le gouvernement.

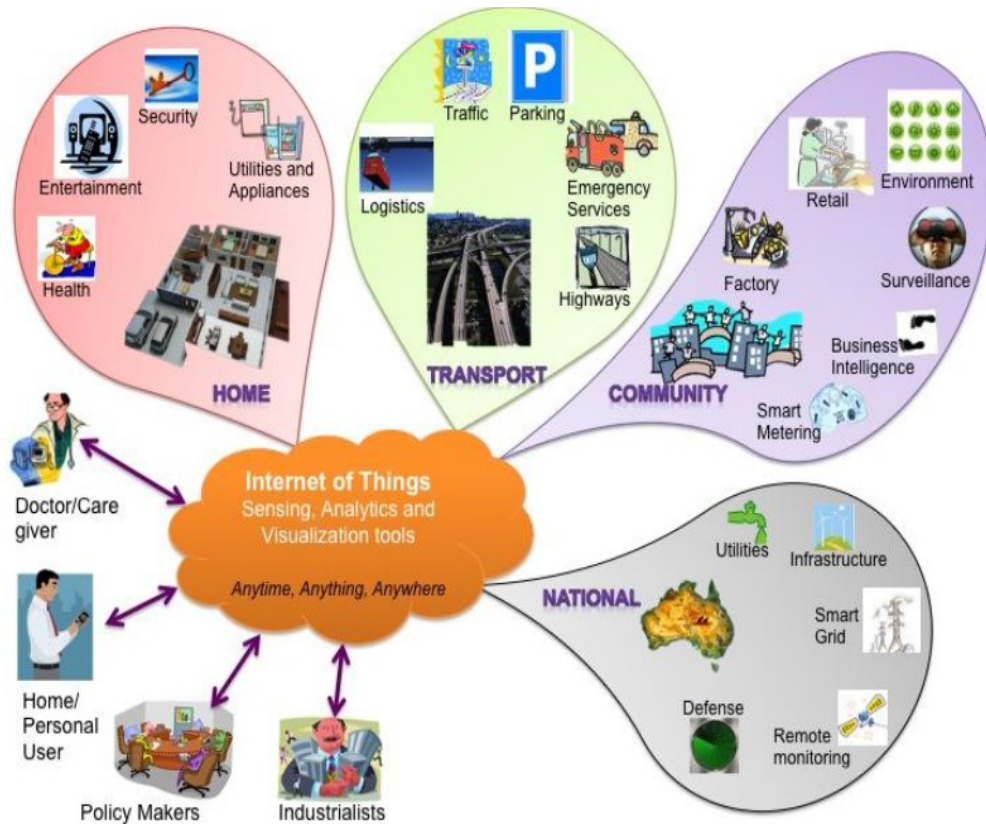


Figure 2.3 : Les domaines d'Internet of Things [26]

Nous pouvons affirmer que l'Internet peut être connecté à n'importe quel objet. Ainsi, les domaines d'applications de l'IoT sont multiples. On cite, à titre d'exemples, l'industrie, la santé, l'éducation et la recherche. Cependant, il sera possible dans le futur de trouver le concept de l'IoT n'importe où, n'importe quand et à la disposition de tout le monde.

Pessemier a argumenté que « L'IoT consiste en un monde de données (énormes), qui, si elles sont exploitées correctement, contribueront à répondre aux problèmes d'aujourd'hui, notamment dans les domaines suivants : aérospatial, aviation, automobile, télécommunications, construction, médical, autonomie des personnes handicapées,

pharmaceutiques, logistiques, gestion des chaînes d'approvisionnements, fabrication et gestion du cycle de vie des produits, sécurité, sûreté, surveillance de l'environnement, traçabilité alimentaire, agriculture et élevage. » [27].

2.4. Web of Things (WoT)

Le Web et ses technologies étant largement disponibles et populaires, la prochaine étape logique consiste alors à l'utiliser comme plateforme de communication pour les objets intelligents. Lier des objets aux pages Web a été la proposition de plusieurs chercheurs. Incorporer des objets intelligents dans une architecture de services Web normalisés en utilisant les normes SOAP, WSDL ou l'UDDI est une autre façon d'utiliser le Web [28].

2.4.1 Définition du WoT

L'Internet of Things (IoT) est considéré parmi les domaines de recherche les plus actifs. Avec l'exploration des différents points de vue et l'analyse des différentes méthodes, on constate le développement rapide et la propagation phénoménale de l'IoT, en citant comme exemple la renaissance du Web of Things (WoT), qui permet principalement de réduire les barrières entre le monde physique et le monde virtuel [17].

Le WoT propose de faire connecter les dispositifs physiques par les protocoles et les normes du Web et de les utiliser comme des ressources dans le développement d'applications Web [29].

Le WoT peut se définir comme étant « un concept informatique qui décrit un avenir où les objets du quotidien sont entièrement intégrés avec le Web. Les systèmes

informatiques qui permettent la communication avec le Web en sont une condition préalable. Ces dispositifs intelligents seraient alors en mesure de communiquer les uns avec les autres en utilisant les standards du Web existants. » [30].

2.4.2 Spécifications de WoT

D'après les définitions du Web of Things (WoT), la solution d'inclure l'architecture du Web classique à des objets physiques connectés à l'Internet rend leurs contributions plus intelligentes et efficaces dans notre vie physique et virtuelle. Ainsi, le WoT permet d'exploiter des données enregistrées par des ressources embarquées grâce aux services Web, tels que le protocole HTTP et l'URI, qui présentent des spécifications pour le WoT.

Le HTTP (Hypertext Transfer Protocol) est un protocole utilisé dans le WoT qui permet l'accès aux fichiers situés sur le réseau Internet. Il fonctionne selon un principe de requête et réponse : « le client transmet une requête comportant des informations sur le document demandé et le serveur renvoie le document s'il est disponible ou, le cas échéant, un message d'erreur » [31].

L'URI (Uniform Resource Identifier) est « une chaîne de caractères structurée permettant d'identifier de manière unique une ressource dans un espace de nom donné. Cette ressource peut être désignée soit par un URN (Uniform Resource Name) soit par une URL (Uniform Resource Locator). URN et URL sont des sous-ensembles d'URI. Un URN permet d'identifier une ressource par son nom même lorsque celle-ci n'est plus disponible. Une URL permet de localiser une ressource » [31]. Comme pour le protocole HTTP, l'URI représente une technologie de base pour le Web qui est devenue incontournable pour le WoT.

2.5 Conclusion

Tout au long de ce chapitre, nous avons délibérément choisi des définitions simples et variées pour chaque outil et concept adopté dans notre cadre théorique, dans le but d'enlever l'ambiguïté et de démystifier certaines confusions, surtout entre le Web of Things (WoT) et l'Internet of Things (IoT) d'un côté, et entre le Service Web et le service communiqué dans le Web, de l'autre.

Dans le chapitre suivant, nous exposerons les approches dont les outils et concepts sont liés à notre sujet de recherche.

CHAPITRE 3

L'ÉTAT DE L'ART

L'Internet of Things (IoT) et la technologie Web sont les outils majeurs de la nouvelle ère de transformation digitale. À travers l'IoT, on recherche le potentiel des nouvelles technologies et leurs impacts dans l'amélioration continue du monde réel. Il est important de dresser un état de l'art retranscrivant avec tact les divers travaux déjà réalisés par nos prédécesseurs pour apporter ou même lever le voile sur des solutions qui, de prime abord, ne paraîtraient pas aussi évidentes quant à notre problématique. Pour cela, nous développerons cette partie en trois plans distincts et succincts qui auront le mérite d'apporter des réponses à notre problématique.

Tout d'abord, nous aborderons la découverte de services, ensuite la sélection de services qui va mener à l'étape cruciale d'échange de données. Nous apporterons également des analyses détaillées afin de mieux cerner le concept du Web of Things (WoT), mais aussi l'IoT ainsi que les modèles collaboratifs.

3.1 Découverte de services

La place de plus en plus prépondérante qu'occupe l'Internet dans notre vie sociale est devenue incontestable.

L'Internet of Things (IoT) est un concept qui a émergé au cours de ces dernières années en ayant tiré parti de l'avènement de l'Internet. L'IoT a radicalement bouleversé notre manière de voir, de penser, de communiquer, mais surtout d'interagir avec notre

environnement qui est devenu de plus en plus complexe, et pour cause la majorité des agents économiques ont dû s'adapter à cet évènement. Par exemple, une étude faite en 2014 a démontré que près de 20 % des entreprises mondiales (1500 interrogées) investissent aujourd'hui dans l'IoT contre 17 % seulement en 2013. Aussi, 14 % d'entre elles indiquent avoir placé les capteurs sensoriels au sommet de leur priorité pour les cinq prochaines années. Ainsi, l'opérateur AT&T a mis en place un système de « bagages intelligents » via des systèmes LTE et GPS. Il est ainsi maintenant possible de suivre ses bagages en temps réel avec un téléphone intelligent ou une tablette. De son côté, la ville de San Jose s'est équipée d'un réseau de capteurs pour mesurer le trafic ainsi que la pollution atmosphérique et sonore [32].

Prenons le cas de Washington, DC. La ville a demandé à des développeurs travaillant à l'extérieur du gouvernement de créer une application permettant aux citoyens de signaler en temps réel les endroits où se trouvent des nids-de-poule à l'aide de leur téléphone intelligent dans le but d'accroître la capacité de la ville à reconnaître et régler les problèmes sur ses routes. Les économies réalisées au cours de la première année représentent 2,5 millions de dollars [33].

Ainsi, cette réalité implique une nouvelle phase de changement. La méthodologie de traitement de données nécessite la création et la mise à disposition de nouveaux outils de découvertes plus performants et plus précis pour gérer au mieux l'énorme flux de services dans l'Internet [34]. L'importance de la phase de découverte dans le cadre de l'IoT est donc un des premiers postulats de la raison pratique auxquels on doit porter une attention particulière. Comme le soulignaient les chercheurs [35] [36] qui préfèrent orienter l'évolution du Web de façon à servir les intérêts de l'IoT, en commençant par Hadjila *et al.*

[35], la découverte des services est une approche représentative d'un modèle qui exploite l'interface du service Web, tout en prenant en compte les entrées et les sorties, sans oublier l'ontologie de domaine. Dans le cadre de leur théorie, ils ont proposé une automatisation constante du principe même de la découverte de service Web. Le principe est relativement simple; il se base sur la notion du Web sémantique. Cela consiste à introduire une mesure de similarité [35] dans un algorithme de correspondance pour calculer le score de similitude entre la demande et les services Web modélisés. Cette mesure présente le mérite d'être relativement simple comparée avec d'autres mesures et accompagnée d'une complexité peu exigeante en rapport avec l'ontologie de domaine. Cependant, du fait de sa simplicité, elle est limitée si l'on intègre la notion de distance sans oublier qu'elle exclut la stimulation conceptuelle. Dans le même contexte, Varguez-Moo *et al.* [36] ont proposé une approche méthodologique de la découverte des services dans le Web, qui pourrait être cernée par l'usage d'un algorithme d'apprentissage automatique. L'algorithme exécute la fonction de découverte par une méthodologie intuitive de classification des différents services Web. Le langage sémantique de description de service Web étant alors inné, propre à l'algorithme, il est appelé SAWSDL (Semantic Annotation for WSDL). Ce dernier est capable de tenir compte des différents aspects fonctionnels du service demandé, mais également d'autres aspects moins évidents tels que la fiabilité, les performances, la disponibilité, l'intégrité, la sécurité et le prix.

Quant à eux, tournés plus vers le concept d'IoT, les chercheurs Paganelli & Parlanti [37] ont proposé un éventail de réponses basé sur une méthodologie d'approche pair-à-pair. Il est important de souligner que dans l'ensemble, les approches de découverte de services dans l'IoT ne supportent généralement qu'un seul attribut. Par exemple, un

identificateur d'objet représenté par une chaîne de caractères lui permettant d'identifier un objet. Cependant, l'approche pair-à-pair n'est pas limitée à cela. Basée sur l'adoption d'un schéma d'indexation et de classification pour la découverte de services, elle se veut multidimensionnelle en allant même jusqu'à pouvoir traiter des requêtes complexes. Le principal inconvénient technique de cette approche est la confusion que peut porter l'utilisation d'un même nœud. En effet, chaque nœud peut à la fois représenter un client et un serveur, ce qui peut engendrer une demande de service de manière directe (client à client). Un autre inconvénient majeur est que nous ne sommes pas à l'abri d'un nombre de clients restreints, voire insuffisants, mais également d'un nombre limité de réponses.

De la même manière, les recherches effectuées par Altmann *et al.* [38] ont toutefois apporté une autre façon de percevoir la découverte de service dans l'IoT. Cette approche repose sur la puissance des tables de hachage distribuées (DHT). Leur principe est relativement simple, mais efficace, car le DHT est une technique qui facilite la localisation ainsi que le téléchargement des fichiers dans un réseau distribué. Cette méthode consiste à dénicher tous les fournisseurs de services, puis à attribuer des valeurs de hachage pour chacun pour enfin résoudre les collisions de hachage.

3.2 Sélection de services

Nous entrons dans la deuxième phase de notre cadre théorique qu'est la phase de sélection. Une fois que la phase méthodologique d'approche de la découverte a été bien détaillée, force est de constater qu'une sélection rigoureuse dans une liste de services proposée sur Internet se révèle primordiale. Notre démarche précédente dans la phase de découverte ouvre donc la porte au processus de sélection qui n'est que l'expression de

réponses pour permettre une meilleure sélection possible parmi une liste de services relative aux exigences postulées dans la demande de l'utilisateur. Nous aborderons donc, par la suite, la problématique de la sélection et les diverses méthodologies affiliées qui ont toujours été au cœur même de divers courants de pensée, mais aussi d'un nombre important d'articles scientifiques traitant du sujet. D'ailleurs, un grand nombre de chercheurs ont fait des contributions majeures au développement et à la meilleure compréhension de la phase de sélection.

Dans leur travail, les chercheurs Makhluhian *et al.* [39] ont présenté un modèle de sélection global qui permet de classer les services des fournisseurs tout en respectant le niveau de la QoS (Quality of Service) et la qualité fonctionnelle. Cette approche se déroule en trois phases distinctes. Premièrement, la mise à niveau qui se fait par prétraitement afin de normaliser les valeurs de qualité de service. Deuxièmement, la sélection qui s'opère par le biais d'une classification locale : elle classe les différents services des fournisseurs admissibles en fonction des différents niveaux de qualité de service. Pour terminer, une approche basée sur l'aspect fonctionnel est sollicitée pour déterminer de manière intelligible l'utilité de chaque service en tenant compte des résultats fournis par la dernière phase de classification et de leur qualité fonctionnelle grâce à la mesure de la similarité sémantique (Figure 3.1).

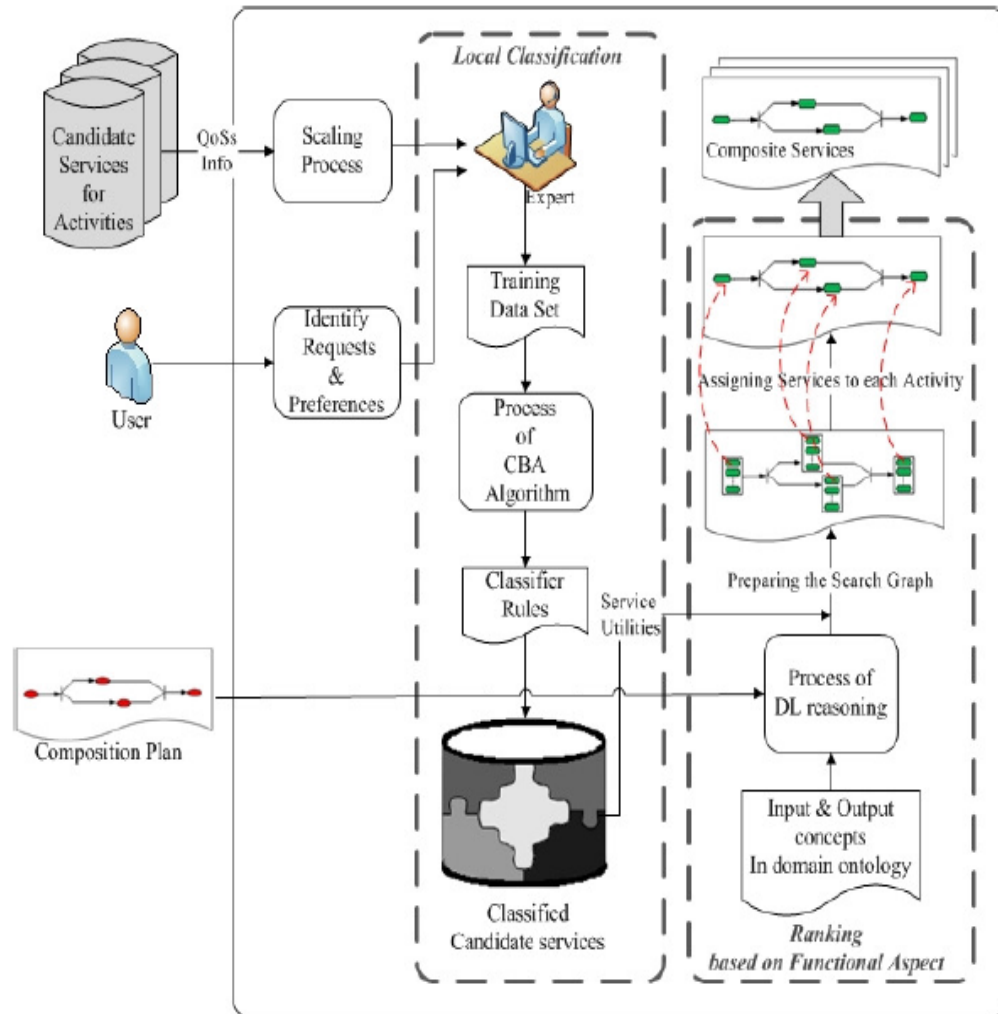


Figure 3.1 : Le processus de contour de la méthode de sélection proposée[39]

Par contre, si l'on se réfère aux travaux des chercheurs Wang *et al.* [40], l'approche de sélection par la QoS serait plus efficace via un premier artéfact du *Cloud* qui permettrait d'effectuer un calcul optimal sur l'incertitude de la QoS et donc d'optimiser les résultats redondants tout en extrayant des services fiables. Cette approche se nomme SSCM et est composée de deux phases principales :

- La première phase représente le calcul de l'incertitude. Dans cette phase, il s'agit de transformer les valeurs de qualité de service quantitatives en valeurs de qualité

de service qualitatives. Ensuite, selon le niveau d'incertitude, un service Web avec toujours de bonnes QoS peut être distingué de ces services si celles-ci dispose d'une grande variance de sa QoS.

