

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université de Blida 1**  
**Faculté des Sciences**  
**Département de l'informatique**



**Mémoire de fin d'études en vue de l'obtention du diplôme de**  
**MASTER EN INFORMATIQUE**  
**Option : Systèmes informatiques et Réseaux**

**Thème**

Solution pour le problème de portabilité dans les environnements  
de Cloud Computing de type Software as a Service

**Réalisé par :**

-BADAQUI Lamia.  
-BIRI Camelia.

**Devant le jury :**

-Mme CHIKHI Imane (Présidente).  
-Mme GHEBGHOUB Yasmina (Examinatrice).  
-Mme MANCER Yasmine (Promotrice).

Promotion :2020/2021

## **Remerciement**

Nous voulons d'abord remercier Allah, qui nous a donné la force  
et le courage pour réaliser ce travail.

Nous remercions également notre promotrice, Mme Mancer, qui  
nous a guidée et aidée tout au long de notre travail.

Nous remercions sincèrement nos chers parents, nos familles et  
tous nos amis.

Enfin, nous tenons à remercier les membres du jury qui ont  
Accepté d'évaluer notre travail.

## **Dédicace**

Je dédie ce modeste travail :

A mes parents, la prunelle de mes yeux et mon pilier qui m'ont soutenu et encouragé tout le long de mes études, aucun hommage ne pourrait être à la hauteur de l'amour dont ils ne cessent de me combler. Que dieu leur procure bonne santé et longue vie.

A ma chère sœur jumelle Lynda.

A mes chère frère Zineddine et Mebarek.

A mon binôme : Badaoui Lamia ainsi qu'à toute sa famille.

A mes amis, en souvenir de notre sincère et profonde amitié et des moments agréables que nous avons passés ensemble.

A tous ceux qui me sens chers et que j'ai omis de citer.

Je ne manquerai pas de remercier tous les professeurs qui m'ont suivi pendant le cursus universitaire.

**Camelia**

## Dédicace

Je dédie ce projet :

A ma chère mère,

A mon cher père,

Qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

A mes chères sœurs HANAN, ZOU et YOUTA,

A ma chère sœur HABIBA et son mari HAMZA et ma nièce RAHMA,

Pour leurs soutiens moraux et leurs conseils précieux tout au long de mes études.

A ma chère binôme CAMELIA BIRI,

Pour son entente et sa sympathie.

A mes chères amies RANIA et RIMA,

Pour leurs aides et supports dans les moments difficiles.

**Lamia**

## Résumé

Malgré son apparition récente, le Cloud Computing a connu une montée rapide en popularité. Le rapport qualité/prix des services de Cloud Computing a séduit beaucoup de consommateurs qui désirent avoir les performances de certaines ressources sans devoir en procurer de manière définitive. Cependant, les défis auxquels le Cloud Computing doit faire face sont également en constante augmentation et la portabilité en est un d'eux.

La portabilité est un problème épineux qui empêche une adoption plus large des services de Cloud Computing à cause du verrouillage des fournisseurs (lock-in) d'un côté et du manque de standardisation de l'autre, empêchant ainsi non seulement la migration vers les solutions de Cloud Computing mais également la collaboration entre les différents services.

Notre but est de proposer une solution à ce défi pour les services de Cloud Computing public de type Software as a Service(SaaS). Cette solution se divise en deux étapes, une étape pour le mapping des services Cloud en utilisant les ontologies et les mesures de similarité et une autre étape pour la portabilité entre les services compatibles en utilisant les services Web. Notre solution est destinée aux fournisseurs qui souhaitent mettre à disposition des services où la portabilité est souhaitée.

Dans notre solution, les ontologies servent à décrire les inputs et outputs des services Cloud. Après la création des ontologies, nous allons calculer les correspondances entre les services Cloud en utilisant les mesures de similarité syntaxique et sémantique pour définir les compatibilités. Après avoir défini les correspondances entre les services, la portabilité serait réalisée en utilisant les services Web.

**Mots Clés:** Cloud Computing, portabilité, Software as a Service(SaaS), lock-in, mapping, ontologies, mesure de similarité, services Web.

## Abstract

Despite its recent emergence, Cloud Computing has experienced a rapid rise in popularity. The quality / price ratio of Cloud Computing services has attracted many consumers who want the performance of certain resources without having to provide them permanently. However, the challenges facing Cloud computing are also constantly increasing and portability is one of them.

Portability is a thorny issue preventing wider adoption of Cloud computing services due to vendor lock-ins on one side and lack of standardization on the other, thus preventing not only migration to vendors. Cloud Computing solutions but also collaboration between different services.

Our goal is to provide a solution to this challenge for Software as a Service (SaaS) Cloud Computing services. This solution is divided into two stages, a stage for the mapping of Cloud services using ontologies and similarity measures and another stage for portability between compatible services using web services. Our solution is intended for suppliers who wish to provide services where portability is desired.

In our solution, ontologies are used to describe the inputs and outputs of Cloud services. After creating the ontologies, we will calculate the correspondences between the Cloud services using the measures of syntactic and semantic similarity to define the compatibilities. After defining the mappings between the services, the portability would be achieved by using the web services.

**Keywords:** Cloud Computing, portability, Software as a Service (SaaS), Lock-in, mapping, ontologies, similarity measurement, web services.

## ملخص

على الرغم من ظهورها مؤخرًا، فقد شهدت الحوسبة السحابية ارتفاعًا سريعًا في شعبيتها. اجذب نسبة الجودة / السعر لخدمات الحوسبة السحابية العديد من المستهلكين الذين يرغبون في أداء موارد معينة دون الحاجة إلى توفيرها بشكل دائم. ومع ذلك، فإن التحديات التي تواجه الحوسبة السحابية تتزايد باستمرار وإمكانية النقل واحدة منها.

تعد قابلية النقل مشكلة شائعة تمنع التبني الأوسع لخدمات الحوسبة السحابية بسبب قيود البائعين من ناحية ونقص التوحيد القياسي من ناحية أخرى، وبالتالي منع الانتقال إلى بائعين حلول الحوسبة السحابية ولكن أيضًا بين الخدمات المختلفة.

ينقسم هذا الحل إلى مرحلتين، مرحلة لرسم خرائط الخدمات السحابية باستخدام مقاييس الأنطولوجيا والتشابه ومرحلة أخرى لقابلية النقل بين الخدمات المتوافقة باستخدام خدمات الويب. حلنا مخصص للموردين الذين يرغبون في تقديم خدمات حيثما تكون قابلية النقل مطلوبة.

في حلنا، تُستخدم الأنطولوجيات لوصف مدخلات ومخرجات الخدمات السحابية ثم حساب مراسلاتهم باستخدام مقاييس التشابه النحوية والدلالية. بعد تحديد المراسلات بين الخدمات، سيتم تحقيق قابلية النقل باستخدام خدمات الويب.

**الكلمات المفتاحية:** الحوسبة السحابية، قابلية النقل، البرمجيات SaaS، القفل، رسم الخرائط، الأنطولوجي، قياس التشابه، خدمات الويب.

## Sommaire :

Introduction générale

I : Chapitre 1 : Introduction au Cloud Computing	4
<b>I.1 Introduction</b>	<b>5</b>
<b>I.2 Historique</b>	<b>5</b>
<b>I.3 Définitions du Cloud Computing</b>	<b>8</b>
<b>I.4 Les acteurs du Cloud Computing</b>	<b>11</b>
<b>I.5 Éléments technologique du Cloud Computing</b>	<b>12</b>
<b>I.6 Accord de niveau de service (SLA)</b>	<b>14</b>
<b>I.7 Caractéristique du Cloud Computing</b>	<b>14</b>
<b>I.8 Modèles de service du Cloud Computing</b>	<b>16</b>
I.8.1 Infrastructure en tant qu'un Service (Infrastructure as a Service (IaaS))	17
I.8.2 Plateforme en tant qu'un Service (PaaS pour Platform as a Service)	18
I.8.3 Les logiciels en tant que services (SaaS Software as a Service)	19
I.8.4 XaaS	20
<b>I.9 Modèles de déploiement du Cloud Computing</b>	<b>21</b>
I.9.1 Cloud public « Public Cloud »	21
I.9.2 Cloud privé « Private Cloud »	22
I.9.2.1 Les Clouds privés internes	22
I.9.2.2 Les Clouds privés externes	23
I.9.3 Cloud Communautaire « Community Cloud »	24
I.9.4 Cloud hybride « Hybrid Cloud »	25
<b>I.10 Secrets du succès du Cloud</b>	<b>26</b>
<b>I.11 Avantages et limites du Cloud Computing</b>	<b>27</b>
I.11.1 Avantages du Cloud Computing	27
I.11.2 Limites du Cloud Computing	28
<b>I.12 Conclusion</b>	<b>29</b>
II. Chapitre II : La portabilité dans le Cloud Computing	30
<b>II.1 Introduction</b>	<b>31</b>
<b>II.2 Définition de la portabilité</b>	<b>31</b>
<b>II.3 Types de portabilité</b>	<b>32</b>
<b>II.4 Le verrouillage des fournisseurs (lock-in)</b>	<b>32</b>
<b>II.5 Raisons de la portabilité</b>	<b>33</b>
<b>II.6 Notions importante pour la portabilité des données</b>	<b>33</b>
II.6.1 Standardisation	33
II.6.2 Migration	34
II.6.3 Mapping	34



<b>II.7</b>	<b>Portabilité et interopérabilité :</b>	<b>35</b>
<b>II.8</b>	<b>Solutions existantes pour la portabilité dans le Cloud Computing :</b>	<b>35</b>
II.8.1	Solution de Thabet [43] :	35
II.8.2	Solution de Lupşe et al [71]:	37
II.8.3	Solution Benhsayen and Ettalbi [72] :	38
II.8.4	Solution Kwon et al [73] :	40
II.8.5	Solution Bouzerzour et al [74] :	40
<b>II.9</b>	<b>Discussion :</b>	<b>42</b>
<b>II.10</b>	<b>Conclusion :</b>	<b>43</b>
<b>III.</b>	<b>Chapitre III : Conception de notre solution</b>	<b>44</b>
<b>III.1</b>	<b>Introduction :</b>	<b>45</b>
<b>III.2</b>	<b>Analyse des besoins :</b>	<b>45</b>
<b>III.3</b>	<b>Schéma global de la solution proposée :</b>	<b>47</b>
<b>III.4</b>	<b>Description générale de notre approche</b>	<b>49</b>
<b>III.5</b>	<b>Description détaillée de l'approche proposée</b>	<b>49</b>
III.5.1	Phase de mapping :	50
III.5.1.1	Etape 1 : Représentation de chaque service Cloud en une ontologie	50
III.5.1.2	Etape 2 : Extraction des inputs et outputs des services Cloud.	50
III.5.1.3	Etape 3 : Calcul de similarité entre les services Cloud	51
III.5.1.3.1	Mesure de similarité syntaxique :	52
III.5.1.3.1.1	L'indice de Dice :	52
III.5.1.3.1.2	L'indice de Jaccard :	53
III.5.1.3.2	Mesure de similarité sémantique :	54
III.5.1.3.2.1	Mesures basées sur les chemins :	54
III.5.1.3.2.1.1	Wu et Palmer	55
III.5.1.3.2.1.2	Path :	55
III.5.1.3.2.2	Mesures basées sur les nœuds :	56
III.5.1.3.2.2.1	Mesure de Lin	57
III.5.1.4	Etape 4 : Agrégation des résultats	57
III.5.2	Phase de portabilité des données :	60
<b>III.6</b>	<b>Conclusion :</b>	<b>60</b>
<b>IV.</b>	<b>Chapitre IV : Implémentation de la solution</b>	<b>61</b>
<b>IV.1</b>	<b>Introduction</b>	<b>62</b>
<b>IV.2</b>	<b>Ressources matérielles :</b>	<b>62</b>
<b>IV.3</b>	<b>Outils et environnement de développement</b>	<b>62</b>
<b>IV.4</b>	<b>Présentation de notre système :</b>	<b>68</b>
IV.4.1	Phase 1 : phase de mapping :	68
IV.4.2	Phase 2 : Portabilité des données entre les services Cloud.	77
<b>IV.5</b>	<b>Evaluation de la solution par rapport à l'état de l'art :</b>	<b>80</b>
<b>IV.6</b>	<b>Conclusion :</b>	<b>81</b>

**Conclusion générale et perspectives.**

**Bibliographie.**

**Annexe A : Notions sur les ontologies.**

**Annexe B : Service Web.**

## Liste des figures :

Figure I. 1 : Evolution et générations de l'informatique[5].	7
Figure I. 2:Caractéristique du Cloud Computing.	16
Figure I. 3:Les couches du Cloud Computing [32].	17
Figure I. 4:Modèle de déploiement d'un Cloud public[42].	22
Figure I. 5:Modèle de déploiement d'un Cloud privé [42].	24
Figure I. 6:Modèle de déploiement d'un Cloud communautaire [42].	25
Figure I. 7:Modèle de déploiement d'un Cloud hybride [42].	26
Figure II. 1: Approche proposer basé sur les Système Multi-Agent (SMA)[49].	36
Figure II. 2: Architecture d'interopérabilité sémantique des ressources IaaS [82].	39
Figure III. 1: Diagramme de cas d'utilisation.	46
Figure III. 2:Schéma global de notre solution.	48
Figure IV. 1:Exemple d'ontologie du service Cloud laboratoire.	69
Figure IV. 2: Interface principale de notre application.	70
Figure IV. 3:Interface login de l'administrateur.	70
Figure IV. 4:Interface après le login de l'administrateur.	71
Figure IV. 5:Interface pour ajouter une ontologie.	72
Figure IV. 6:Valeurs de similarité des différentes mesures pour le domaine Business.	73
Figure IV. 7:Valeurs de similarité des différentes mesures pour le domaine médicale.	73
Figure IV. 8:Valeur de similarité final pour le domaine business.	74
Figure IV. 9: Valeur de similarité final pour le domaine médicale.	74
Figure IV. 10: Services compatible avec leur mesure pour le domaine business.	75
Figure IV. 11: Services compatible avec leur mesure pour le domaine médicale.	75
Figure IV. 12: Interface login des fournisseurs de services Cloud.	76
Figure IV. 13:Interface d'un fournisseur de services après identification.	76
Figure IV. 14: Service compatible avec le service laboratoire.	76
Figure IV. 15:WSDL du service Web de l'application laboratoire.	78
Figure IV. 16: Interface du service doctor pour la récupération du résultat de test.	79
Figure IV. 17:Interface correspondante si le test est négatif.	79
Figure IV. 18: Interface correspondante si le test est positif.	80
Figure 1:Les couches de l'OWL.	101
Figure 2:Message SOAP [128].	107
Figure 3:Structure d'une description WSDL 1.1 [127].	109
Figure 4:Architecture de base des services Web [127].	112

**Liste des Tableaux :**

<i>Tableau II. 1: Tableau comparatif</i> .....	42
<i>Tableau III. 1: Les acteurs et leur rôle</i> .....	45
<i>Tableau III. 2: Description textuelle des cas d'utilisation</i> .....	47

**Liste des Algorithmes :**

*Algorithme III. 1: Algorithme chargement des ontologies et d'extraction des inputs et des outputs .....51*  
*Algorithme III. 2: Algorithme calcul de similarité et compatibilité entre les ontologies.....59*

## **Liste des acronymes :**

**ASP** : Application Service Provider.

**API**: Application Programming Interface.

**AWS**: Amazon Web service.

**CRM**: Customer Relationship Management.

**CSA**: Cloud Security Alliance.

**CSP**: Cloud Service Provider.

**DBaaS**: Database as a Service.

**EC2**: Elastic Compute Cloud.

**ESB**: Enterprise Service Bus.

**GCSD**: Generic Cloud Service Description.

**HTTP**: Hypertext Transfer Protocol.

**IaaS**: Infrastructure as a Service.

**IBM**: International Business Machines.

**ID**: Identifiant.

**IT**: Information Technologie.

**Java EE** : Java, Enterprise Edition.

**Java SE** : Java Standard Edition.

**JAWS**: Java API for Wordnet Searching.

**JCP**: Java Community Process.

**JDBC**: Java Database Connectivity.

**JDK** : Java Development Kit.

**JRE** : Java Runtime Environment.

**JSP**: Java Server Pages.

**MVC**: Model-View-Controller.

**NoSQL:** Not Only SQL.

**NIST:** National Institute of Standards and Technology.

**OCCI:** Open Cloud Computing Interface.

**OWL:** Ontology Web Language.

**OWL DL:** OWL Description Logics.

**OWL-S:** Ontology Web Language for Service.

**PaaS:** Platform as a Service.

**RDF:** Resource Description Framework.

**RDFS:** RDF Schéma.

**SaaS:** Software as a Service.

**SLA:** Service-Level Agreements.

**SMA:** Système Multi Agent.

**SOA:** Service Oriented Architecture.

**SOAP:** Simple Object Access Protocol.

**SQL:** Structured Query Language.

**S3:** Simple Storage Service.

**UDDI:** Universal Description Discovery and Integration.

**URI:** Uniform Ressource Identifier.

**VPN:** Virtual Private Network.

**WSDL:** Web Services Description langage

**WS-I:** Web Service Interoperability.

**W3C:** World Wide Web Consortium.

**WS4J:** Wordnet Similarity for Java.

**XML:** Extensible Markup Language.

## **Introduction générale**

Face à l'augmentation continue des coûts de mise en place et de maintenance des systèmes informatiques, les entreprises externalisent de plus en plus leurs services informatiques en les confiant à des entreprises spécialisées comme les fournisseurs de Cloud Computing. L'intérêt principal de cette stratégie pour les entreprises réside dans le fait qu'elles ne paient que pour les services effectivement consommés.

Avec la rapidité de l'utilisation des services Cloud par les entreprises, en nombre comme en fonctionnalités, le besoin des entreprises d'utiliser plusieurs Cloud à la fois ou de migrer carrément vers un autre fournisseur en cas de besoin pour une meilleure qualité de services, devient une nécessité incontournable. À cet effet, les fournisseurs de services Cloud devraient collaborer, d'une part pour mieux satisfaire les besoins des clients, d'autre part, pour éviter le problème du lock-in, qui met en question l'adoption du Cloud par les entreprises utilisatrices. Pour cela, il faut assurer la portabilité entre les services Cloud pour améliorer la productivité des entreprises et de contribuer au développement des Cloud.

### **Problématique :**

Malgré le fort engouement que suscite le Cloud Computing, son adoption et sa vulgarisation se heurtent à plusieurs obstacles. Le manque de standardisation a fait émerger beaucoup de problèmes relatifs au nombre grandissant de fournisseurs de services Cloud d'un côté, et d'un autre côté, à l'hétérogénéité des technologies, plateformes et modèle de données utilisées.

Delà, une solution pour le problème de portabilité dans des environnements de Cloud Computing est nécessaire.



## **Objectifs du travail**

L'objectif général de notre travail consiste à proposer une approche pour le problème de portabilité dans des environnements de Cloud Computing de type Software as a Service. Cette approche se divise en deux étapes ; premièrement le mapping entre les services Cloud en utilisant les ontologies et les mesures de similarité ensuite assurer la portabilité pour les services Cloud compatibles en utilisant les services Web.

## **Organisation du mémoire**

Pour mener à bien notre mémoire, nous avons organisé notre travail en quatre chapitres :

### **Chapitre 1 : Introduction au Cloud Computing**

Le premier chapitre présente les notions de base du Cloud Computing telles que l'historique, la définition du Cloud Computing, les acteurs, les modèles de livraison et de déploiement des services de Cloud Computing, les avantages, les inconvénients, les caractéristiques du Cloud mais aussi les concepts liés au Cloud Computing tels que les éléments technologiques, les accords de niveau de service et les secrets de succès du Cloud Computing.

### **Chapitre 2 : La portabilité dans le Cloud Computing**

Le deuxième chapitre présente la portabilité dans le Cloud Computing et les concepts qui y sont liés. Mais aussi une étude comparative des solutions existantes pour la portabilité sera faite.

### **Chapitre 3 : Conception de notre solution**

Ce chapitre, consiste à concevoir une solution pour la portabilité entre les différents services Cloud de type Software as a Service. Nous présentons dans ce chapitre les deux étapes essentielles qui sont le mapping en utilisant les ontologies et les mesures de similarité et la portabilité des données en utilisant les services Web.

### **Chapitre 4 : Implémentation et test du système**

Dans ce dernier chapitre, nous avons présenté l'environnement de développement de notre solution, les outils utilisés, et une présentation visuelle de l'implémentation de notre application Java EE.

# **Chapitre 1 : Introduction au Cloud Computing**

## **I.1 Introduction :**

Dans le présent chapitre on va présenter un état de l'art sur le Cloud Computing pour avoir une vue générale en présentant tout ce qui est considéré important dans ce domaine.

Nous allons présenter ce qu'est le « Cloud Computing » dès sa genèse, son histoire, sa définition, les acteurs, les modèles de livraison et de déploiement des services de Cloud Computing, les avantages, les inconvénients, les caractéristiques du Cloud mais aussi les concepts liés au Cloud Computing tels que les éléments technologiques, les accords de niveau de service et les secrets de succès du Cloud Computing.

## **I.2 Historique :**

Techniquement, le concept de Cloud Computing est loin d'être nouveau, il est même présent depuis des décennies. On en trouve les premières traces dans les années 1960, quand John McCarthy<sup>1</sup> affirmait que cette puissance de traitement informatique serait accessible au public dans le futur [1]. Mais la première apparition de la notion de Cloud, au sens du Cloud Computing, date d'une conférence Microsoft à propos du lancement de sa nouvelle plateforme « .Net », en 2001 où Microsoft présente le Cloud comme l'endroit où seront hébergés les « Web services »[1] .

Dans les années 1990, un autre concept avait déjà préparé le terrain au Cloud Computing. Il s'agit de l'ASP (Application Service Provider) qui permettait au client de louer l'accès à un logiciel installé sur les serveurs distants d'un prestataire, sans installer le logiciel sur ses propres machines. Le Cloud Computing ajoute à cette offre la notion d'élasticité avec la possibilité d'ajouter de nouveaux utilisateurs et de nouveaux services d'un simple clic de souris[2].

---

<sup>1</sup> *John McCarthy* (né le 4 septembre 1927, à *Boston, Massachusetts*) informaticien et mathématicien américain est le principal pionnier de l'intelligence artificielle. A la fin des années 1950, il a créé avec *Fernando Cobarto* la technique du temps partagé, qui permet à plusieurs utilisateurs d'employer simultanément un même ordinateur.

Il est communément admis que le concept de Cloud Computing a été initié par le géant Amazon en 2002. Le cybermarchand avait alors investi dans un parc informatique afin de pallier les surcharges des serveurs dédiés au commerce en ligne constatées durant les fêtes de fin d'année. A ce moment-là, Internet comptait moins de 600 millions d'utilisateurs [2] mais la fréquentation de la toile et les achats en ligne étaient en pleine augmentation. En dépit de cette augmentation, les ressources informatiques d'Amazon restaient peu utilisées une fois que les fêtes de fin d'année étaient passées. Ce dernier a alors eu l'idée de louer ses capacités informatiques le reste de l'année à des clients pour qu'ils stockent les données et qu'ils utilisent les serveurs [2][3].

Amazon propose l'accès à des ressources informatiques comme un service avec un paiement à la consommation (Utility Computing) en utilisant plusieurs dizaines de milliers de serveurs (Fermes de Serveurs) répartis sur plusieurs sites dans le monde (Grid Computing) et utilise la technologie de virtualisation permettant l'élasticité [2][4].

Auparavant, seuls les superordinateurs permettaient de fournir cette puissance et étaient principalement utilisés par des gouvernements, des militaires, des laboratoires et des universités.

Désormais, des entreprises comme Google fournissent des applications qui exploitent le même type de puissance et sont accessibles à tout moment, de n'importe où et par tout à chacun via Internet.

Le Cloud Computing est la cinquième génération de technologies de calcul (Figure I. 1) après le mainframe, le PC, le client serveur et les services Web [5] :

— **1re génération (MainFrame)** : Un ordinateur central ou un macroordinateur (mainframe computer) est un ordinateur de grande puissance de traitement et qui sert d'unité centrale à un réseau de terminaux.

- **2me génération (PC)** : Apparu en 1970s, l'utilisateur installe/accède à ses applications directement sur son PC sans avoir besoin d'une tierce partie (réseau ou serveur).
- **3me génération (Client-Serveur)** : Dans ce type d'architecture, l'application est divisée en 2 entités : le fournisseur de service appelé Serveur et le consommateur de service appelé Client. Les deux entités doivent être connectées par un réseau.
- **4me génération (www)** : Le Web permet d'héberger des applications (dites applications Web) sur des serveurs distants. On peut accéder à ces applications via le réseau internet en utilisant un navigateur.
- **5me génération (Cloud Computing)** : Apparu au début du 21me siècle, il consiste à utiliser des serveurs informatiques distants par l'intermédiaire d'un réseau, généralement Internet, pour héberger/fournir des applications, ces serveurs sont loués à la demande.

Figure I. 2 illustre l'évolution et les générations de l'informatique.