- La deuxième phase, quant à elle, représente la sélection de service. Phase à travers laquelle nos chercheurs ont adopté une programmation mixte (MIP) pour résoudre le problème d'optimisation de sélection de service sur la base des services obtenus[40].

Comme nous pouvons le constater, plusieurs travaux et recherches ont pris en considération les informations QoS. En effet, la notion de QoS est importante d'autant qu'elle représente un facteur majeur dans le domaine du service Web. Dans le cadre du présent mémoire, et selon nos recherches personnelles, nous estimons que le QoS ou qualité de service fait sûrement partie des meilleurs critères pour nous aider à classifier les services. Cependant, le QoS ne s'avère ni idéal ni le meilleur critère à retenir lorsqu'il s'agit de répondre de manière adaptée au besoin d'un utilisateur.

Pour citer par exemple, les chercheurs Mohana & Dahiya [41] qui, à travers leurs documents de recherche, ont proposé différentes approches de sélection de service Web, toutes catégorisables en deux aspects : sémantique « la recherche du service est basée sur la connaissance des attributs, de telle sorte que les demandes soient compris et répondu en fonction de leur signification.», et non-sémantique « la recherche du service est basée sur la syntaxe. Elle exige que les données soient structurées selon le format requis et les demandes seront traitées en conséquence ». Afin de découvrir le meilleur service Web, cette approche préconisait une nouvelle technique adaptative offrant une réponse dynamique à la qualité des changements de service en retenant certains critères comme la

fiabilité au niveau de la qualité de service. Cette approche révèle un véritable potentiel qui se résume notamment au niveau des changements dynamiques des attributs de qualité des services Web.

Comme pour la sélection des services Web, la phase de sélection dans l'IoT revêtait également un centre d'intérêt majeur pour plusieurs chercheurs dont les études approfondies ont porté fruit. Les chercheurs Khezrian *et al.* [42] ont ouvert le début d'une réflexion qui a marqué le début d'un tournant positif. Néanmoins, cette ébauche de solution n'a pu ni être appliquée ni être présentée dans le milieu de la littérature. Par contre, les chercheurs Wang *et al.* [43] ont apporté leurs connaissances en la matière jusqu'à ce qu'ils viennent proposer leur approche d'amélioration du processus de sélection de service. Cette démarche s'inscrit en trois phases distinctes : 1) vérifier la rétroaction de l'utilisateur pour 2) pouvoir ensuite l'ajuster et l'évaluer vis-à-vis d'autres rétroactions et finalement 3) détecter les évaluations malveillantes en adoptant la méthode de la somme cumulée. L'approche proposée met en évidence l'importance de la mesure de la réputation du service pour pouvoir sélectionner celui qui semble être pertinent. La limite de cette théorie réside dans la possibilité de générer de fausses alarmes en raison de la somme cumulée. De plus, il n'y a pas de directives claires sur ce qui serait le meilleur choix pour certaines rétroactions sélectionnées.

3.3 Échange d'informations

Après avoir discuté des travaux de nos prédécesseurs pour les deux phases de découverte et de sélection de service au niveau des services Web et de l'Internet of Things (IoT), nous passons à la phase d'échanges d'informations.

Élément fondamental, mais aussi étape cruciale de toute notre démarche méthodologique, où nous présentons diverses solutions adoptées par plusieurs chercheurs avant de prétendre pouvoir présenter notre modèle théorique collaboratif dans l'IoT.

Dans leur travail, Jeong *et al.* [44] ont indiqué qu'un échange homogène de données entre les services Web est préalable à la réussite de l'interopérabilité des services Web. Pour assurer une collaboration équitable entre les services Web et faciliter leurs échanges de données, une approche basée sur le mappage sémantique a été proposée par ces chercheurs. Elle consiste à analyser de manière approximative la cartographie de l'information et mesurer la similarité structurale entre les documents XML. L'avantage de cette proposition découle des avantages de la méthode Kernel utilisée pour la classification des documents XML. Par contre, l'inconvénient majeur réside dans le manque de précision de leur mappage sémantique, ce qui rend difficile l'accès aux informations.

Xu *et al.* [45] ont développé une méthode d'échanges de données basée sur les protocoles XML et les architectures des services Web, lesquelles s'avèrent adaptables et personnalisables. Cette méthode agence et unifie différentes plateformes publiques d'échanges de données et plusieurs nœuds de transferts selon le besoin de l'utilisateur. Cette proposition offre l'avantage d'une solution nouvelle pour le transfert et le partage de l'information via des systèmes hétérogènes. Les utilisateurs peuvent ainsi se connecter aux différentes plateformes via un seul système afin d'obtenir les données nécessaires voulues.

Une autre approche qui vaut la peine d'être mentionnée est celle de Kubler *et al.* [46] qui soulignent que l'IoT nécessite plusieurs interfaces de communications avancées et normalisées pour faciliter les interactions entre tous types d'objets. Afin d'affirmer ce

point, ils ont présenté une stratégie basée sur la création de modèles pair-à-pair (P2P) multi-agents de données synchronisées en utilisant les principes des interfaces normalisées du Quantum Life cycle Management (QLM). Les points forts de cette étude sont : le développement d'un modèle générique de communication qui peut être utilisé avec toute autre interface similaire à QLM et la gestion de la réplication des données entre divers objets.

3.4 Internet of Things

L'évolution croissante de l'Internet of Things (IoT) et sa forte présence dans de multiples domaines expliquent le nombre important d'études existantes sur ce phénomène. Les chercheurs Hachem *et al.* [47] dans leur article expliquent les défis nombreux de l'IoT liés à l'évolutivité, l'hétérogénéité des composantes et la nature hautement dynamique et imprévisible de la topologie du réseau. Ils présentent une ontologie globale pour l'IoT basée sur le concept du SOM (Service-oriented Middleware) qui peut assurer la flexibilité et l'interopérabilité entre les composants de l'IoT.

D'après Kumar *et al.* [48], l'IoT est un environnement servant à interconnecter des objets, sauf s'ils sont adressables et identifiables. Dans ce sens, ils ont mené une analyse comparative des différentes topologies utilisées pour interconnecter des capteurs de données en tenant compte des différents protocoles d'authentification du RFID (Radio Frequency Identification); cette étude a pour but d'améliorer le choix des utilisateurs de la RFID en fonction de l'ampleur du projet.

D'un autre point de vue, Liang & Han-Chieh [49] présentent l'IoT comme une nouvelle architecture du réseau, qui a gagné rapidement une attention particulière dans le

domaine de la télécommunication. Par conséquent, ils ont proposé une nouvelle architecture du trafic multimédia qui, selon eux, va optimiser la sécurité des données transférées; cette architecture est basée sur une approche qui subdivise le traitement du trafic multimédia dans l'IoT en trois catégories : la communication, le calcul et les services.

3.5 Web of Things

Dans cette partie de notre mémoire, nous faisons souvent référence au terme Web of Things (WoT). Il est donc totalement justifié de présenter quelques travaux sur ce concept. Nous citons, à titre d'exemple, Dillon *et al.* [50] qui argumentent que le Web a évolué dans les dernières années d'un simple protocole reliant des pages Web par des liens hypertextes à une plateforme pour applications professionnelles. Ils ajoutent aussi que, parallèlement à cela, les technologies ont évolué jusqu'à un stade où elles sont aptes à être connectées au Web. Cette fusion de technologies crée de nouvelles opportunités pour le développement d'applications idoines. Tout cela montre le caractère prolifique du WoT, qui au cours des dernières années s'est développé en parallèle avec l'IoT et qui lui a permis de voir le jour.

Cheng *et al.* [51] montrent que le WoT a mis à la disposition des gens différentes façons d'obtenir de l'information en instaurant divers moyens de communication avec les objets intelligents ; ils ont également souligné l'importance de déployer de nouveaux mécanismes de découverte, d'interprétation sémantique et de modélisation de contexte dans ce nouvel environnement.

Il appert donc que si nous avons porté un intérêt grandissant pour l'IoT et les diverses solutions proposées pour sa mise en œuvre, nous nous devons de porter une attention particulière de la même importance envers le WoT et ses applications. Cela peut paraître

moins évident, même s'il existe un lien direct entre les deux concepts. En effet, d'après Mayer *et al.* [52], le Web of Things a connu une série de réussites qui a encouragé davantage l'utilisation des technologies et des principes de base du Web comme des méthodes et des solutions pertinentes pour les problèmes de l'IoT.

Suite à cela, le WoT a engendré des défis tout aussi intéressants et innovants les uns que les autres. Par exemple Dillon *et al.* [50] ont proposé un cadre qui vise à intégrer et universaliser l'utilisation de l'informatique ubiquitaire (UCDs) avec le Web, afin d'interagir efficacement avec les technologies en ligne et de les faire coopérer au bon moment via des applications adaptées.

Cependant, il est nécessaire de mentionner que divers travaux de recherches ont exploité différentes technologies du Web afin de développer des applications pour le WoT. Shin [53] a proposé une approche d'intégration des dispositifs du monde réel sur le Web, ce qui lui permet d'être facilement combiné avec d'autres ressources, tant virtuelles que physiques. Les plateformes des WoT permettent d'exercer plusieurs fonctions. Dans son étude, Shin [53] a mentionné quelques-unes de ses fonctions comme la fonction de gestion, la fonction d'interface, la fonction de sécurité, mais également la fonction du lien qui permet de créer automatiquement un code d'interface pour les différents utilisateurs. Ces derniers peuvent ainsi accéder aux différents services disponibles des divers objets intelligents présents sur la plateforme. En effet, les applications et les plateformes des WoT ont été parmi les principales problématiques débattues dans les articles de recherche comme dans celui de Mainetti *et al.* [54] qui ont considéré la connaissance approfondie des plateformes comme une nécessité primordiale dans la compréhension des WoT. Ils ont donc développé une architecture logicielle qui permet de créer, mais aussi de contrôler les

applications Mash-up (application composite développée à partir de plusieurs services ou applications Web distincts [55]) de manière simple et évolutive. Il faut noter que, comme pour l'IoT, dans le monde du WoT il est nécessaire d'accorder de l'importance aux plateformes existantes et aux diverses applications pour pouvoir se servir de manière optimale et efficace du WoT.

3.6 Modèles collaboratifs de partage de données

3.6.1 IoT et les modèles collaboratifs

Les préoccupations actuelles des chercheurs sont dirigées vers l'IoT et son influence sur notre vie quotidienne, spécialement après l'augmentation du nombre d'appareils connectés autour du monde qui aujourd'hui dépasse même le nombre d'êtres humains [56]. Ainsi, pour contribuer à cette évolution majeure, les chercheurs doivent proposer des solutions à la fois pertinentes, mais en plus efficaces qui permettraient de relever les défis qu'apporte l'IoT.

Même si plusieurs approches ont été proposées pour la collaboration dans l'IoT, celle-ci met en interaction une multitude d'objets hétérogènes ayant des fonctionnalités multiples. Ainsi, il devient nécessaire de présenter des approches capables de gérer différents modèles de données. Dans leur étude, De *et al.* [57] ont proposé des modèles sémantiques pouvant être liés les uns aux autres, dont l'avantage est la description des différents composants de l'IoT ainsi que la réalisation de l'interopérabilité au niveau des données et des services. Ces modèles permettent de décrire les données à caractère spatial, temporel ou thématique.

Selon Angulo-Lopez & Jimenez-Perez [5], l'utilisation des applications qui soutiennent l'IoT est une solution pour découvrir de nouvelles technologies et par conséquent, ils ont proposé un cadre basé sur un système multi-agent qui soutient et supporte les besoins des applications complexes dans un environnement distribué. Selon ces mêmes auteurs, ce cadre est un moyen utile pour simplifier la mise en œuvre et le développement des applications.

3.6.2 WoT et les modèles collaboratifs

La théorie de Guinard & Trifa [6] se base sur le succès de l'architecture REST dans le Web. Par conséquent, ils ont proposé une approche pour l'intégration du Web dans les dispositifs du monde réel. Leur approche repose sur la transformation des dispositifs en ressources selon les principes de REST afin qu'elle soit utilisée directement par le protocole HTTP. Ainsi, les interactions REST peuvent être exploitées pour créer rapidement de nouveaux prototypes capables de combiner les mondes physique et virtuel. En effet, l'objectif principal de ces chercheurs était de fournir des directives pratiques sur le WoT et de trouver la meilleure façon de la mettre en pratique afin d'en tirer le meilleur parti.

Dans le même contexte, Guinard *et al.* [58] ont déduit que le Web of Things (WoT) est une suite logique de l'Internet of Things (IoT). Dans ce sens, ils ont proposé d'intégrer des serveurs Web embarqués dans chaque objet intelligent avec une architecture REST afin de conserver et réutiliser les protocoles du Web comme le HTTP pour l'identification des ressources et le transport des services, ce qui fait des objets intelligents une partie intégrante du Web.

Quant aux chercheurs Ishaq *et al.* [59], ils ont représenté une approche qui utilise la technologie des services Web intégrés, à savoir le protocole CoAP (Constrained Application Protocol) qui permet la découverte, la connectivité continue, mais aussi la détection automatique des nouveaux nœuds de services déployés dans le réseau. Cela est fait par la combinaison avec le système de noms de domaine (DNS) afin de permettre l'utilisation du FQDN (Fully Qualified Domain Names) avec n'importe quel utilisateur. Le FQDN étant le nom de domaine complet pour un ordinateur spécifique, ou l'hôte, sur Internet.

3.6.3 Synthèse

Nous commençons donc cette section par une brève synthèse qui sera représentée par un tableau comparatif mettant en exergue les différentes approches pour les services dans l'IoT. Nous avons défini six critères qui caractérisent ces approches, avec, entre autres, les limites qui nous permettent de déterminer les défis de la recherche de notre travail.

Tableau 3.1 : Tableau comparatif des approches

	De <i>et al.</i> [57]	Angulo-Lopez & Jimenez-Perez [5]	Guinard & Trifa [6]	Guinard <i>et al.</i> [58]	Ishaq <i>et al.</i> [59]
SW	sémantique		Service Web standard	Service Web standard	Service Web intégré
Architecture			REST	REST	COAP
Passerelle		Dynamique	Intelligente	Intelligente	
Type de modèle	sémantique	Multi-agent			
Contrôle	Oui	Oui	Non	Oui	Non
Sécurité	Non	Oui	Non	Oui	Non
Limite	Limitation au niveau de la recherche des ressources.	Un concept qui n'a pas été validé ou testé dans le monde réel.	Manque des outils du traitement de données.	Un modèle qui ne prend pas en compte les objets à capacité limitée	Absence de sécurité. Absence de contrôle.

Le présent tableau synthétise les différentes propositions étudiées dans les sections précédentes (section 3.5.1 et 3. 5.2) en identifiant les principaux aspects auxquels elles sont liées. De *et al.* [57], dans le cadre de leur théorie, proposent une approche sémantique du Service Web par rapport à une approche multi-agent [5]. Les principaux points clés divergents entre ces deux théories relèvent surtout du critère de sécurité traduisant de manière tacite les limites entre les deux approches. D'une part, De *et al.* [57] ont dû voir la limite de leur approche sémantique au niveau de la recherche des données, tandis que Angu

lo-Lopez & Jimenez-Perez [5] n'ont pu voir leurs travaux de recherches validés. En résumé, nous remarquons que l'approche de Guinard & Trifa, [6] est caractérisée par un investissement du principe REST, lequel facilite la gestion et le traitement de données à travers une simple séparation entre les informations et leur présentation. Cependant, la deuxième approche de Guinard *et al.* [58] reste toujours limitée, malgré les efforts présentés, car il manque la généralisation de cette approche dans le monde réel. Dans l'approche de Guinard & Trifa [6] la passerelle peut prendre en charge plusieurs types de dispositifs à travers une architecture pilote. Tandis que dans l'approche de Guinard *et al.* [58] l'utilisation de la passerelle est caractérisée par sa représentation des ressources dans différents formats pour faciliter son intégration à d'autres applications Web. L'approche de Guinard *et al.* [58] se distingue aussi par sa capacité à s'appairer avec des appareils permettant de surveiller et de contrôler le système. Finalement, l'approche d'Ishaq *et al.* [59] est la plus récente et représente une composante importante qui facilite l'utilisation réelle de services Web incorporés afin de pouvoir construire le WoT. Toutefois, cette approche a une faiblesse en termes de sécurité ; aussi, nous pouvons remarquer que la configuration et la gestion des réseaux de capteurs ont été faites de façon purement manuelle. Cependant, malgré l'importance du service Web dans le WoT, ces différentes approches se limitent par l'analyse et le stockage de grandes quantités d'informations tout en présentant des solutions relatives à des besoins très spécifiques.

Afin d'améliorer ces approches et pallier les lacunes qu'elles présentent, nous essayons toujours, dans le cadre de notre recherche, de traiter les différents défis présents au niveau du partage des données. Nous avons constaté qu'il est encore difficile de parler d'une solution pertinente pour gérer le traitement de données. Ainsi, nous essayons dans

cet essai de résoudre ce point pour pouvoir finalement présenter au mieux un modèle collaboratif valide.

3.7 Conclusion

Dans le présent chapitre, nous nous sommes intéressés aux principales approches utilisées dans le domaine de l'Internet of Things (IoT). Nous avons commencé par une présentation par section des différentes phases. Soit, pour citer ci-dessus : la découverte, la sélection et enfin l'échange de données au niveau de service Web et de l'IoT. Puis, nous avons développé de manière explicite le concept du Web of Things (WoT). Pour finir, nous avons exposé les différentes approches collaboratives dans le domaine de l'IoT et de WoT. Chaque approche dûment expliquée et présentée dans ce chapitre recèle des avantages, mais également des inconvénients. Au final, la combinaison des différentes techniques existantes semble être la solution optimale afin de pouvoir prétendre traiter le problème. À cela vient s'ajouter la possibilité d'effectuer une description de partage plus adaptée afin de permettre une meilleure collaboration des données dans l'IoT. Comme nous l'avons mentionné, notre objectif principal était de construire un modèle dans le cadre théorique qui permettrait la collaboration et le partage de données dans l'IoT.