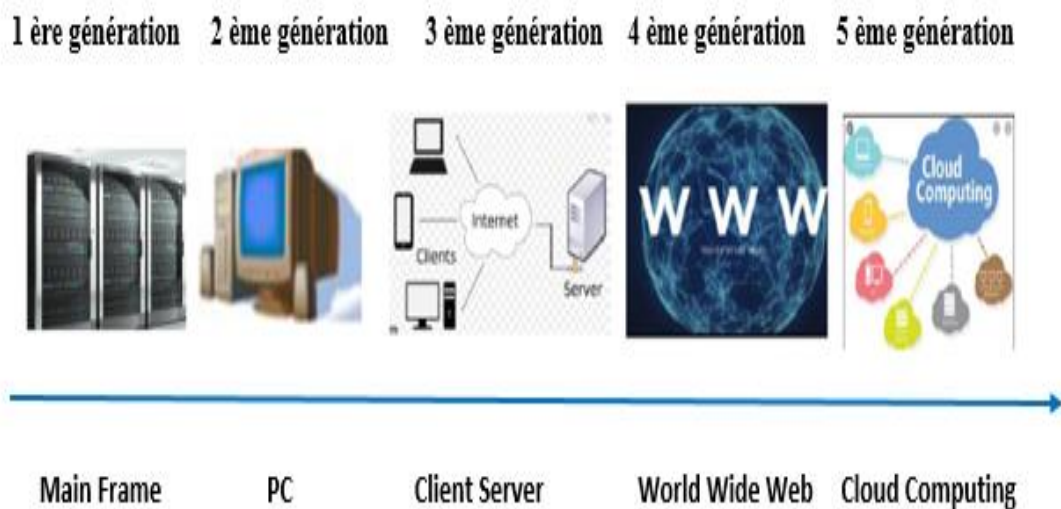


Figure I. 1 : Evolution et générations de l' informatique[5].

### I.3 Définitions du Cloud Computing :

L'informatique dans le nuage est plus connu sous sa forme anglo-saxonne : « Cloud Computing », mais il existe de nombreux synonymes francophones tels que : « infonuagique » ou encore « informatique dématérialisée ».

Le terme Cloud Computing a été inventé par des acteurs du monde de l'informatique (Google, Amazon et Microsoft) pour désigner un nouveau modèle de consommation de services informatiques. Du point de vue technique, ce nouveau modèle n'est pas une nouvelle technologie révolutionnaire en soi. Il s'agit plutôt de l'association et de l'intégration de plusieurs technologies, dont il a hérité des caractéristiques et des fonctionnalités telles que la virtualisation, les systèmes distribués, la grille informatique et le Computing autonome. Le Cloud Computing est en pleine évolution et ses définitions, caractéristiques, cas d'usages et technologies seront progressivement affinés [6][7].

Aujourd'hui, il existe de nombreuses tentatives de définition de ce modèle, mais la définition du NIST<sup>2</sup> (National Institute of Standards and Technology) est la plus largement acceptée .

- **Selon le NIST [8]**, le Cloud Computing est défini comme suit: *«Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable Computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction ».*

Autrement dit, Le Cloud Computing est un modèle permettant d'accéder à travers le réseau, à la demande, à un ensemble de ressources informatiques partagées et configurables (les réseaux, les serveurs, des capacités de stockage, des

---

<sup>2</sup> NIST: National Institute of Standards and Technology

applications, et les services), qui sont rapidement mobilisables et libérables avec le moindre effort d'administration ou d'intervention du fournisseur.

Cette définition, corroborée par la plupart des grands acteurs de l'informatique dématérialisée [9], affirme que l'informatique dans les nuages est un paradigme en pleine évolution à l'heure d'aujourd'hui.

- **Selon Gartner** [10], le Cloud est défini comme suit: « *Cloud is a style of Computing where scalable and elastic IT-related capabilities are provided as a service to external customers using Internet technologies*».

Autrement dit, le Cloud Computing est un type d'informatique dans lequel des capacités très évolutives et élastiques sont fournies sous forme de service à plusieurs clients via les technologies Internet.

- **Selon I. Foster** [11] : Le Cloud Computing est un paradigme informatique distribuée qui est conduit par des échelles économiques, dans lequel un pool de ressources virtualisées, une évolutive dynamique, une puissance de calcul, un moyen de stockage, des plates-formes et des services sont fournis à la demande, à des clients externes sur Internet.

- **Selon Cisco** [12], le leader mondial des technologies réseaux, le Cloud Computing est une plateforme de mutualisation informatique fournissant aux entreprises des services à la demande avec l'illusion d'une infinité de ressources.

- **Selon Amazon Web Services** [13] : « *Le Cloud Computing est la mise à disposition de ressources informatiques à la demande via Internet, avec une tarification en fonction de votre utilisation. Au lieu d'acheter, de posséder et de gérer des serveurs et des centres de données physiques, vous pouvez accéder à votre guise aux services technologiques, tels que la puissance de calcul, le*



*stockage et les bases de données, d'un fournisseur Cloud tel qu'Amazon Web Services (AWS<sup>3</sup>) »*

- **Selon IBM<sup>4</sup> [14]** : « *Cloud Computing is on-demand access, via the internet, to Computing resources—applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more—hosted at a remote data center managed by a Cloud services provider (or CSP<sup>5</sup>)..* ». Autrement dit le Cloud Computing est un accès à la demande, via Internet, aux ressources informatiques hébergées dans un centre de données distant géré par un service Cloud fournisseur (ou CSP).

A partir de ces définitions, nous constatons que le Cloud Computing est basé sur la technologie de virtualisation, qui consiste à partager des ressources informatiques configurables à titre d'exemple, les réseaux, les serveurs, stockage de base de données, puissance de calcul, etc. Le Cloud fournit aussi à ses clients des services à la demande, instantanés et élastiques avec facturation à l'usage.

On parle de Cloud Computing lorsqu'il est possible d'accéder à des données ou à des programmes, tels que le calcul, l'Access et stockage de données à des endroits inconnus depuis l'internet, plutôt que via le disque dur d'un ordinateur. Il s'oppose ainsi à la notion de stockage local, consistant à entreposer des données ou à lancer des programmes depuis le disque dur. En utilisant cette technologie, les clients pourraient louer les services requis via des navigateurs Web [15].

Après avoir défini le concept de Cloud Computing, nous tenons à présenter dans les sections suivantes les acteurs du Cloud, ces éléments technologiques, accord de niveau de service(SLA) , ses caractéristiques, ses modèles de services et de déploiement ainsi que ses secrets du succès et enfin ses avantages et inconvénients.

---

<sup>3</sup> AWS : Amazon Web service

<sup>4</sup> IBM : International Business Machines.

<sup>5</sup> CSP: Cloud Service Provider.

#### I.4 Les acteurs du Cloud Computing :

L'architecture de référence du Cloud Computing de l'organisme NIST définit cinq acteurs majeurs: le consommateur du Cloud, le fournisseur du Cloud, l'auditeur, le Cloud broker et le transporteur de Cloud que nous détaillons comme suit [8] :

- **Consommateur** : « *une personne ou un organisme qui entretient une relation d'affaires avec le fournisseur de services Cloud et utilise le service fourni* ».

Autrement dit, le consommateur choisit le service fourni par le fournisseur de services Cloud Computing, établit un contrat de niveau de service, puis utilise le service [16].

- **Fournisseur** : « *Une personne, une organisation qui est l'entité responsable de mettre à disposition des services en Cloud aux tiers intéressés* ».

Autrement dit, le fournisseur crée, gère des services et les fournit aux clients à travers le réseau [16].

- **Auditeur** : « *un tiers qui peut procéder à une évaluation des services du Cloud, des opérations des systèmes d'information, de la performance et de la sécurité de l'implémentation du Cloud* ».

Autrement dit, les auditeurs sont chargés d'évaluer la confidentialité et les performances des services du Cloud et leur conformité [16].

- **Broker (courtier)**: « *une entité qui gère l'utilisation, la performance et la livraison de services du Cloud et négociant les relations entre les fournisseurs et le consommateur du Cloud* ».

Autrement dit, le broker est un intermédiaire entre le consommateur et le fournisseur de service Cloud. Il permet aux utilisateurs du Cloud de communiquer à

leur place avec le fournisseur de service et de sélectionner le meilleur service parmi un ensemble de services existants [16].

- **Transporteur** : « *l'intermédiaire qui fournit la connectivité et le transport des services Cloud du fournisseur vers le consommateur* ».

Autrement dit, il présente l'intermédiaire qui fournit la connectivité et transporte les services du fournisseur du Cloud vers le consommateur à travers le réseau [16][17].

## I.5 Éléments technologique du Cloud Computing :

Les technologies et les infrastructures de base nécessaires pour construire un Cloud, indépendamment de son type sont :

- **Infrastructure** :

L'infrastructure informatique du Cloud est constituée d'un assemblage de serveurs, d'espaces de stockage et de composants réseau organisés de manière à permettre une croissance incrémentale supérieure à celle que l'on obtient avec les infrastructures classiques. Ces composants doivent être sélectionnés pour leur capacité à répondre aux exigences d'extensibilité, d'efficacité, de robustesse et de sécurité [18].

- **Réseaux IP** :

Dans une infrastructure de Cloud, le réseau non seulement connecte les utilisateurs au Cloud, mais sert également à l'interconnexion interne du Cloud [18].

- **La Virtualisation** :

La virtualisation a été la première pierre vers l'ère du Cloud Computing. En effet, cette notion permet une gestion optimisée des ressources matérielles en partitionnant une ressource physique en plusieurs ressources virtuelles, par exemple : un serveur, un espace de stockage ou un réseau lors de la création des machines virtuelles. Elle permet d'intégrer les différents serveurs de façon plus flexible pour faciliter l'utilisation [1] [19].

Le but de la virtualisation est de permettre la transparence d'utilisation, l'efficacité d'exploitation des ressources, d'assurer le fonctionnement des différents services et la séparation entre les multiples utilisateurs impliqués dans un matériel physique [18].

La virtualisation permet de minimiser les coûts grâce à la possibilité de mettre en marche plusieurs machines virtuelles sur une seule machine physique donc une exploitation maximale de la machine [1].

En résumé, le Cloud s'appuie sur la technologie de virtualisation pour atteindre le but de fournir les ressources informatiques selon l'utilité de façon dynamique.

Certains des principaux avantages de la virtualisation, qui sont indigènes au Cloud, sont les suivants [19] :

- Une facturation basée sur l'utilisation (la tarification des services) et non pas sur la capacité matérielle fixe.
- Le déploiement rapide de serveurs supplémentaires.
- La séparation de la clientèle des emplacements de serveurs physiques.
- L'utilisation repose sur Service-Level Agreements (SLA).
- La tolérance aux pannes.
- La mobilité des applications entre les serveurs et les centres de données.
- Des réseaux flexibles, agiles menant à d'importantes réductions de coûts.

- **Interfaces de service :**

L'interface de service, placée entre le fournisseur et le client, est un élément de différenciation du Cloud. Elle représente un contrat qui fait respecter la proposition de valeur décrite par des SLA et des conditions tarifaires. Si le Cloud semble nouveau, c'est principalement en raison de cette interface. Elle représente la valeur d'un fournisseur et sert de base à la concurrence[18].

## **I.6 Accord de niveau de service (SLA) :**

« Accord de niveau de service » ou « Service-Level Agreement » ou « contrat de niveau de service » [19] est un contrat entre un fournisseur de service et un consommateur établie après des négociations concernant la qualité de service attendue par le consommateur, la régulation des ressources , le contrôle des coûts, les paramètres de livraison, la maintenabilité, les niveaux de disponibilité, l'exploitation, la performance ou autres attributs du service, comme les sanctions en cas de violation du contrat ainsi que tout critère auquel un prestataire informatique s'engage à répondre [20].

Autrement dit, il s'agit d'une clause contractuelle qui définit les objectifs précis et la qualité du service à attendre de la part du prestataire signataire.

## **I.7 Caractéristique du Cloud Computing :**

Le Cloud Computing se différencie des systèmes traditionnels par les cinq caractéristiques essentielles suivantes définie en octobre 2011, par l'organisme américain « National Institute of Standards and Technology » (NIST) [8] :

- **Libre-service à la demande :**

La mise en œuvre des services Cloud est entièrement automatisée et c'est l'utilisateur, au moyen d'une interface Web (console de commande), qui met en place et gère la configuration à distance [6][8], en temps réel, en fonction des besoins, et sans nécessiter d'intervention humaine [7].

- **Accès réseau large bande (Universel) :**

Les services Cloud sont à disposition de l'utilisateur universellement et simplement à travers le réseau, et prennent en charge des plates-formes clients hétérogènes telles que les appareils mobiles et les stations de travail [8]. En effet, les grands fournisseurs répartissent leurs centres de traitement de données sur la planète

pour fournir une disponibilité de 99.99%, avec un accès aux systèmes en moins de 50 ms de n'importe quel endroit du monde [21].

- **Mutualisation des ressources :**

Les ressources (bande passante réseau, machines virtuelles, mémoire, puissance de traitement, capacité de stockage, applications, etc.) sont mises en commun pour desservir plusieurs clients à l'aide d'un modèle multi-locataire [6]. Avec ce modèle, un même serveur peut servir à des dizaines de clients différents du fait que les ressources sont affectées dynamiquement et réaffectés en fonction des besoins et des clients [8]. Les économies d'échelle réalisées avec ce principe permettent de réduire significativement les coûts du fournisseur et donc les dépenses des utilisateurs [22].

- **Elasticité ou mise à l'échelle rapide :**

L'une des caractéristiques essentielles du Cloud Computing est l'élasticité des ressources [23] qui sont provisionnées et libérées à la demande. Cela garantira que les applications auront exactement la capacité dont elles ont besoin à tout moment [24][25].

- **Facturation à l'usage :**

Les systèmes du Cloud contrôlent et optimisent automatiquement l'utilisation des ressources en se basant sur le modèle « Payez ce que vous consommez », l'utilisation des services Cloud est automatiquement surveillée, contrôlée et rapportée [22], offrant la transparence pour le fournisseur de services Cloud et leurs clients. La facturation, en générale prend en compte différents éléments tels que le type de fonctionnalités demandées, la capacité de traitement, le nombre d'utilisateurs, la capacité de stockage des données et le niveau de service SLA [8].

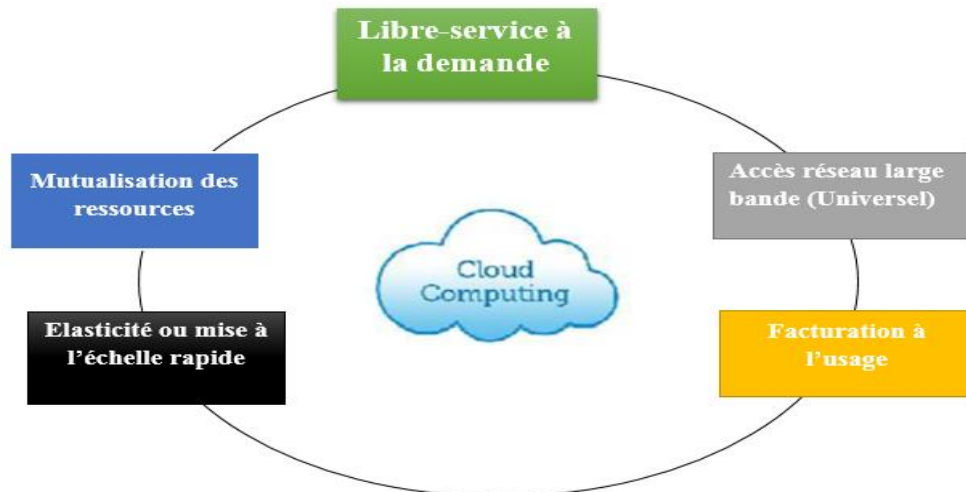


Figure I. 2:Caractéristique du Cloud Computing.

### I.8 Modèles de service du Cloud Computing:

Les services fournis dans le cadre de l'informatique dématérialisée sont catégorisés en trois sous-ensembles selon leur modèle de livraison : les logiciels en tant que services (SaaS Software as a Service), les plates-formes en tant que services (PaaS Platform as a Service) et les infrastructures en tant que services (IaaS Infrastructure as a Service) [8] [26].

Chaque modèle de service offre différents niveaux de contrôle, de flexibilité, de gestion et représente une partie différente de la pile du Cloud Computing [6][7]. Ces niveaux sont reliés entre eux de manière à ce que les niveaux les plus bas servent de base sur laquelle les autres niveaux sont fondés.

Le choix de la solution la plus adaptée dépend des besoins, compétences et budget des entreprises ou des individus [22].

La figure I.3 représente les différentes couches du Cloud Computing de la couche la moins visible pour les utilisateurs finaux à la plus visible. L'infrastructure as a Service (IaaS) est plutôt gérée par les architectes réseaux, la couche PaaS est destinée aux développeurs d'applications et finalement le logiciel comme un service (SaaS) est le « produit final » pour les utilisateurs.

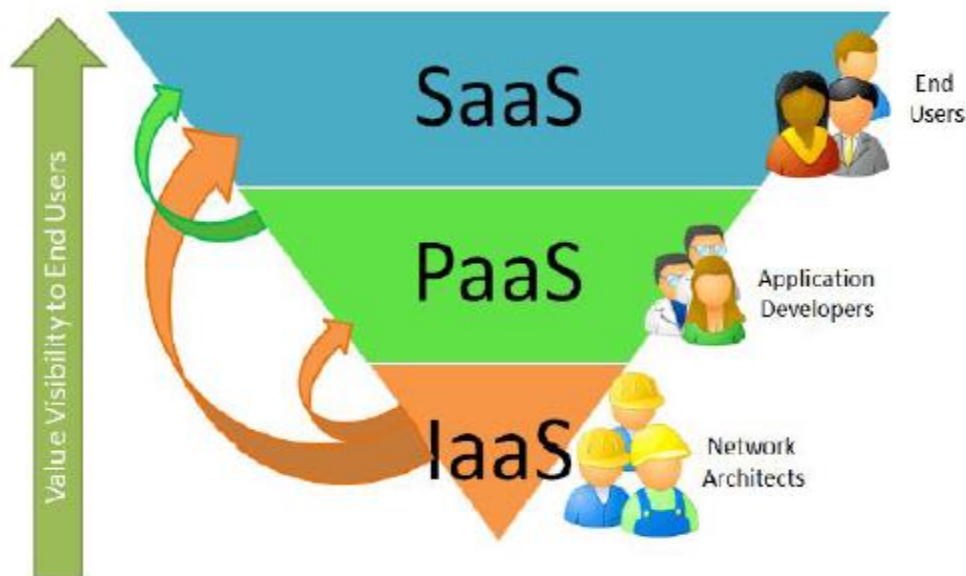


Figure I. 3:Les couches du Cloud Computing [32].

### I.8.1 Infrastructure en tant qu'un Service (Infrastructure as a Service (IaaS)) :

C'est un modèle de déploiement des infrastructures qui fournit à la demande un ensemble de services de niveau bas (serveurs, réseaux, espace de stockage, bande passante etc.) dont la plus grande partie est localisée à distance dans des Datacenter [6] [27].

IaaS fournit des services de calcul tel que EC2<sup>6</sup>, du stockage tel que S3<sup>7</sup> (Simple Storage Service), et des bases de données en ligne tel que SimpleDB<sup>8</sup> qui est un magasin de données NoSQL [8].

L'infrastructure en tant que service est fournie par des sociétés expertes en conception et gestion d'infrastructure informatique. Ces sociétés investissent dans des centres de traitement de données qui sont composés de plusieurs dizaines de milliers de machines. Ces centres sont construits de façon à optimiser les coûts de maintenance,

<sup>6</sup> Amazon EC2 : <https://aws.amazon.com/fr/ec2/>

<sup>7</sup> Amazon S3 : <https://aws.amazon.com/fr/s3/>

<sup>8</sup> Amazon SimpleDB : <https://aws.amazon.com/fr/simpledb/>



d'alimentation, de refroidissement et d'administration de l'infrastructure matérielle [28] [29].

Les raisons du succès de l'infrastructure en tant que service sont les suivantes :

- L'illusion d'une capacité de calcul, de stockage et de communication infinie de la part des utilisateurs du nuage. Internet devient alors une place de marché où l'infrastructure informatique est distribuée, et consommée en tant que marchandise.
- Réduction des coûts de maintenance des équipements.
- Personnalisation et flexibilité des services selon les besoins des clients par exemple permettre aux entreprises de commencer avec peu de ressources, puis d'augmenter si le besoin s'en fait sentir.

### **I.8.2 Plateforme en tant qu'un Service (PaaS pour Platform as a Service) :**

PaaS est la plate-forme d'exécution, de déploiement et de développement des applications. Le fournisseur offre une plateforme sous forme d'environnement complet sur laquelle des développeurs ou éditeurs de logiciels peuvent développer et déployer des applications en utilisant des langages et outils de programmation fournis par le fournisseur en ligne[8] [30]. Le déploiement des solutions PaaS est automatisé et évite à l'utilisateur d'avoir à acheter des logiciels ou d'avoir à réaliser des installations supplémentaires [31] [32].

Les principaux fournisseurs de PaaS sont : Microsoft avec AZURE, Google avec Google App Engine et Orange Business Services[33].

Certaines solutions PaaS permettent aux non-développeurs de créer des applications Web en utilisant des outils visuels plutôt qu'un langage de programmation, et certains donnent le meilleur des deux (outils visuels et langage de

programmation) c à d utiliser les outils visuels pour créer des applications et le langage de programmation pour étendre les fonctionnalités si nécessaire [32].

Le PaaS met à disposition un environnement d'exécution avec lequel les entreprises n'auront pas à se soucier ni de la gestion ni du contrôle de l'infrastructure sous-jacente (les systèmes d'exploitation, le réseau, les serveurs et le stockage) mais se concentrer sur le déploiement et la gestion de leurs propres applications [26].

Les raisons du succès de la plateforme en tant que service sont les suivantes :

- La diminution du temps de développement (l'installation et la maintenance sont assurées par le prestataire).
- Création et déploiement facile et rapide des applications.
- Les applications fournies par la plateforme seront mises à jour régulièrement donc à la pointe des meilleures technologies.

### **I.8.3 Les logiciels en tant que services (SaaS Software as a Service) :**

Ce modèle du service offre un produit final qui est exécuté et géré par le prestataire de services. Avec une offre SaaS, les utilisateurs n'ont pas à songer à la maintenance du service ; à la gestion de l'infrastructure sous-jacente ; les mises à jour, le déploiement, le stockage et la sauvegarde qui sont du ressort du fournisseur de services [7][34]. Un exemple courant d'application SaaS concerne les messageries Web (Gmail, Yahoo mail, etc...), dans lesquelles l'utilisateur peut envoyer et recevoir des e-mails sans avoir à gérer des fonctionnalités ou à effectuer la maintenance des serveurs et des systèmes d'exploitation sur lesquels elles s'exécutent[7].

Dans ce modèle, le logiciel est offert sous la forme d'un service. Les utilisateurs consomment ces services à la demande sans les acheter, avec une facturation à l'usage réel (pas d'achat d'une licence). Il n'est plus nécessaire pour l'utilisateur d'effectuer les installations, les mises à jour ou encore les migrations de données [35] [36].

Il n'y a donc aucun prérequis sur le poste client si ce n'est d'avoir un accès réseau au Cloud (en général Internet) pour pouvoir accéder aux services à travers des interfaces clients : navigateurs Web ou des interfaces de programme [26].

Les prestataires de solutions SaaS les plus connus sont Microsoft qui offre Office365 (outils collaboratifs), Google avec Gmail et YouTube ou encore les réseaux sociaux Facebook et Twitter [33].

Les solutions SaaS constituent la forme la plus répandue de Cloud Computing.

Les raisons du succès du logiciel en tant que service sont les suivantes :[32] :

- Les applications sont disponibles à l'échelle mondiale, donc réduction du temps de développement de celle-ci.
- Flexibilité, extensibilité des logiciels fournis.
- Gestion des mises à jour par les fournisseurs de service, c'est-à-dire des applications avec les versions les plus récentes sont fournis.

Mais il ya aussi un autre modèle qui est XaaS.

#### **I.8.4 XaaS:**

X signifie « Anything » (n'importe quoi) ; donc XaaS signifie « n'importe quoi en tant que Service ». Cette appellation veut dire que n'importe quel concept peut être adapté en service de Cloud Computing. D'ailleurs le XaaS a été définie comme « toute technologie fournie sur Internet qui était auparavant fournie sur site » [37].

On trouve d'autres acronymes autres que le IaaS, PaaS, SaaS qui sont :

- **BaaS** :Blockchain as a service[38].
- **DBaaS** : Database as a Service, ou Base de Données en tant que Service [39].
- **STaaS** : Storage as a Service, ou stockage en tant que Service.
- **DaaS** : Desktop as a Service, ou Station de travail en tant que Service.
- **FaaS** : Function as a Service, ou Fonction en tant que Service.
- **CaaS** : Containers as a Service, ou Centeneur en tant que Service.
- **NaaS** :Network as a Service, ou Réseau en tant que Service.
- **HaaS** :Hardware as a Service, ou Matériel en tant que service.
- **VPNaaS** : VPN<sup>9</sup> as a Service, ou VPN en tant que service.

---

<sup>9</sup> VPN: Virtual Private Network.

## **I.9 Modèles de déploiement du Cloud Computing :**

Le groupe de travail NIST a identifié quatre modèles de déploiement pour le Cloud [8] à savoir : privé, public, communautaire ou hybride, qui correspondent à des usages différents[40].

### **I.9.1 Cloud public « Public Cloud » :**

Dans un Cloud public, l'environnement est entièrement détenu par le fournisseur de service qui met à disposition ses services Cloud (des machines virtuelles, des applications, des capacités de stockage, etc...) au grand public via le réseau public. Le Cloud public peut être gratuit ou fonctionner selon le paiement à la consommation.

Dans le domaine du Cloud, et du Cloud public en particulier, la sécurité est la clé de voûte. L'infrastructure étant partagée avec de nombreux clients, la sécurité est de la responsabilité du fournisseur qui en assure la gestion. Il doit donc tout mettre en œuvre pour garder la confiance de ses utilisateurs.

L'avantage de ce genre d'architecture est d'être facile à mettre en place, pour des coûts relativement raisonnables. La charge du matériel, des applicatifs, de la bande passante est couverte par le fournisseur. De cette manière ce modèle permet de proposer une souplesse et une évolutivité accrue afin de répondre rapidement au besoin. Les freins à l'adoption de ces technologies restent en effet inchangés. Toutes tailles confondues, les entreprises jugent la sécurité, la confidentialité et l'immaturation comme les faiblesses majeures.

Un nuage public ne signifie pas que les données d'un utilisateur sont publiquement visible car les fournisseurs de Cloud public offrent généralement des mécanismes de contrôle d'accès pour les utilisateurs [41].

AWS (Amazon Web Services), IBM Bluemix, Google Cloud et Windows Azure sont des exemples de Cloud publics [21].

La figure I.4 représente le modèle de déploiement Cloud public.

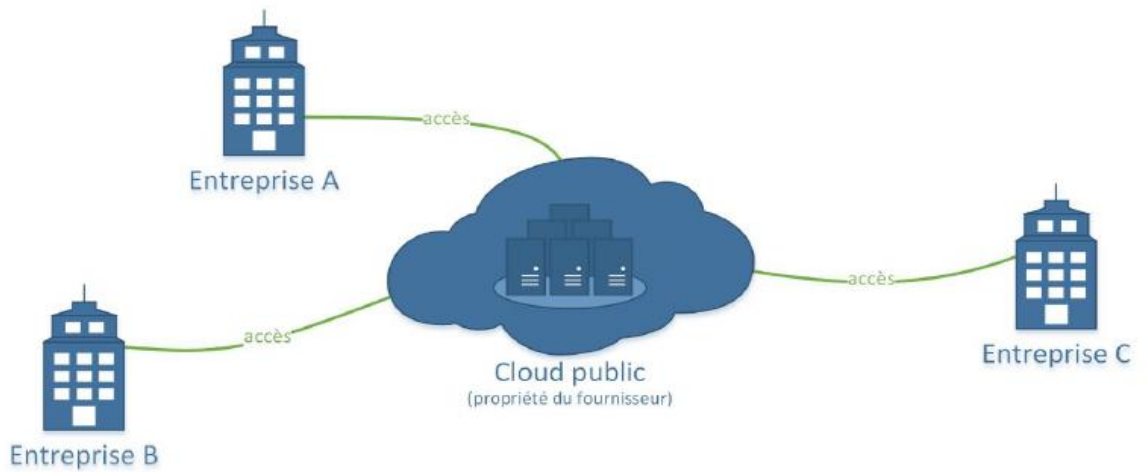


Figure I. 4:Modèle de déploiement d'un Cloud public[42].

## I.9.2 Cloud privé « Private Cloud » :

Le terme Cloud privé est utilisé pour décrire l'infrastructure qui émule le Cloud Computing sur un réseau privé. Le Cloud privé a pour ambition d'offrir certains avantages du Cloud Computing tout en limitant ses inconvénients. L'infrastructure du Cloud privé est provisionnée pour une utilisation exclusive par une seule organisation avec plusieurs consommateurs lui donnant ainsi d'avantage de contrôle, de protection et de confidentialité.

Elle peut être gérée par l'organisation elle-même ou par une tierce partie [6]. La notion de « Cloud privé » peut-être divisée en deux : le Cloud privé interne et le Cloud privé externe.

### I.9.2.1 Les Clouds privés internes :

L'architecture est hébergée dans les locaux de l'entreprise. Ce Cloud privé interne est à l'usage de plusieurs consommateurs appartenant à cette entreprise qui est propriétaire de l'infrastructure. Elle peut également être partagée ou mutualisée de

façon privée avec les filiales. Les Clouds privés internes sont gérés par l'entreprise elle-même.

### **I.9.2.2 Les Clouds privés externes :**

Le Cloud privé externe suit la même logique que le Cloud privé interne. La différence est que l'architecture est hébergée chez un prestataire. Elle est entièrement dédiée à l'entreprise et accessible via des réseaux sécurisés ou VPN.

La sécurité supplémentaire fournie par le Cloud privé est idéale pour tout type d'organisation ou d'entreprise ayant besoin de stocker et de traiter des données privées, ou d'exécuter des tâches sensibles. Le déploiement de Cloud privé semble être dirigé principalement par les grandes organisations et agences gouvernementales, plutôt que les petites entreprises et les utilisateurs finaux. La différence de coûts de démarrage est une des principales raisons pour cela [43][8].

Eucalyptus, OpenNebula et OpenStack sont des exemples de solutions proposées pour la mise en place d'un Cloud privé [8].

La figure I.5 représente le modèle de déploiement Cloud privé.

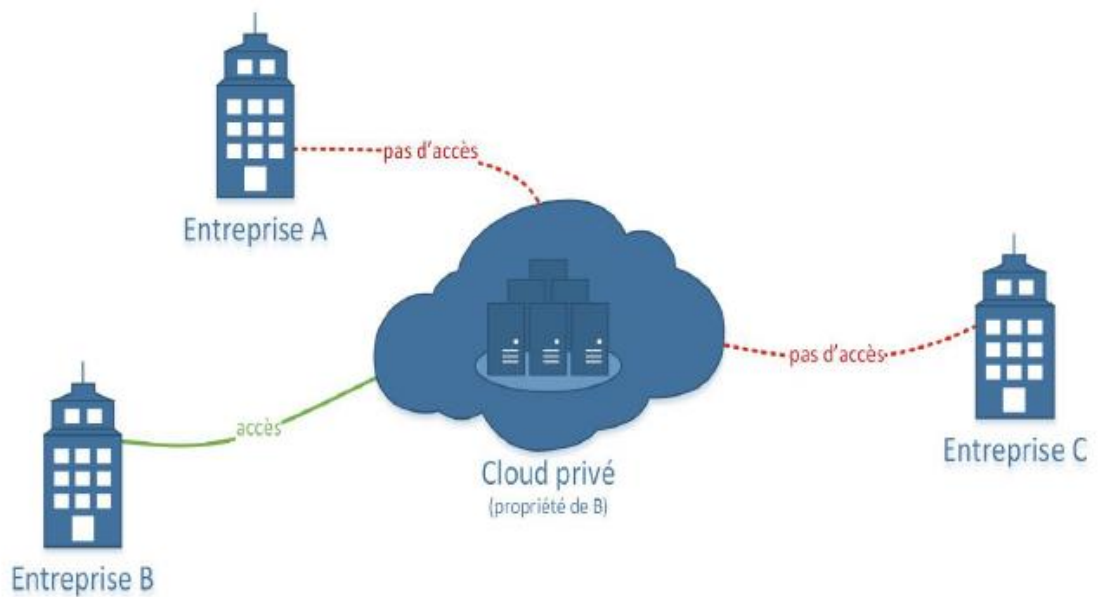


Figure I. 5:Modèle de déploiement d'un Cloud privé [42].

Les deux autres modèles de déploiement (communautaire et hybride) qui sont présentés ci-dessous, sont situés entre les Clouds public et privé [44].

### I.9.3 Cloud Communautaire « Community Cloud »

Le Cloud de type communautaire est un modèle de déploiement multi-tenant partagé entre plusieurs entreprises ou organisations. Ces communautés ont des exigences semblables et réunissent leurs moyens humains et financiers pour atteindre leurs objectifs communs.

L'infrastructure du Cloud communautaire est provisionnée pour une utilisation exclusive par une communauté spécifique de consommateurs qui partagent les mêmes domaines d'intérêts. Elle peut être gérée par les organisations membres ou par un prestataire externe et peut être placée dans les locaux ou à l'extérieur [6].

Un exemple de Cloud Communautaire est OpenCirrus formé par HP, Intel, Yahoo, et autres [41] [45].

La figure I.6 représente le modèle de déploiement Cloud communautaire.

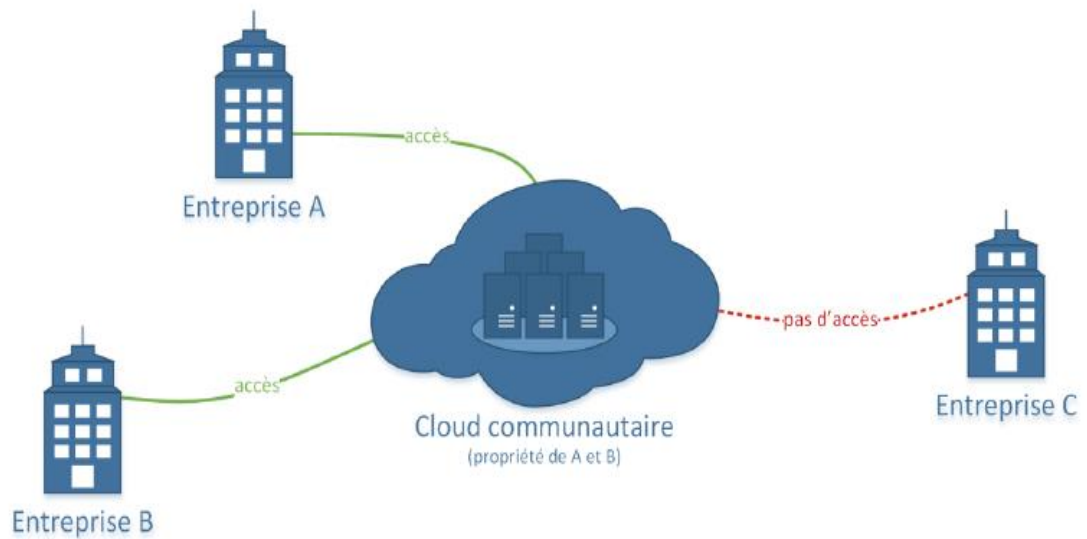


Figure I. 6:Modèle de déploiement d'un Cloud communautaire [42].

#### I.9.4 Cloud hybride « Hybrid Cloud »

Comme son nom l'indique, un Cloud hybride consiste en la combinaison de plusieurs modèles de déploiement de Clouds (privé, public et communautaire) [21] [36]. Avec un Cloud hybride, une entreprise peut tirer parti de la simplicité et du faible coût d'un Cloud public pour héberger des services classiques ne requérant pas de précautions particulières, tout en créant son propre Cloud privé.

Dans un Cloud hybride, les Clouds publics, privés ou communautaires restent des entités uniques, mais sont reliés entre eux par une technologie normalisée ou propriétaire qui permet la portabilité des données et des applications.

Le Cloud hybride permet de différencier le lieu de traitement des données selon qu'elles soient stratégiques ou pas : les données sensibles pourront alors être traitées dans les murs de l'entreprise alors que les autres le seront par un Cloud public plus rentable.



En effet, toutes les études affirment que le Cloud hybride est une solution idéale pour l'entreprise [46], en permettant à cette dernière de se rapprocher des clients et des utilisateurs distants, tout en conservant le contrôle et la protection des données dans l'environnement sur site.

La figure I.7 représente le modèle de déploiement Cloud hybride.

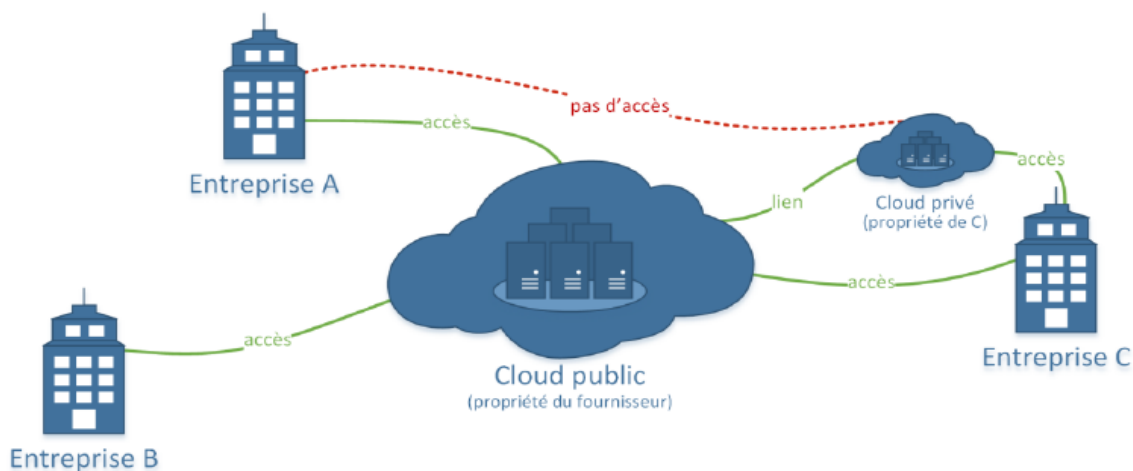


Figure I. 7:Modèle de déploiement d'un Cloud hybride [42].

### I.10 Secrets du succès du Cloud :

Le Cloud représente la solution idéale permettant d'allier les bénéfices de l'internet et les technologies informatiques. On peut résumer le succès du Cloud dans les points suivants [34][47].

- Transfert de risques vers le fournisseur.
- Simplicité d'accès et d'utilisation de services complexes.
- Pas d'investissement initial.
- Facturer à l'usage (Pay As You Go).
- Puissance de calcul élastique.

## **I.11 Avantages et limites du Cloud Computing :**

Chaque technologie possède des avantages et des limites :

### **I.11.1 Avantages du Cloud Computing**

- Le Cloud Computing offre aux développeurs et aux services informatiques la possibilité de se concentrer sur l'essentiel et d'éviter les tâches indifférenciées telles que l'approvisionnement, la maintenance et la planification des capacités [40][48].
- La réduction des coûts est une motivation dans l'adoption massive du Cloud Computing [40], puisque les clients payent que ce qu'ils utilisent comme ressources [6].
- La diminution des besoins en administration informatique, en fournissant une informatique auto-disponible, à faible coût et comportant peu de risques [45][49].
- Un autre point fort du Cloud Computing est l'accessibilité des ressources ( données, applications, etc.). En effet, ce modèle permet aux utilisateurs un accès facile et rapide à leurs données de n'importe où, à tout moment et de n'importe quel type de périphérique relié à Internet (ordinateur, Smartphone, tablette, etc.) [40][48][46].
- La flexibilité est un autre avantage du Cloud Computing [45][40], du fait que les services sont extensibles et peuvent être ajustés à tout moment en fonction des besoins.
- Une liberté totale, puisque les clients ne sont liés à leur fournisseur par aucun engagement à long terme.

- La disponibilité de la dernière version, cela évite de devoir penser à effectuer des mises à jour manuellement et à devoir s'assurer qu'on a accès à la même version de l'application partout où l'on est.

### **I.11.2 Limites du Cloud Computing :**

En dépit de ses nombreux avantages, le Cloud Computing crée également de nouveaux risques, qui viennent se greffer aux problèmes traditionnels [50] [51] :

- La sécurité est certainement l'un des enjeux majeurs du Cloud Computing et prend une place centrale dans toutes discussions concernant ce paradigme surtout le problème de fiabilité de la bande passante vers le fournisseur, ce qui rend les données vulnérables et à la portée des cybercriminels [52].
- Un autre grand problème introduit par le Cloud Computing est la perte du contrôle sur les données physiques. La CSA<sup>10</sup> [51] affirme que l'externalisation des données et les confier à un tiers externe n'est pas une mince affaire, notamment pour les entreprises.
- La nécessité d'une connexion internet de haut débit puisque le Cloud Computing ne fonctionne pas avec une connexion internet à faible débit, une coupure peut priver l'entreprise de tous les accès au Cloud et donc à toutes ses applications et données[53].
- Malgré une connexion internet rapide, avec un débit garanti, certaines applications Web peuvent s'avérer très lentes à cause de la surcharge puisque des centaines de personnes utilisent la même application au même moment donc elles peuvent être plus limitées que des applications fonctionnant sur les propres ordinateurs de l'entreprise.

---

<sup>10</sup> CSA :Cloud Security Alliance

- Cadre légal : les données transférées dans le Cloud ne sont pas forcément présentes sur le territoire national, elles peuvent l'être, comme elles peuvent être dans un autre pays. Par conséquent, sauf mention contraire du prestataire de service, on ne sait pas précisément à quel endroit sont stockées les données. De plus, on n'a aucun accès physique à ces données.
- Pérennité du service : il est nécessaire de se demander si l'hébergeur Cloud va durer dans le temps. Cet élément est important à prendre en compte car un changement d'hébergeur peut prendre du temps, et peut nécessiter un recodage des applications.
- Productivité des salariés : il est nécessaire que les employés de l'entreprise sachent se servir du Cloud. En effet, même si l'entreprise fait des économies sur le stockage et le traitement d'informations, parfois les employés passent plus de temps pour leurs tâches à cause du Cloud, ce qui risque de faire perdre plus que de gagner.
- Difficulté de réaliser la portabilité entre les différents services Cloud en raison de l'absence de standardisation et cela revient que le Cloud est un domaine relativement jeune dans le monde de l'informatique.

## **I.12 Conclusion**

Le premier chapitre vise à donner une description globale de la technologie de Cloud Computing. Ainsi son développement remarquable ces dernières années qui suscite de plus en plus l'intérêt des différents utilisateurs de l'internet et de l'informatique. C'est un nouveau modèle économique qui, en effet, promet un changement dans le mode d'investissement et d'exploitation des ressources.

Dans le chapitre suivant, nous allons explorer l'un des inconvénients auxquels le Cloud Computing est confronté, il s'agit de la portabilité.

## **Chapitre II : La portabilité dans le Cloud Computing**

## II.1 Introduction :

Dans ce chapitre, nous allons commencer par présenter l'axe central de notre travail qui est la portabilité. Dans un premier lieu, nous définissons la portabilité, type de portabilité, le verrouillage des fournisseurs et les raisons de la portabilité. Ensuite, nous mettons en relief la notion de standardisation, migration, Mapping et une comparaison entre la portabilité et l'interopérabilité.

Nous terminons ce chapitre par une étude de quelques travaux abordant la problématique de la portabilité, en proposant un tableau comparatif.

## II.2 Définition de la portabilité :

Il existe plusieurs définitions de la portabilité dans la littérature, on cite quelques-unes :

**Selon** [53] la portabilité est définie comme étant *«la capacité de déplacer une image dans un état bas d'un hôte à un autre, et de la démarrer à sa destination.»*

**Selon** [54]: *« La portabilité concerne la capacité de déplacer une entité d'un système à un autre afin qu'elle soit utilisable sur le système cible ».*

**Selon** [55] la portabilité des Clouds est la capacité des données et des composants des applications d'être facilement déplacés et réutilisés, peu importe, le fournisseur, le système d'exploitation, le stockage, le format et l'API.

Dans notre travail, on va suivre les deux dernières définitions.

### II.3 Types de portabilité :

Il existe trois type de portabilité :

- **La portabilité des applications** : C'est-à-dire que les applications peuvent être déplacé d'un environnement Cloud à un autre sans avoir à les traiter au préalable [56], c à d réutilisation des applications plus facilement entre différentes plateformes (PaaS) et infrastructures Cloud (IaaS).
- **La portabilité des plateformes** : est considérée comme une réutilisation facile des plateformes soit par le transfert complet de plateforme ou des composants de la plateforme sur une infrastructure (IaaS) Cloud [57].
- **La portabilité des données** : c'est la possibilité de déplacer des données d'un service du Cloud Computing vers un autre sans avoir à ressaisir les données de nouveau ou les modifier. Généralement les services source et cible doivent avoir la même syntaxe et sémantique des données, sinon un mapping des données doit être fait [56][54].

La portabilité des données s'avère être difficile pour le SaaS et nécessite des transformations de formats de données. Cependant, elle ne cause pas de problèmes pour les modèles IaaS et PaaS dont les données sont contrôlées par les utilisateurs, sauf dans le cas de PaaS où les fournisseurs utilisent les technologies des bases de données particulières.

La principale préoccupation découlant de l'absence de portabilité des données est l'effet de verrouillage des utilisateurs [58].

### II.4 Le verrouillage des fournisseurs (lock-in) :

Parmi les problèmes découlant des nombreuses solutions de Cloud Computing, il y a le verrouillage, où les données et les applications peuvent difficilement être déplacées vers d'autres systèmes (donc verrouillées), forçant les utilisateurs du Cloud

à s'appuyer sur un fournisseur de services sans avoir la possibilité de changer de fournisseur. Et même s'ils le peuvent, ça sera une opération très coûteuse en raison de l'incompatibilité des nuages, chacun sa propre API, son format de description et son propre format de SLA [59] [60] [61].

La résolution de ce verrouillage du fournisseur repose sur la capacité de faire travailler ensemble différents fournisseurs de services Cloud [62], pour permettre la réalisation de l'interopérabilité dans les environnements de Cloud Computing [63].

## **II.5 Raisons de la portabilité :**

La portabilité est nécessaire pour plusieurs raisons[64] :

- Raisons techniques : pour l'exploitation de l'avantage de l'élasticité et du concept de paiement à l'usage.
- Raisons économiques : protection des investissements des utilisateurs c'est-à-dire pas de réécriture des services et des applications lors de changement du service Cloud.
- Raison juridiques : pour éviter de dépendre d'un seul fournisseur de service.

## **II.6 Notions importante pour la portabilité des données**

Dans cette section, nous allons présenter des notions qui sont importante pour la portabilité des données :

### **II.6.1 Standardisation :**

La standardisation dans le Cloud Computing vise à fournir une approche unifiée pour les services du Cloud à travers les différents prestataires. Ces standards permettent d'éliminer les barrières d'interopérabilité et de portabilité entre les environnements de Cloud Computing en décrivant le Cloud, l'interface de communication et le format des données [65].



Ainsi, il n'y aurait plus besoin d'apporter des modifications aux données, aux applications et charges de travail pendant leur déplacement d'un environnement Cloud à un autre.

Le domaine du Cloud Computing n'est pas standardisé et cela revient que c'est un domaine relativement jeune dans le monde de l'informatique, malgré que beaucoup de groupes et organisations travaillent dessus [66] [67].

### **II.6.2 Migration :**

La migration est le processus de déplacement d'entités telles que des données, des applications ou des charges de travail vers un environnement de Cloud Computing [53][68] [69].

Il existe trois types de migration dans le Cloud Computing :

- **Cloud to Cloud** : c'est le passage d'un Cloud à un autre. Par exemple si une entreprise trouve un autre service plus performant que celui qu'elle utilise déjà, elle va migrer vers ce nouveaux service.
- **Local to Cloud** : c'est le passage d'un environnement local vers un environnement Cloud, pour profiter de plusieurs avantages : le paiement à l'usage et l'élasticité.
- **Cloud to exit** : c'est le passage d'un environnement Cloud vers un site ou centre de données local.

### **II.6.3 Mapping :**

Aujourd'hui, les données proviennent de nombreuses sources, et les mêmes points de données peuvent être définis différemment par chacune de ces sources.

Le mapping des données aussi appelé cartographie des données consiste à mettre en correspondance les champs des données sources et les champs cibles de destination.

Le mappage des données résout les différences entre deux systèmes, ou modèles de données, pour que les données soient exactes et utilisables lorsqu'elles sont transférées de la source à la cible.

Le mappage de données est la première étape pour faciliter la migration, l'intégration des données et d'autres tâches de gestion des données.

Le mapping de données se fait à travers des outils de mapping et de cartographie tels que les ontologies [70].

## **II.7 Portabilité et interopérabilité :**

L'interopérabilité est définie comme étant la facilité de migration des applications, des données, des charges de travail et de compréhension des formats des données entre différents fournisseurs de services Cloud [53] [53].

L'un des plus grands défis de l'interopérabilité est la portabilité et l'échange de données. La conversion des données du format du système source au format requis par le système cible peut demander beaucoup d'efforts car les fournisseurs de services Cloud Computing n'utilisent pas tous les mêmes modèles et technologies [54][56]. Donc pour pouvoir interopérer des services Cloud, il faut assurer la portabilité entre eux.

## **II.8 Solutions existantes pour la portabilité dans le Cloud Computing :**

### **II.8.1 Solution de Thabet [43] :**

Dans cette approche, l'auteur a considéré une entreprise qui utilise deux fournisseurs Cloud. Le premier offre les applications informatiques décisionnelles afin

de réaliser des tableaux de bord et des rapports aidant à prendre des décisions pour ses clients et le deuxième offre les services CRM<sup>11</sup> pour la gestion de relation client.

Chaque fournisseur utilise sa propre technologie, son modèle et son format de données.

Pour résumer cette approche repose sur SMA<sup>12</sup> (système multi-agent) qui comporte cinq types d'agents illustré dans la figure suivante :

- Un agent qui représente chaque fournisseur de services Cloud.
- Un agent coordinateur qui sert à coordonner l'interaction entre tous les agents.
- Un agent descripteur qui sert à décrire tous les services offerts dans un registre.
- Un agent mobile qui assure la portabilité des données.
- Un agent adaptateur qui permet de résoudre le problème d'hétérogénéité syntaxique des données échangées entre les Clouds.

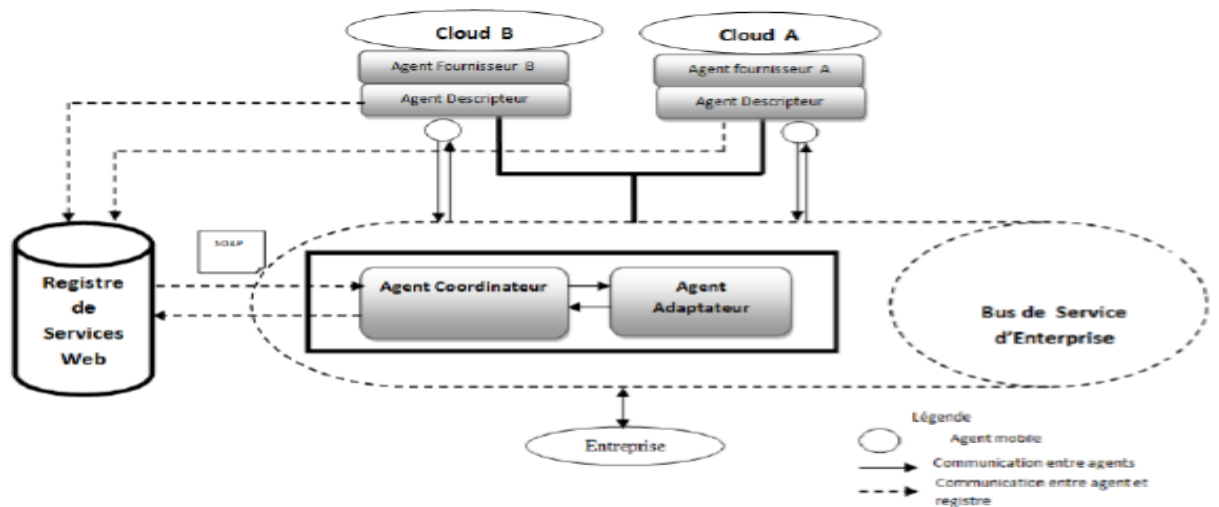


Figure II. 1: Approche proposer basé sur les Système Multi-Agent (SMA)[49].