CHAPITRE 4

CADRE THÉORIQUE

Nous venons de voir dans le chapitre précédent intitulé « L'état de l'art », les enjeux et défis à relever dans le domaine de l'Internet of Things (IoT). Ces enjeux, qui revêtent une importance capitale, peuvent être regroupés en trois phases : phase de découverte des services, phase de sélection du service et finalement la phase d'échange de données.

Dans notre cadre théorique, nous proposons donc un modèle de coopération entre tous les éléments qui constituent l'IoT, afin d'améliorer l'échange des données facilitant la réalisation des services. Nous l'avons appelé le modèle collaboratif pour l'IoT. Cette collaboration se concrétise dans une tâche individuelle entre deux objets ou dans une tâche commune entre plusieurs objets dans le but de produire un service quelconque, tout en respectant les trois phases évoquées ci-dessus.

4.1 Modèle collaboratif

Le monde de l'Internet of Things (IoT), de par sa diversité et sa complémentarité en matière d'objets, s'inscrit dans une relation croissante entre le nombre d'objets adhérents et la capacité de trouver les ressources d'informations et les services afin de résoudre une tâche individuelle ou commune. Par conséquent, notre travail se concentre sur le développement d'un modèle théorique capable de favoriser la communication et l'échange entre ces divers objets.

Dans cette optique, nous avons suivi une démarche simple s'appuyant sur une stratégie typique des services centralisés (service-centric) qui ne repose pas sur la centralisation des ressources, mais cherche à personnaliser les solutions aux besoins de chaque consommateur. Cette démarche se reflète dans un modèle flexible et évolutif qui vise à faciliter la collaboration et le partage de données dans le monde de l'IoT. Le modèle se base sur une stratégie de propagation requête et réponse d'un objet à un autre à la recherche du service approprié.

La réalité est un peu plus complexe : établir une relation entre des millions d'objets nécessite tout un système qui gère le contrôle d'accès, la sécurité – surtout dans les interactions machine-à-machine (M2M), plus vulnérables lorsqu'elles sont amenées à déployer une communication Wi-Fi ou Bluetooth sans avoir recours à l'Internet) – l'équité, l'identification, et également l'authentification et l'intégrité des données échangées. Un exemple modeste qui présente la simple collaboration entre deux objets pouvant collaborer, soit d'une manière directe soit indirecte (Figure 4.1), montre l'admissibilité de notre modèle. Dans le cas d'un service direct, les étapes de la découverte, la sélection et l'échange d'informations s'exécutent d'une manière directe par de simples requêtes/réponses. En contrepartie, lorsque notre modèle contient plus que deux objets et que le demandeur ne possède pas un accès direct au service, l'un des objets prend le rôle d'intergiciel.

Donc, nous distinguons deux types de services, direct et indirect, où les critères suivants permettent d'arrêter le choix sur le service concerné : le contrôle d'accès entre les objets ou les groupes d'objets, l'accessibilité du service pour l'objet demandeur, la

confiance entre le demandeur et le point final (le fournisseur), en passant par l'intergiciel, s'il y a lieu.

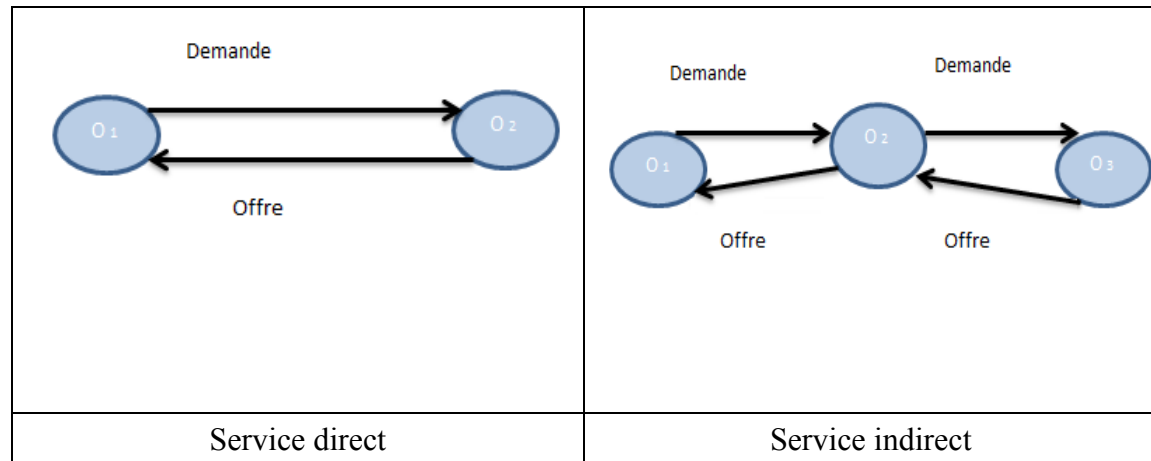


Figure 4.1 : Comparaison entre un service satisfait directement et indirectement

Notre cadre théorique consiste à assembler les objets de façon dynamique à l'intérieur de groupes selon la tâche commune affectée à ce groupe, le but étant de favoriser la collaboration entre ces objets en les mettant sous un même groupe pour l'accomplissement d'une tâche donnée.

Ainsi, notre modèle sera constitué d'objets individuels et de plusieurs groupes, ayant chacun sa contribution au sein de l'IoT. L'échange de services peut ainsi prendre les trois aspects suivants :

- d'un objet à un autre : chaque objet peut collaborer avec un autre objet pour réaliser une tâche.
- d'un objet à un groupe : un objet nécessite de collaborer avec de nombreux autres objets pour accomplir sa tâche.
- d'un groupe à un autre : de nombreux objets sont amenés à collaborer avec de nombreux autres objets de groupes divers pour exécuter une tâche.

Ceci, tout en respectant les mêmes principes directeurs : phases de découverte, de sélection et d'échange de données.

De ce qui a été cité précédemment, nous pouvons déduire que notre plan n'est fonctionnel, sauf si tous les éléments qui constituent notre modèle sont en mesure de collaborer entre eux. C'est-à-dire, l'objet ou le groupe qui détient le service demandé (le point final) a la capacité de transmettre son service au demandeur à travers le réseau. Cela devient possible si le point final est doté d'un CPU/RAM (on parle alors d'un nœud intelligent), sans quoi il faudra utiliser un serveur mandataire pour que l'objet (ou le groupe d'objets) puisse se connecter à l'Internet (et l'on parle alors de nœud non intelligent). Notre modèle pourrait supporter ces deux cas en même temps, c'est-à-dire des groupes mixtes formés de nœuds intelligents et d'autres non intelligents qui collaborent entre eux.

4.2 Phase de découverte

La phase de découverte de service dans notre cadre théorique repose conjointement sur la propagation du couple requête et réponse entre les objets et sur une politique de contrôle d'accès basique. Ces axes sont décrits ci-dessous.

4.2.1 Demande de service

Dans cette phase, l'objet interroge son réseau sur la disponibilité du service reçu en lançant une requête vers les objets ou les groupes d'objets qui seront sélectionnés selon le contexte de la demande.

Lancée sur le Web, la requête doit se conformer à l'architecture d'une requête HTTP classique, hormis les descriptions dans le contenu qui précisent les caractéristiques du service demandé et sa source.

4.2.1.1 Service simple

a) Type de la demande

Selon notre modèle, l'envoi de la demande explore deux modes, le premier est appelé actif, présenté sur la séquence de la Figure 4.2 dans lequel un objet transmet une demande ponctuelle (une seule fois) à un point final qui pourra, à son tour, répondre à cette demande.

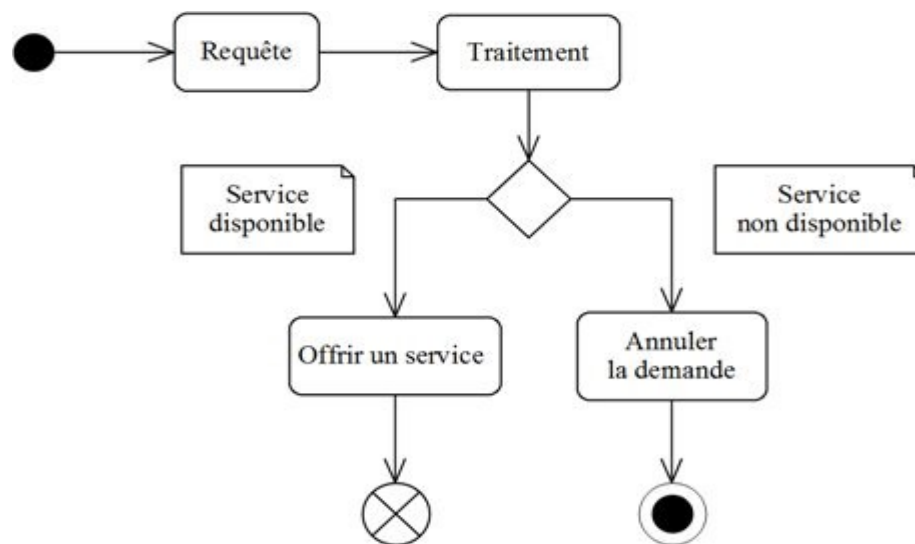


Figure 4.2 : Demande de type actif

Le deuxième est le mode passif qui, quant à lui (Figure 4.3), détaille les objets liés par une entente permettant la répétition du service demandé. Ce type est plus fréquent entre les objets du même groupe, où le demandeur reçoit passivement et régulièrement des notifications sur la disponibilité du service par le biais du point final.

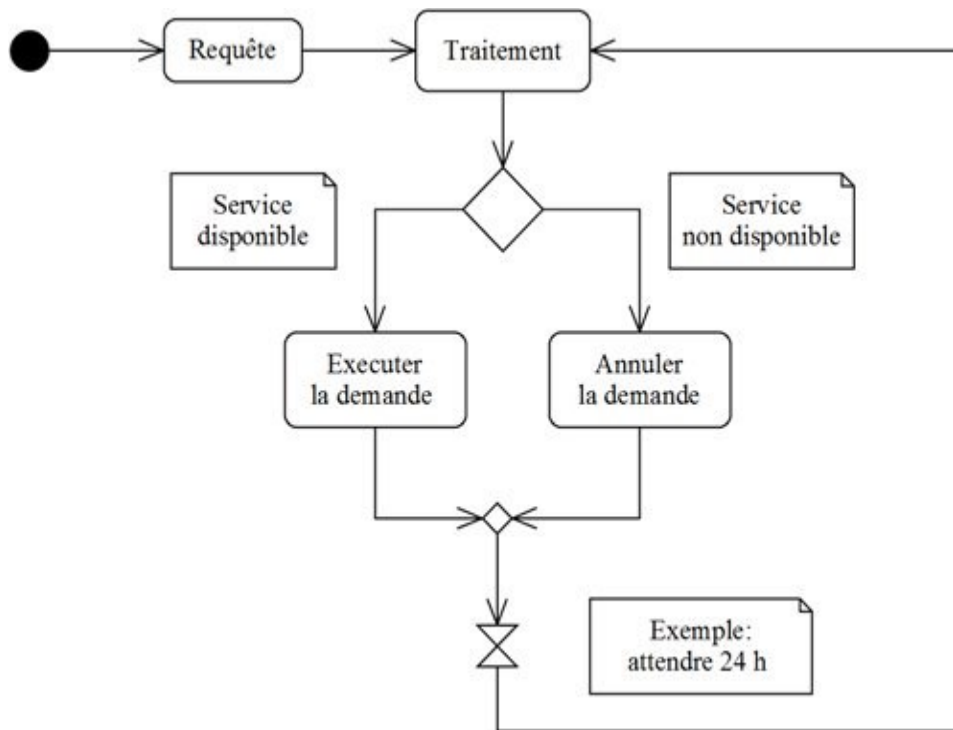


Figure 4.3 : Demande passive

b) Contenu de la demande

Dans cette section, nous décrivons les trois parties formant la demande de service, par ordre logique :

- **Le contexte** : contient des données basées essentiellement sur l'identification et la géolocalisation de l'objet demandeur, dont nous proposons d'utiliser deux axes, X et Y, permettant de définir respectivement l'altitude et la longitude, afin de déterminer le positionnement géographique des objets.

- **Les données :** contiennent des métadonnées qui informent le point final sur le type de données recherchées. Cette partie est très flexible, étant donné qu'elle détient les informations liées à la nature du service demandé.
- **La fréquence :** désigne le nombre de fois où un service sera collecté au cours d'une période déterminée. Nous pouvons nous reporter à l'un des trois types de fréquence suggérés ci-dessous, selon le besoin du demandeur :
 - envoyer le service chaque fois que le demandeur lance une nouvelle requête;
 - envoyer le service chaque fois que le point final génère de nouvelles valeurs pour le service concerné;
 - envoyer le service chaque fois que le point final génère de nouvelles valeurs pour le service concerné dans une période déterminée.

4.2.1.2 Services circulaires

La circularité est une approche plus en phase avec la réalité. C'est pourquoi notre modèle théorique s'oriente vers la demande au niveau des services circulaires. Une demande érigée sur la base d'un service simple (voir la section 4.2.2.1) où viennent s'ajouter deux points primordiaux, notamment la circularité et la matrice, qui sont explicitées ci-dessous.

a) Circularité

Dans un système décentralisé et non structuré bâti sur un réseau hybride, une découverte des services qui se propage d'un nœud à un autre en quête du service voulu, et qui maintient en correspondance les identifiants (IDs) des nœuds (objet) et leurs adresses

physiques, est primordiale pour chaque objet, et ce, afin de construire un historique personnalisable sous forme de listes réparties en deux catégories : une liste noire qui présente les objets qui ne sont pas favorables de répondre aux prochaines demandes, et une liste blanche qui présente, en ordre décroissant, les objets favorables de répondre à une prochaine demande.

Comme nous l'avons formulé précédemment, une demande peut être acheminée vers un point final par le biais d'un objet ou de plusieurs intermédiaires lorsque le demandeur concerné ne connaît pas le point final. Dans ce cas, il convient de joindre au contexte de la demande une donnée informant le point initial de l'éventualité d'accepter ou pas la réception du service par un autre objet lui étant lié, si cet autre objet n'est pas enregistré dans la liste noire des demandeurs.

Cette démarche doit veiller au respect du champ d'échange du demandeur présenté par les listes noires et blanches à chaque fois qu'elle passe d'un objet à un autre. Par conséquent, au final, la demande renferme une liste de la succession des identifiants (IDs) rattachés aux objets dans lesquels elle a circulé et qui est conclue par l'identifiant du point final.

Le but est donc de favoriser la création de groupes, faciliter la détermination des listes noires et blanches et anticiper le trajet le plus court dans la sélection du service (voir la section 4.3.2). Ce comportement est dépeint selon la procédure de l'algorithme 1 (Algorithme 4.1) qui est précédé par les définitions suivantes :

1. Dans notre contexte, un acteur est un ensemble d'objets détenus et gérés par le même utilisateur. Ainsi, cet ensemble d'objets et les services que chaque objet

peut offrir est représenté par : $\{(O_1, S_1, \text{rating}_1), (O_2, S_2, \text{rating}_2), \dots, (O_i, S_i, \text{rating}_i)\}$ pour $i \in [1, n]$, tandis que $\text{rating}_i \in [1..10]$ présente la note du service S_i .

2. Les crochets ([]) sont utilisés pour faire référence à un élément se trouvant à une position donnée dans une liste. Par exemple, $A[1]$ se réfère au premier élément de A .
3. Le point (.) est utilisé pour faire référence à un composant d'un élément. Par exemple, $A[1].O$ se réfère à l'élément O du premier élément de A .

Algorithme 1

```
1: procedure PROCESSSERVICEREQUEST
2:   INITIALIZATION:
3:    $wListTemp \leftarrow wList - \{aSender\}$ ;
4:   if  $aSender \in bList$  or  $aCurrent \in Rsd.IDS$  then
5:     exit
6:   end if
7:   if  $SERVICEMATCH(Rsd, aCurrent)$  then
8:     RESPONDOSERVICEREQUEST ( $Rsd, aSender$ )
9:   end if
10:  for all  $a \in wListTemp$  do
11:     $Rsd.IDS \leftarrow Rsd.IDS + \{aCurrent\}$ ;
12:    FORWARDTHESERVICEREQUEST( $Rsd, a$ )
13:  end for
14: end procedure
```

Algorithme 4.1 :Processus de la requête du service

Il en ressort que dans l'algorithme 1, la procédure de la demande de service repose sur la définition de la requête dans notre cadre théorique où une requête de service est représentée par un couple $Rsd = (S, IDS)$ qui détient des informations concernant les

données demandées et une liste d'identifiants d'objets par lequel la demande a déjà passé où :

- IDS = (ID1, ID2, ..., IDn) tandis que IDn fait référence au point final.

Lors de l'étape d'initialisation, l'expéditeur entame la création manuelle d'une liste blanche temporaire (wListTemp) recensant l'ensemble des acteurs qui sont déjà connus par lui et qui sont favorables à répondre à sa présente requête. Cette liste sera ensuite utilisée par la boucle visible entre les lignes 10 et 13 afin de pouvoir transmettre la requête aux acteurs qu'elle contient. La primitive `SERVICEMATCH ()` vérifiera si le service recherché correspond ou pas au service offert par l'acteur actuel (aCurrent). Si oui, la fonction `RESPONDOTOSERVICEREQUEST ()` est appelée à répondre à la requête. Cependant, quel que soit le résultat de la fonction `SERVICEMATCH ()`, la requête sera expédiée aux acteurs dans la wListTemp comme on le voit ci-dessus. Il est à noter que si l'expéditeur (aSender) est dans la liste noire de l'acteur actuel ou que celui-ci a déjà reçu la requête, la procédure s'arrête et sort sans être traitée.

b) Matrice de services

Une matrice de services est un tableau à entrées croisées qui sert à combiner le service et son fournisseur. Donc, chaque objet établit deux listes d'éléments : une liste des services offerts directement ou indirectement, et une liste des objets qui fournissent ces services en s'incluant lui-même en première ligne (Tableau 4.1).

Tableau 4.1 : Matrice de services de l'objet O1

	S ₁	S ₃	S ₇	S ₁₉	
O ₁	•			•	Services offerts par O ₁
O ₃			•		
O ₄		•			Services offerts par liste blanche O ₁
O ₇		•	•		

On suppose que le demandeur veut interagir simultanément avec un nombre indéterminé d'objets par le biais des objets de sa liste blanche, jusqu'à ce que l'objectif de sa requête soit atteint au moyen d'un ou de plusieurs points finaux. Un mécanisme dédié à la classification des trajets de découverte, du plus court au plus long est imposé, non en référence à la distance, mais plutôt au nombre le plus petit possible d'objets intermédiaires.