L'entreprise utilise le fournisseur A pour faire ses rapports d'analyse pour déterminer les clients présentant une probabilité d'achat. Donc afin d'invoquer le

<sup>11</sup> CRM: Customer Relationship Management.

<sup>12</sup> SMA: Système Multi Agent.

fournisseur A, l'agent coordinateur accède au registre de services pour avoir les adresses des services. Une fois l'agent coordinateur a obtenu l'adresse, il invoque le fournisseur A qui a besoin des données qui concerne les clients et les produits, qui sont stockées chez le fournisseur B.

Pour satisfaire les besoins de l'entreprise, l'agent fournisseur envoie une requête à l'agent coordinateur qui vérifie auprès de l'agent adaptateur si les données requises sont disponibles dans sa base de données (l'auteur a pris en considération le cas où les données ne sont pas disponibles). L'agent adaptateur envoie alors une requête en indiquant que les données ne sont pas disponibles. L'agent coordinateur transmet la requête du fournisseur A au fournisseur B en lui envoyant les données requises. L'agent fournisseur B active l'agent mobile et lui transmet la requête contenant toutes les données requises par le fournisseur A.

L'agent mobile se clone et migre au sein du Cloud, d'une base de données à une autre, afin de rechercher les données nécessaires. Une fois les données trouvées par les agents mobiles, elles seront transmises à l'agent adaptateur, il va extraire les données à travers son parseur XML<sup>13</sup> puis il génère une base de données selon le modèle de données défini par le fournisseur A. Une fois la génération de la base de données achevée, l'agent adaptateur envoie une requête contenant toutes les données nécessaires au fournisseur A.

Le fournisseur A intègre les données et offre les services à l'entreprise, il les transmet à l'agent coordinateur qui à son tour transmet les services à l'entreprise.

Dans cette approche [43], l'auteur a proposé de déployés les agents Coordinateur et Adaptateur, dans un bus ESB<sup>14</sup>. Ce bus offre une solution distribuée.

## **II.8.2 Solution de Lupşe et al [71]:**

---

<sup>13</sup> XML : Extensible Markup Language.

<sup>14</sup> ESB :Entreprise Service Bus.

Cette solution est proposée entre deux application : pédiatrie et gynécologie. Le flux d'informations entre les composants dans ce modèle est le suivant : l'application Pédiatrie envoie un XML avec l'ID<sup>15</sup> de la mère, l'ID du bébé et la date de naissance du bébé, l'application gynécologie lit la requête XML, identifie les données nécessaires et les convertit en XML puis les envoie au service de pédiatrie où les données sont lues et renseignées dans le dossier du bébé.

Pour une sécurité accrue, la solution suggérée consiste en une architecture privée basée sur le Cloud c à d un Cloud privée où les applications et le stockage de données peuvent être trouvés dans chaque centre de données privé de l'hôpital (un au service pédiatrique et un au service gynécologie).

### **II.8.3 Solution Benhssayen and Ettalbi [72] :**

La solution proposée est un cadre visant à maximiser le niveau d'interopérabilité sémantique entre Ressources IaaS, dans un environnement multi-Cloud. Il se compose de :

- Composants interagissant avec les fournisseurs Cloud (IAAS Resource Requestor Component) qui constitue l'interface de framework avec les fournisseurs Cloud, ses principales fonctions c'est de demander les ressources IaaS disponibles depuis le fournisseurs Cloud périodiquement et envoyer les réponses reçues (format XML ) au composant (Resource Persistency) afin de créer des ontologies liées à OWL<sup>16</sup>.
- Composants (Resource Persistency), assure la persistance et la mise à jour des réponses XML reçues, et leurs ontologies OWL en les stockant dans (Knowledge Base).
- Composant (Resource Mapping composants) reçoit l'ontologie OWL liée à chaque réponse nouvelle ou mise à jour du fournisseur Cloud ensuite procède à la cartographie de chaque ontologie OWL en utilisant les ontologies des normes IAAS stockées dans

---

<sup>15</sup> ID : IDentifiant.

<sup>16</sup> OWL: Ontology Web Language

le composant Knowledge Base tel que OCCI<sup>17</sup> afin de récupérer les description sémantique de chaque ressource Cloud dans l'ontologies standards. Les résultats sont stockés dans (Resource DataBase Component) sous forme de RDF<sup>18</sup>.

-Composant Constitue l'interface du framework avec les consommateurs Cloud (composants IaaS Resource Request Listener).

La figure 2 présente l'architecture générale du Framwork .

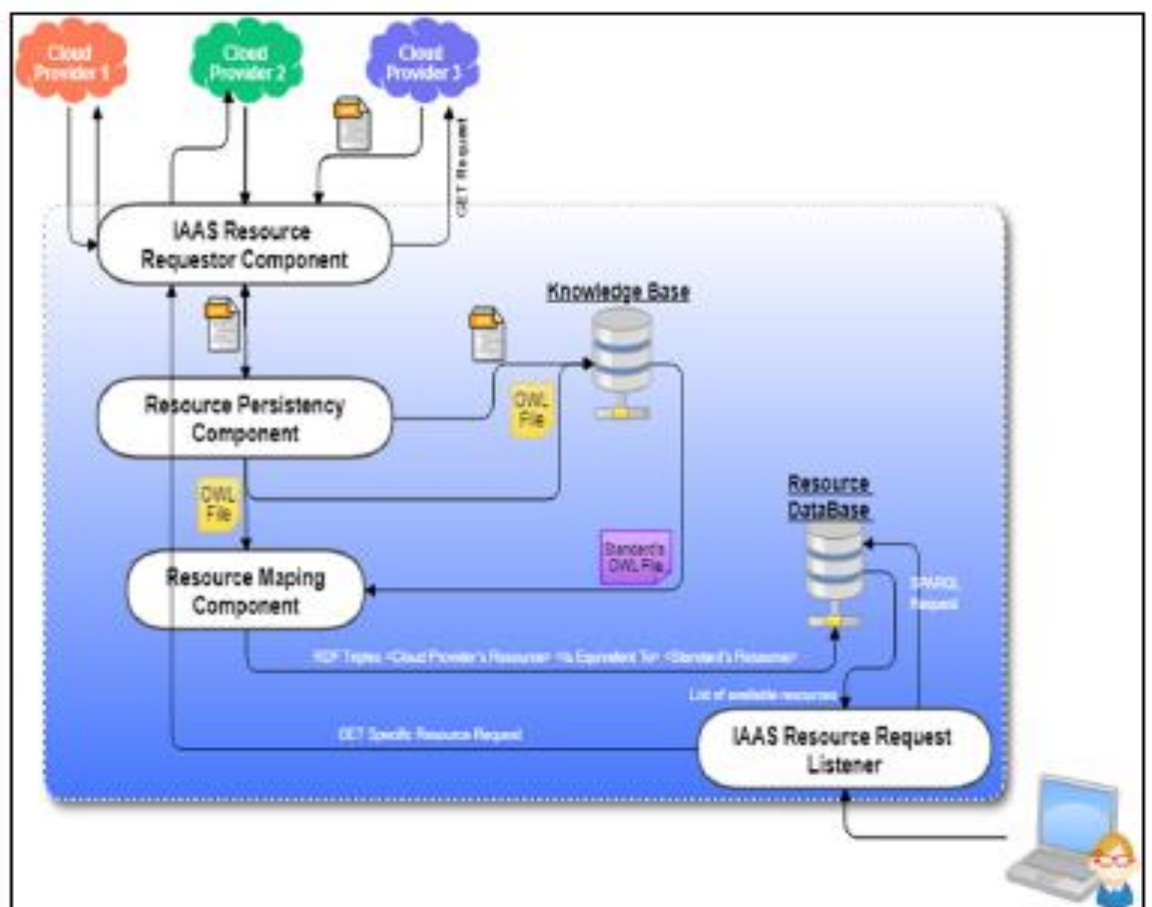


Figure II. 2: Architecture d'interopérabilité sémantique des ressources IaaS [82].

<sup>17</sup> OCCI: Open Cloud Computing Interface

<sup>18</sup> RDF: Resource Description Framework

#### **II.8.4 Solution Kwon et al [73] :**

Dans cet article, ils ont proposé Mobility Handicapped System pour prévenir les accidents de la circulation sur les passages piétons et pour fournir un service de notification en cas de situation dangereuse car la vitesse de marche des personnes à mobilité réduite est plus lente que celle du grand public.

Quand le système détecte le signal de danger comme les chutes, il transmet les informations sur l'état de santé au centre d'urgence.

Dans les systèmes à mobilité réduite, lors de l'envoi de messages de notification aux centres d'urgence et aux hôpitaux, des problèmes d'hétérogénéité des données surviennent en raison des différents formats de schémas et de données pour les centres d'urgence et les hôpitaux.

Par conséquent, le système doit résoudre l'hétérogénéité de données en mappant le format des données. Pour résoudre ce problème, ils ont conçu un système d'interopérabilité des données basé sur DBaaS<sup>19</sup> (Database as a Service).

DBaaS est proposé comme l'une des technologies de Cloud Computing. Cela utilise efficacement une base de données conçue différemment. En d'autres termes, il n'est pas nécessaire de reconfigurer la base de données ou le système, ce qui réduit le temps et l'argent.

#### **II.8.5 Solution Bouzerzour et al [74] :**

Dans cette approche, les auteurs ont proposées une transformation de différents langages de description de service Cloud hétérogène à une GCSD<sup>20</sup> pour assurer l'interopérabilité des services Cloud.

---

<sup>19</sup> DBaaS:Database as a Service.

<sup>20</sup> GCSD : Generic Cloud Service Description.

Cette GCSD est une nouvelle description de service Cloud général proposé dans cet article qui permet de rendre les descriptions interopérables en utilisant un modèle de pivot.

La transformation dans ce cas se fait de OWL-S<sup>21</sup> a GCSD.

L'OWL-S est une ontologie basée sur OWL pour les services Web, elle a trois parties :

- The Service Profile : c'est une description des aspects généraux (le nom de service, description textuelle...).
- The Service Model (Process model): fournit une description détaillées (entrées, sorties, conditions préalables, résultats).
- The grounding: décrit la manière de l'invocation de service.

Les auteurs ont proposé des règles de mappage séparés pour chaque partie d'OWL-S.

Le tableau II.1 représente un tableau comparatif des différentes solutions étudiées.

---

<sup>21</sup> OWL-S : Ontology Web Language for Service.



<i>Critères</i>	<b>Solution à base de Cloud</b>	<b>Modèle de service utilisé</b>	<b>Modèle de déploiement utilisé</b>	<b>Nombre de fournisseurs Cloud</b>	<b>Aspect Mapping</b>	<b>Format d'échange</b>
<b>Solutions effectués</b>						
Solution de Thabet [43]	<i>oui</i>	<i>SaaS</i>	<i>public</i>	2	<i>pas traité</i>	<i>Agent (SMA)</i>
Solution de Lupşe et al [71]	<i>oui</i>	<i>SaaS</i>	<i>privé</i>	2	<i>pas traité</i>	<i>Xml</i>
Solution Benhssayen and Ettalbi [72]	<i>oui</i>	<i>IaaS</i>	-	<i>plusieurs</i>	<i>traité</i>	<i>ontologie</i>
Solution Kwon et al [73]	<i>non</i>	-	-	-	<i>traité</i>	<i>DBaaS</i>
Solution Bouzerzour et al [74]	<i>oui</i>	-	-	<i>plusieurs</i>	<i>pas traité</i>	<i>ontologie</i>

Tableau II. 1: Tableau comparatif.

## II.9 Discussion :

Le tableau ci-dessus (Tableau II.1) représente une comparaison des travaux étudiés :

La solution de Thabet [43] se base sur les systèmes multi agents dans un environnement Cloud public pour les services SaaS, alors que la solution de Lupşe et al [71] est une solution proposer pour le Cloud privé pour les services SaaS en utilisant un format d'échange XML.

Concernant la solution Benhssayen and Ettalbi [72] et la solution Bouzerzour et al [74] se sont des solutions pour le Cloud avec plusieurs fournisseurs, leur format d'échange se base sur les ontologies.

Par contre la solution Kwon et al [73] n'est pas proposé pour le Cloud mais son format d'échange est basé sur DBaaS .

Parmi les solutions étudiées, il y a que la solution de Benhssayen and Ettalbi [72] qui prend en charge l'aspect de mapping pour les services Cloud.

## **II.10 Conclusion :**

Dans ce chapitre, nous avons présenté la définition de la portabilité, type de portabilité, le verrouillage des fournisseurs et les raisons de la portabilité.

Ensuite, nous avons mis en relief la notion de standardisation, migration, mapping et une comparaison entre la portabilité et l'interopérabilité.

Nous avons terminé ce chapitre par une étude de quelques travaux abordant la problématique de portabilité, le tous résumer dans un tableau comparatif et une discussion.

## **Chapitre III : Conception de notre solution**

### III.1 Introduction :

La conception d'un système est une étape cruciale pour l'évaluation des besoins d'un utilisateur. L'objectif de ce chapitre est de présenter une solution à notre problématique.

Dans ce chapitre, nous allons présenter la conception de notre solution. Nous commençons d'abord par présenter l'analyse des besoins avec le diagramme de cas d'utilisation, ensuite le schéma global de notre système et son fonctionnement, en expliquons chaque étape de ce schéma avec les pseudo algorithmes correspondants.

### III.2 Analyse des besoins :

Dans cette section, nous allons définir les besoins de notre système en présentant les acteurs qui interagissent avec le système et ses cas d'utilisations.

#### Les acteurs du système :

Un acteur est une entité externe qui peut être un utilisateur humain, un dispositif matériel ou un autre système. Cette entité interagit avec un système en lui fournissant des données en entrée et en déclenchant ses fonctionnalités.

Dans notre approche, les acteurs sont des utilisateurs qui utilisent ce système de mapping qui sont les fournisseurs de service et l'administrateur.

Le tableau *III.1* présente ces utilisateurs et leur rôle associé :

Acteur	Rôle
Fournisseur de service	L'entité qui veut voir les compatibilités de ses services.
Administrateur	L'entité qui va déclencher le processus de mapping.

Tableau III. 1: Les acteurs et leur rôle.

Le diagramme de cas d'utilisation suivant montre les actions possibles de chaque acteur :

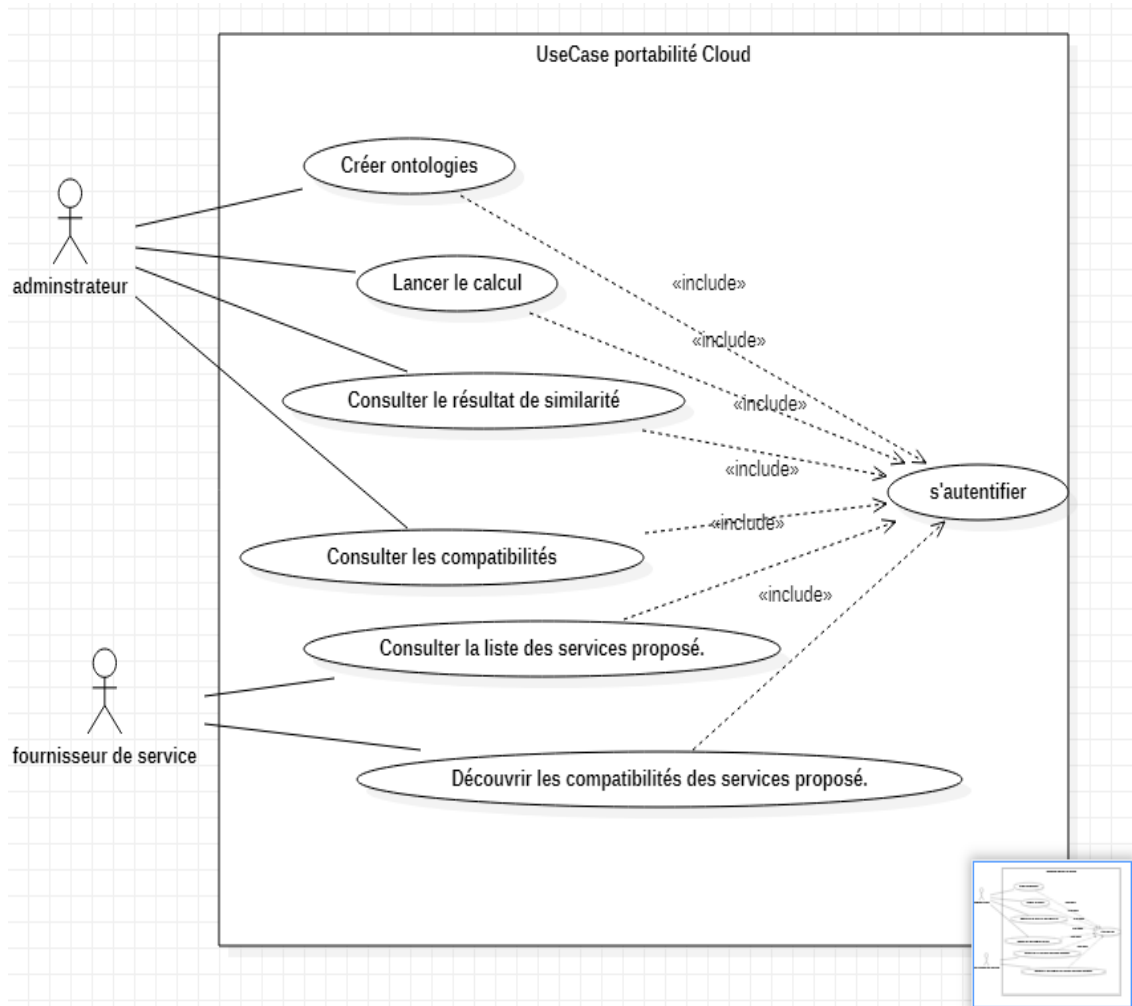


Figure III. 1: Diagramme de cas d'utilisation

Le tableau III.2 présente une description détaillée de chaque cas d'utilisation :

<b>Cas d'utilisation</b>	<b>Description textuelle</b>
S'authentifier.	Le fournisseur de services et l'administrateur ne peuvent accéder au système que s'ils en le droit, ils doivent donc entrer un nom d'utilisateur et un mot de passe.
Créer ontologies .	L'administrateur va créer une ontologie pour chaque service Cloud.
Lancer le calcul.	L'administrateur lance le calcul de similarité en utilisant les différentes mesures.
Consulter le résultat de similarité.	L'administrateur consulte le résultat de calcul de similarité entres les différentes ontologies.
Consulter les compatibilités des services Cloud.	L'administrateur consulte les compatibilités entres les services Cloud.
Consulter la liste des services proposés.	Le fournisseur de service peut consulter la liste de ses services Cloud proposés.
Découvrir les compatibilités des services proposés.	Le fournisseur de service peut découvrir les compatibilité de leurs services Cloud proposés.

Tableau III. 2:Description textuelle des cas d'utilisation.

### **III.3 Schéma global de la solution proposée :**

Notre système consiste à effectuer la portabilité des données entre des services Cloud public de type Software as a Service.

Nous allons commencer par présenter le schéma global de notre solution proposé illustré dans la figure suivante :

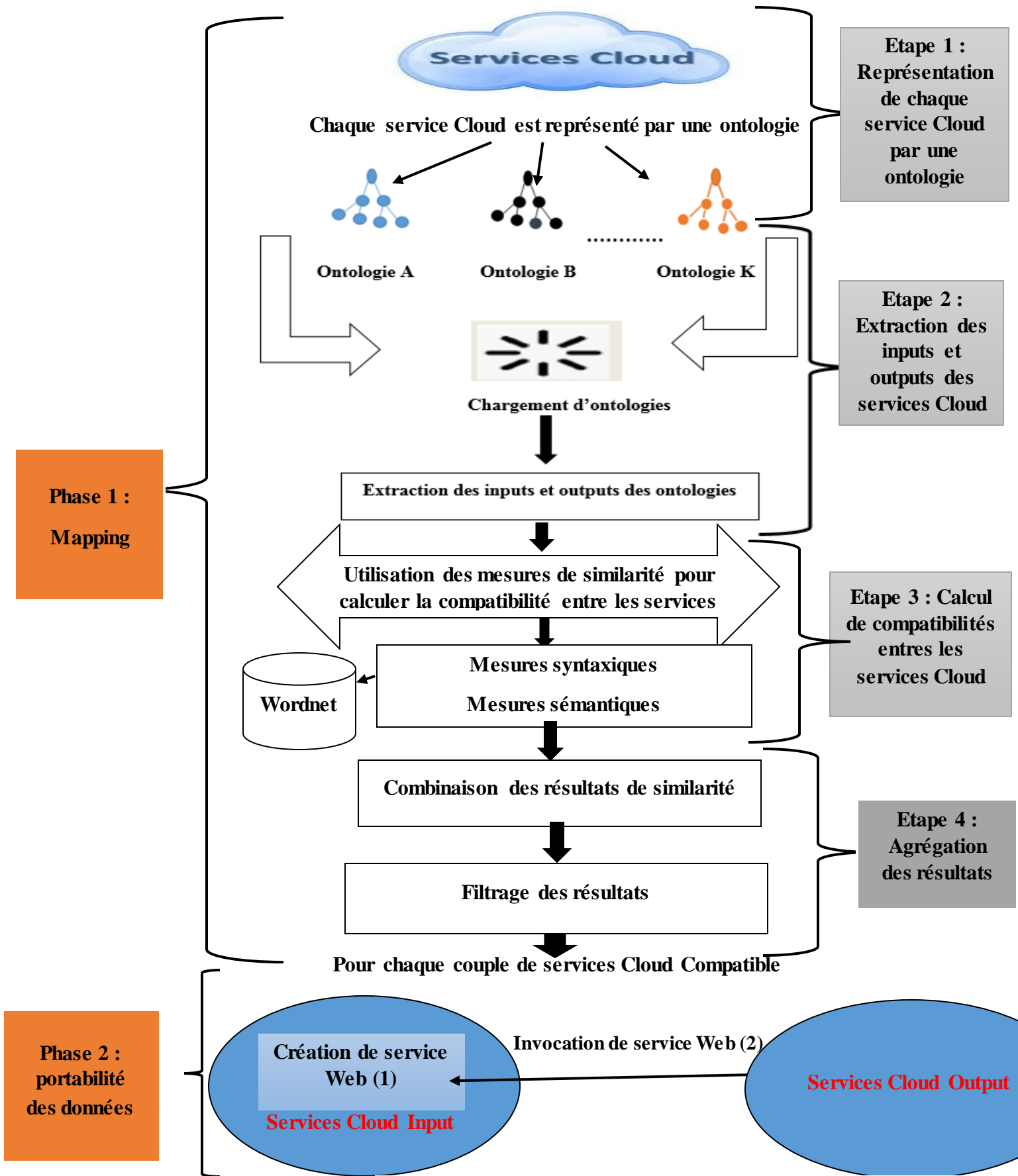


Figure III. 2:Schéma global de notre solution.

### **III.4 Description générale de notre approche**

Notre approche est composée de deux phases, chaque phase présente une ou plusieurs étapes.

#### **Phase 1 : Le mapping**

Avant de faire la portabilité entre deux services Cloud A et B, tel que les outputs de A doivent être utilisés comme inputs de B, il faut s'assurer que réellement les outputs de A et les inputs de B sont compatibles, et cela en utilisant les outils de mapping tel que les ontologies. Chaque ontologie va représenter un service Cloud en définissant ses inputs et outputs.

Cette phase est réalisée en quatre étapes :

- Etape 1 : Représentation de chaque services Cloud en une ontologie.
- Etape 2 : Extraction des inputs et outputs des services Cloud.
- Etape 3 : calcul de similarité entre les services Cloud.
- Etape 4 : Agrégation des résultats.

#### **Phase 2 : Portabilité**

Après la réalisation de la phase de mapping, il serait possible de faire la portabilité entre les services Cloud compatibles et cela en utilisant les services Web.

### **III.5 Description détaillée de l'approche proposée**

L'approche que nous avons proposée consiste à assurer la portabilité des données entre les services Cloud, mais avant cela, il faut que les services Cloud soient compatibles.

Dans cette section, nous allons expliquer en détail l'approche proposée.



### **III.5.1 Phase de mapping :**

#### **III.5.1.1 Etape 1 : Représentation de chaque service Cloud en une ontologie**

La première étape à réaliser c'est la représentation des inputs et outputs de chaque service Cloud en une ontologie. Ces ontologies sont écrites dans le format OWL qui est un langage ontologique, capable de décrire formellement la signification de la terminologie employée dans les documents Web. OWL constitue le premier niveau nécessaire du Web sémantique après RDF [75] (voir annexe A : Notions sur les ontologies).