À cette fin, nous proposons une approche théorique basée sur les matrices de découverte, où chaque objet doit partager sa matrice de services via des serveurs de stockage en réseau. Mis à part le stockage des matrices, ces serveurs ont pour fonction la simulation des trajets possibles entre le demandeur et les points finaux. Ce procédé réduit énormément le trafic des requêtes envoyées afin d'éviter la surcharge du réseau et la perte de temps.

Exemple : nous présentons ci-dessous (Figure 4.4) l'exemple de simulation des trajets, où les objets sont présentés par des cercles et les liens de la liste blanche de chaque objet sont présentés par des flèches bidirectionnelles.

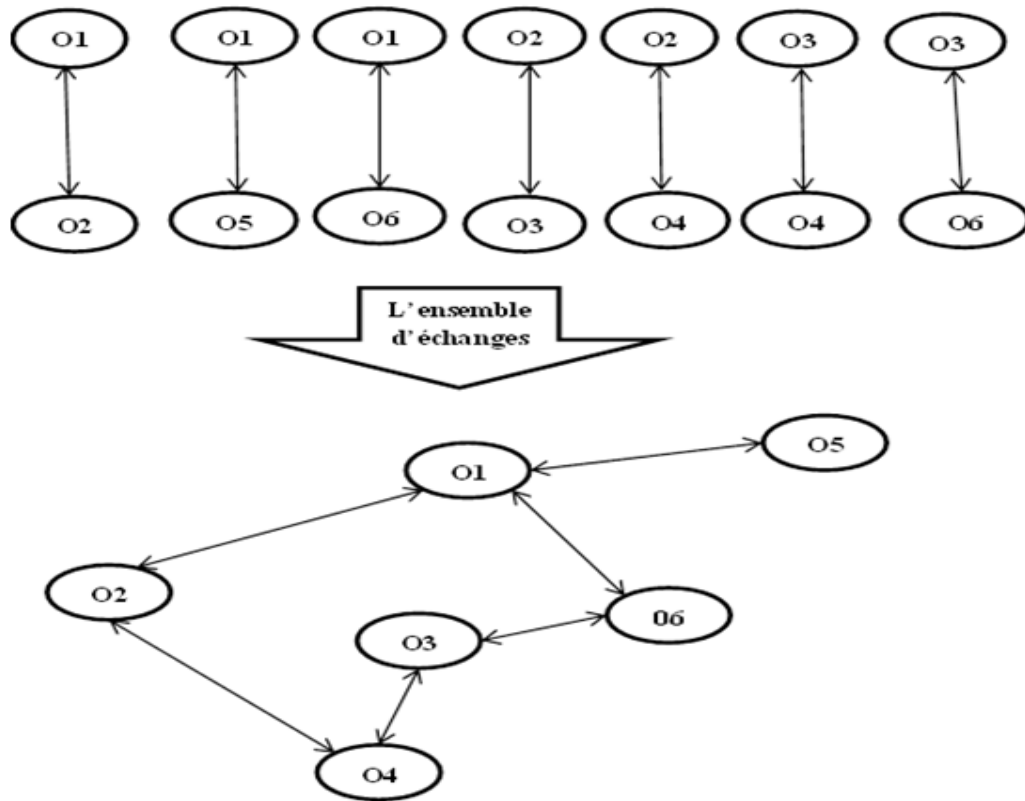


Figure 4.4 : Exemple sur la simulation des trajets

4.2.2 Réponse de la demande

Après avoir traité la requête reçue, le point final lance une réponse vers le demandeur pour l'informer sur les critères du service disponible. Cette réponse prend en compte tous les détails prescrits dans le contenu de la demande.

Conçu pour faciliter son traitement auprès du demandeur, le format de la réponse est similaire au format de la demande. Elle contient le contexte qui contient les informations de géolocalisation et d'autres informations sur l'objet fournisseur, les informations sur des services similaires, si disponibles, et la même liste d'identifiants (IDs)

rattachée aux objets à travers lesquels la requête a passé, en plus de l'ID de celui qui a répondu à la demande.

Algorithme 2

```
1: procedure PROCESSSERVICERESPONSE
2: if  $Rsd.IDS[aCurrent] > 1$  then
3:   ForwadTheServiceResponse ( $Rsd, Rsd.IDS[aCurrent - 1]$ )
4: else
5:   return ( $Rsd$ )
6: end if
7: end procedure
```

Algorithme 4.2 : Procédure de Réponse à une Demande de Service

Une réponse de service est un couple $Rrd = (R, IDs)$ qui détient des informations concernant les données offertes et une liste d'identifiants par lesquels la réponse a déjà passé en commençant par le point final où :

$IDs = (IDn, IDn-1, \dots, ID1)$ tandis que $ID1$ fait référence au demandeur

La procédure de l'algorithme 2 se résume par le renvoi d'une réponse à l'acteur actuel ($aCurrent$) ayant atteint sa destination (ligne 5). Dans le cas contraire, elle est transmise à l'objet dans $Rsd.IDS$ qui se situe exactement avant l'acteur actuel en utilisant la fonction `ForwadTheServiceResponse ()` (ligne 3).

4.2.3 Contrôle d'accès

L'Internet des objets remet en question les anciens mécanismes de pare-feu et de contrôle d'accès pour protéger un terminal, surtout en ce qui concerne l'autorisation, l'authentification et la confiance par apport aux services échangés.

Notre cadre théorique ne peut être intégré en l'absence d'une politique de contrôle d'accès qui tient compte des critères mentionnés ci-dessus. Nous avons donc développé une approche basée sur un principe très simple et connu de tous : « les amis de mes amis sont mes amis », ce principe devant être appliqué en utilisant les listes blanches et noires. Tandis que la liste blanche inclut les objets avec lesquels un objet a bâti des liens de confiance à travers un historique d'échanges positifs qui prend en compte le nombre et la qualité de services transmis, la liste noire, quant à elle, répertorie les objets avec lesquels la collaboration et le partage de données demeurent défavorables, suite à un historique d'échanges que l'objet juge insatisfaisant, unilatéral... etc. Ainsi, notre modèle de propagation est dicté par un réseau de connaissances et non par la topologie du réseau. Il convient de noter que les listes blanches et noires sont actualisées manuellement par les utilisateurs selon l'évaluation de l'historique d'échanges.

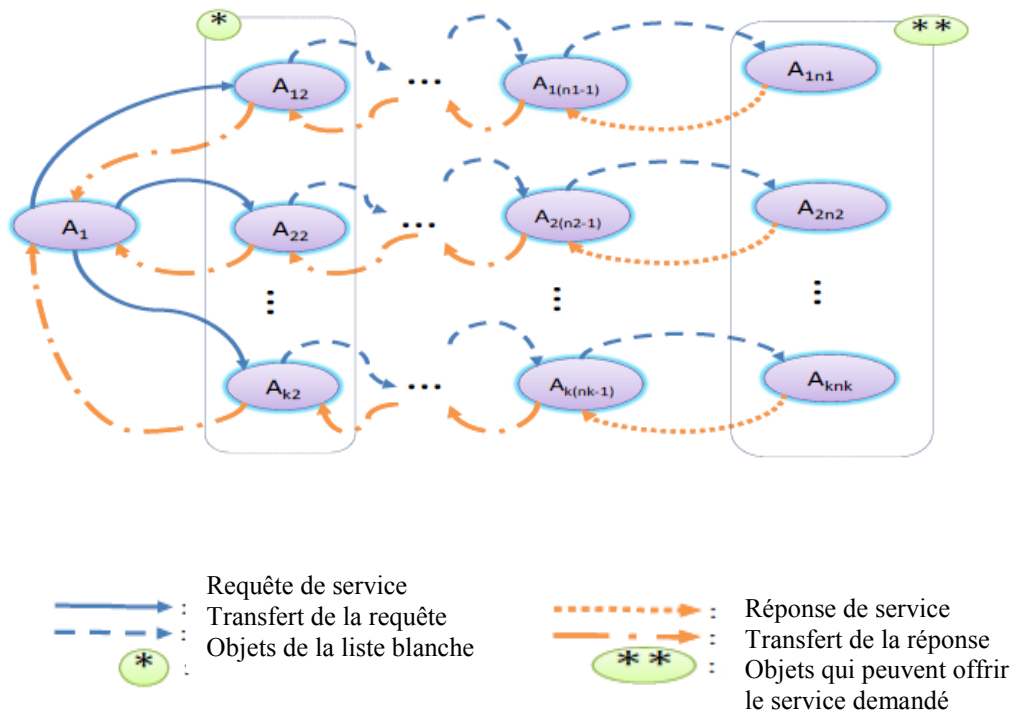


Figure 4.5 : Découverte de service par propagation

La Figure 4.5 révèle le processus de découverte dont la fonction se décline ci-dessous. D'abord, un objet formule une demande de service qui décrit le service qu'il recherche. Ensuite, la demande est envoyée à tous les objets classifiés dans sa liste blanche. En admettant qu'un récepteur soit en mesure de répondre à cette demande, il renvoie alors une réponse décrivant le service qu'il offre correspondant au service demandé. Pour finir, il transmet la demande à tous les objets classifiés dans sa liste blanche. Le même processus est effectué par les autres récepteurs, de façon à ce que la demande soit récursivement propagée.

L'objet qui a formulé la demande peut recevoir de nombreuses réponses de service émanant des différents objets. Le début de la deuxième phase se concrétise donc en donnant lieu à la sélection d'un service parmi toutes les réponses proposées.

4.3 Phase de sélection

La sélection de service est engagée immédiatement après qu'un demandeur ait reçu une réponse et s'arrête suite à un certain nombre ou un certain laps de temps qui se soit écoulé. Elle se fait en deux étapes : la présélection des candidats valables et la sélection du service demandé.

4.3.1 Présélection

Dans ce qui précède, nous avons discuté et analysé le contenu de la demande sur la base des caractéristiques et de la fréquence du service demandé sur lequel l'étape de présélection se base pour valider les correspondances exactes entre la demande et le service offert.

Les caractéristiques du service demandé et sa fréquence sont sujettes à changement par le demandeur et à la diligence de ce dernier, selon la nature du service voulu. Dans ce qui suit, nous allons présenter une méthode mathématique simple qui peut s'intégrer à plusieurs langages de programmation et qui couvre presque toutes les éventualités d'un traitement de réponse à cette étape de présélection.

Notre méthode repose sur la définition même du service dans notre cadre théorique, où un service S est défini comme étant un couple qui apporte les données offertes et les informations sur le point final ($data, ctx$), à savoir que :

- $data = (d, frq, ops)$ présente respectivement les données offertes, leur fréquence de régénération possible et d'autres données que nous désignons comme étant optionnelles, où :
 - $d = (t, u)$ le type et l'unité des données.
 - $frq = (début, fin, crn)$ la date du début, la date de la fin et crn spécifie un horaire d'exécution périodique.
 - ops peut être dédiée à d'autres critères que nous présentons en forme de couple (attribut, valeur) du service.
- $Ctx = (lat, lon, ops)$ présente respectivement l'altitude, la longitude de la position géographique du point final et d'autres informations optionnelles (attribut, valeur) de ce dernier.

Maintenant que nous avons attribué une formulation mathématique à chaque concept qui constitue le service, nous suggérons la proposition suivante : $s1$ et $s2$ sont la requête et la réponse du service respectivement, $s2$ une offre de service valide pour $s1$, noté $s2 \Rightarrow s1$ seulement si les conditions suivantes sont satisfaites :

- s1 est située près de s2, la distance géographique qui sépare ces deux objets émetteurs est égale ou plus petite à un seuil déterminé.
- $s1.data.d.t = s2.data.d.t$,
- $s1.data.d.u = s2.data.d.u$,
- $s1.data.frq.début \geq s2.data.frq.début$,
- $s1.data.frq.fin \leq s2.data.frq.fin$,
- $s1.data.frq.crn \subseteq s2.data.frq.crn$,
- $\forall op1 \in s1.data.ops, \exists op2 \in s2.data.ops$ tels que : $op1.attribute = op2.attribute \wedge op1.value (\subseteq \vee =) op2.value$,
- $\forall op1 \in s1.data.ctx.ops, \exists op2 \in s2.data.ctx.ops$ tels que : $op1.attribute = op2.attribute \wedge op1.value (\subseteq \vee =) op2.value$.

L'étape de présélection commence instantanément, après qu'un objet a reçu la première réponse; son résultat se traduit par une liste incrémentée en services au fur et à mesure qu'un service est validé. Elle s'arrête à un seuil prédéterminé de temps ou de nombre, autrement dit elle permet de choisir les services qui participent à l'étape de sélection.

4.3.2 Sélection

La sélection est un processus qui permet de choisir le service le mieux adapté parmi ceux qui ont été validés dans l'étape de présélection. Ce processus fait intervenir d'autres critères comme le nombre des requêtes déployées pour atteindre le point final et la note

moyenne de chaque acteur participant afin d'affiner le choix pour une meilleure qualité de service et une vitesse d'exécution plus rapide.

Notre processus de sélection commence instantanément après qu'un premier service ait été retenu dans la liste de présélection et s'arrête avec le dernier service ajouté, pour retourner une valeur d'évaluation globale de chaque service. Cette valeur est calculée en combinant trois outils :

- le premier calcule la note moyenne des acteurs participants :

$$avgScoreOfActor(validS List[K].IDS[i]) = \frac{\sum_{j=1}^{|\text{validS List}[K].IDS[i].O|} (\text{validS List}[K].IDS[i].O[j].score)}{|\text{validS List}[K].IDS[i].O|}$$

- le deuxième calcule, en fonction de la note moyenne des acteurs participants, la note moyenne du trajet parcouru, ce dernier n'étant nul autre que le nombre de requêtes déployées :

$$avgScoreOfPath(validS List[K]) = \frac{\sum_{i=1}^{|\text{validS List}[K].IDS|} avgScoreOfActor(validS List[K].IDS[i])}{|\text{validS List}[K].IDS|}$$

- pour qu'enfin la fonction Max retienne la note la plus élevée qui déterminera le service sélectionné :

$$Max_{k=1}^{|\text{validS List}|} avgRatingOfPath(validSlist[K])$$

Où : $validS List = \{S \in sList \mid S \Rightarrow S \text{ requête}, \text{ et } |X| \text{ représente la cardinal de } X$

Avec ce processus de sélection, nous pouvons nous retrouver dans le cas où plus d'une offre de service possède la même note maximale. Nous sélectionnerons alors la première offre arrivée. Cela dit, nous devons retourner un avis à l'acteur offrant ce service pour commencer la phase d'échange d'information.

4.4 Phase d'échange d'information

Dans cette phase, le demandeur reçoit la livraison du service sélectionné selon ses critères, avec la possibilité de voir un cas de livraison du service pouvant s'appliquer plus d'une fois, et selon les conditions de fréquence indiquées dans la requête de service. La réalisation du service est entamée à partir d'une date de début et peut être effectuée à la manière d'une boucle répétitive, comme indiqué dans la demande. Par exemple : l'obtention à fréquence journalière de la température de la ville de Montréal à partir de 20 janvier 2016.

Dans notre modèle théorique, nous proposons d'utiliser le système pair-à-pair (P2P) (Figure 4.6) où ce dernier détient un modèle client-serveur dans lequel l'acteur peut jouer à la fois le rôle de client ou de nœud serveur. Le système pair-à-pair se dote d'un avantage non négligeable qui lui permet de diminuer le risque de tomber en panne. En effet, lorsqu'une panne se produit en utilisant un réseau traditionnel, les clients sont immédiatement bloqués. Par contre, avec un système pair-à-pair, si un ordinateur venait à tomber malencontreusement en panne, les autres ordinateurs sont correctement authentifiés pour prendre le relais.

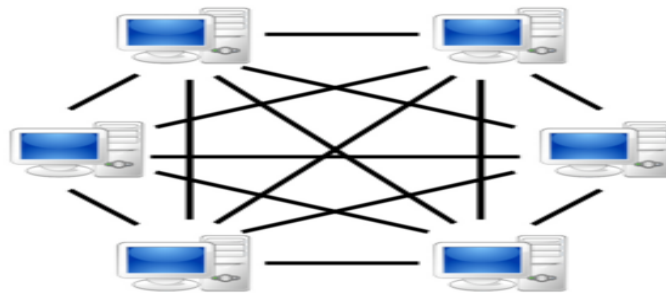


Figure 4.6 : Pair à Pair

L'utilisation du système pair-à-pair (P2P) peut satisfaire nos deux cas d'échange : direct et indirect. L'échange indirect présente le cas centralisé où les connexions traversent un objet intermédiaire appelé nœud commun. L'objet reçoit sa réponse d'un autre objet connecté à ce nœud commun. L'échange direct, quant à lui, présente le cas décentralisé où les connexions se font de manière directe.

a) Échange direct

Le diagramme des séquences UML (Unified Modeling Language) présenté dans la Figure 4.7 détaille les actions effectuées par un modèle pair-à-pair lors de l'échange direct de données. Cet échange se fait seulement d'un objet à un autre, d'un objet à un groupe ou d'un groupe à un autre.

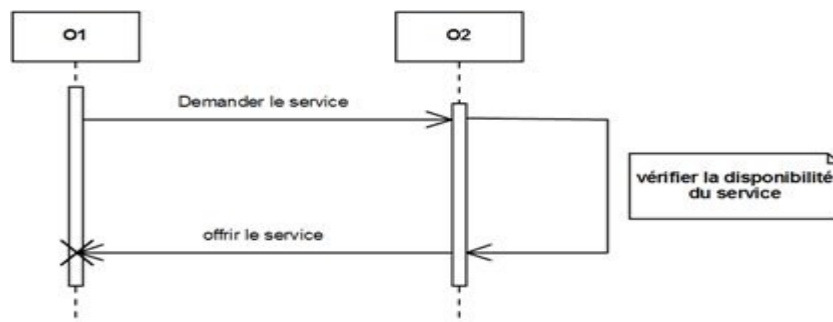


Figure 4.7 L'échange direct

Toujours dans cette figure, l'objet O₁ envoie sa requête de service vers l'objet O₂. Celui-ci vérifie la disponibilité du service demandé puis offre directement le service à l'objet O₁.

b) Échange indirect

La Figure 4.8 montre l'échange indirect des données reposant sur une séquence d'actions détaillées dans le diagramme des séquences UML.

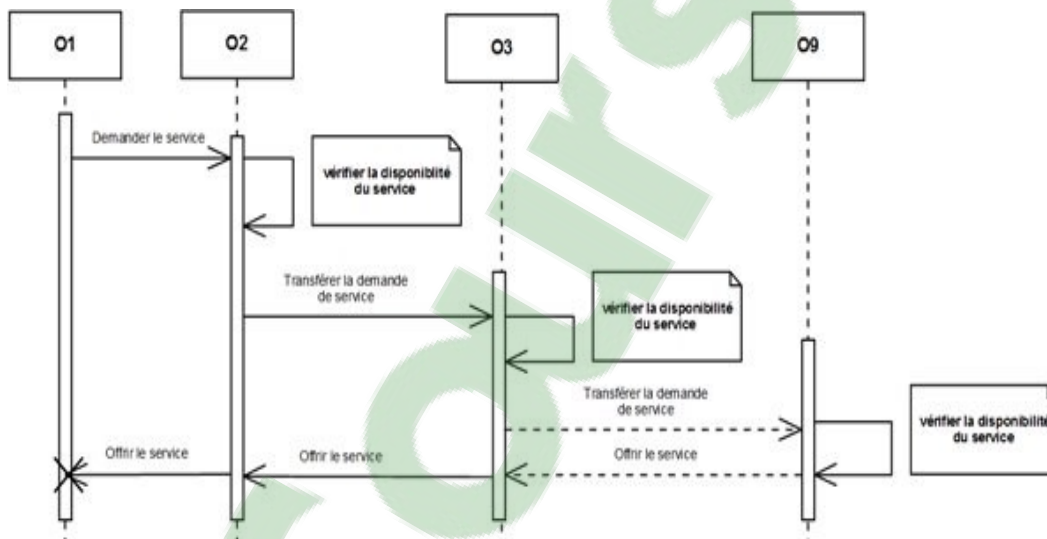


Figure 4.8 : L'échange indirect

Avant de lancer sa requête pour chercher le service, le demandeur principal O₁ doit déclarer explicitement s'il est intéressé à recevoir le service par l'intermédiaire d'un autre objet lié à l'objet visé. Puis il envoie sa requête vers l'objet O₂, lequel effectue une recherche dans sa matrice de services afin de trouver le service, et, le cas échéant, il envoie la requête vers l'objet O₃ qui peut offrir le service, directement ou indirectement. Ce dernier vérifie à son tour et envoie la requête vers l'objet O_n... etc. jusqu'à ce qu'une

réponse positive apparaisse, O_9 dans notre exemple. Le service S_9 sera donc réacheminé vers O_1 en respectant le même trajet emprunté par la requête de l'objet O_1 .

4.5 Conclusion

Le présent chapitre s'est fixé pour objectif de présenter un nouveau modèle théorique pour la collaboration et le partage de données dans l'Internet of Things (IoT). Notre modèle théorique se compose de trois étapes majeures résumées schématiquement à la Figure 4.9 et répétées à chaque itération.

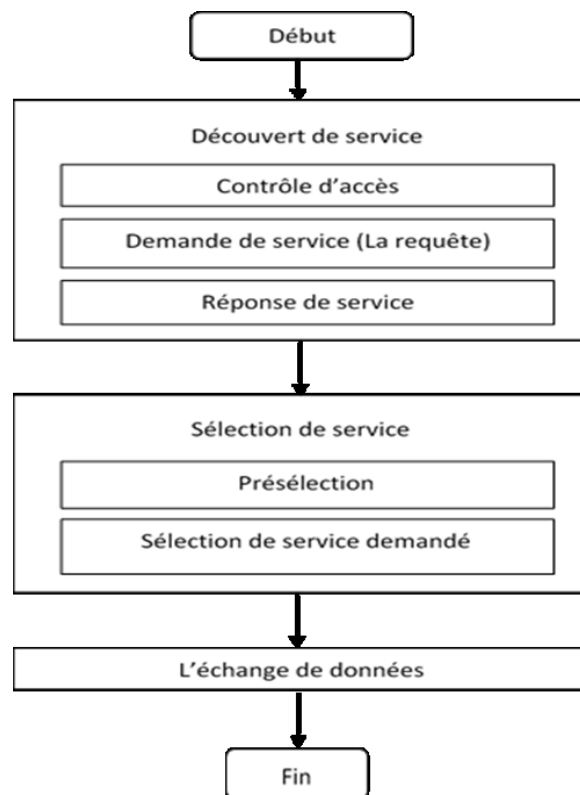


Figure 4.9 : Déroulement de notre modèle

La première étape est celle de la découverte de service à travers laquelle l'acteur peut chercher un service en lançant un ensemble d'informations formant la requête, vers les

objets potentiellement capables d'offrir le service concerné. La réponse retournée confirme la présence et l'accessibilité du service demandé.

La deuxième étape consiste à sélectionner le service demandé. Lorsque le demandeur reçoit de manière effective un certain nombre de réponses, il commence par valider la correspondance entre sa requête et les services offerts. Puis, il sélectionne le service selon la distance la plus courte entre lui et l'acteur qui répond à sa demande (l'objet final) et selon l'évaluation moyenne de chaque acteur participant.

La troisième et dernière étape a pour but de livrer le service comme décrit, selon l'offre de service qui a été choisie.

Le chapitre suivant aborde l'implémentation de notre cadre théorique et expose les résultats des expérimentations qui ont pour finalité de valider les différents concepts de notre modèle.

CHAPITRE 5

IMPLÉMENTATION D'UN PROTOTYPE ET TESTS

Le présent chapitre traite de la traduction pratique de notre modèle théorique afin de rendre opérationnelles nos propositions énoncées au chapitre précédent. Dans ce modèle, nous avons intégré un système contribuant à la collaboration des objets intelligents au sein de l'Internet of Things.

D'abord, nous présentons notre langage dédié (Domain Specific Language) entièrement développé dans un environnement IDE-Eclipse qui sert à traduire les concepts de notre modèle en langage de programmation. Ensuite, nous détaillons l'expérimentation réalisée en laboratoire sur une nouvelle technologie qui permet de convertir la plupart de nos objets ordinaires en objets connectés. Enfin, nous présentons notre système virtuel de collaboration pour la surveillance et le partage des données climatologiques, qui est en fait un site Web dynamique.

5.1 Domain Specific Language: IoTCollab

Nous avons développé un langage spécifique à un domaine (DSL) portant la dénomination IoTCollab, conçu pour faciliter l'intégration des différents concepts introduits par notre modèle théorique de partage de données à haut niveau d'abstraction. Le DSL nous aide à éviter de modifier les langages existants pour définir des systèmes de partage dans l'Internet of Things (IoT) basé sur le modèle théorique proposé.

En ce qui concerne l'utilisation, nous avons distingué deux types de DSL [60] :

- Le DSL interne, étant considéré comme un ensemble de couches ou de méthodes particulières pratiquées sur un langage de programmation traditionnel afin de lui donner un caractère spécifique et efficace. L'idée est de pouvoir profiter de l'infrastructure et des outils de l'environnement de développement de langages généralistes comme Java, C #, Python, Ruby, ...etc.
- Le DSL externe, dont la programmation est entièrement indépendante, ayant sa propre infrastructure et sa propre syntaxe, et pour lequel il faut écrire un analyseur complet afin de pouvoir le traiter.

Dans notre cadre théorique, nous avons opté pour l'utilisation de Xtext (version 2.6) [61] comme plateforme de développement de DSL. Xtext a été choisi principalement pour son ouverture et son intégration alliées à d'autres outils de modélisations libres et aux générateurs de code tels que l'EMF [62] et l'Xtend [63].

La mise en œuvre de l'IoTCollab passe par deux étapes : tout d'abord, il faut développer la grammaire du langage. Ensuite, élaborer un ensemble de modèles pour la création du code. Alors que la grammaire est utilisée pour représenter le métamodèle du langage et analyser ses concepts, les modèles sont quant à eux utilisés pour générer des artefacts comme une sortie d'un modèle (génération de code).

Une vue partielle de la grammaire de IoTCollab est représentée dans la Figure 5.1 ci-dessous.



```
IOTCollabDSL.xtext | Terminals.xtext | InternalIoTCollabDSL.g
IOTCollabModel:
  "Model" name-ID ':'
  "Users:" users+=User+ "EndUsers"
  "IOTs:" iots+=Iot+ "EndIOTs"
  "Services:" services+=Service+ "EndServices"
  "Actors:" actors+=Actor+ "EndActors"
  "EndModel"
;

Service:
  "Service" name-ID ':'
  "Data:" data=Data "End Data"
  "Context:" context=Context "EndContext"
  "EndService"
;

Context:
  "Latitude:" lat=DECIMAL
  "Longitude:" lon=DECIMAL
  ops+=Options*
;
```

Figure 5.1 : Une vue partielle de la grammaire de l'IoTCollab[64]

La Figure 5.2 montre une vue du système de partage de données complètement écrit en utilisant un langage IoTCollab.

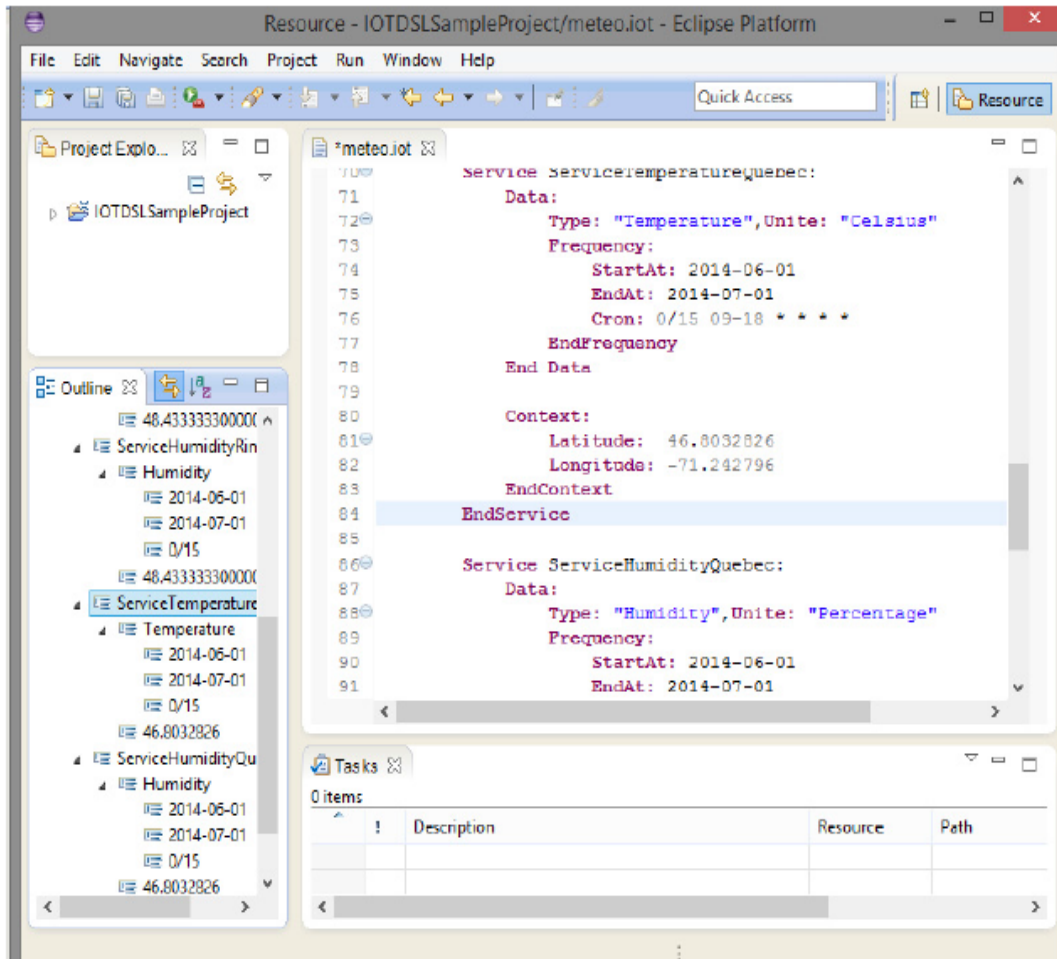


Figure 5.2 : Une vue partielle d'un système de partage de données de météorologie écrit utilisant le langage IoTCollab[64]

5.2 Expérimentation en laboratoire

La phase d'implémentation en laboratoire est inévitable pour cerner l'objectif de notre cadre théorique, mais tout aussi importante pour que nous puissions avancer dans notre recherche, puisqu'elle nous permet d'implémenter notre modèle dans un environnement réel afin de le valider et d'en dégager les principales limites auxquelles il sera confronté.

Notre projet en laboratoire consiste à développer un service de météorologie pour le partage de données, en implémentant plusieurs capteurs de température et d'humidité dans différentes régions, connectés à un serveur Web local accessible par Internet afin de partager les données collectées entre différents utilisateurs. Pour que notre projet puisse voir le jour, il faut passer par deux étapes essentielles : i) construire un capteur connecté, en associant un microcontrôleur muni d'une connexion Wifi et d'une mémoire à un capteur de température et d'humidité, puis développer un programme qui permettrait de lire les données émises par le microcontrôleur (cette partie du projet a été faite par notre collègue du premier cycle Alain Cariou qui nous a donné son consentement à reproduire n'importe quel contenu de son rapport dans ce mémoire); ii) créer un serveur local, pour chaque capteur installé, dans lequel on développera une application pair-à-pair en utilisant notre langage de programmation IoTCollab pour le partage de données. Cette partie a été assez compliquée à réaliser en raison du manque de temps et de matériel. C'est la raison pour laquelle nous avons eu recours à une expérimentation virtuelle.

5.2.1 Projet du capteur connecté

5.2.1.1 SparkCore

Un SparkCore est par définition un microcontrôleur programmable de très petite taille auquel est associé un module Wifi. Voici les caractéristiques de ses composants [65] :

- Le microcontrôleur est de type STM32F01 avec un cœur ARM Cortex-M3 32-bit fonctionnant à 72 MHz, possédant 128 ko de Flash et 20 ko de mémoire vive (RAM);

- Le contrôleur Wifi est un contrôleur Wifi Texas Instruments SimpleLink CC3000 compatible avec les réseaux 802,11 b/g, d'une portée de 30 à 90 m (100-300 pieds);
- Il dispose aussi d'une puce fournissant 2 Mo de Flash externe;
- 8 entrées/sorties numériques;
- 8 entrées analogiques;
- Une prise micro-USB supportant un voltage de 3,3 V.

Fonctionnant sur le même principe qu'un Arduino (carte de circuits imprimés sur lesquels il est possible de brancher toutes sortes d'appareils; cette carte se programme sur l'ordinateur via un câble USB)[66], différents éléments peuvent être ajoutés au SparkCore afin d'en faire un puissant outil de développement pour réaliser des projets divers et variés. Entièrement libre, il dispose aussi d'un service au *Cloud* permettant au SparkCore d'être connecté à Internet pour ainsi être accessible de n'importe où grâce à son application mobile Tinker et à son API REST.

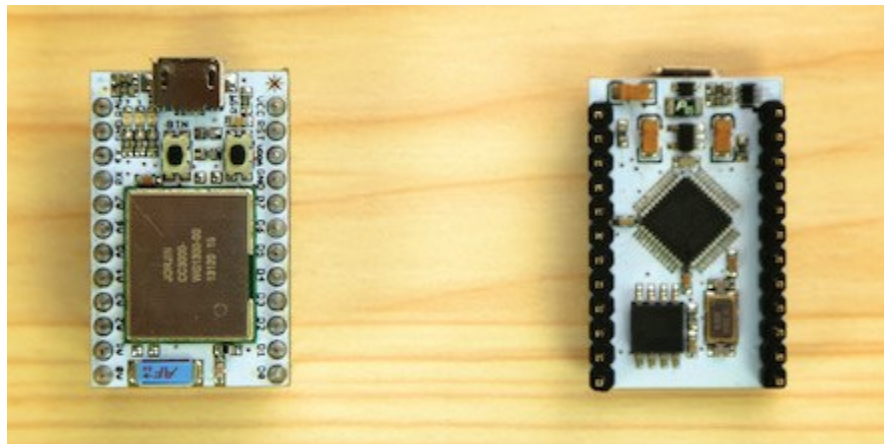


Figure 5.3 : Image d'un Spark Core tirée du kick starter du projet [67]

5.2.1.2 Outils nécessaires

Plusieurs outils sont nécessaires à la réalisation et au test du projet, à savoir :

1) un mobile ou une tablette fonctionnant sous Android ou iOS permettra d'utiliser l'application Tinker, l'application mobile officielle du SparkCore qui nous permet d'interagir avec le firmware de celui-ci à travers le *Cloud*, pour contrôler et lire les tensions de ces 16 broches (8 numériques et 8 analogiques) sans écrire une seule ligne de code. Bien évidemment, il est possible de s'en passer en développant notre code firmware personnel et de communiquer avec le SparkCore sans traverser le *Cloud*.

2) Un compte Spark est aussi nécessaire pour pouvoir utiliser l'interface de développement disponible en ligne pour les utilisateurs du SparkCore. Nous avons donc créé un compte pour ce projet avec l'identifiant :

- Adresse courriel : spark.uqar@gmail.com

3) Un capteur DHT22 est également utilisé pour mesurer la température et l'humidité ambiantes.

4) un logiciel permettant de se connecter en série au SparkCore permettra de communiquer localement avec ce dernier et de l'enregistrer. Dans notre cas, nous avons préféré utiliser CoolTerm pour sa simplicité.