#### **III.5.1.2 Etape 2 : Extraction des inputs et outputs des services Cloud.**

Dans cette étape, on doit tout d'abord charger les ontologies, ensuite parcourir toute la hiérarchie de ces ontologies afin d'extraire tous les concepts inputs et outputs comme l'indique le pseudo-algorithme suivant :

**Algorithme 1 de chargement des ontologies et d'extraction des inputs et des outputs :**

**Entrée : Les ontologies**

**Sortie : Liste inputs et outputs**

**Début**

1/- /\*chargement des ontologies\*/

**Lire** (ontologie)

2/- /\*extraction des inputs des ontologies dans une liste A\*/

**A = extractionInput ();**

3/- /\*parcourir la liste des inputs A\*/

**Pour j← 1 a A.length () faire**

4/- /\*ajouter les inputs dans la base de données \*/

**addInput ();**

**finPour**

5/- /\*extraction des outputs des ontologies dans une liste B\*/

**B = extractionOutput ();**

6/- /\*parcourir la liste des outputs B\*/

**Pour j← 1 a B.length () faire**

7/- /\*ajouter les outputs dans la base de données \*/

**addOutput ();**

**finPour**

**fin pour**

Algorithme III. 1:Algorithme chargement des ontologies et d'extraction des inputs et des outputs

Il existe plusieurs moyens pour évaluer la similarité entre deux entités. Dans notre travail, notre choix s'est basé sur l'utilisation des mesures de similarité syntaxique et sémantique [76][77] [78].

Nous détaillons dans ce qui suit les différentes mesures de similarité utilisées :

### **III.5.1.3.1 Mesure de similarité syntaxique :**

Pour évaluer la correspondance entre les inputs et outputs de deux ontologies, nous avons appliqué la distance de DICE et l'indice de Jaccard, qui calculent la similarité entre deux paires de chaînes de caractères.

#### **III.5.1.3.1.1 L'indice de Dice :**

L'indice de Dice est défini par le nombre des objets communs multipliés par 2 sur le nombre total d'objets. La mesure de Dice est donc définie par la formule suivante :

$$SimDice (X, Y) = \frac{2 * |X \cap Y|}{|X| + |Y|} \quad (3.1)$$

Le résultat de cette mesure est compris entre 0 et 1, plus la valeur est proche de 1 plus les paires de chaînes sont similaires. Si le résultat est égal à 1, cela signifie que les paires sont identiques.

On considère X comme étant un concept d'une ontologie et Y un autre concept d'une autre ontologie.

La similarité entre X et Y est calculée en traitant chaque caractère qui compose les deux concepts.

**Exemple :**

X: animaux. (a n i m a u x)

Y: animal. (a n i m a l)

$X \cap Y$  représente le nombre de caractères en commun entre X et Y = 5.

$|X|$  Représente le nombre de caractères du mot X = 7.

$|Y|$  Représente le nombre de caractères du mot Y = 6.

$$SimDice(X, Y) = \frac{2 * 5}{7 + 6} = 0.76 \quad (3.2)$$

Nous avons aussi utilisé l'indice de Jaccard comme une autre mesure de similarité syntaxique.

**III.5.1.3.1.2 L'indice de Jaccard :**

L'indice de Jaccard ou coefficient de Jaccard est le rapport entre la cardinalité (la taille) de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles. La mesure de Jaccard est donc définie par la formule suivante :

$$SimJaccard(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \quad (3.3)$$

**Exemple :**

X : animaux. (a n i m a u x)

Y : animal. (a n i m a l)

$X \cap Y$  Représente le nombre de caractères en commun entre X et Y = 5.

$$X \cup Y = |X| + |Y| - |X \cap Y| \quad (3.4)$$

$X \cup Y$  Représente la cardinalité de l'union entre X et Y = 8.

$$SimJaccard(X, Y) = \frac{5}{8} = 0.625 \quad (3.5)$$

### **III.5.1.3.2 Mesure de similarité sémantique :**

Ces mesures consistent à utiliser des ressources externes (dictionnaires, taxonomies...). Elles permettent de déterminer la similarité entre deux entités.

Plusieurs types de ressources peuvent être utilisées, notre choix s'est porté sur WordNet.

WordNet est le produit d'un projet de recherche à l'Université de Princeton. C'est une base de données lexicales en ligne contenant des noms, verbes et adjectifs en anglais. Ces entités sont organisées en ensemble de synonymes (appelés synsets) reliées par différentes relations sémantiques. Chaque synset regroupe tous les termes dénotant un concept donné [76].

La similarité est calculée à partir des liens sémantiques déjà existants dans les ressources externes [79].

L'objectif des mesures de similarité sémantique est d'évaluer la proximité sémantique et de déterminer le degré de correspondance entre les concepts qui se voient attribuer une métrique basée sur la ressemblance de leur signification / contenu sémantique [79].

Nous avons utilisés deux approche de similarité sémantique [79] [80] :

#### **III.5.1.3.2.1 Mesures basés sur les chemins :**

Dans le cadre de cette méthode, la similarité est calculée en fonction de la distance entre les concepts et leurs positions dans la taxonomie. Elle est totalement basée sur la représentation graphique de Wordnet.

Parmi les mesures utiliser classifiés sous cette approche :

### III.5.1.3.2.1.1 Wu et Palmer

La métrique de similarité de Wu et Palmer est basée sur la notion « the least common super-concept » c'est-à-dire le concept commun le plus éloigné de la racine dans la taxonomie Wordnet. Cette mesure reflète que plus on descend dans la hiérarchie, plus les nœuds sont proches sémantiquement.

La mesure de Wu et Palmer est donc définie par la formule suivante :

$$SimWuPalmer(N1, N2) = \frac{2 \times depth(N)}{depth(N1) + depth(N2)} \quad (3.6)$$

Où N est le concept commun à N1 et N2 le plus éloigné de la racine,

depth (N) c'est le nombre d'arcs qui sépare N de la racine et depth (Ni) représente le nombre d'arcs qui sépare Ni de N.

**Exemple:**

N1: Name.

N2: Username.

$$SimWuPalmer(N1, N2) = 0.93 \quad (3.7)$$

On a utilisé une autre mesure de similarité sémantique basé sur les chemins qui est Path.

### III.5.1.3.2.1.2 Path :

Cette mesure est basée sur le plus court chemin des deux concepts dans la taxonomie Wordnet.

La mesure Path est définie par la formule suivante :

$$SimPath(N1,N2) = 2 \times deep\_max - len(N1,N2) \quad (3.8)$$

Pour une version spécifique de la taxonomie WordNet, *deep\_max* est une valeur fixe qui représente la profondeur et *len(N1,N2)* est une fonction du plus court chemin entre N1 et N2.

**Exemple :**

N1 : Name.

N2 : Username.

$$SimPath(N1,N2) = 0.5 \quad (3.9)$$

### III.5.1.3.2.2 Mesures basées sur les nœuds :

Ces mesures se base sur la notion du contenu informationnel (CI) qui mesure la pertinence d'un concept dans le corpus [78].

Dans cette approche, plus les deux concepts partagent des informations communes, plus les concepts sont similaires [79].

Le CI est calculé comme suit :

$$CI(synset) = \log(p(synset)) \quad (3.10)$$

Où :

$P(Synset)$  est la probabilité de trouver un synset ou un de ses descendants dans le corpus, il est calculé par la formule suivante :

$$P(synset) = \frac{frequence(synset)}{N} \quad (3.11)$$

Où :

$frequence(Synset)$  c'est le nombre d'apparition du synset.

N est le nombre total de Synsets .

Parmi les mesures basées sur le contenu informationnel, nous avons optées pour l'utilisation de la mesure de Lin.

#### III.5.1.3.2.2.1 Mesure de Lin

La mesure de Lin retourne le contenu informationnel (CI) du plus bas ancêtre commun (LCS) de deux concepts donnés N1 et N2.

La mesure Lin est donc définie par la formule suivante :

$$SimLin(N1, N2) = \frac{2 \times CI(LCS(N1, N2))}{CI(N1) + CI(N2)} \quad (3.12)$$

**Exemple :**

N1 :Name.

N2 :Surname.

$$SimLin(N1, N2) = 0.0 \quad (3.13)$$

#### III.5.1.4 Etape 4 : Agrégation des résultats

Après le calcul de similarité entre les différents concepts provenant des différentes ontologies, la dernière étape de la phase de mapping est l'agrégation des résultats (combinaison des mesures de similarité et le filtrage).

- **Combinaison des mesures de similarité :**

La similarité globale est calculée par la combinaison des similarité syntaxique et sémantique en utilisant la moyenne pondérée.

$$SIM_{final} = \frac{\alpha_1 \times SIM_{Sem} + \alpha_2 \times SIM_{Syn}}{\alpha_1 + \alpha_2} \quad (3.14)$$



Sachant que :

$$SIM_{Sem} = \frac{SimWuPalmer + SimPath + SimLin}{3} \quad (3.15)$$

$$Sim_{Syn} = \frac{SimDice + SimJaccard}{2} \quad (3.16)$$

Et  $\alpha_1$  et  $\alpha_2$  sont des valeurs de pondération.

- **Filtrage**

Le filtrage consiste à déterminer une valeur de seuil bien précise de sorte qu'on garde que les couples d'ontologie qui ont une valeur de similarité supérieure à la valeur seuil et on élimine les couples qui ont une valeur inférieure.

Nous allons maintenant résumer tout ce qui a été dit dans un pseudo-algorithme :

**Algorithme 2 de calcul de similarité et compatibilité entre les ontologies :**

**Entrée : Liste des inputs et outputs.**

**Sortie : Valeur de compatibilité des ontologies.**

**Début**

1/- /\* calcul de similarité entre les différents inputs et outputs de toutes les ontologies\*/

**Pour (toutes les ontologies existante)**

2/- /\*selection des inputs d'une ontologie dans une liste C à partir de la base de données\*/

**C=SelectInputByIdO (IdO) ;**

3/-/\*selection des outputs d'une ontologie dans une liste D à partir de la base de données \*/

**D= SelectOutputByIdO (IdO) ;**

4/- /\*parcourir la liste des inputs\*/

**Pour i ← 1 a C.length () faire**

5/- /\*parcourir la liste des outputs\*/

**Pour j ← 1 a D.length () faire**

6/- /\*calculer la similarité entre chaque input et output\*/

**Calculer\_similarité (input,output) ;**

**finPour**

**finPour**

7/- /\*enregistrer les résultats dans la base de données \*/

**addSim () ;**

**finPour**

**finPour**

**fin Pour**

8/- /\*comparer la valeur de similarité avec un seuil \*/

**If valSim > seuil**

9/- /\*enregistrer les résultats de compatibilité des services dans une table dans la base de données\*/

**AddResult() ; FIN**

Algorithme III. 2: Algorithme calcul de similarité et compatibilité entre les ontologies.

### **III.5.2 Phase de portabilité des données :**

Après la fin de la phase de mapping, nous avons réalisé la portabilité des données entre les services Cloud compatibles en utilisant les Services Web.

Chaque service Cloud qui possède des outputs compatibles avec les inputs d'un autre service Cloud va proposer un service Web. Ce service Web sera invoqué pour réaliser la portabilité des données.

Dans notre travail, on a opté pour l'utilisation des services Web se basant sur le protocole SOAP et utilisant UDDI et WSDL [81][82], leur atout principale est de se reposer sur le langage XML(eXtensible Markup Language) qui est un standard (voir annexe B : service Web).

En plus, les services Web se basent aussi sur l'utilisation des normes actuelles d'Internet comme le protocole HTTP<sup>22</sup>.

### **III.6 Conclusion :**

Dans ce chapitre, nous avons présenté la conception de notre solution proposée au problème de portabilité dans le Cloud public pour les services software as a service.

Nous avons commencé par présenter une analyse des besoins du système puis un schéma global qui résume notre approche, ensuite nous avons détaillé les deux phases avec les différentes étapes suivies pour la réalisation de notre solution.

Dans le chapitre suivant, nous allons montrer en détail comment nous avons implémenté notre solution en donnant les outils utilisés et le résultat final.

---

<sup>22</sup> HTTP: Hypertext Transfer Protocol.

## **Chapitre IV : Implémentation de la solution**

## **IV.1 Introduction**

Dans le chapitre précédent, nous avons présenté la conception de notre solution. Dans ce chapitre nous allons passer à son implémentation.

Ce chapitre va comporter une partie pour spécifier les ressources matérielles et logicielles utilisées ainsi que des imprimes écrans qui montrent chaque interface de notre solution suivant leurs cas d'utilisation.

## **IV.2 Ressources matérielles :**

Afin de réaliser notre application, nous avons utilisé les ressources matérielles suivantes :

- Ordinateur portable HP 6550b.
- Processeur Intel(R) Core (TM) i5 CPU M 480 @2.67GHz 2.67GHz.
- Mémoire RAM 4.00 Go (3.80 Go utilisable).
- Système d'exploitation 64 bits, processeur x64.
- Windows 7 Professionnel.

## **IV.3 Outils et environnement de développement**

### **Eclipse :**

Eclipse IDE est un environnement de développement intégré écrit en java publié par IBM, gratuit et disponible pour la plupart des systèmes d'exploitation [83].

La spécificité d'Eclipse IDE vient du fait que son architecture est entièrement développée autour du concept de plug-ins : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-ins [84].

## **JDK :**

JDK<sup>23</sup> ou Java Development Kit est un environnement de développement qui permet aux utilisateurs de créer des applications, des applets et des composants en java. JDK contient des outils qui peuvent être utilisés non seulement pour le développement mais aussi pour le test des programmes qui fonctionnent sur la plateforme Java [85].

Le kit de développement Java contient le JRE<sup>24</sup> (Java Runtime Environment) qui est la plate-forme sur laquelle les programmes Java s'exécutent [86] un compilateur Java qui transforme le fichier source java en un fichier bytecode [87], ainsi que de nombreuses API JAVA.

## **Java EE :**

Java EE<sup>25</sup> (Java Platform, Entreprise Edition) est une plateforme qui permet la création d'applications Web construite sur le langage Java et la plateforme Java SE<sup>26</sup> (Java Standard Edition). Java EE dispose d'un ensemble d'APIs performantes qui servent à faciliter le développement d'une application Web tout en réduisant le temps de développement et sa complexité. Cette plateforme a été développée au sein du programme JCP<sup>27</sup> (Java Community Process ) selon des standards de stabilité et de compatibilité [88].

Java EE est un standard utilisé par de nombreuses sociétés à travers le monde professionnel et celui de l'Open Source.

---

<sup>23</sup> JDK : Java Development Kit.

<sup>24</sup> JRE : Java Runtime Environment.

<sup>25</sup> Java EE : Java Entreprise Edition.

<sup>26</sup>Java SE : Java Standard Edition.

<sup>27</sup> JCP : Java Community Process.

**Tomcat :**

Tomcat est un logiciel open source qui représente une implémentation des technologies Java Servlet, JSP<sup>28</sup> (Java Server Pages), Java Expression Language ainsi que Java WebSocket [89]. C'était un projet de Sun Microsystems puis devenu celui d'Apache Foundation.

Tomcat est un serveur Web et un conteneur de servlet, utilisé pour le déploiement d'applications développées selon le modèle MVC<sup>29</sup> (Model-View-Controller) et se basant sur les technologies de servlets et de JSP(Java Server Pages) [90].

Tomcat est caractérisé par sa stabilité et sa capacité à travailler avec d'autres serveurs Web [91].

**Protégé :**

Protégé est un éditeur d'ontologies libre et gratuit développé par l'université de Stanford. Il est riche en fonctionnalités permettant la création, l'édition et la visualisation d'ontologies dans les différents langages d'ontologies tel que : RDF-Schéma, OWL, ...etc. Cela est rendu possible grâce à l'utilisation de plugins qui sont disponibles en téléchargement pour la plupart de ces langages.

L'outil Protégé possède une interface utilisateur graphique lui permettant de manipuler aisément tous les éléments d'une ontologie : classe, propriété, instance...etc. Il peut être utilisé dans n'importe quel domaine où les concepts peuvent être modélisés en une hiérarchie des classes.

Protégé bénéficie d'une large communauté et est disponible sous la forme d'une application de bureau ainsi que d'une version accessible en ligne appelée WebProtégé [92].

---

<sup>28</sup> JSP : Java Server Pages.

<sup>29</sup> MVC : Model-View-Controller.

## **WampServer :**

WampServer est une plateforme de développement Web sous Windows de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP.

WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases de données [93].

## **SQL :**

SQL<sup>30</sup> (Structured Query Language) est un langage d'interrogation commun et standardisé pour l'accès et la manipulation de bases de données relationnelles. Les instructions SQL sont des instructions à la base de données et permettent aux utilisateurs d'effectuer diverses tâches allant de l'insertion, la mise à jour et la suppression des tables à la récupération de données spécifiques et au contrôle de l'accès à la base de données [94] .

Les instructions SQL peuvent être utilisées directement par les systèmes de gestion de base de données, mais elles peuvent également être intégrées dans du code écrit dans des langages comme Java.

## **API JDBC :**

L'API JDBC<sup>31</sup> ou Java Database Connectivity est une bibliothèque java qui permet l'accès aux données stockés sur une multitude de supports allant des fichiers Excel aux serveurs de bases de données relationnelles [95].

---

<sup>30</sup> SQL : Structured Query Language.

<sup>31</sup> JDBC : Java Database Connectivity.



## **Wordnet :**

WordNet est une ontologie pour la langue anglaise développée depuis 1985 par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. WordNet est une des bases de données lexicales en ligne contenant des noms, verbes et adjectifs. Ces entités sont organisées en ensemble de synonymes (appelés synsets), reliées par différentes relations sémantiques. Chaque synset représente un sens, un concept de la langue anglaise. Chacun d'eux contient tous les mots synonymes pouvant exprimer le sens auquel il fait référence. Les liens sémantiques ne relient alors pas les mots entre eux mais les synsets auxquels les mots sont affectés [76][77].

La première version de Wordnet diffusée remonte à juin 1991. Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise [96].

Des versions de WordNet pour d'autres langues existent, mais la version anglaise est cependant la plus complète à ce jour.

Afin d'implémenter notre travail, nous avons utilisé WordNet de version 2.1. Le choix s'est porté sur cette base de données pour diverses raisons : c'est la base de données la plus riche et la plus générale qui contient tous les domaines et en plus elle utilise langue anglaise qui est la langue la plus utilisée dans le monde.

Nous avons utilisés les API du WordNet : ws4j et jaws .

## **JAWS API**

JAWS<sup>32</sup> ou bien Java API for Wordnet Searching est une API qui permet aux utilisateurs d'ajouter à leur applications la possibilité de recherche et de récupération de données depuis la base de données Wordnet [97].

---

<sup>32</sup> JAWS: Java API for Wordnet Searching.

Cette API est caractérisée par sa simplicité et sa rapidité.

JAWS est compatible avec les versions 2.1 et 3.0 de la base de donnée Wordnet et elle est fonctionnelle sur JAVA 1.4 et les versions ultérieurs [97].

#### **API WS4J :**

WS4J<sup>33</sup> ou bien Wordnet Similarity for Java est une API java qui prend en charge des algorithmes de calcul de similarité sémantique [97]. Cette API est conçue pour être utilisée avec les instances de la base de données lexicales WordNet.

#### **OWL API :**

OWL API est une API Java qui permet la création, l'analyse, la manipulation et la sérialisation des ontologies OWL. Elle est publiée sous une double licence open source : LGPL et Apache. Elle est utilisée depuis 2003[98].

#### **API Jena :**

Jena est une API Java open source développée dans les laboratoires HP en année 2000 avant d'être adoptée par la fondation Apache en 2010. Elle entre en jeu dans la création d'applications du Web sémantique [99].

Cette API fournit des méthodes pour créer, interroger et manipuler des données en RDF, RDFS et OWL selon les recommandations du W3C. Jena permet la manipulation des ontologies en OWL et RDFS et elle comporte également un moteur d'inférence basé sur les règles pour effectuer des opérations de raisonnement sur ces ontologies [99].

---

<sup>33</sup> WS4J : Wordnet Similarity for Java.

#### **IV.4 Présentation de notre système :**

Notre travail se divise en deux phases :

##### **IV.4.1 Phase 1 : phase de mapping :**

La première phase est une application Web Java EE, qui permet de faire le mapping entre les services Cloud. Dans notre solution, nous avons utilisé des services Cloud de deux domaines différents qui sont le domaine médical et le domaine business.

Dans chaque domaine, nous avons plusieurs services Cloud qui sont les suivants :

- Domaine Business: service CRM, service ERP et service stock.
- Domaine médicale : service docteur, service laboratoire, service gynécologie et service pédiatrie.

Pour déterminer la compatibilité entre les services Cloud, nous avons utilisé les outils de mapping tels que les ontologies.

Chaque service Cloud correspond à une ontologie, chaque ontologie représente les inputs et outputs d'un service Cloud, créé en utilisant l'outil Protégé.

La figure *IV.1* représente un exemple d'ontologie pour le service Cloud laboratoire du domaine médical.

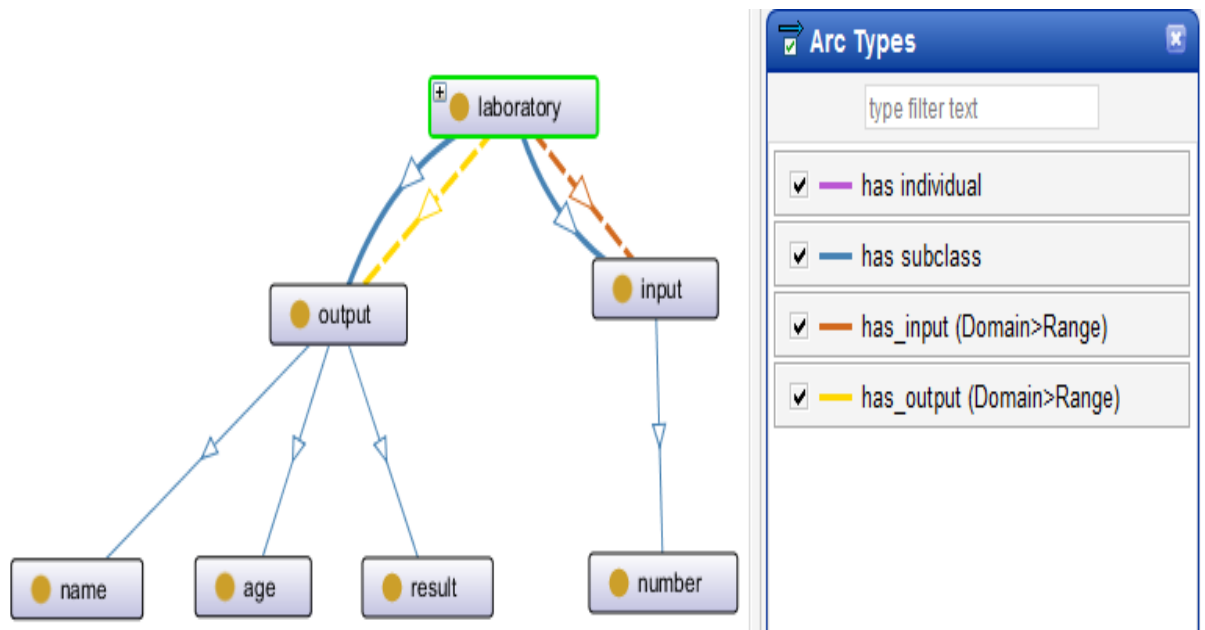


Figure IV. 1:Exemple d'ontologie du service Cloud laboratoire.

Pour mesurer la compatibilité entre les services représentés par les ontologies, nous avons utilisé les mesures de similarité syntaxique et sémantique.

Pour les mesures syntaxiques, on a opté pour l'utilisation de la mesure de Dice et de Jaquard.

Pour les mesures sémantiques, on a utilisé la mesure de Lin, WuPalmer et Path qui utilisent la base de données Wordnet.

Notre application est destinée aux fournisseurs de services pour consulter les services compatibles avec leurs services proposés, mais aussi utilisée par l'administrateur qui déclenche le processus de mapping, en calculant les similarités entre les inputs et outputs des services Cloud.

La figure IV.2 représente l'interface principale de notre application.

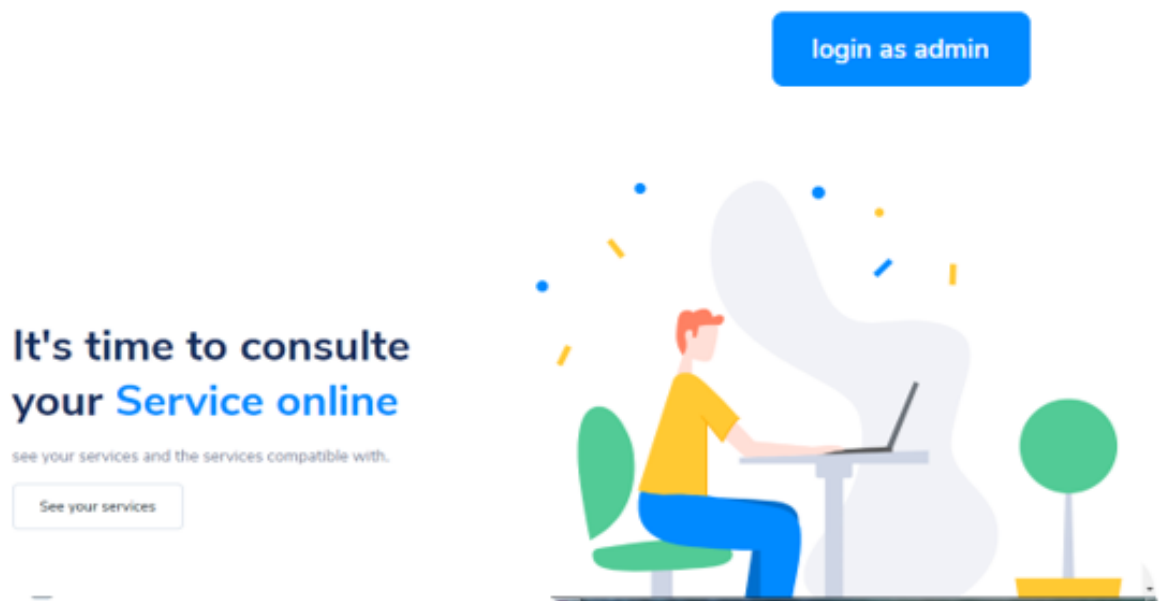


Figure IV. 2: Interface principale de notre application.

La figure IV.3 représente l'interface login de l'administrateur pour accéder à l'interface représenté dans la figure IV.4.