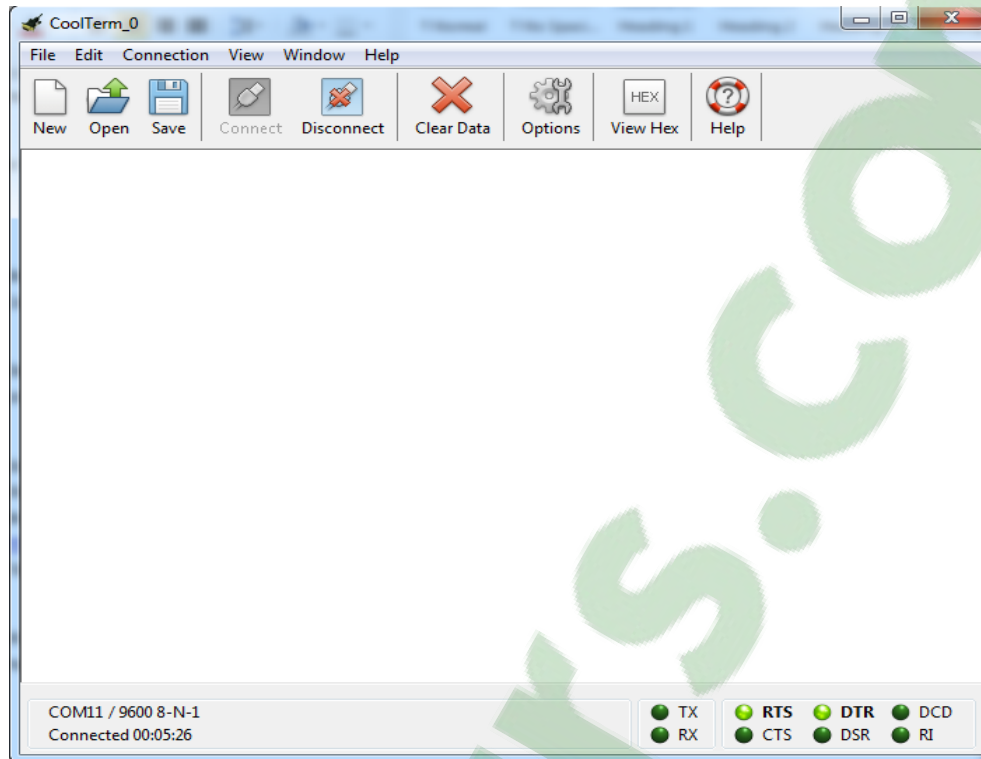


Figure 5.4: Interface de CoolTerm sous Windows

5.2.1.3 Connexion au SparkCloud

Maintenant que les principaux outils sont présentés, l'étape suivante est le « claiming » du Core. Cette étape consiste à lier le SparkCore au SparkCloud, ce dernier rassemblant tous les services de communication et de développement disponibles en ligne par Spark, qui nous permettent de communiquer et de reprogrammer notre SparkCore via notre compte Spark. C'est à partir de cette étape que le microcontrôleur de notre SparkCore démarre et maintient une connexion TCP avec le SparkCloud. Bien qu'il semble possible de joindre le SparkCore directement avec l'API Tinker, nous n'avons pas réussi à le faire. Nous montrerons donc uniquement la procédure par câble USB.

Pour commencer, nous nous connectons en série au SparkCore via l'application CoolTerm. Une fois le port de connexion correctement établi, nous pouvons récupérer l'identifiant du SparkCore qui sert à créer notre compte personnel sur le SparkCloud (Figure 5.5).

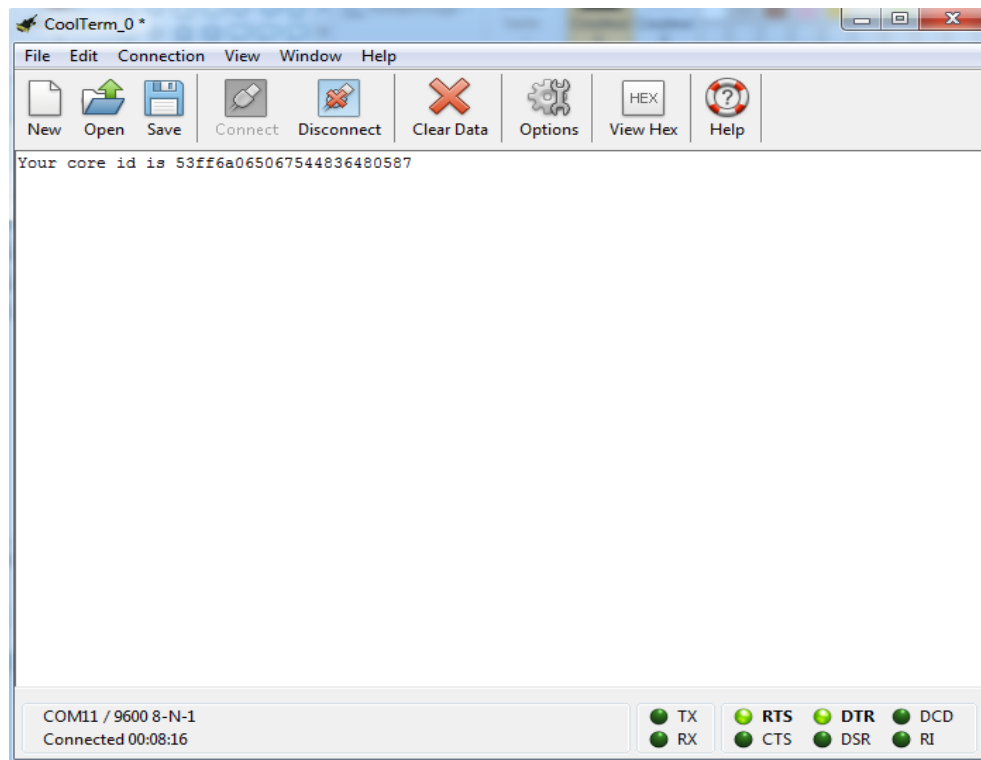


Figure 5.5 : Identifiant du SparkCore

Ensuite, nous configurons le Wifi du SparkCore en saisissant le SSID et le mot de passe de notre connexion Wifi, ce qui permet au SparkCore de se connecter à l'Internet.

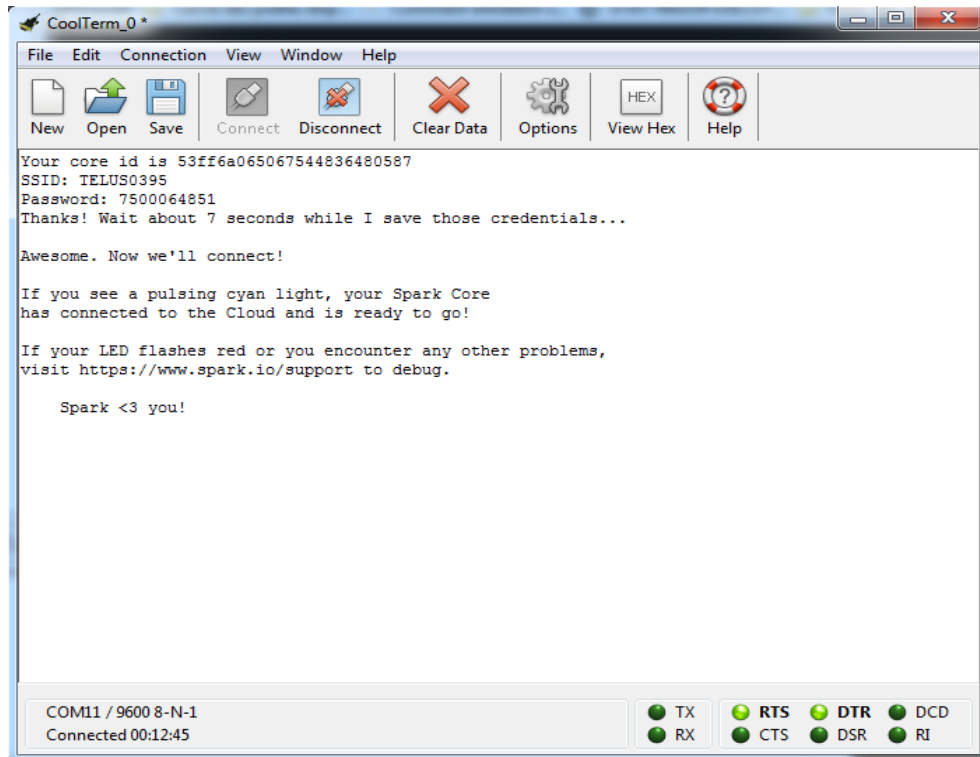


Figure 5.6 : Connexion du SparkCore à internet

Finalement, il suffit d'enregistrer le SparkCore dans le compte Spark avec son identifiant, pour que la connexion SparkCore/SparkCloud s'établisse.

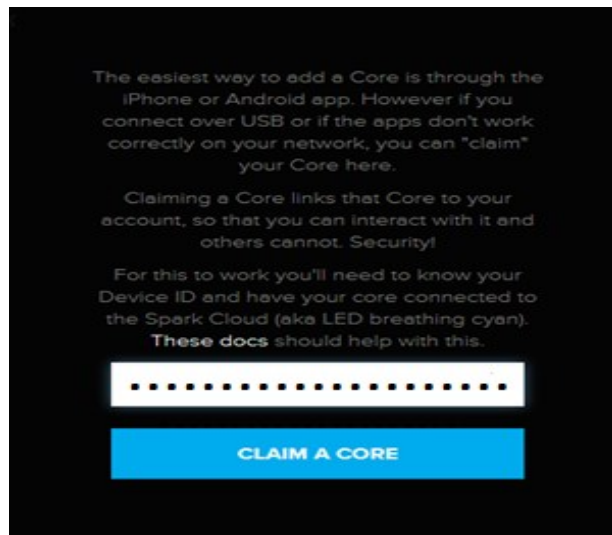


Figure 5.7 : Identification du SparkCore au SparkCloud

5.2.1.4 Test préliminaire

Afin de vérifier le bon fonctionnement des différentes manipulations effectuées, nous avons réalisé un test en utilisant l'application Tinker et une lampe DEL reliée au pin numérique D2 du SparkCore. Ce test consiste à modifier la tension électrique envoyée vers la lampe par la simple manipulation de la broche respective au pin D2 au niveau de l'application, comme le montrent les figures ci-dessous :

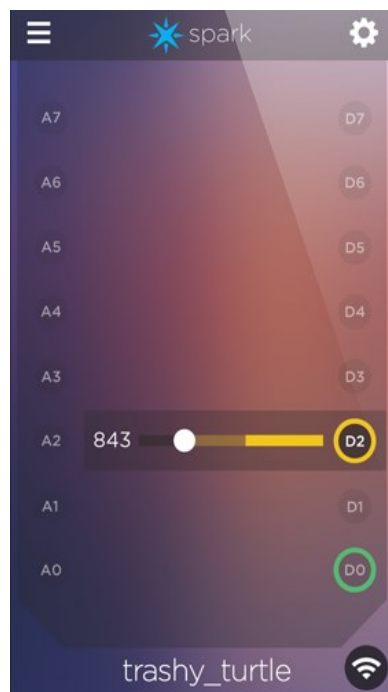


Figure 5.8 : Contrôle des différents pins du Spark Core [32]

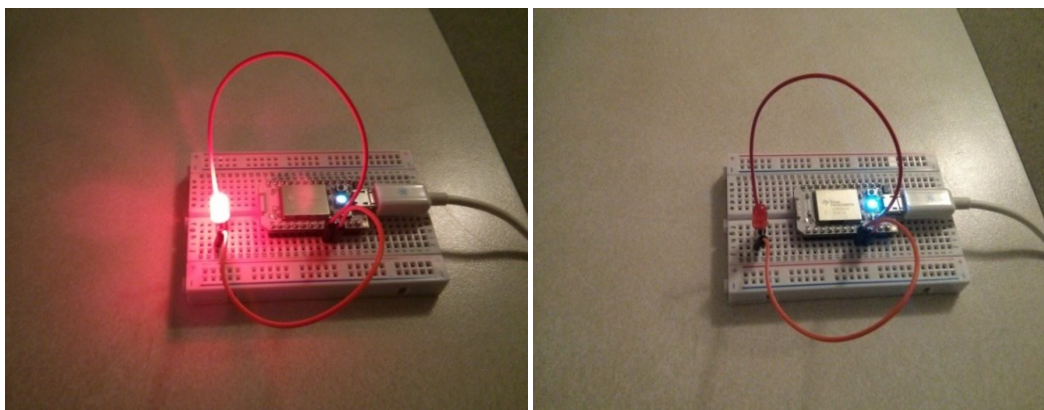


Figure 5.9 : DEL activée et DEL désactivée

5.2.1.5 Test du capteur

a) Branchement

Le branchement du capteur est simple. Dans notre cas, nous disposons de trois fils :

- Le fil rouge est le fil d'alimentation : branché à la borne 3V3.
- Le fil noir est le fil de mise à la terre : branché à la borne GND.
- Le fil jaune est le fil de transfert de données : branché à la borne qui sera réservée à cette fonction. Dans notre cas nous avons choisi arbitrairement le pin digital 2 (Figure 5.10).

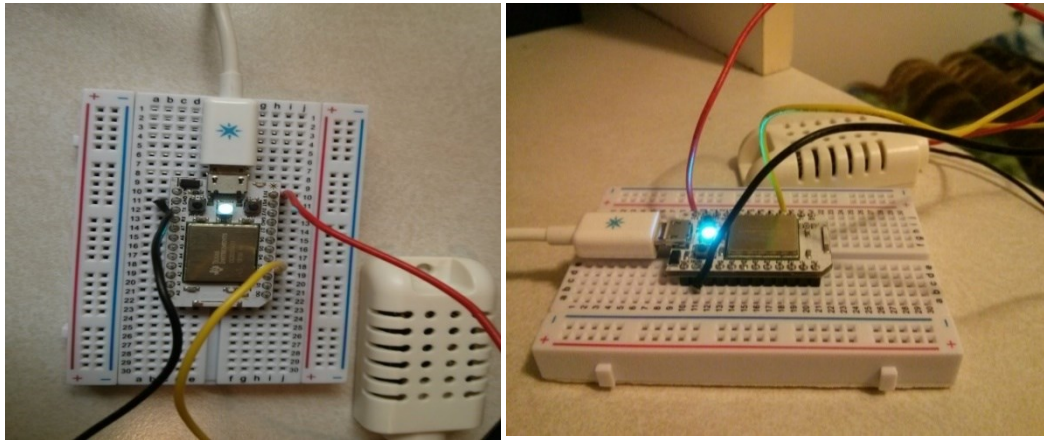


Figure 5.10 : Montage du capteur et branchement des fils

b) Présentation du code

Il faut savoir qu'une application SparkCore se compose de deux fonctions principales. La fonction *setup()* qui est appelée au lancement du programme servant à initialiser une partie des données et à effectuer les premières actions. Mais aussi la fonction *loop()* qui contient la boucle principale du programme. Cette fonction est, comme son nom l'indique, appelée en boucle afin de répéter les mêmes actions durant toute la durée de vie du programme. Elle constitue la colonne vertébrale de ce dernier.

Afin de pouvoir utiliser le capteur DHT22, la librairie DHT (écrite en CPP) a été ajoutée au programme. Elle a été développée par Adafruit, le fabricant de ces capteurs. Elle contient des fonctions permettant de déterminer facilement la température (en Celsius et Fahrenheit), le taux d'humidité et l'indice de chaleur. À partir de cela, il ne reste plus qu'à flasher le programme sur le SparkCore et à ouvrir une connexion en série, pour afficher les données. Ci-dessous le résultat obtenu avec le premier programme :

```
Hit a key to begin.  
Hit a key to begin.  
Beginning of the measurement with DHT sensor.  
Humidity: 37.70 %Temperature: 25.50 *C 77.90 *F.Heat index: 78.54 *F  
Humidity: 37.40 %Temperature: 25.50 *C 77.90 *F.Heat index: 78.52 *F  
Humidity: 37.30 %Temperature: 25.50 *C 77.90 *F.Heat index: 78.52 *F  
Humidity: 37.30 %Temperature: 25.40 *C 77.72 *F.Heat index: 78.43 *F  
Humidity: 37.20 %Temperature: 25.50 *C 77.90 *F.Heat index: 78.51 *F  
Humidity: 37.10 %Temperature: 25.40 *C 77.72 *F.Heat index: 78.42 *F  
Humidity: 37.10 %Temperature: 25.40 *C 77.72 *F.Heat index: 78.42 *F  
Humidity: 37.00 %Temperature: 25.40 *C 77.72 *F.Heat index: 78.41 *F
```

Figure 5.11 : Résultats du capteur DHT22

5.2.2 Projet du serveur local

Ce projet consiste à utiliser la librairie TCPClient implémentée nativement par le SparkCore. Cette librairie permet de connecter aisément un client TCP à un serveur. Ainsi, au lieu d'afficher les données en série sur le terminal de l'utilisateur, le programme va créer ici une instance d'un client et le connecter au serveur sur lequel il transmettra les données récupérées grâce au capteur. La fonction *setup()* initialise le client et le connecte au serveur tandis que la fonction *loop()* récupère les données et les envoie à chaque tour de boucle si le client est toujours connecté au serveur. À partir de cela, l'application de partage et de collaboration prend le relais. Malheureusement, ce projet n'a pas pu être testé à cause des problèmes liés au développement du serveur et de l'application pair-à-pair.

5.3 Expérimentations et simulations

En somme, notre projet consiste à réaliser un système collaboratif de partage de données météorologiques entre plusieurs utilisateurs dispersés dans différentes régions. Un utilisateur dans notre modèle théorique est un acteur qui retient des données générées par des objets (capteurs), ces données pouvant être échangées entre différents acteurs à

travers un réseau non structuré (décentralisé) où le routage des requêtes et des réponses se fait par propagation sélective. En tentant de reproduire ce concept dans une expérience réelle, nous avons échoué en raison de plusieurs contraintes, notamment le manque de dispositifs (un seul capteur et un seul SparkCore) et l'ampleur de la tâche à réaliser (application pair-à-pair). Suite à cela, et afin d'appréhender les idées, nous avons créé un site Web où nous simulons le fonctionnement de notre application de partage pair-à-pair.

Le site en question englobe toutes les fonctionnalités d'échanges et de collaborations de notre application et les traduit sous forme d'un système communautaire de coopération. Le but est de construire une plateforme où différents utilisateurs s'identifient et offrent leurs services contre d'autres services.

5.3.1 Outils de développement

Plusieurs langages de programmation sont utilisés dans le développement de notre site Web, présentés ci-dessous en quatre catégories :

- Langages de programmation
 - PHP 5.5.8 : langage de développement du site, plus précisément pour dynamiser le site
 - JavaScript : pour avoir importé des données externes comme les météos d'un autre site vers notre site
 - HTML5 : Langage pour l'affichage
 - CSS3 : langage de mise en forme (design)
- Base des données
 - MySQL 5.6.15 : stockage des données

- Serveurs Web
 - EasyPHP DevSERVER v14.1 VC11 : serveur de déploiement
 - Apache version 2.4.7 : serveur HTTP
- Module complémentaire
 - API Google Maps : pour pouvoir intégrer une carte dans le site

5.3.2 Prototype

Après avoir détaillé l'environnement de développement, nous présentons dans cette section les interfaces d'utilisation de notre site Web.

5.3.2.1 Identifications et localisation

L'interface d'accueil (Figure 5.12) présente deux espaces : un espace de « Connexion » si l'utilisateur a déjà son propre compte et un espace d'« Inscription » pour tout nouvel utilisateur souhaitant s'inscrire dans la base de données. Cet espace contient plusieurs champs de saisie, dont l'« Adresse », qui seront utilisés pour la localisation une fois le compte validé.

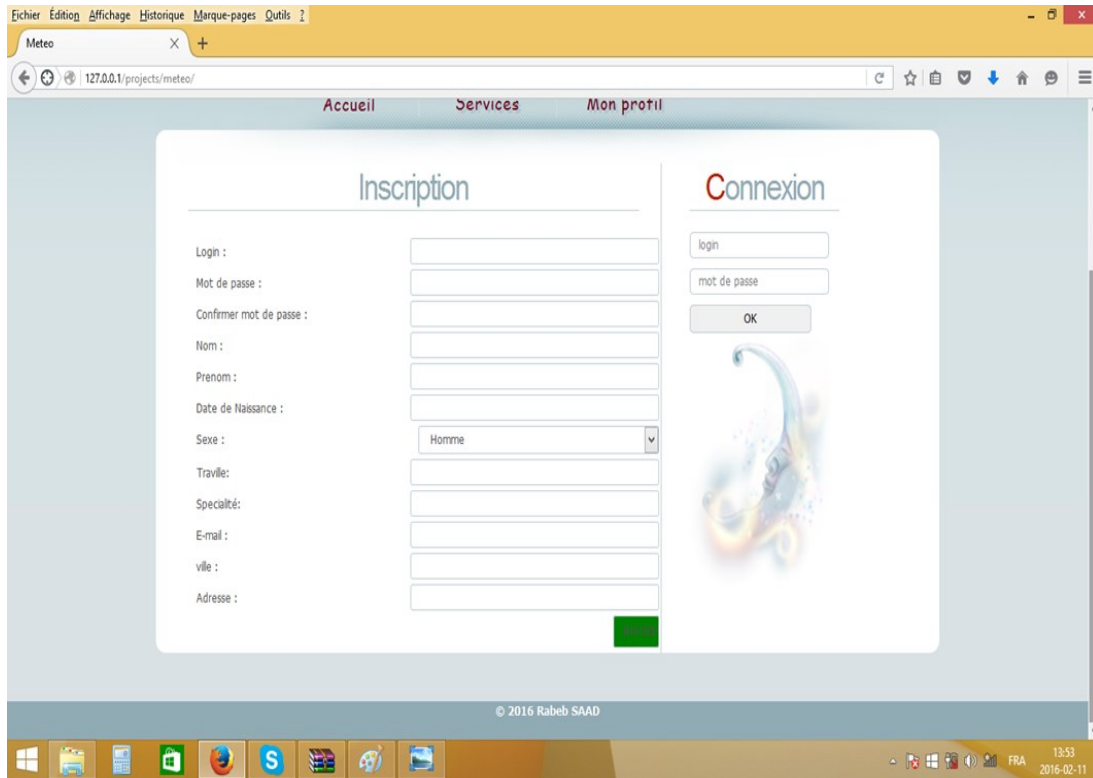


Figure 5.12 : Interface principale de notre application

En s'identifiant, l'utilisateur est redirigé vers l'interface de localisation (Figure 5.13) qui lui permet de visualiser l'emplacement de chaque utilisateur avec un aperçu sur son identité. Un espace de recherche par nom ou par adresse a été ajouté à cette interface, puisqu'il sera d'une grande utilité dans le cas où il y a plusieurs utilisateurs.

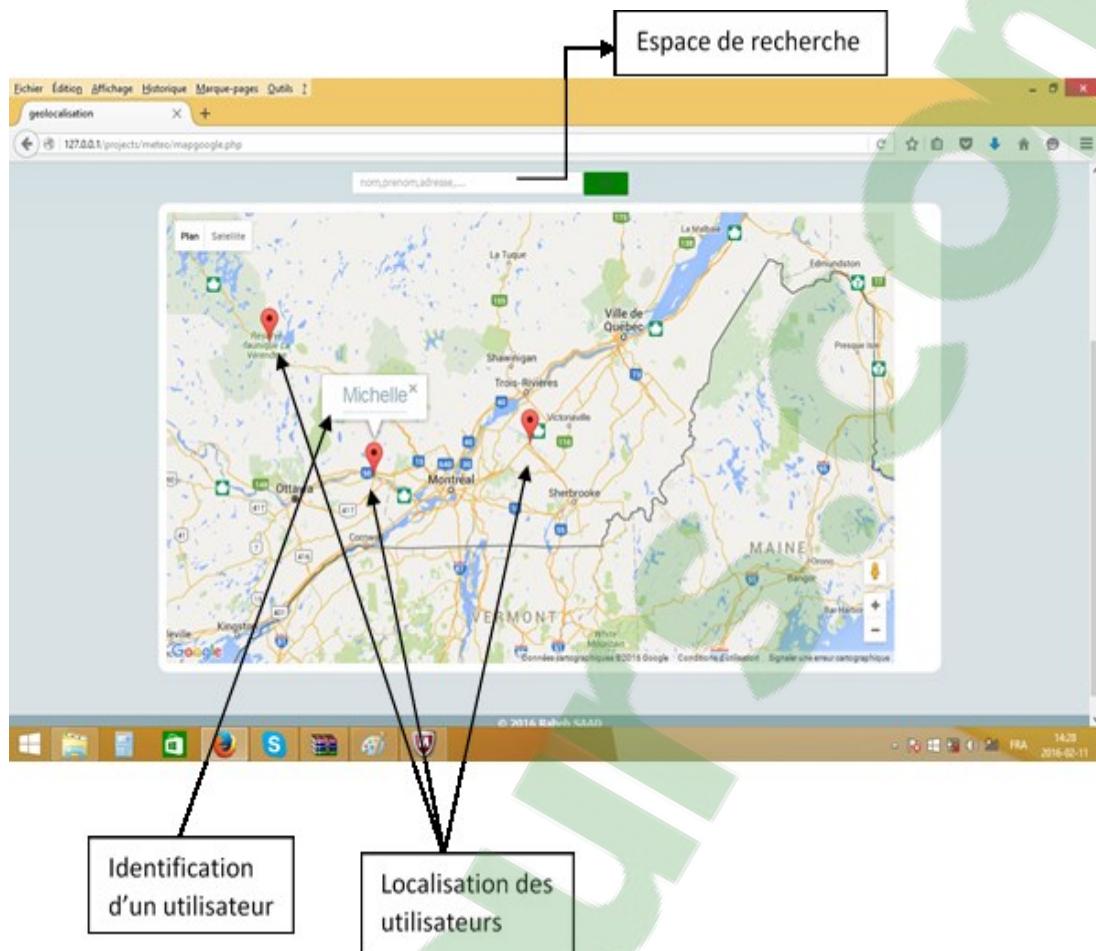


Figure 5.13 : Interface de localisation Google Maps.

5.3.2.2 Demande de services

Pour effectuer une demande, l'utilisateur doit accéder à la section « Services » à partir de l'interface de son profil, laquelle affiche toutes les informations personnelles enregistrées par lui lors de son inscription en plus des données générées par sa source d'actualités météorologiques (Figure 5.14).

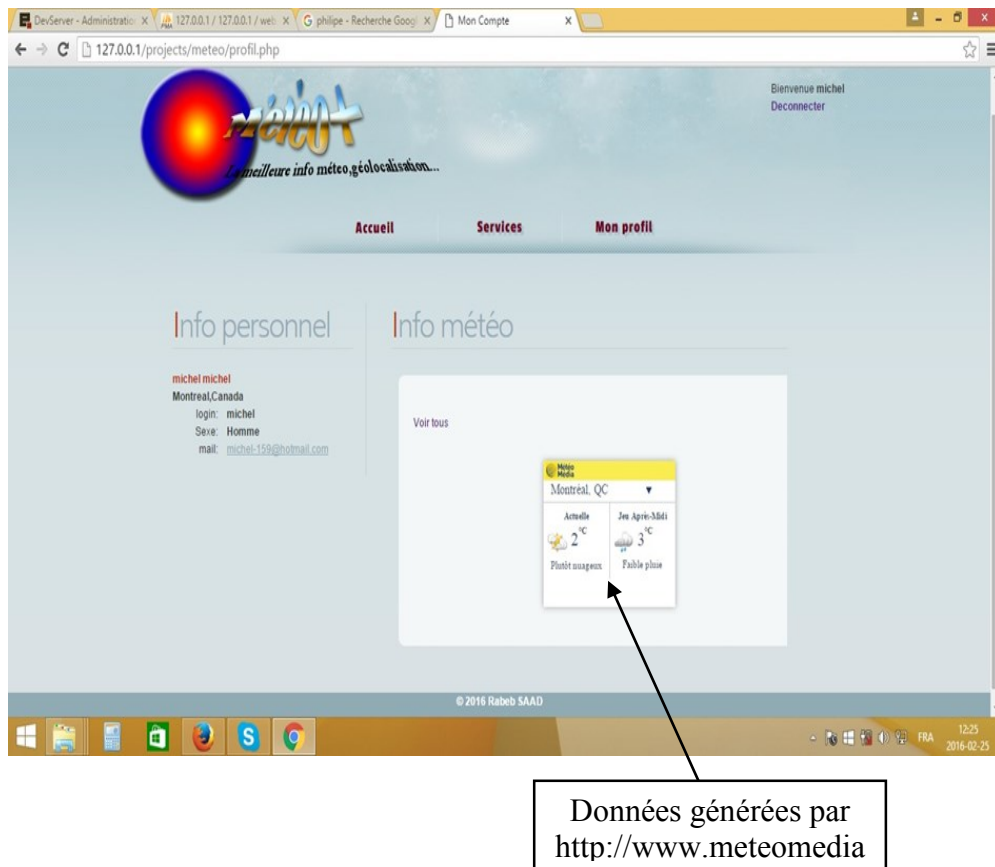


Figure 5.14 : Profile utilisateur

Rendu sur l'interface « Services », l'utilisateur peut formuler sa demande de service grâce à un formulaire de demande adapté pour couvrir le maximum possible de critères de service, afin de faciliter son traitement auprès des autres utilisateurs. Son formulaire de demande contient : l'objective de sa demande, la date de début et la date de fin entre lesquelles il aura besoin du service, la fréquence d'envoi, la manière dont il sera servi, soit directement, soit indirectement, par d'autres utilisateurs liés à ceux qui seront contactés (Figure 5.15).

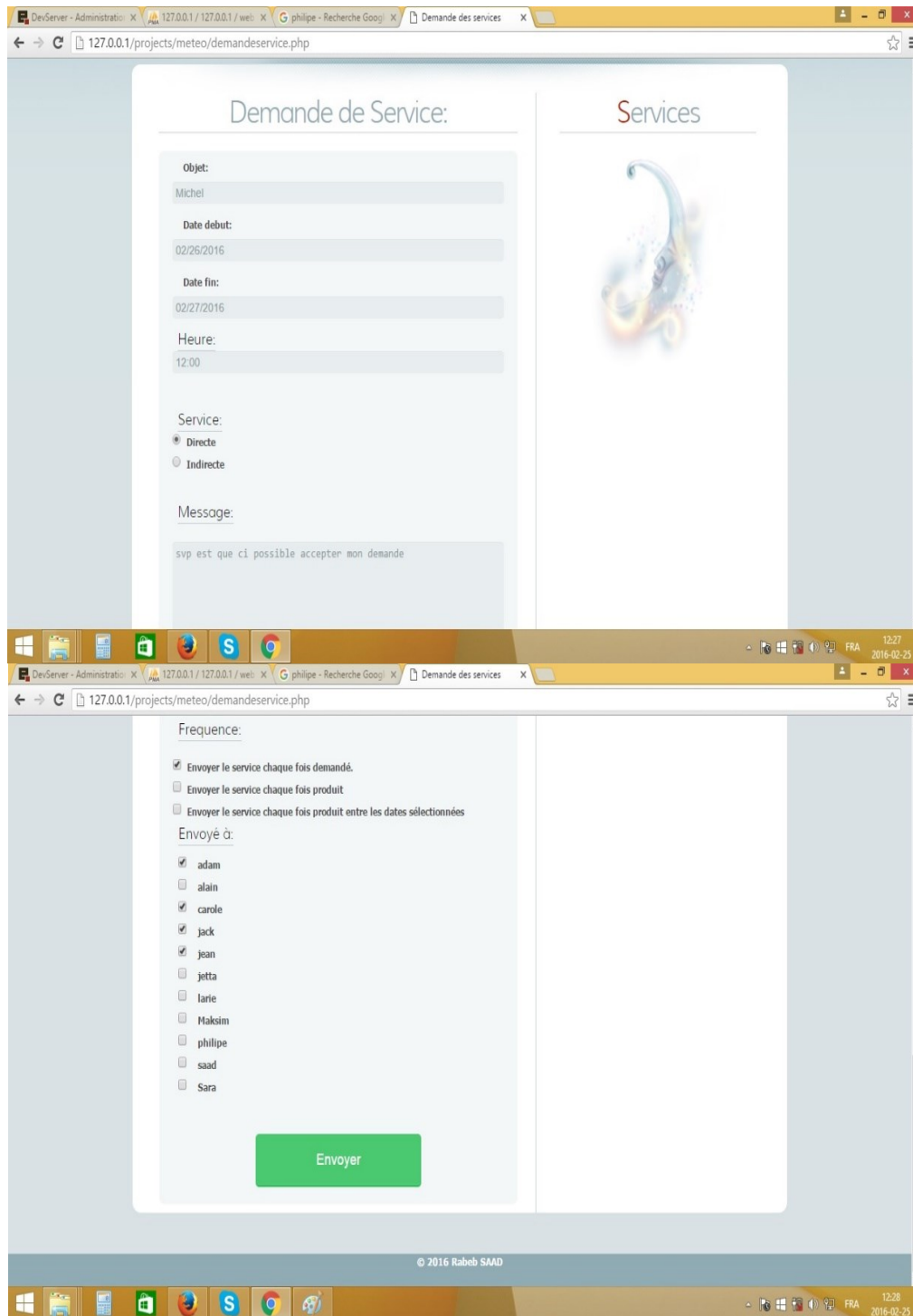


Figure 5.15 : Demande de service

5.3.2.3 Service reçu

Les services reçus ont la même conception de la demande, ce qui facilite leur évaluation auprès du demandeur (Figure 5.16). Ce dernier peut recevoir plusieurs réponses pour une seule demande et, grâce à un système de notation par étoiles, il pourra attribuer une valeur de satisfaction pour chaque service. Cette valeur ne sera pas retournée vers le serveur, car elle sert plutôt à l'évaluation de l'historique d'échanges du demandeur avec d'autres utilisateurs.

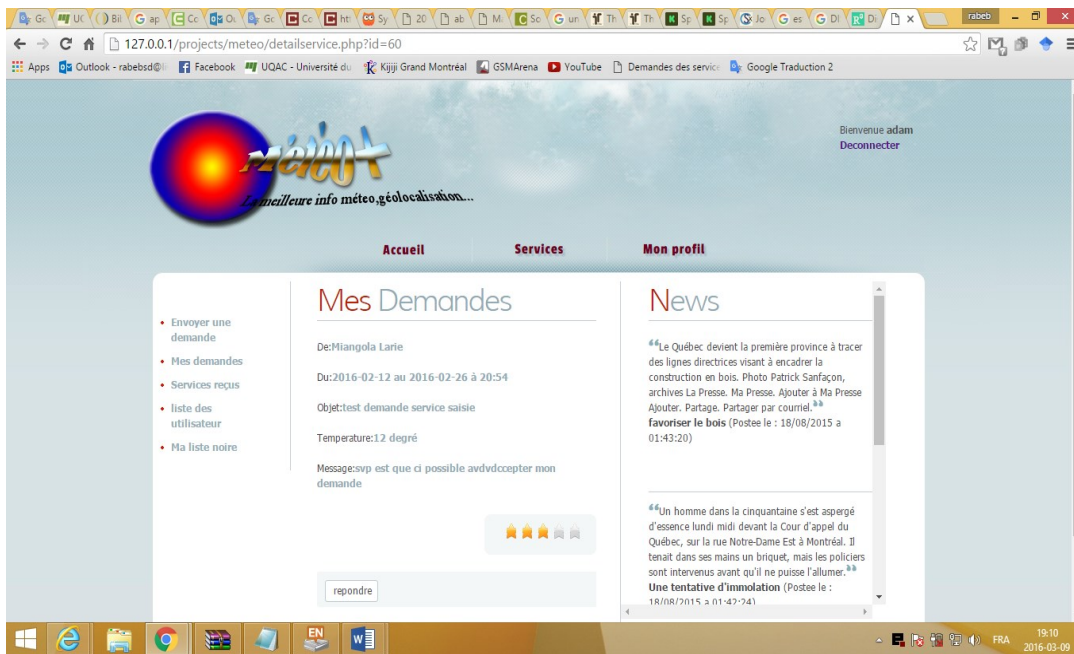


Figure 5.16 : Demandes reçues

5.4 Conclusion et résultats obtenus

Avant de présenter les résultats de nos expérimentations, il faut souligner que notre cadre théorique vient de mettre en œuvre un nouveau concept de communication et de collaboration dans l'Internet of Things (IoT), lui-même un nouveau concept.

L'implémentation de ce concept s'est avérée très enrichissante et instructive, notamment en ce qui a trait aux limites de nos théories et de celles des nouvelles technologies développées pour le monde de l'IoT.