The image shows a login form titled "Login" in a bold, dark blue font. Below the title, the text "Sign in to your account to continue." is displayed in a smaller, grey font. The form contains two input fields: "Username" and "Password". Each field has a light blue placeholder icon on the left and a light grey border. Below the input fields is a prominent blue button with the text "Sign In" in white.

Figure IV. 3:Interface login de l'administrateur.

La figure IV.4 représente l'interface de l'administrateur après login.

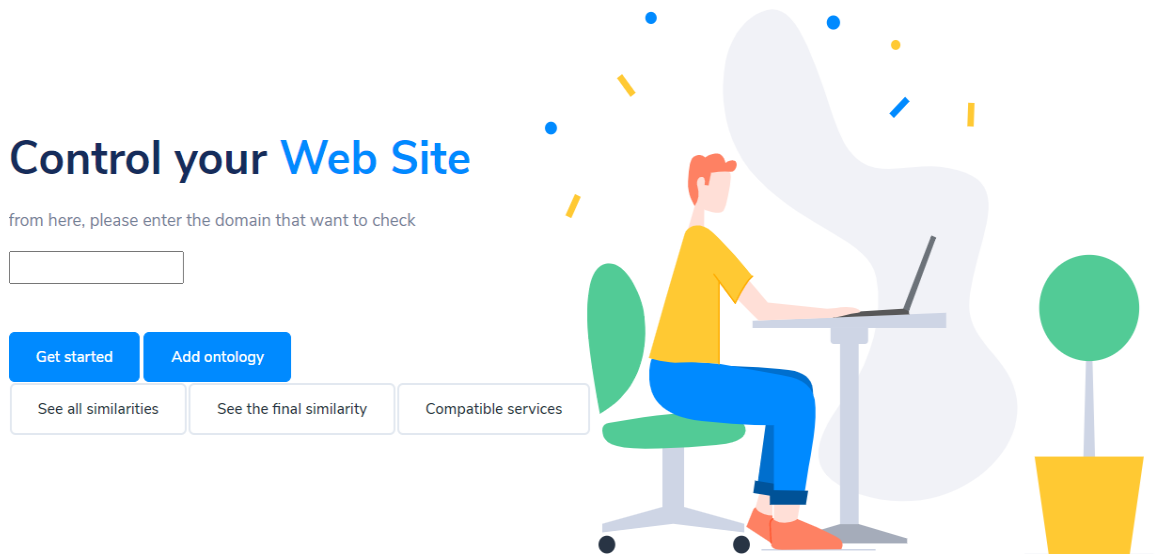


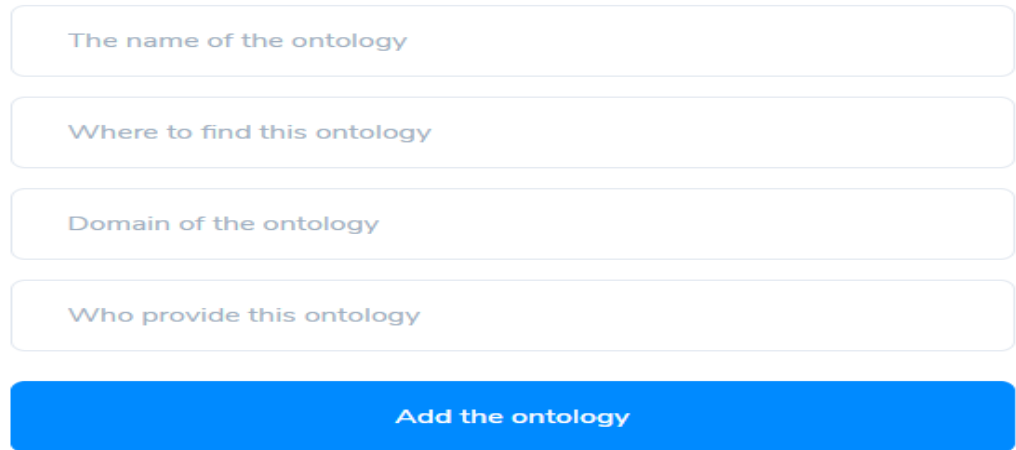
Figure IV. 4:Interface après le login de l'administrateur.

Dans l'interface précédente indiqué par la figure IV.4, l'administrateur peut ajouter des ontologies, calculer les similarités entre les différentes ontologies et peut voir le résultat de calcul et les compatibilités entre les services appartenant à un domaine.

Pour ajouter des ontologies dans le système, la figure suivante IV.5 sera vu par l'administrateur.

## Add ontology

please enter the required information for adding the ontology



The screenshot shows a web interface for adding an ontology. It consists of four stacked input fields with light blue borders and rounded corners. The first field is labeled 'The name of the ontology', the second 'Where to find this ontology', the third 'Domain of the ontology', and the fourth 'Who provide this ontology'. Below these fields is a prominent blue button with white text that says 'Add the ontology'.

Figure IV. 5:Interface pour ajouter une ontologie.

Pour le calcul de similarité, on a opté pour l'utilisation d'une formule qui combine les mesures de similarité syntaxique et sémantique indiqué par la formule (3.13).

Nous avons aussi choisi d'utiliser des poids ( $\alpha_1, \alpha_2$ ) afin d'améliorer la précision de notre système en avantageons les mesures de similarité qui se base sur la sémantique avec les valeurs suivantes :

$$\alpha_1 = 0.8 \text{ et } \alpha_2 = 0.2$$

Les figures IV.6 et IV.7 présentent les valeurs de similarité des différentes mesures pour les domaines Business et médicale.

Nom Service input	Nom Service output	sim LIN	sim PATH	sim PALMER	sim DICE	sim JAQUARD
ERP	CRM	0.11669925914446895	0.16666666666666666	0.4	0.4444444444444444	0.2857142857142857
stock	CRM	0.1605336084467319	0.24999999999999997	0.48888888888888893	0.06666666666666667	0.037037037037037035
CRM	ERP	0.21731511326570185	0.2643939393939394	0.46948529411764706	0.14195804195804196	0.10625
stock	ERP	0.13628299723694748	0.22777777777777777	0.44027777777777777	0.03333333333333333	0.018518518518518517
CRM	stock	0.0	0.0	0.0	0.0	0.0
ERP	stock	0.0	0.0	0.0	0.0	0.0

Figure IV. 6: Valeurs de similarité des différentes mesures pour le domaine Business.

Nom Service input	Nom Service output	sim LIN	sim PATH	sim PALMER	sim DICE	sim JAQUARD
gynecology	pediatric	0.17519188801386296	0.2537545787545788	0.42314118629908104	0.22104978354978355	0.16527777777777777
doctor	pediatric	0.0	0.0	0.0	0.08333333333333333	0.05
laboratory	pediatric	0.10921186561959956	0.16057692307692306	0.3338540544422897	0.13446969696969697	0.09444444444444444
pediatric	gynecology	0.0	0.0	0.0	0.0	0.0
doctor	gynecology	0.0	0.0	0.0	0.33333333333333333	0.2
laboratory	gynecology	0.0	0.0	0.0	0.06060606060606061	0.03333333333333333
pediatric	doctor	0.01705991535071313	0.07133838383838384	0.24523809523809526	0.04545454545454546	0.025
gynecology	doctor	0.016807854372249038	0.11342384467384466	0.32371974430797956	0.05681818181818182	0.036111111111111115
laboratory	doctor	0.2681413558280344	0.31316137566137564	0.4337606837606838	0.20265151515151514	0.15952380952380954
pediatric	laboratory	0.0	0.07692307692307693	0.25	0.0	0.0
gynecology	laboratory	0.03460400365515438	0.11363636363636365	0.39868421052631575	0.05128205128205129	0.027777777777777776
doctor	laboratory	0.0	0.0	0.0	0.0	0.0

Figure IV. 7: Valeurs de similarité des différentes mesures pour le domaine médicale.

Les figure IV.8 et IV.9 représentent la valeur de similarité finale entre les services du domaine business et médicale.



Nom Service input	Nom Service output	Similarity	Service domaine
ERP	CRM	0.6927244866806546	business
stock	CRM	0.7402787386092374	business
CRM	ERP	0.810597085813439	business
stock	ERP	0.6538412126043728	business
CRM	stock	0.0	business
ERP	stock	0.0	business

Figure IV. 8: Valeur de similarité final pour le domaine business.

Nom Service input	Nom Service output	Similarity	Service domaine
gynecology	pediatric	0.7589356347195305	medical
doctor	pediatric	0.02666666666666667	medical
laboratory	pediatric	0.5286971027938782	medical
pediatric	gynecology	0.0	medical
doctor	gynecology	0.10666666666666667	medical
laboratory	gynecology	0.018787878787878787	medical
pediatric	doctor	0.2810000246326628	medical
gynecology	doctor	0.3817470132691172	medical
laboratory	doctor	0.8844857971351399	medical
pediatric	laboratory	0.26153846153846155	medical
gynecology	laboratory	0.45335162806623286	medical
doctor	laboratory	0.0	medical

Figure IV. 9: Valeur de similarité final pour le domaine médicale.

Nous avons aussi défini un seuil pour le filtrage des résultats pour chaque domaine :

- Le domaine médical c'est 0.6.
- Le domaine business c'est 0.8.

Les figures *IV.10* et *IV.11* représentent les services compatibles avec leur mesure pour le domaine business et médicale.

Service input	Service output	Nom Service input	Nom Service output	Similarity	Service domaine
5	6	CRM	ERP	0.810597085813439	business

Figure IV. 10: Services compatible avec leur mesure pour le domaine business.

Service input	Service output	Nom Service input	Nom Service output	Similarity	Service domaine
2	1	gynecology	pediatric	0.7589356347195305	medical
4	3	laboratory	doctor	0.8844857971351399	medical

Figure IV. 11: Services compatible avec leur mesure pour le domaine médicale.

Concernant les fournisseurs de services, cette application va leur permettre de voir les services compatibles avec leurs services Cloud proposés.

La figure *IV.12* représente l'interface login des fournisseurs de services Cloud.

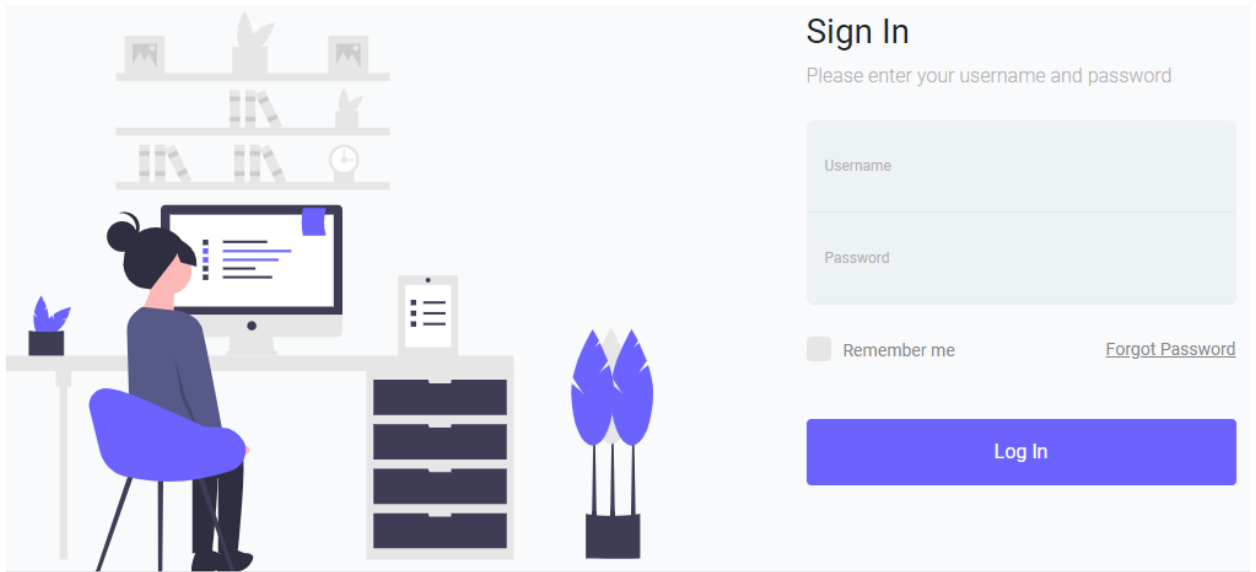


Figure IV. 12: Interface login des fournisseurs de services Cloud.

Après la connexion d'un fournisseur, une interface de ces services proposé est affichée avec la possibilité de voir leur compatibilité (voir la figure IV.13 et IV.14).

## Your services

Service name	Service domain	see the services compatible with
pediatric	medical	<a href="#">compatible with</a>
laboratory	medical	<a href="#">compatible with</a>
CRM	business	<a href="#">compatible with</a>
stock	business	<a href="#">compatible with</a>

Figure IV. 13:Interface d'un fournisseur de services après identification.

Nom Service input	Nom Service output	Similarity	Service domaine
laboratory	doctor	0.8844857971351399	medical

Figure IV. 14: Service compatible avec le service laboratoire.

#### **IV.4.2 Phase 2 : Portabilité des données entre les services Cloud.**

Après le mapping des services Cloud et l'identification de leur compatibilité, la portabilité des données serait possible entre les services Cloud compatibles en utilisant les services Web.

Une portabilité des données entre le service laboratoire et docteur est possible comme l'indique la figure *IV.14*.

Le service laboratoire permet d'enregistrer les informations des malades et le résultat de leur test PCR s'il est positif ou négatif. Le services doctor a besoin de ce résultat pour la consultation du malade. Donc une portabilité des données entre ces services est nécessaire.

Pour réaliser la portabilité des données entre le service laboratoire et le service doctor, nous avons utilisé les services Web.

Le service laboratoire va proposer un service Web qui serait invoqué par le service doctor pour récupérer le résultat du test d'un malade, et cela est possible en connaissant le WSDL du service Web.

La figue *IV.15* représente le WSDL du service Web proposer par l'application laboratoire.

```

-<definitions targetNamespace="http://webservice/" name="prelevement">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://webservice/" schemaLocation="http://localhost:8585/?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="resultat">
    <part name="parameters" element="tns:resultat"/>
  </message>
  <message name="resultatResponse">
    <part name="parameters" element="tns:resultatResponse"/>
  </message>
  <portType name="prelevementSante">
    <operation name="resultat">
      <input wsam:Action="http://webservice/prelevementSante/resultatRequest" message="tns:resultat"/>
      <output wsam:Action="http://webservice/prelevementSante/resultatResponse" message="tns:resultatResponse"/>
    </operation>
  </portType>
  <binding name="prelevementSantePortBinding" type="tns:prelevementSante">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="resultat">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="prelevement">
    <port name="prelevementSantePort" binding="tns:prelevementSantePortBinding">
      <soap:address location="http://localhost:8585"/>
    </port>
  </service>
</definitions>

```

Figure IV. 15:WSDL du service Web de l'application laboratoire.

Dans les figures suivantes, nous allons présenter les interfaces du services doctor qui va invoquer le service Web proposé par laboratoire pour récupérer les résultats de test de leurs patients.

La figure *IV.16* représente l'interface qui va invoquer le service Web pour la récupération des résultats en renseignant le Numéro prélèvement du patient.

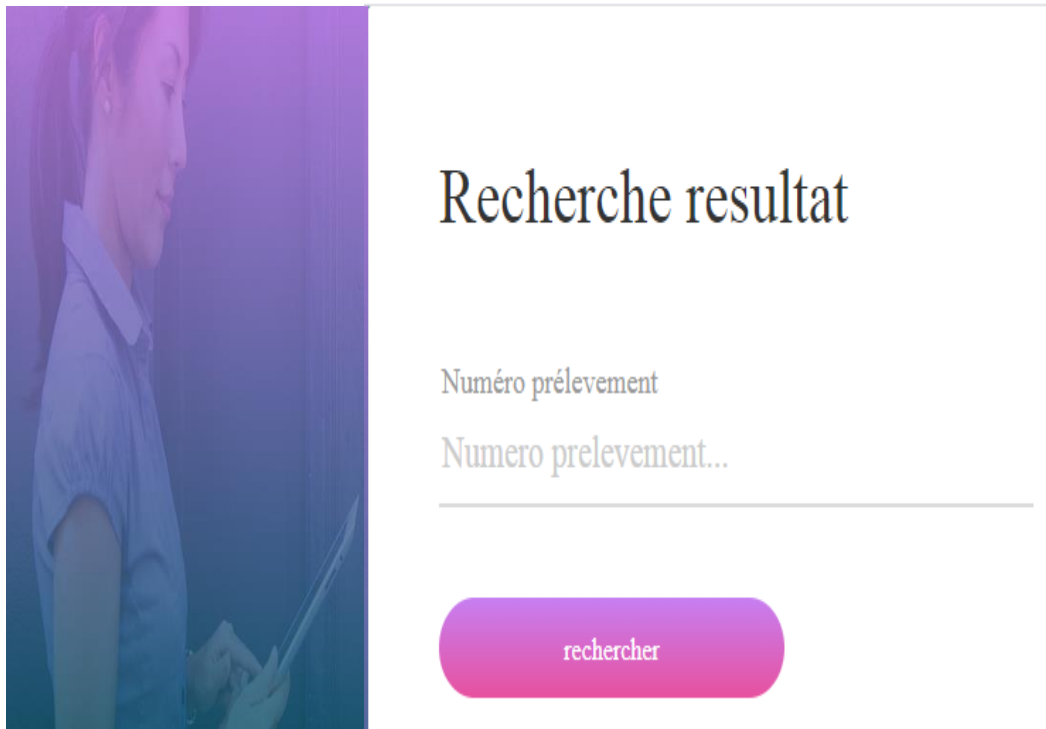


Figure IV. 16: Interface du service doctor pour la récupération du résultat de test.

Selon le résultat du test du patient, une des interfaces indiquées par les figures *IV.17* et *IV.18* sera affichés.

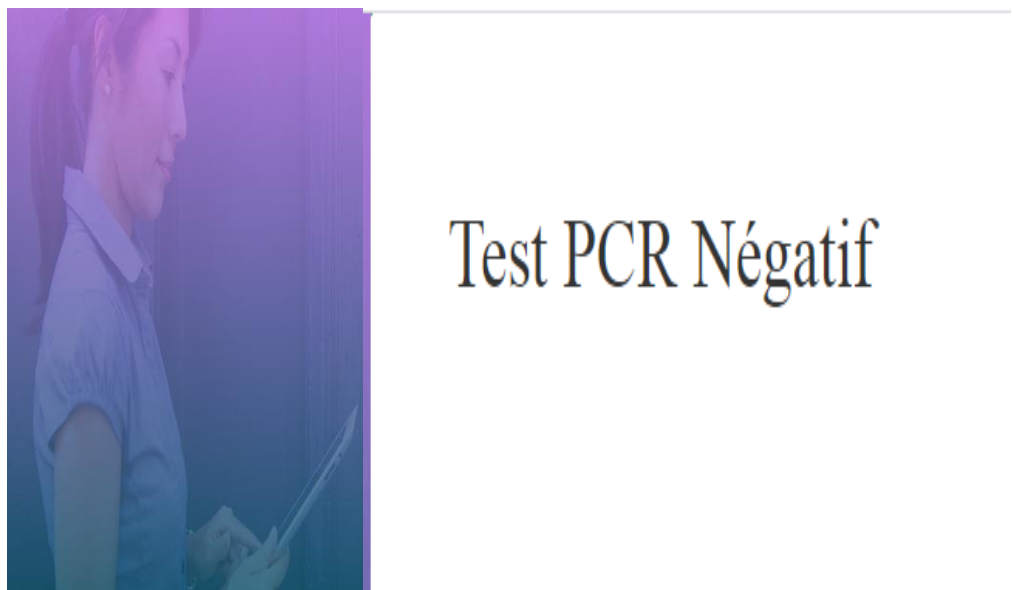


Figure IV. 17:Interface correspondante si le test est négatif.

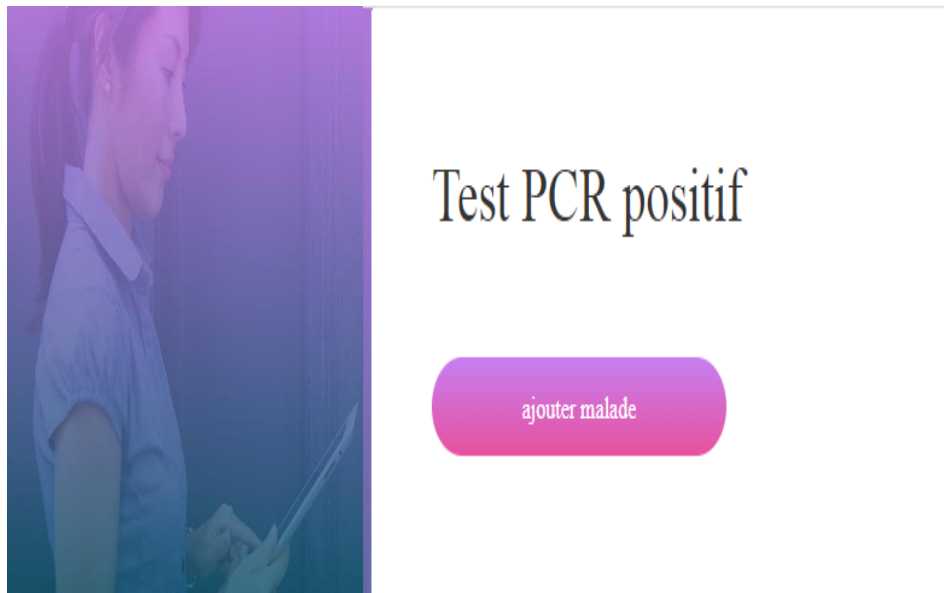


Figure IV. 18: Interface correspondante si le test est positif.

Après que le médecin consulte le résultat du test ; si le test est positif, il peut enregistrer les informations du patient avec le traitement prescrit.

#### **IV.5 Evaluation de la solution par rapport à l'état de l'art :**

Pour évaluer notre solution, nous avons décidé de la comparer avec les solutions traitant le problème de portabilité que nous avons étudié dans le chapitre III selon les critères d'évaluation utilisés dans l'étude comparative. Ceci est la seule façon qui nous permet de montrer notre apport car il n'existe pas de benchmarks relatifs à notre problématique.

Cette évaluation est présentée dans le tableau ci-dessous :

	<b>Solution à base de Cloud</b>	<b>Modèle de service utilisé</b>	<b>Modèle de déploiement utilisé</b>	<b>Nombre de fournisseurs Cloud</b>	<b>Aspect mapping</b>	<b>Format d'échange</b>
Notre solution	oui	SaaS	public	plusieurs	traitée	Xml (service Web)

Tableau IV. 1: Evaluation de notre solution.

#### **IV.6 Conclusion :**

Dans ce chapitre, nous avons présenté la mise en œuvre de notre solution qui aborde le problème de portabilité des données entre les services du Cloud public de type SaaS.

Au début, nous avons présenté les ressources matérielles sur lesquelles notre solution a été réalisée. Puis, nous avons présenté les outils utilisés et l'environnement de développement. Ensuite, nous avons présenté les interfaces de notre application et de leurs fonctionnalités.



## **Conclusion générale et perspectives :**

Le but principal de ce travail était de réaliser une solution pour la portabilité des données entre les services de Cloud Computing de type SaaS.

Nous avons commencé par étudier les concepts de base de Cloud Computing et de portabilité. Ensuite, nous nous sommes lancées dans l'étude des travaux existants ayant proposé des solutions sur notre problématique en les analysant et en les comparant.

Notre solution consiste à faire le mapping des inputs et outputs des différents services Cloud pour définir leur compatibilité en utilisant les outils de mapping tel que les ontologies et les mesures de similarité, ensuite faire la portabilité des données entre les services compatibles.

Notre solution prend la forme d'une application Java EE accessible, facile à déployer et à utiliser.

Comme un travail de recherche n'est jamais achevé, nous donnons quelques perspectives :

Dans notre approche, nous avons travaillé sur deux domaines pour les services Cloud, ça serait intéressant de travailler avec plus de domaines.

En effet, il nous a été difficile de simuler notre approche sur des Clouds réels étant donné que les services SaaS sont toujours fournis par un tiers. Toutefois, il est important de déployer l'approche proposée sur des Clouds réels afin de mieux concrétiser l'approche proposée.

## **Bibliographie :**

- [1] J. w Rittinghouse and J. f Ransome, *Cloud Computing : Implementation, Management, and Security*, CRC Press. 2016.
- [2] “le Cloud Computing une nouvelle filière fortement structurante,” *Direction régionale des entreprises, de la concurrence, de la consommation du travail et de l’emploi.*, 2012. [www.idf.directe.gouv.fr](http://www.idf.directe.gouv.fr).
- [3] K. Rubenstein, “Cloud Computing in Life Sciences R&D,” Cambridge Healthtech Institute(CHI), 2010.
- [4] L. Saïd, “Gestion De Tolerances Aux Fautes Dans Les Cloud Computing Avec CloudSim,” Thèse de doctorat Université Ahmed Ben Bella d’Oran, 2012.
- [5] S. Rajan and A. Jairath, “Cloud computing: The fifth generation of computing,” in *Proceedings - 2011 International Conference on Communication Systems and Network Technologies, CSNT 2011*, 2011, pp. 665–667, doi: 10.1109/CSNT.2011.143.
- [6] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: State-of-the-art and research challenges,” *J. Internet Serv. Appl.*, vol. 1, no. 1, pp. 7–18, 2010, doi: 10.1007/s13174-010-0007-6.
- [7] S. Chhabra and V. S. Dixit, “Cloud Computing: STATE OF THE ART AND SECURITY ISSUES,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 40, no. 2, 2015, doi: 10.1145/2735399.2735405.
- [8] P. Mell and T. Grance, “The NIST-National Institute of Standards and Technology- Definition of Cloud Computing,” *NIST Spec. Publ. 800-145*, p. 7, 2011.
- [9] M. Creeger, “CTO Roundtable: Cloud Computing,” *Commun. ACM*, vol. 52, no. 8, pp. 50–56, 2009, doi: 10.1145/1536616.1536633.
- [10] “Definition of Cloud Computing - Gartner Information Technology Glossary.”