Le travail pratique décrit dans ce chapitre présente le premier pas vers la réalisation de notre perspective, particulièrement dans les trois volets suivants :

D'abord : la mise en place d'un langage dédié qui répond à toutes nos approches proposées et qui nous permet de les intégrer dans différentes interfaces de programmations;

Ensuite : la réalisation de notre premier objet connecté, ce qui nous a permis d'explorer les nouvelles technologies dans le domaine de l'Internet of Things, dont le SparkCore et sa capacité de convertir tout objet à base d'un système électronique en un objet connecté, d'où nous avons pu récupérer des données émises par un capteur météorologique. Cette technologie est capable même d'embarquer un serveur Web à part entière, ce qui ouvre le champ pour le développement d'applications embarquées de communication et d'échange de données;

Enfin : les résultats du test virtuel sur le site Web créé ont été satisfaisants, car celui-ci nous a permis de manifester notre concept de découverte de services, où nous avons intégré : l'identification, la géolocalisation, la propagation de la demande et la conception du service. Ce prototype de l'application que nous voulons développer nous a été très instructif en matière de développement et de programmation.

CHAPITRE 6

CONCLUSION

L'émergence de l'Internet of Things (IoT) offre un grand potentiel pour le développement de nouveaux services et applications connectant le monde physique au monde virtuel. Le problème abordé dans ce mémoire est comment exploiter ces services qui peuvent être offerts par différents appareils. Pour ce faire, nous avons mis en exergue un modèle théorique de collaboration et de partage de données dans l'IoT. Ce modèle est destiné à favoriser les interactions automatiques entre des objets par la mise en place d'une stratégie permettant la propagation transitive du couple requête et réponse dans un réseau d'objets. Nous résumons par la suite nos réalisations ainsi que nos travaux de recherche futurs.

6.1 Contribution

L'objectif de ce mémoire est de construire un modèle théorique qui apporte une nouvelle vision sur la collaboration dans le monde de l'Internet of Things (IoT). Ce modèle est basé sur la décentralisation du système de recherche et de récupération des données à l'inverse des travaux réalisés jusqu'à maintenant, qui se sont focalisés sur la réalisation de plateformes ou d'applications pour exploiter et contrôler les objets connectés. Nous avons aussi développé un langage DSL appelé IoTCollab dédié à la rédaction et l'intégration des différents concepts introduits dans notre modèle collaboratif.

Notre travail vise à profiter des technologies Web actuelles pour faciliter la communication, le transport et l'acheminement des données entre les objets. Cela nous a menés à développer une approche de collaboration selon les trois phases suivantes.

- premièrement, la phase de découverte, basée sur la propagation de la requête à travers les croisements des réseaux de connaissance de chaque objet, selon son contexte et selon une politique simple de contrôle d'accès, où nous avons développé un algorithme.
- deuxièmement, la phase de sélection, basée sur deux mécanismes qui se complètent l'un l'autre permettant ainsi de sélectionner le service approprié parmi ceux trouvés dans la phase de découverte. Le premier valide les services qui correspondent aux exigences de la demande, le deuxième calcule la note moyenne de chaque service selon un certain nombre de critères.
- et la phase finale consiste à l'échange de données. Nous avons donc proposé l'utilisation des principes du pair à pair, puisque l'objet peut être à la fois client et serveur dans le même service. Le concept du pair à pair est plus pertinent dans les cas de services complexes ou composés, car il permet de satisfaire le client par le biais d'une ou de plusieurs ressources en même temps.

En parallèle, nous avons développé un langage spécifique à un domaine (DSL) nommé IoTCollab pour faciliter la programmation et l'intégration des différents concepts introduits dans notre modèle à un haut niveau d'abstraction. Cela ayant été fait dans le but d'affiner les langages présents afin d'être en mesure de définir des systèmes de partage basés sur notre modèle théorique proposé.

6.2 Perspectives

6.2.1 Limites

De manière substantive, notre approche méthodologique répond aux différents objectifs préalablement fixés. Pourtant, il reste encore certaines limites sur lesquelles nous devons travailler pleinement dans un avenir proche. Il serait judicieux de pouvoir tester notre modèle dans un cadre expérimental pratique afin de souligner les limites auxquelles il sera soumis. Nous pouvons tout de même constater qu'il existe d'autres limites comme la possibilité d'une surcharge du réseau induite par notre stratégie de propagation, et le manque de mécanismes de contrôle d'accès pour couvrir d'autres aspects de sécurité comme l'identification, l'authentification, l'intégrité des données, etc. Ainsi, il reste encore des pistes de recherche à explorer pour peaufiner et optimiser notre modèle.

6.2.2 Travaux futurs

Comme nous pouvons le constater, notre projet pour l'avenir vise une autonomie complète des objets, dans le sens où ils peuvent collaborer sans intervention humaine, en restant fidèle à notre modèle actuel et en explorant d'autres pistes de recherche et de pratique sur les différents concepts proposés.

Malgré les réalisations que nous avons réussi à mener à bien, nous pensons qu'il reste encore du chemin à faire pour améliorer et explorer de nouvelles pistes :

D'abord, l'approche de propagation nécessite le développement et l'implémentation d'une application serveur/client capable de reformuler les besoins d'un objet en une requête envoyée dans le réseau.

Ensuite, l'approche du contrôle d'accès dépend d'une répartition manuelle des objets selon une analyse visuelle de leur contribution; nous prévoyons développer un outil automatique avec plus de critères capables d'attribuer une note moyenne pour chaque objet selon laquelle il sera classé et répertorié.

De plus, l'adoption ou le développement d'un système de sécurité qui prévoit les risques auxquels tout système basé sur l'Internet doit se confronter comme l'usurpation d'identité et la non-intégrité des données.

Finalement, la version actuelle du modèle suppose que les entités impliquées partagent une horloge commune. Il serait intéressant de travailler sur un cas plus général où les entités ne sont pas synchronisées.

BIBLIOGRAPHIE

- [1] U. Arijit, B. Soma, J. Joel, B. Vijayanand, and L. Sachin, "Negotiation-based privacy preservation scheme in internet of things platform," Proceedings of the First International Conference on Security of Internet of Things, Kollam, India, 2012.
- [2] B. Christophe, M. Boussard, M. Lu, A. Pastor, and V. Toubiana, "The web of things vision: Things as a service and interaction patterns," *Bell Labs Technical Journal (John Wiley & Sons, Inc.)*, vol. 16, no. 1, pp. 55-61, 2011.
- [3] B. Fabian, "Secure name services for the Internet of Things," Thèse, Humboldt-Universität zu Berlin, Wirtschaftswissenschaftliche Fakultät, 2008.
- [4] A. Nawaf, S. A. Anthony, and A. Akbar Sheikh, "Application of ZigBee and RFID Technologies in Healthcare in Conjunction with the Internet of Things," Proceedings of International Conference on Advances in Mobile Computing ; Multimedia, Vienna, Austria, 2013.
- [5] A.-L. Priscila and J.-P. Guillermo, "Collaborative Agents Framework for the Internet of Things," Workshop Proceedings of the 8th International Conference on Intelligent Environments, 2012.
- [6] D. Guinard and V. Trifa, "Towards the web of things: Web mashups for embedded devices," Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain, 2009.
- [7] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. Souza, *et al.*, "SOA-based integration of the internet of things in enterprise services," Web Services, 2009. ICWS 2009. IEEE International Conference on, 2009.
- [8] M. Han and H. Zhang, "Business intelligence architecture based on internet of things " *Journal of Theoretical & Applied Information Technology*, vol. 50, no. 1, pp. 90-95, 2013.
- [9] Cisco, "L'Internet des objets (IoT)", 16 déc. 2015;
<http://www.cisco.com/web/FR/solutions/trends/iot/overview.html>.
- [10] N. Mehdi, "Demain, l'Internet des objets", 18 déc. 2015;
<http://www.strategie.gouv.fr/publications/demain-linternet-objets>.
- [11] E. Dave, *L'Internet des objets Comment l'évolution actuelle d'Internet transforme-t-elle le monde ?*, Cisco IBSG éd., 2011, 13 p.
- [12] G. Hubert, "L'internet des objets "web"", 20 déc. 2015;
<http://blogs.lesechos.fr/internetactu-net/l-internet-des-objets-web-a13507.html>.
- [13] G. Babin and M. Leblanc, "Les Web Services et leur impact sur le commerce B2B," CIRANO Centre interuniversitaire de recherche en analyse des organisations 2003.

- [14] F. Abouzaid, "Analyse formelle d'orchestrations de services Web," Doctora Génie informatique et génie logiciel, École Polytechnique de Montréal, 2010.
- [15] L. Haiyan, L. Yanyan, X. Zenggang, Y. Zhen, and T. Kun, "Study and Application of Urban Flood Risk Map Information Management System Based on SOA," *Journal of Software (1796217X)*, vol. 10, no. 2, pp. 180-189, 2015.
- [16] B. Soukkarieh, "Technique de l'internet et ses langages: vers un système d'information web restituant des services web sensibles au contexte," Doctora, Informatique Université de Toulouse, Université Toulouse III-Paul Sabatier, 2010.
- [17] D. Zeng, S. Guo, and Z. Cheng, "The web of things: A survey," *Journal of Communications*, vol. 6, no. 6, pp. 424-438, 2011.
- [18] J. Alliot, "Qu'est-ce que le "Middleware", " Note 2003.
- [19] J. Lisa, "Oracle® Fusion Middleware Concepts Guide ", 20 déc.2015; http://docs.oracle.com/cd/E28280_01/core.1111/e10103/title.htm.
- [20] The IT Crew, "Conseil en Systèmes d'Information, Urbanisation, Architectures et Expertise JEE The Best Way to Predict The Future Is To Invent IT™", 20 déc. 2015; <http://it.fr/index.php/tutoriels/java-ee/128-qu-est-ce-qu-un-intergiciel-middleware.html>
- [21] P.-J. Benghozi, S. Bureau, F. Massit-Folléa, C. Waroquiers, and S. Davidson, *L'internet des objets: quels enjeux pour l'Europe*, Éd. de la Maison des sciences de l'homme éd., 2009, 66 p.
- [22] M. Weill and M. Souissi, "L'Internet des objets: concept ou réalité?," dans *Réalités industrielles*, ESKA, Éd., Les Annales des Mines. ESKA, 2010, pp. 90-96.
- [23] les experts Ooreka, "Système RFID : définition et fonctionnement d'un système RFID ", 22 déc. 2015; <http://rfid.comprendrechoisir.com/comprendre/systeme-rfid>.
- [24] J. A. Stankovic, "Wireless sensor networks," *IEEE Computer Society*, vol. 41, no. 10, pp. 92-95, 2008.
- [25] N. Daniel, R. Marcel, and K. Daniel, *Livre blanc Machine To Machine enjeux et perspectives*: Orange Business Services, Syntec informatique, Fing, 2006, 40 p.
- [26] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645-1660, 2013.
- [27] J. D. Pessemier, "Une réflexion sur « L'internet des Objets » (IdO) ou « Internet of Things » (IoT)," 2015.

- [28] D. Guinard, V. Trifa, F. Mattern, and E. Wilde, "From the internet of things to the web of things: Resource-oriented architecture and best practices," dans *Architecting the Internet of Things*, Springer, 2011, pp. 97-129.
- [29] d. A. J. Nailton Vieira, d. A. Jeferson Lima, C. Ramon Dias, J. Leandro, A. Silva, P. Geroge Pacheco, *et al.*, "Web of Things Gateway: a Performance Evaluation," Proceedings of the 21st Brazilian Symposium on Multimedia and the Web, Manaus, Brazil, 2015.
- [30] Techopedia, "What is the Web of Things (WoT)? - Definition from Techopedia", 25 janv. 2015; <https://www.techopedia.com/definition/26834/web-of-things-wot>.
- [31] C. Sébastien, "Tutorial HTTP " Laboratoire de transmission de données / EIG, Rapport 21 2004.
- [32] B. Oscar and I.-N. Service, "20% des entreprises investissent dans l'Internet des objets", 10 janv.2016; <http://www.lemondeinformatique.fr/actualites/lire-20-des-entreprises-investissent-dans-l-internet-des-objets-58263.html>.
- [33] D. Robert, "Commencer petit pour atteindre des gros succès avec l'Internet des objets", 10 janv.2016; <http://www.directioninformatique.com/commandite/commencer-petit-pour-atteindre-des-gros-succes-avec-linternet-des-objets>.
- [34] K. Mohebbi, S. Ibrahim, M. Khezrian, K. Munusamy, and S. G. H. Tabatabaei, "A comparative evaluation of semantic web service discovery approaches," Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services, 2010.
- [35] F. Hadjila, A. Chikh, and A. Belabed, "Automated discovery of web services: an interface matching approach based on similarity measure," Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications, Amman, Jordan, 2010.
- [36] M. Varguez-Moo, F. Moo-Mena, and V. Uc-Cetina, "Use of classification algorithms for semantic web services discovery," *Journal of Computers*, vol. 8, no. 7, pp. 1810-1814, 2013.
- [37] F. Paganelli and D. Parlanti, "A DHT-Based Discovery Service for the Internet of Things," *Journal of Computer Networks & Communications*, pp. 1-11, 2012.
- [38] V. Altmann, J. Skodzik, P. Danielis, J. Mueller, F. Golasowski, and D. Timmermann, "A DHT-Based Scalable Approach for Device and Service Discovery," Embedded and Ubiquitous Computing (EUC), 2014 12th IEEE International Conference on, 2014.
- [39] M. Makhluhian, S. M. Hashemi, Y. Rastegari, and E. Pejman, "Web service selection based on ranking of qos using associative classification," *International Journal on Web Service Computing (IJWSC)*, vol. 2, pp. 1-14, 2012.

- [40] S. Wang, Z. Zheng, Q. Sun, H. Zou, and F. Yang, "Cloud model for service selection," Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on, 2011.
- [41] R. Mohana and D. Dahiya, "Approach and impact of a protocol for selection of service in web service platform," *SIGSOFT Softw. Eng. Notes*, vol. 37, no. 1, pp. 1-6, 2012.
- [42] M. Khezrian, A. Jahan, W. M. N. Wan Kadir, S. Ibrahim, and M. K. Khan, "An Approach for Web Service Selection Based on Confidence Level of Decision Maker," *PLoS ONE*, vol. 9, no. 6, pp. 1-14, 2014.
- [43] S. Wang, Q. Sun, H. Zou, and F. Yang, "Reputation measure approach of web service for service selection," *IET Software*, vol. 5, no. 5, p. 466, 2011.
- [44] B. Jeong, D. Lee, J. Lee, and H. Cho, "Support for seamless data exchanges between web services through information mapping analysis using kernel methods," *Expert Systems with Applications*, vol. 36, no. 1, pp. 358-365, 2009.
- [45] J. Xu, G. Xiao, J. Lu, Q. Liang, and J. Shen, "Customizable data exchange based on Web service," e-Business Engineering, 2009. ICEBE'09. IEEE International Conference on, 2009.
- [46] S. Kubler, I. D. Nargund, K. Fr, #228, mling, and W. Derigent, "Peer-to-Peer Data Synchronization Agents," Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 03, 2014.
- [47] S. Hachem, T. Teixeira, and V. Issarny, "Ontologies for the Internet of Things," Proceedings of the 8th Middleware Doctoral Symposium, 2011.
- [48] A. Kumar, K. Gopal, and A. Aggarwal, "Cost and Lightweight Modeling Analysis of RFID Authentication Protocols in Resource Constraint Internet of Things," *Journal of Communications Software & Systems*, vol. 10, no. 3, pp. 179-187, 2014.
- [49] Z. Liang and C. Han-Chieh, "Multimedia traffic security architecture for the internet of things," *IEEE Network*, vol. 25, no. 3, pp. 35-40, 2011.
- [50] T. S. Dillon, A. Talevski, V. Potdar, and E. Chang, "Web of things as a framework for ubiquitous intelligence and computing," dans *Ubiquitous Intelligence and Computing*, Springer, 2009, pp. 2-13.
- [51] C. Cheng, Z. Chunhong, Q. Xiaofeng, and J. Yang, "The Social Web of Things (SWoT)- Structuring an Integrated Social Network for Human, Things and Services," *Journal of Computers*, vol. 9, no. 2, pp. 345-352, 2014.
- [52] S. Mayer, D. Guinard, and E. Wilde, "Third International Workshop on the Web of Things (WoT 2012)," Proceedings of the Third International Workshop on the Web of Things, Newcastle, United Kingdom, 2012.

- [53] W.-C. Shin, "Design and Embodiment of Web of Things Platform," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 2, pp. 309-318, 2014.
- [54] L. Mainetti, V. Mighali, L. Patrono, and P. Rametta, "Discovery and Mash-up of Physical Resources through a Web of Things Architecture," *Journal of Communications Software & Systems*, vol. 10, no. 2, pp. 124-134, 2014.
- [55] M. David, "Lexique Comprendre les mots d'internet et son e-langage", 09 déc. 2015; <http://www.eclairer.net/lexique/#m>.
- [56] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, vol. 1, p. 14, 2011.
- [57] S. De, P. Barnaghi, M. Bauer, and S. Meissner, "Service modelling for the Internet of Things," Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on, 2011.
- [58] D. Guinard, V. Trifa, and E. Wilde, "A resource oriented architecture for the web of things," *Internet of Things (IOT) 2010*, 2010.
- [59] I. Ishaq, J. Hoebeke, J. Rossey, E. De Poorter, I. Moerman, and P. Demeester, "Enabling the web of things: facilitating deployment, discovery and resource access to IoT objects using embedded web services," *International Journal of Web and Grid Services* 6, vol. 10, no. 2-3, pp. 218-243, 2014.
- [60] M. Fowler, *Domain-specific languages*, Copyrighted Material éd.: Pearson Education, 2010, 581 p.
- [61] M. Eysholdt and H. Behrens, "Xtext: implement your language faster than the quick and dirty way," *Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion*, 2010.
- [62] C. G. Richard, *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*, Copyrighted Material éd.: Addison-Wesley Professional, 2009, 706 p.
- [63] Xtend, "Java 10, today!", 27 mai. 2014; <http://www.eclipse.org/xtend/>.
- [64] M. Adda and R. Saad, "A data sharing strategy and a DSL for service discovery, selection and consumption for the IoT," *Procedia Computer Science*, vol. 37, pp. 92-100, 2014.
- [65] B. Arnaud, "Présentation du Spark Core", 10 mar.2015; <https://www.ekito.fr/people/presentation-du-spark-core/>.
- [66] T0B1NUX, "Informatique ou électronique, telle est la question", 30 mar. 2015; <https://infoelec.wordpress.com/2013/03/24/arduino-quest-ce-que-cest/>.

- [67] Spark IO, "Spark Core: Wi-Fi for Everything (Arduino Compatible)", 12 déc. 2015; <https://www.kickstarter.com/projects/sparkdevices/spark-core-wi-fi-for-everything-arduino-compatible/description>.