- <https://www.gartner.com/en/information-technology/glossary/cloud-computing> (accessed Apr. 09, 2021).
- [11] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud Computing and Grid Computing 360-Degree Compared,” *Cloud Comput. Grid Comput. 360-Degree Comp.*, vol. 5, 2009, doi: 10.1109/gce.2008.4738445.
- [12] “Qu’est-ce que le cloud computing ? - Définition du cloud computing - Cisco.” [https://www.cisco.com/c/fr\\_fr/solutions/cloud/what-is-cloud-computing.html](https://www.cisco.com/c/fr_fr/solutions/cloud/what-is-cloud-computing.html) (accessed Apr. 09, 2021).
- [13] “AWS | Qu’est-ce que le cloud computing – Les avantages du cloud.” <https://aws.amazon.com/fr/what-is-cloud-computing/> (accessed May 26, 2021).
- [14] “Qu’est-ce que le cloud computing? | IBM.” <https://www.ibm.com/cloud/learn/cloud-computing> (accessed May 26, 2021).
- [15] “Cloud Computing – Définition, Avantages et Exemples d’utilisation.” <https://www.lebigdata.fr/definition-cloud-computing> (accessed Apr. 09, 2021).
- [16] A. Chauhan, P. Suneja, S. Kumar, R. Sahota, and L. Rachna, “Comparative analysis of different actors in cloud computing reference model,” pp. 297–300, 2014.
- [17] S. M. Habib, S. Hauke, S. Ries, and M. Mühlhäuser, “Trust as a facilitator in cloud computing: A survey,” *J. Cloud Comput. Adv. Syst. Appl.*, vol. 1, no. 1, pp. 1–18, Aug. 2012, doi: 10.1186/2192-113X-1-19.
- [18] V. J. R. Winkler, *La Sécurité dans le Cloud: Techniques pour une informatique en nuage sécurisée*, Pearson Ed. 2013.
- [19] R. Buyya, S. K. Garg, and R. N. Calheiros, “SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions,” in *Proceedings - 2011 International Conference on Cloud and Service Computing, CSC 2011*, 2011, pp. 1–10, doi: 10.1109/CSC.2011.6138522.
- [20] P. Wieder, J. M. Butler, W. Theilmann, and R. Yahyapour, *Service Level*

*Agreements for Cloud Computing*, Springer. 2011.

- [21] L. Ronald, Krutz, and R. dean Vines, “A Comprehensive Guide to Secure Cloud Computing,” *Wiley*, 2010.
- [22] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, “The characteristics of cloud computing,” *Proc. Int. Conf. Parallel Process. Work.*, pp. 275–279, 2010, doi: 10.1109/ICPPW.2010.45.
- [23] Y. Gong, Z. Ying, and M. Lin, “A survey of cloud computing,” *Proc. 2nd Int. Conf. Green Commun. Networks 2012 (GCN 2012)*, vol. 3, pp. 79–84, 2013, doi: 10.1007/978-3-642-35470-0\_10.
- [24] J. Rosenberg and A. Mateos, “The Cloud at Your Service,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2011.
- [25] B. Rochwerger *et al.*, *An Architecture for Federated Cloud Computing*, Wiley. 2011.
- [26] A. Sivakumar, “A Survey on Cloud Computing Models and it’s Applications,” *Int. Res. J. Eng. Technol.*, vol. 6, no. 12, pp. 1922–1928, 2019.
- [27] D. Rani and R. K. Ranjan, “A Comparative Study of SaaS , PaaS and IaaS in Cloud Computing,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 6, pp. 158–161, 2014.
- [28] A. . Khurana, S., Verma, “Comparison of cloud computing service models: SaaS, PaaS, IaaS,” *Int. J. Electron. Commun. Technol.*, no. 7109, pp. 29–32, 2013.
- [29] A. Greenberg, J. Hamilton, D. Maltz, and P. Patel, “The cost of a cloud: research problems in data center networks,” *ACM SIGCOMM - Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, 2008, doi: 10.1145/1496091.1496103.
- [30] K. Kavitha, “Study on Cloud Computing Model and its Benefits, Challenges,” *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. 1, pp. 2423–2431, 2014.
- [31] M. . Kavis, *Architecting the cloud: design decisions for cloud computing*

*service models (SaaS, PaaS, AND IaaS)*, Wiley. 2014.

- [32] M. Williams, *A Quick Start Guide to Cloud Computing: Moving Your Business Into the Cloud*, Kogan Page. 2010.
- [33] F. EuroCloud, *L'évolution maîtrisée vers le IaaS/PaaS*. 2011.
- [34] "Modèles et types de cloud computing | AWS."  
<https://aws.amazon.com/fr/types-of-cloud-computing/> (accessed May 18, 2021).
- [35] R. Buyya, S. Pandey, and C. Vecchiola, "Cloudbus toolkit for market-oriented cloud computing," *Proceeding 1st Int. Conférence Cloud Comput. (CLOUDCOM 09)*, pp. 24–44, 2009, doi: 10.1007/978-3-642-10665-1\_4.
- [36] A. Shawish and M. Salama, "Cloud Computing : paradigms and technologies," *Inter-cooperative Collect. Intell. Tech. Appl. .Springer Berlin Heidelb.*, pp. 39–67, 2014.
- [37] "The realities of XaaS(Everything-as-a-Service) -Bits & Bytes."  
<https://www.bittitan.com/blog/realities-everything-service-xaas/> (accessed Jun. 07, 2021).
- [38] "Blockchain as a Service: Enterprise-Grade BaaS Solutions."  
<https://101blockchains.com/blockchain-as-a-service/amp/> (accessed Jun. 08, 2021).
- [39] E. Alomari, A. Barnawi, and S. Sakr, "CDPort: A Portability Framework for NoSQL Datastores," *Arab. J. Sci. Eng.*, vol. 40, no. 9, pp. 2531–2553, 2015, doi: 10.1007/s13369-015-1703-0.
- [40] M. Armbrust *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4. pp. 50–58, Apr. 01, 2010, doi: 10.1145/1721654.1721672.
- [41] R. Hill, L. Hirsch, P. Lake, and S. Moshiri, *Guide to Cloud Computing*. London: Springer London, 2013.
- [42] S. Benfriha and S. Bellifa, "Étude De La Corrélation Entre La Topologie Et

Les Performances Dans Le Cloud.,” Mémoire de master Université Abou Bekr Belkaid -Tlemcen, 2019.

- [43] M. Thabet, “Une approche à base d’agents et de composants pour l’interopérabilité des Clouds,” Thèse de doctorat Université Constantine2 Abdelhamid Mehri ,Constantine, 2015.
- [44] A. Youssef, “Exploring Cloud Computing Services and Applications,” *J. Emerg. Trends Comput. Informations Sci.*, vol. 3, no. 6, pp. 838–847, 2012.
- [45] R. Buyya, J. Broberg, and A. Goscinski, *Cloud computing: Principles and paradigms*, Wiley. 2011.
- [46] L. F. B. Soares, D. A. B. Fernandes, J. V. Gomes, M. M. Freire, and P. R. M. Inácio, “Cloud security: State of the art,” in *Security, Privacy and Trust in Cloud Systems*, Jun. 2014, pp. 3–44, doi: 10.1007/978-3-642-38586-5\_1.
- [47] H. MATAALLAH, “Vers un nouveau modèle de stockage et d’accès aux données dans les Big Data et les Cloud Computing,” Thèse de doctorat Université ABOU-BEKR BELKAID - Tlemcen, 2018.
- [48] D. C. Chou, “Cloud computing: A value creation model,” *Comput. Stand. Interfaces*, vol. 38, pp. 72–77, 2015, doi: 10.1016/j.csi.2014.10.001.
- [49] “Le marché du cloud atteindra 554 milliards de dollars en 2021 selon IDC.” <https://www.lebigdata.fr/marche-du-cloud-2021-idc> (accessed Feb. 22, 2021).
- [50] T. Dillon, C. Wu, and E. Chang, “Cloud computing: Issues and challenges,” *Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA*, pp. 27–33, 2010, doi: 10.1109/AINA.2010.187.
- [51] “CSA Top Threats Working Group.” <https://cloudsecurityalliance.org/research/working-groups/top-threats/> (accessed Feb. 22, 2021).
- [52] M. Ali, S. U. Khan, and A. V. Vasilakos, “Security in cloud computing: Opportunities and challenges,” *Inf. Sci. (Ny)*, vol. 305, pp. 357–383, 2015, doi: 10.1016/j.ins.2015.01.025.

- [53] S. Dowell, A. Barreto, J. B. Michael, and M. T. Shing, "Cloud to cloud interoperability," in *Proceedings of 2011 6th International Conference on System of Systems Engineering*., 2011, pp. 258–263, doi: 10.1109/SYSOSE.2011.5966607.
- [54] C. Baudoin, E. Dekel, and M. Edwards, "Interoperability and Portability for Cloud Computing :A Guide," *Cloud Stand. Cust.*, pp. 1–20, 2014.
- [55] N. Loutas, E. Kamateri, and K. Tarabanis, "A Semantic Interoperability Framework for Cloud Platform as a Service," *IEEE Third Int. Conf. Cloud Comput. Technol. Sci. CloudCom*, pp. 280–287, 2011, doi: 10.1109/CloudCom.2011.45.
- [56] M. . D. Wallace and B. Tushar, "Interoperability and Portability," *Cloud Security Alliance Group 4*, 2011.  
<https://studylib.net/doc/6968414/interoperability-and-portability> (accessed Jun. 15, 2021).
- [57] M. Kostoska, M. Gusev, and S. Ristov, "An overview of cloud portability," *Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 159, pp. 248–254, 2015, doi: 10.1007/978-3-319-27072-2\_32.
- [58] B. Engels,, "Data portability among online platforms," *Internet Policy Rev.*, vol. 5, no. 2, pp. 1–17, 2016, doi: 10.14763/2016.2.408.
- [59] M. Thabet, M. Boufaïda, and K. Fabrice, "An approach for developing an interoperability mechanism between cloud providers," *Int. J. Space-Based Situated Comput.*, vol. 4, no. 2, pp. 88–99, 2014.
- [60] G. A. Lewis, "Role of Standards in Cloud-Computing Interoperability," *Syst. Sci. (HICSS),2013 46th Hawaii Int. Conf. Syst. Sci.*, no. 10, pp. 1652–1661, 2013, doi: 10.1109/HICSS.2013.470.
- [61] P. Harsh, F. Dudouet, R. G. Cascella, Y. Jegou, and C. Morin, "Using open standards for interoperability issues,solutions, and challenges facing cloud computing," in *Proceedings of the 2012 8th International Conference on Network and Service Management, CNSM*, 2012, pp. 435–440.

- [62] M. Thabet and M. Boufaïda, "An Agent-Based Architecture and a Two-Phase Protocol for The Data Portability in Clouds," *2013 27th Int. Conf. Adv. Inf. Netw. Appl. Work.*, pp. 785–790, 2013, doi: 10.1109/WAINA.2013.129.
- [63] B. Rashidi, M. Sharifi, and T. Jafari, "A Survey on Interoperability in the Cloud Computing Environments," *Int. J. Mod. Educ. Comput. Sci.*, vol. 5, no. 6, pp. 17–23, 2013, doi: 10.5815/ijmecs.2013.06.03.
- [64] D. Petcu and A. . Vasilakos, "Portability in Clouds : Approaches and Research Opportunities.," *Scalable Comput. Pract. Exp.*, vol. 15, no. 3, pp. 251–270, 2014, doi: 10.12694/scpe.v15i3.1019.
- [65] J. K. Wang, J. Ding, and T. Niu, "Interoperability and Standardization of Intercloud Cloud Computing," *arXiv Prepr. arXiv1212.5956*, 2012.
- [66] M. Moravcik, P. Segec, and M. Kontsek, "Overview of Cloud Computing Standards," *ICETA 2018 - 16th IEEE Int. Conf. Emerg. eLearning Technol. Appl. Proc.*, no. 11, pp. 395–402, 2018, doi: 10.1109/ICETA.2018.8572237.
- [67] R. Sandhu and I. Chana, "Cloud computing Standardization Initiatives: State of Play," *Int. J. Cloud Comput. Serv. Sci.*, vol. 2, no. 5, pp. 351–362, 2013, doi: 10.11591/closer.v2i5.4894.
- [68] M. Kostoska, M. Gusev, S. Ristov, and K. Kirovski, "Cloud computing interoperability approaches - Possibilities and challenges," *CEUR Workshop Proc.*, vol. 920, pp. 30–34, 2012.
- [69] "What is cloud migration? An introduction to moving to the cloud." <https://searchcloudcomputing.techtarget.com/definition/cloud-migration> (accessed Jun. 15, 2021).
- [70] Y. K. Hooi, M. F. Hassan, and A. M. Shariff, "A survey on ontology mapping techniques," *Lect. Notes Electr. Eng.*, 2014.
- [71] O.-S. Lupșe, M. M. Vida, and S.-T. Lăcrămioara, "Cloud Computing and Interoperability in Healthcare Information Systems," *INTELLI 2012 TheThe First Int. Conf. Intell. Syst. Appl.*, pp. 81–85, 2012.



- [72] K. Benhssayen and A. Ettalbi, "Semantic Interoperability Framework for IAAS Resources in Multi-Cloud Environment," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 21, no. 2, 2021.
- [73] T. Kwon, J. Lee, and K. Jung, "A Study of Data Interoperability System using DBaaS for Mobility Handicapped," *Int. J. Internet, Broadcast. Commun.*, vol. 11, no. 1, pp. 97–102, 2019.
- [74] N. E. H. Bouzerzour, S. Ghazouani, and Y. Slimani, "Cloud interoperability based on a generic cloud service description: Mapping OWL-S to GCSD," *Proc. Work. Enabling Technol. Infrastruct. Collab. Enterp. WETICE*, pp. 70–75, 2020.
- [75] "OWL - Semantic Web Standards." <https://www.w3.org/OWL/> (accessed Aug. 26, 2021).
- [76] L. Meng, R. Huang, and J. Gu, "A Review of Semantic Similarity Measures in WordNet," *Int. J. Hybrid Inf. Technol.*, vol. 6, no. 1, pp. 1–12, 2013.
- [77] S. Iltache, "Modélisation ontologique pour la recherche d'informations: évaluation de la similarité sémantique de textes et application à la détection de plagiats.," Thèse de doctorat Université de Toulouse, 2018.
- [78] T. Slimani, B. Ben Yaghlane, and K. Mellouli, "Une extension de mesure de similarité entre les concepts d' une ontologie," *4rth Int. Conf. Sci. Electron. Technol. Inf. Telecommun.*, pp. 1–10, 2007.
- [79] A. SIAGH and C. DEROUCHE, "Similarité sémantique entre concepts: Application à la recherche d'images," Thèse de doctorat Université Kasdi Merbah Ouagla, 2016.
- [80] A. Ali, H. Alquhayz, R. Marg, S. Vihar, and N. Delhi, "Semantic Similarity Measures Between Words: a Brief Survey," *Sci. Int. J.*, vol. 30, no. 6, pp. 907–914, 2018.
- [81] B. Amrane, "Méthode de recherche de services Web basée sur l'analyse formelle de concepts," Thèse de doctorat Université Ahmed Ben Bella Oran, 2015.

- [82] H. Kadima and V. Monfort, *Web Service : technique, démarche et outils XML, WSDL, SOAP, UDDI*, Rosetta Ne. 2003.
- [83] “What is Eclipse?” <https://www.educative.io/edpresso/what-is-eclipse> (accessed Aug. 27, 2021).
- [84] “🔍 Eclipse (logiciel) : définition et explications.” <https://www.techno-science.net/definition/517.html> (accessed Aug. 27, 2021).
- [85] “Java SE Development Kit 8 - Downloads.” <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html> (accessed Aug. 27, 2021).
- [86] “Java SE Runtime Environment 8 - Downloads.” <https://www.oracle.com/java/technologies/javase-jre8-downloads.html> (accessed Aug. 27, 2021).
- [87] “javac - Java programming language compiler.” <https://docs.oracle.com/javase/7/docs/technotes/tools/windows/javac.html> (accessed Aug. 27, 2021).
- [88] “Java Platform, Enterprise Edition (Java EE) | Oracle Technology Network | Oracle.” <https://www.oracle.com/java/technologies/java-ee-glance.html> (accessed Aug. 27, 2021).
- [89] “Apache Tomcat® - Welcome!” <http://tomcat.apache.org/> (accessed Aug. 27, 2021).
- [90] S. Agrawal and R. Gupta, “Development and comparison of open source based web gis frameworks on wamp and apache tomcat web servers,” *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, 2014.
- [91] K. Kumar, “Review on Tomcat Security & Role of Java Security Manager(JSM) in Apache Tomcat,” Thèse de Doctorat University Montfort-Angleterre, 2012.
- [92] “protégé.” <https://protege.stanford.edu/> (accessed Aug. 27, 2021).
- [93] “WampServer, la plate-forme de développement Web sous Windows -

- Apache, MySQL, PHP.” <https://www.wampserver.com/> (accessed Aug. 27, 2021).
- [94] “SQL Developer.” <https://www.oracle.com/database/technologies/appdev/sqldeveloper-landing.html> (accessed Aug. 27, 2021).
- [95] “Java JDBC API.” <https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/> (accessed Aug. 27, 2021).
- [96] A. Budanitsky and G. Hirst, “Evaluating wordnet-based measures of lexical semantic relatedness,” *Comput. Linguist.*, vol. 32, no. 1, pp. 13–47, 2006, doi: 10.1162/coli.2006.32.1.13.
- [97] M. A. Finlayson, “Java libraries for accessing the Princeton Wordnet: Comparison and evaluation,” *Proc. 7th Glob. Wordnet Conf.*, pp. 78–85, 2014.
- [98] M. Horridge and S. Bechhofer, “The owl api: A java api for owl ontologies,” *Semant. Web*, vol. 2, pp. 11–21, 2011.
- [99] “Apache Jena - What is Jena?” [https://jena.apache.org/about\\_jena/about.html](https://jena.apache.org/about_jena/about.html) (accessed Aug. 27, 2021).
- [100] T. R. Gruber, “A translation approach to portable ontology specifications,” *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, 1993.
- [101] W. Borst., “Construction of engineering ontologies for knowledge sharing and reuse,” *Cent. Telemat. Inf. Technol. (CTIT)*, 1997.
- [102] N. Noy and D. McGuinness, “Ontology Development 101: A Guide to Creating Your First,” *Knowl. Syst. Lab.*, vol. 32, 2001.
- [103] C. Roche, “Ontology: A Survey,” *IFAC Proc. Vol.*, vol. 36, no. 22, pp. 187–192, 2003, doi: 10.1016/S1474-6670(17)37715-7.
- [104] N. Hernandez, “Ontologie de domaine pour la modélisation du contexte en recherche d’information,” Thèse de doctorat Université Paul Sabatier-France, 2006.

- [105] M. RICHARD, “Apports de la modélisation ontologique pour le partage des connaissances en psychiatrie,” Thèse de doctorat Université Pierre et Marie Curie France, 2017.
- [106] “Extensible Markup Language (XML).” <https://www.w3.org/XML/> (accessed Jul. 15, 2021).
- [107] G. Antoniou and F. Van Hamelen, *A semantic Web primer*, MIT press. 2012.
- [108] “Schema - W3C.” <https://www.w3.org/standards/xml/schema> (accessed Aug. 26, 2021).
- [109] “RDF - Semantic Web Standards.” <https://www.w3.org/RDF/> (accessed Aug. 26, 2021).
- [110] M. LADJENEF, “système à base d’ontologie pour l’aide à la maintenance des machines,” Thèse de doctorat université Mohamed Boudiaf - M’SILA, 2018.
- [111] “RDF Vocabulary Description Language 1.0: RDF Schema.” <https://www.w3.org/TR/2004/REC-rdf-schema-20040210/> (accessed Aug. 26, 2021).
- [112] D. Kalibatiene and O. Vasilecas, “Survey on ontology languages,” *Lect. notes Bus. Inf. Process.*, vol. 90, pp. 124–141, 2011.
- [113] “Qu’est-ce que la SOA (architecture orientée services) ? | IBM.” <https://www.ibm.com/cloud/learn/soa> (accessed Aug. 25, 2021).
- [114] J.-M. Francisco, “Modélisation des architectures logicielles dynamiques : application à la gestion de la qualité de service des applications à base de services web,” Thèse de doctorat- Institut National Polytechnique de Toulouse, 2007.
- [115] “Web Services Architecture.” <https://www.w3.org/TR/ws-arch/> (accessed Aug. 24, 2021).
- [116] “Qu’est-ce qu’un webservice ? -Documentation IBM.” <https://www.ibm.com/docs/en/cics-ts/5.2?topic=services-what-is-web-service> (accessed Aug. 26, 2021).

- [117] S. Chemaâ, “Une approche de composition de services Web à l’aide des Réseaux de Petri orientés objet,” Thèse de doctorat Université Abdelhamid Mehri Constantine 2, 2014.
- [118] F. Hadjila, “Composition et interopération des services web sémantiques,” Thèse de doctorat Université de Tlemcen, 2014.
- [119] “Web Service Definition Language (WSDL).”  
<https://www.w3.org/TR/wsdl.html> (accessed Aug. 26, 2021).
- [120] S. HAMIDA, “Une approche basée agent mobile pour le mservice web sémantique,” Thèse de doctorat Université Mohamed Khider – BISKRA, 2014.
- [121] A. Zulqarnain and A. Samreen, “Dynamic Service Discovery, Composition and Reconfiguration in a Model Mapping Business Process to Web Services,” *J. Comput. Commun.*, pp. 24–39, 2016.

## Annexe A : Notions sur les ontologies

### **Définition :**

Les ontologies sont des ressources conceptuelles présentant des structures formelles issues du domaine de la philosophie, elles ont été ensuite introduites en informatique et leur principal objectif est de modéliser les connaissances d'un domaine donné.

Selon Gruber [100]: « une ontologie est une spécification explicite d'une conceptualisation ».

D'après Borst [101] : « une ontologie est une spécification explicite et formelle d'une conceptualisation partagée ».

Autrement dit une ontologie doit être compréhensible par la machine, le type de concept utilisé et les contraintes sur leur utilisation doivent être explicitement définis et ne doit pas être restreint à certains individus mais acceptée par un groupe.

L'ontologie constitue un modèle de données représentant un ensemble de concepts dans un domaine ainsi que des relations entre ces concepts. La forme la plus courante selon laquelle les ontologies sont présentées consiste en une hiérarchie de concepts, organisés des plus généraux aux plus spécifiques.

Les ontologies servent de vocabulaire commun lorsqu'il est nécessaire de partager des informations dans un domaine mais aussi les ontologies sont conçues pour fournir des ensembles de données cohérents et bien structurés qui alimenteraient des programmes et des application [102].

Les ontologies sont utilisées pour la représentation des connaissances, la recherche, l'extraction, l'intégration et le partage d'information dans des domaines variés tels que le Web sémantique, le traitement du langage naturelle, la recherche

d'information, l'intégration des données, les Web services et interopérabilité entre différents systèmes [103].

### **Composants d'une ontologie :**

Les ontologies rassemblent les connaissances propres à un domaine donné. Pour représenter ces connaissances, une ontologie est constituée d'un ensemble de composants de base qui sont : les concepts, les relations, les fonctions, les axiomes et les instances. Nous détaillons ci-dessous ces différents composants [104]:

#### **1-Les concepts :**

Un concept représente un ensemble d'objets et leurs propriétés communes, ils sont appelés aussi termes ou classes de l'ontologie. Ils constituent les éléments de base au sein d'une ontologie.

Les concepts peuvent être une idée, un principe, une notion profonde, Il peut décrire une fonction, un processus de raisonnement, une stratégie, etc.

Ce concept est défini par ses différentes relations avec les autres concepts de l'ontologie, ses attributs et les contraintes qui lui sont associées.

#### **2-Les relations**

Les relations d'une ontologie désignent différentes interactions et corrélations entre les concepts de l'ontologie.

Ces relations permettent de capturer la structuration ainsi que l'interaction entre les concepts, permettant de représenter une grande partie de la sémantique de l'ontologie.

Ces relations incluent les relations de spécialisation-généralisation (sous classes), relations d'agrégation ou de composition (relations is-a, partie de), relations d'association et d'instanciation.

### **3-Les fonctions**

Constituent des cas particuliers de relations, dans laquelle un élément de la relation, (le nième) est défini en fonction des N-1 éléments précédents.

### **4-Les axiomes**

Les axiomes sont des assertions qui sont toujours vraies. Ils combinent des concepts, des relations pour définir des règles d'inférence et permettent de vérifier la validité des informations spécifiées ou de déduire de nouvelles informations.

### **5-Les instances**

Instance c'est l'élément décrit par le concept. Ils représentent les éléments qui véhiculent les connaissances du domaine considéré. Par exemple les individus Mohamed et Amel sont des occurrences du concept Personne.

Dans ce qui suit, nous allons présenter les langages d'ontologies.

### **Langages d'ontologies**

Une des principales décisions à prendre dans le procédé de développement d'ontologie consiste à choisir le langage dans lequel l'ontologie sera exprimée et utilisée.

Il existe plusieurs langages d'ontologie [105]:



## 1-XML et XML schéma

Le langage de balisage extensible (Extensible Markup Language) ou simplement XML est un langage de structuration et de balisage développé à l'origine par un groupe de travail appelé SGML Editorial Review Board en 1996 comme un standard. Ce langage a été élaboré pour faciliter l'échange, le partage et la publication des données à travers le Web. Il est apprécié pour sa simplicité, flexibilité et son efficacité dans l'échange de données sur le Web.

Il constitue le premier langage de description d'ontologies sur le Web. Il rend possible la description des concepts et leurs hiérarchisations sous forme de taxonomies à l'aide des balises. IL a été conçu dans le but d'en faire un langage facilement traitable par les programmes et également par les humains [106].

XML permet de structurer un document en définissant ses propres balises en fonction des besoins et sans tenir compte ni de la signification de cette structure ni des systèmes informatiques qui vont l'exploiter.

XSD, également connu sous le nom de XML Schema Definition, est un langage basé sur XML et dédié à l'expression des contraintes sur les documents XML définis par le W3C. XML Schéma a été développé comme langage qui offre des grammaires qui permettent de spécifier des restrictions sur la structure des documents.

XSD est utilisé pour mettre des restrictions sur les éléments, l'association de type avec des valeurs et pour le contrôle de la qualité des documents (validation XSD) mais aussi la définition de nouveaux types de données [107] [108].

## 2-RDF et RDFS :

RDF est un modèle de représentation et d'échange de données développé par W3C, C'est un modèle de graphe destiné à décrire de façon formelle les ressources Web et leurs métadonnées grâce à sa syntaxe basée sur XML [109].

L'objectif initial de RDF est une bonne représentation et une meilleure exploitation des métadonnées.

Un document RDF est basé sur le triplet (ressources, propriété, valeur) ou encore (sujet, prédicat, objet) tel que [110] [109] :

- **La ressource** (sujet) une entité d'informations pouvant être référencée par un identificateur. Cet identificateur doit être une URI<sup>34</sup> (Uniform Resource Identifier), un exemple de sujet et une page Web.
- **La propriété** (prédicat) représente une propriété ou une relation spécifique utilisée pour décrire une ressource.
- **La valeur** (objet) représente la valeur d'une propriété attribuée à une ressource particulière, elle peut être des ressources ou des littéraux.

RDF est un langage qui permet de définir une ontologie de manière très simple, son inconvénient est qu'il ne supporte pas la vérification de la cohérence des données (vérification que le champ « date de naissance » est vraiment une date par exemple).

RDFS<sup>35</sup> (RDF Schéma) est une évolution de RDF. C'est un méta modèle recommandé par le W3C permettant la définition de schéma/modèle décrivant l'univers sémantique des déclarations RDF. RDFS est le langage de définition de vocabulaires pour RDF, et fournit ainsi un système de typage pour les déclarations RDF [111].

---

<sup>34</sup> URI : Uniform Resource Identifier.

<sup>35</sup> RDFS : RDF Schéma.

Le langage RDFS offre en plus du contrôle la terminologie et la structure des descriptions RDF, la possibilité de raisonner sur liens de types « est-un » (is a) qui existent entre les concepts et les propriétés.

### **3-OWL :**

OWL Ontology Web Language est une recommandation de W3C depuis février 2004. OWL est un langage basé sur la syntaxe XML, très expressif et avec un vocabulaire très riche. Il permet de définir des ontologies de domaines complexes. Il s'agit d'une extension du langage RDF et RDFS ce qui le rend plus puissant pour exprimer la sémantique.

Le langage OWL a été conçu pour être utilisé par les applications qui traitent le contenu de l'information au lieu de la présenter seulement aux utilisateurs. OWL facilite l'interopérabilité au niveau machine du contenu du Web plus que ce qui est déjà supporté par les XML, RDF et *RDF Schema* en fournissant du vocabulaire supplémentaire avec des sémantiques formelles [75].

OWL se compose de trois sous langages d'expressivité croissante [112] :

**OWL Lite** est le sous langage de OWL le plus simple. Il est destiné aux utilisateurs qui ont besoin d'une hiérarchie de concepts simple. L'avantage de ce langage est d'avoir une complexité formelle faible par rapport aux deux autres sous langages d'OWL. C'est la version la plus facile à comprendre et à mettre en œuvre, mais aussi celle qui présente la plus grande restriction d'expressivité.

**OWL DL**<sup>36</sup> est plus complexe que OWL Lite, permettant une expressivité bien plus importante. OWL DL est fondé sur la logique descriptive (d'où son nom, OWL Description Logics), Malgré sa complexité relative face à OWL Lite, OWL-DL offre

---

<sup>36</sup> OWL DL : OWL Description Logics.

une expressivité maximale en garantissant la complétude des raisonnements (calculabilité des inférences) et leur décidabilité (leur calcul se fait en une durée finie). Cette version permet un support de raisonnement très efficace mais, par contre, la compatibilité totale avec RDF n'est plus supportée.

**OWL Full** : est la version la plus complexe d'OWL, mais également celle qui permet le plus haut niveau d'expressivité. Et offre également une compatibilité syntaxique et sémantique complète avec RDF. OWL Full est destiné aux situations où il est plus important d'avoir un haut niveau de capacité de description, quitte à ne pas pouvoir garantir la complétude et la décidabilité des calculs liés à l'ontologie.

Il existe entre ces trois sous langage une dépendance de nature hiérarchique : toute ontologie OWL Lite valide est également une ontologie OWL DL valide, et toute ontologie OWL DL valide est également une ontologie OWL Full valide (voir figure 1).

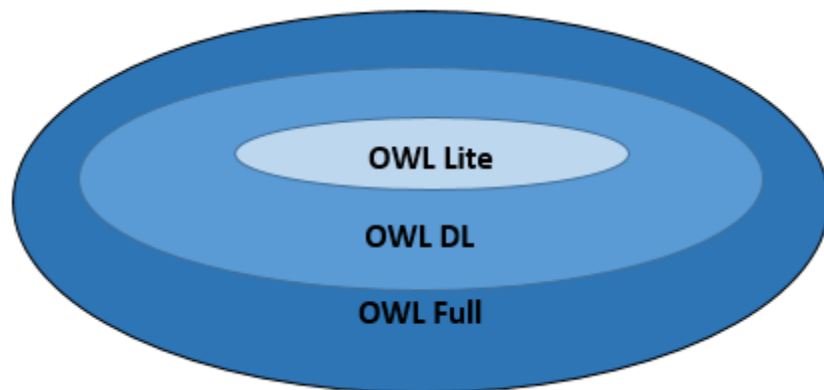


Figure 1:Les couches de l'OWL.

## Annexe B : Services Web

### Services Web et SOA :

Le développement rapide des systèmes d'informations a flamboyé ces dernières décennies à travers la diffusion de l'accès à internet. De nouveaux modèles d'interaction ont été également marqué citons-en : la SOA<sup>37</sup> (service oriented architecture).

### Définition de la SOA :

Il existe plusieurs définitions de la SOA dans la littérature, on cite :

Selon IBM [113]: « *La SOA, ou architecture orientée services, définit un moyen de rendre les composants logiciels réutilisables et interopérables via des interfaces de service. Les services utilisent des normes d'interface communes et un modèle architectural afin qu'ils puissent être rapidement intégrés dans de nouvelles applications. Cela supprime les tâches du développeur d'applications qui a précédemment redéveloppé ou dupliqué des fonctionnalités existantes ou qui devait savoir comment se connecter ou fournir une interopérabilité avec des fonctions existantes* ».

Selon[114] : « *SOA est une approche architecturale permettant la création des systèmes basées sur une collection de services développés dans différents langages de programmation, hébergés sur différentes plates-formes avec divers modèles de sécurité et processus métier. Chaque service représente une unité autonome de traitements et de gestion de données, communiquant avec son environnement à l'aide de messages. Les échanges de messages sont organisés sous forme de contrats d'échange* ».

---

<sup>37</sup> SOA : Service Oriented Architecture.

Autrement dit l'architecture orientée service constitue un style d'architecture basée sur le principe de séparation de l'activité métier en une série de services qui représentent des unités autonomes, identifiables, documentés et accessibles via le réseau destiné à résoudre les problèmes d'hétérogénéité et interopérabilité des logiciels.

Le paradigme SOA possède plusieurs implémentations parmi elle les services Web, qui représentent la concrétisation la plus répandue.

### **Définition services Web :**

La technologie des services Web (en anglais Web services) est une concrétisation de l'architecture orientée service SOA sur le Web. Elle a été initiée par IBM et Microsoft et ensuite normalisée sous l'égide de consortium W3C.

Plusieurs définitions des services Web ont été proposées dans la littérature :

Selon W3C [115]: un service Web est « un système logiciel conçu pour supporter l'interaction interopérable de machine à machine sur un réseau. Il possède une interface décrite en WSDL<sup>38</sup> qui est un format exploitable par la machine. D'autres systèmes interagissent avec les services Web d'une façon prescrite par sa description en utilisant des message SOAP<sup>39</sup>, typiquement en utilisant HTTP<sup>40</sup> avec une sérialisation XML en même temps que d'autres normes du Web ».

Cette définition cite les caractéristiques les plus importantes d'un service Web à savoir l'identifiant (URI), l'interface (WSDL), et l'indépendance par rapport aux plateformes.

---

<sup>38</sup> WSDL : Web Services Description langage.

<sup>39</sup> SOAP: Simple Object Access Protocol.

<sup>40</sup> HTTP: Hypertext Transfer Protocol.

Selon IBM [116]: «*A Web service is a generic term for an interoperable machine-to-machine software function that is hosted at a network addressable location* ».

Selon [82], les services Web sont des applications auto descriptives, modulaires et faiblement couplées qui fournissent un modèle simple de programmation et de déploiement d'applications, basé sur des normes et s'exécutant à travers l'infrastructure Web. Les services Web réalisent des fonctions allant des simples requêtes au processus métier sophistiqués.

Ces définitions affirment que les services Web sont accessibles à travers le Web, en utilisant des protocoles et des formats standards.

### **Les acteurs des services Web :**

Les services Web repose sur trois acteurs [117]:

- **Le fournisseur de service (service provider)** : Correspond au propriétaire de service.
- **Le client (service requestor)** : Correspond au demandeur (consommateur) de service, d'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer le service.
- **L'annuaire de service (Service Registry)** : Entité logicielle intermédiaire entre les clients et le fournisseur de services. Il correspond à un registre de description des services offrant des facilités de publication de services à l'intention des fournisseurs ainsi que des facilités de recherche de services à l'intention des clients.

## **Langages et protocoles utilisés par les services Web**

Les services Web utilisant de nombreux langages et protocoles durant le déploiement et l'invocation des services Web qui sont : SOAP, UDDI<sup>41</sup> et WSDL [81].

En plus, les services Web se basent aussi sur l'utilisation des normes actuelles d'Internet comme le protocole HTTP.

L'atout principal des protocoles SOAP, UDDI et WSDL est de se reposer sur le langage XML (eXtensible Markup Language).

Dans cette section, nous allons présenter d'abord le langage XML. Ensuite, on va définir les langages correspondant aux couches transport, communication, description et à la couche de Découverte et de Publication qui sont HTTP, SOAP, WSDL, UDDI [117][118].

### **1-Le langage XML (eXtensible Markup Language)**

Le langage XML [106] est un format texte simple, très flexible standardisé par le W3C<sup>42</sup> en 1998, largement reconnu et utilisé par de nombreuses entreprises comme format universel d'échange de données. C'est un standard qui permet de décrire des documents structurés transportables sur les protocoles Internet.

XML est un métalangage de représentation de données qui définit un ensemble de règles de formatage pour composer des données valides.

La technologie des services Web a été conçue pour fonctionner dans des environnements totalement hétérogènes. Cependant, l'interopérabilité entre les systèmes hétérogènes demande des mécanismes puissants de correspondance et de gestion des types de données des messages entre les différents participants (clients et fournisseurs). C'est une tâche où les schémas de type de données XML s'avèrent bien adaptés.

---

<sup>41</sup> UDDI: Universal Description Discovery and Integration.

<sup>42</sup> W3C: World Wide Web Consortium.



XML constitue la technologie de base des architectures Web services. Elle leur apporte l'extensibilité et la neutralité vis à vis des plateformes et des langages de développement.

## **2-La couche de Transport**

Cette couche s'intéresse aux protocoles de transport de bas niveau, ces derniers vont transporter les requêtes et les réponses échangées entre services. Le protocole le plus utilisé et recommandé par le consortium WS-I<sup>43</sup> (Web Service Interoperability) est l'HTTP, mais d'autres implémentations peuvent utiliser un autre ensemble de protocoles tels que : FTP, JMS (java messagerie services), SMTP, etc.

## **3-La couche de communication :**

Cette couche spécifie le protocole d'échanges de documents XML entre le service Web et ses clients qui est le protocole SOAP.

SOAP (Simple Object Access Protocol) est un standard du Consortium W3C définissant un protocole de transmission de messages basé sur XML permettant la normalisation des échanges de messages, afin d'assurer l'interconnexion des services Web, dans un environnement distribué et hétérogène, en transportant les paquets de données encapsulés sous forme de texte structuré.

SOAP assure l'interopérabilité entre composants tout en restant indépendant des systèmes d'exploitation et des langages de programmation, donc les clients et serveurs peuvent fonctionner sur n'importe quelle plate-forme et être écrits dans n'importe quel langage à partir du moment où ils peuvent formuler et comprendre des messages SOAP.

SOAP utilise principalement les deux standards HTTP et XML respectivement pour la structure des messages et le transport. :

- HTTP comme protocole de transport des messages SOAP. Il constitue un bon moyen de transport en raison de sa popularité sur le Web.

---

<sup>43</sup> WS-I: Web Service Interoperability.

- XML pour structurer les requêtes et les réponses, indiquer les paramètres des méthodes, les valeurs de retours, et les éventuelles erreurs de traitements.

SOAP est constitué de deux parties (voir figure 1) : une enveloppe XML, et un entête d'un protocole de transport.

L'enveloppe XML contient à son tour deux sous éléments : un entête « header » et un corps appelé body.

- L'entête est facultatif, il peut contenir des informations de sécurité (telles que les signatures électroniques), des informations transactionnelles, des informations de traçabilités...
- La partie body est obligatoire, elle contient les éléments suivants :

-cas d'une requête contient le nom de méthode, avec les données correspondantes.

-cas d'une réponse contient les valeurs de retour ou un message d'erreur.

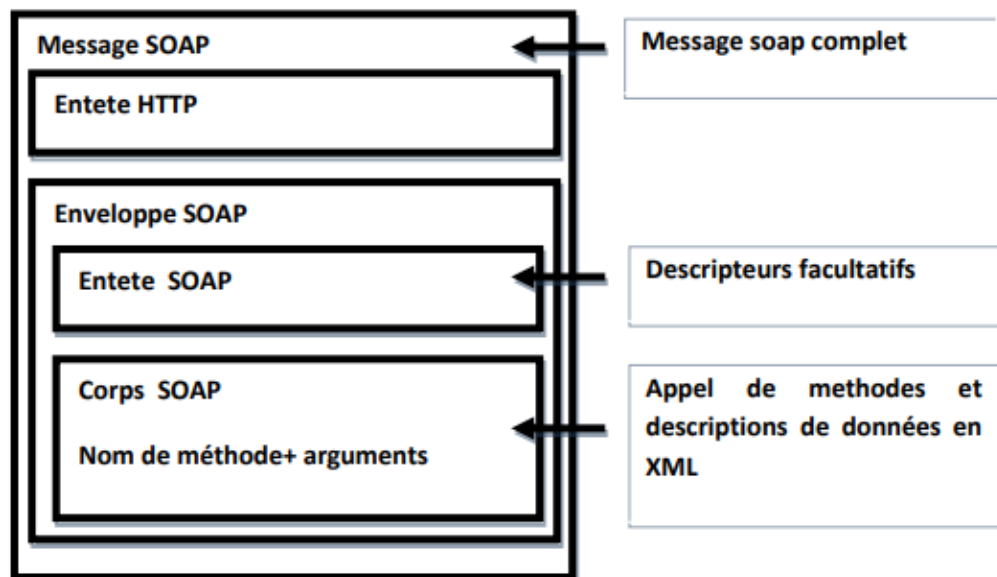


Figure 2:Message SOAP [128].

#### 4- La couche de description :

Le besoin d'une description claire de la communication entre services Web a abouti à des normalisations aux niveaux des messages échangés et des protocoles. Un langage de description s'est basé sur XML est devenu le standard des services Web, c'est le "Web Service Description Language" (WSDL).

WSDL est un standard recommandé par le W3C pour décrire les services Web. Il permet de fournir les spécifications nécessaires à l'utilisation d'un Service Web.

WSDL est un document XML qui décrit les méthodes, les paramètres et les types de retour mais aussi des détails tels que, l'URI du service, les protocoles, les serveurs, les ports utilisés de manière que le client peut localiser le service Web et invoquer une de ses fonctions publiquement accessibles.

Les documents WSDL peut être divisé en deux parties. Une partie pour les définitions abstraites, et une autre pour les descriptions concrètes (voir la figure 2) [119].

La description concrète est composée des éléments qui sont orientés vers le client pour le service physique. Les trois éléments concrets XML présents dans un WSDL sont :

**Binding** : représente une description concrète décrivant le protocole et le format de données utilisés pour les opérations et les messages.

**Port** : représente un point d'accès de services défini par une adresse réseau et une liaison(binding).

**Service** : représente un ensemble de ports (points d'accès) pour permettre l'accès.

La description abstraite est composée des éléments qui sont orientés vers la description des capacités du service Web. Ses éléments abstraits définissent les messages SOAP de façon totalement indépendante de la plate-forme et de la langue. Cela facilite la définition d'un ensemble de services pouvant être implémentés par différents sites Web. Les quatre éléments abstraits XML qui peuvent être définis dans un WSDL sont :

**Type** : cet élément est utilisé pour définir les types de données utilisées pour décrire les messages échangés.

**Message** : l'élément Message représente les données échangées ainsi que les types associés.

**Opération** : cet élément représente la description d'une action, une fonctionnalité, fournie par le service.

**PortTypes** : représente une description des opérations pouvant être exécutées et des messages impliqués

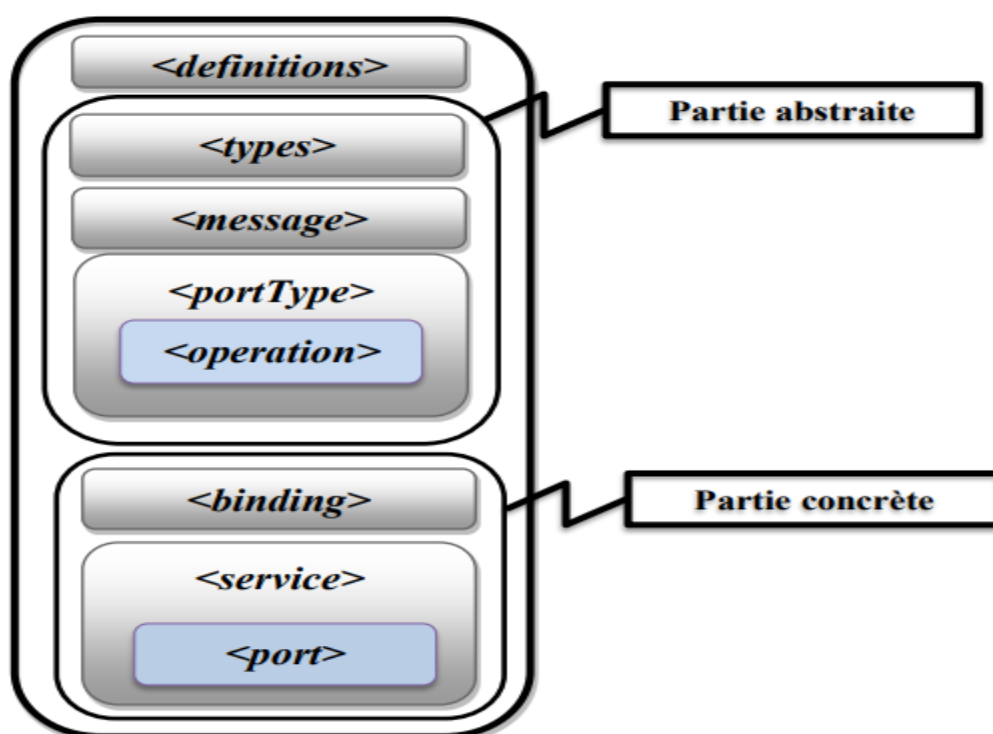


Figure 3:Structure d'une description WSDL 1.1 [127].

## **5-La couche de Découverte et de Publication (UDDI) :**

UDDI Universal Description, Discovery and Integration UDDI est une spécification technique basée sur XML pour la description, la découverte et l'intégration des services Web, Il gère les méta-données des services, l'information sur les fournisseurs de services et les implémentations des services [131].

Un annuaire UDDI est subdivisé en deux parties principales : partie publication ou inscription, et partie découverte. La partie publication regroupe l'ensemble des informations relatives aux entreprises et à leurs services. La partie découverte facilite la recherche d'information contenue dans un UDDI.

Le protocole d'utilisation de l'UDDI offre trois fonctions de base :

- Publish : permet de publier un nouveau service.
- Find : permet d'interroger l'annuaire UDDI.
- Bind : cette fonction gère la façon dont un client peut se connecter et utiliser un service Web une fois celui-ci localisé

Un annuaire UDDI est constitué de pages blanches (nom de l'entreprise, adresse, contacts), de pages jaunes (services classés par catégories industrielles) et de pages vertes (information d'implémentation des services Web proposés).

### **Scénario des services Web :**

Les interactions de base qui existent entre les acteurs des service Web sont les opérations de publication, de recherche, d'invocation [120] [117]comme indiqué dans la figure 3 :

- Publication de service : le fournisseur de service se charge de l'enregistrement et de la publication, de la description de service en WSDL, auprès d'un serveur UDDI, afin qu'elle puisse être découverte et invoquée par des clients potentiels. L'opération s'effectue par l'envoi d'un message (encapsulé dans une enveloppe SOAP). Ce message regroupe la localisation du service, la

méthode d'invocation (et les paramètres associés) ainsi que le format de réponse.

- Recherche et découverte du service : le client interroge d'abord le serveur UDDI dont il connaît l'adresse afin de se renseigner sur les services disponibles correspondant à ses besoins. Le serveur UDDI lui renvoie la liste des possibilités parmi lesquelles il sélectionne l'une d'eux. A ce stade, l'utilisateur ne possède qu'une URL identifiant le service sélectionné.
- Invocation du service : l'utilisateur récupère ensuite une interface WSDL, accessible depuis l'URL, lui permettant la connaissance de l'utilisation du service. De cette interface, il peut générer automatiquement le « proxy » du service, qui est un objet local permettra à l'utilisateur d'accéder au service distant en toute transparence, L'utilisation du service s'accomplit tout simplement en invoquant la méthode du "proxy" correspondant aux besoins de l'utilisateur. Les requêtes et les réponses échangées entre services sont transportés avec des protocoles de bas niveau comme http, SMTP, FTP...etc

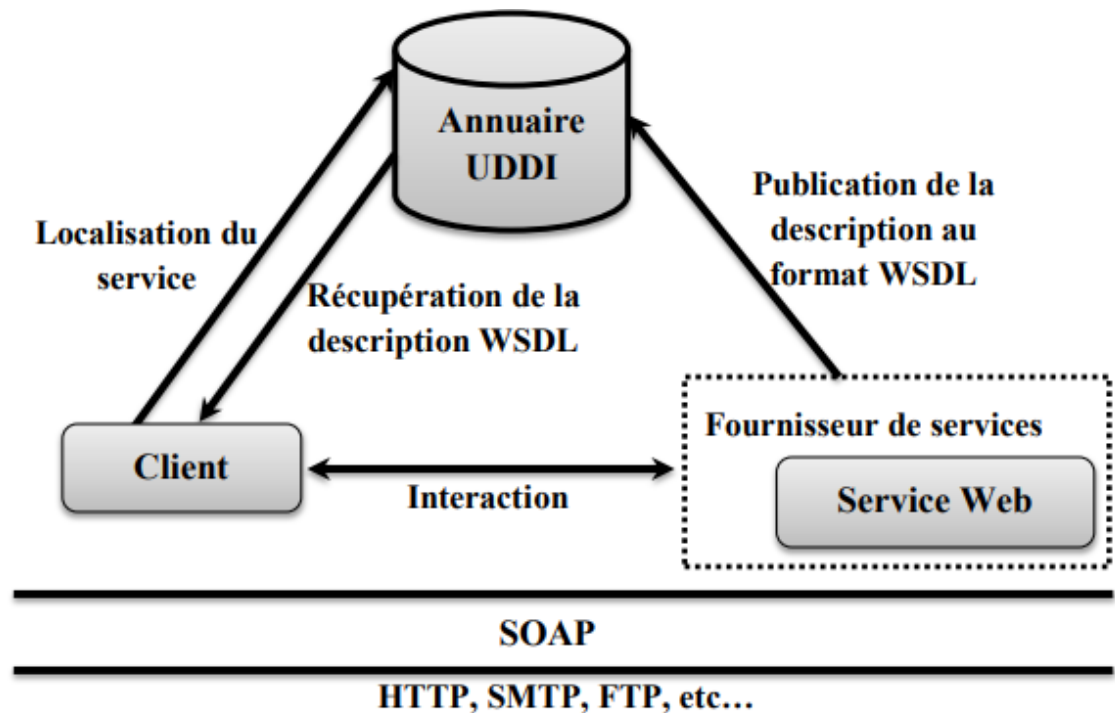


Figure 4:Architecture de base des services Web [127].

#### Avantages des services Web :

Les services Web possède plusieurs avantages [81] [121] :

- Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- Les services Web utilisent des standards et protocoles ouverts.
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges.
- Les Services Web sont utilisables à distance via n'importe quel type de plate-forme.
- Les services Web appartiennent à des applications capables de collaborer entre elles de manière transparente pour l'utilisateur.
- Les services Web permettent d'ignorer l'hétérogénéité entre les différentes applications puisqu'ils décrivent comment transmettre un message (standardisé) entre deux applications, sans imposer comment le construire.

- Les services Web ont la particularité d'être auto-descriptifs c'est-à-dire fournir des informations permettant de comprendre comment les manipuler. La capacité des services à se décrire par eux-mêmes permet d'envisager l'automatisation de l'intégration de services.