

Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE SAAD DAHLEB DE BLIDA 1



FACULTE DES SCIENCES

DEPARTEMENT D'INFORMATIQUE

Mémoire

MASTER ACADEMIQUE

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Systèmes Informatiques et Réseaux

Thème



Description de service Cloud à base d'USDL

Présenté par : AMINATA FOUNE KOMOTA

MAIMOUNA DAMA

Composition Du Jury :

OUKID .S

Pr., université de Blida1

Présidente

CHERIF ZAHAR

Pr., université de Blida1

Examineur

MANCER YASMINE

Pr., université de Blida1

Encadreur

Année Universitaire : 2017/2018

Résumé

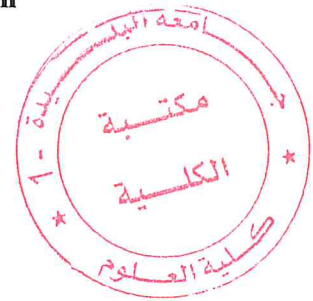
Les langages, les modèles, les ontologies, les normes ont tous été utilisé pour décrire les services du cloud computing, cependant aucun d'entre eux n'est capable de décrire les fonctionnalités spéciales de tous les types de services cloud (IaaS, SaaS, PaaS), du point de vue technique, commercial, opérationnel et sémantique.

La non standardisation de la description des services cloud constitue un grand obstacle dans les processus de découverte, de sélection et de composition pour les entreprises et les particuliers.

Ce mémoire propose une solution pour la description des services cloud prenant en compte tous les aspects (sémantique, opérationnel, business et technique), pour tous les types de service cloud (IaaS, PaaS, SaaS). Le langage de description unifié USDL a été utilisé pour concevoir la solution, USDL ne couvrant que les aspects technique, commercial, opérationnel alors l'aspect sémantique manquant, a été complété en utilisant des ontologies avec le langage OWL-S.

Pour l'expérimentation de la solution proposée, nous avons créé une application java qui permet la création de service et l'utilisation de leurs descriptions pour la découverte et la sélection.

Mots clés : Cloud Computing, Description, USDL, Sélection



Abstract

Languages, models, ontologies, norms; all have been used to describe cloud services, but none of them can describe the special features of all types of cloud services (IaaS, SaaS, PaaS), from a technical, business, operational, and semantic perspective.

The non-standardization of cloud service description is a major obstacle in the discovery, selection and dialing processes for businesses and individuals.

This memory proposes a solution for the description of the cloud services taking into account all the aspects (semantic, operational, business and technical) for all types of cloud service (IaaS, PaaS, SaaS). USDL, the unified description language was used to design the solution; but USDL covering only the technical, commercial and operational aspects, so semantic aspect was completed using ontologies with the OWL-S language.

For the Experimentation of the proposed solution, we created a java application which allows the creation of services and the use of their description for the discovery and selection.

Keywords: Cloud Computing, Description, USDL, Selection

Remerciement

Tout d'abord, nous remercions le bon Dieu, qui nous a donné la force, la patience et la volonté pour accomplir ce modeste travail.

*Nous tenons à exprimer notre gratitude à notre promotrice Mme **MANCER YASMINE** pour son encadrement et sa disponibilité.*

Nous remercions les membres du jury qui nous ont fait l'honneur d'accepter de juger notre travail.

Nous adressons nos sincères remerciements à tous les professeurs et toutes les personnes qui par leurs paroles, leurs efforts, leurs conseils et leurs critiques nous ont guidés durant nos recherches.

Nous remercions nos parents pour leur soutien inconditionnel.

Enfin, nous remercions tous nos amis pour leur sincère amitié et leur confiance.

A tous ces intervenants, nous vous présentons nos remerciements, notre respect et notre gratitude.

SOMMAIRE

INTRODUCTION GENERALE	1
I CHAPITRE I : Cloud Computing.....	3
I.1 Introduction :	3
I.2 Historique :	3
I.3 Définition :	5
I.4 Les caractéristiques du Cloud Computing :	6
I.5 Les types de services du Cloud Computing :	7
I.5.1 IaaS : Infrastructure as a Service :	8
I.5.2 PaaS: Platform as a Service:	8
I.5.3 SaaS: Software as a Service:.....	9
I.5.4 Les avantages et inconvénient des types de services :.....	10
I.6 Les Modèles de déploiement du Cloud Computing :	11
I.6.1 Cloud privé :	12
I.6.2 Cloud public :	12
I.6.3 Cloud communautaire :.....	13
I.6.4 Cloud hybride :	13
I.6.5 Différence entre cloud privée et cloud public :.....	13
I.7 Inter-cloud et Multi-cloud :	14
I.7.1 Inter-Cloud :.....	14
I.7.2 Multi-cloud :	15
I.7.3 Différence entre Inter-Cloud et Multi-cloud :.....	15
I.8 Les avantages et inconvénients du Cloud Computing :.....	16
I.8.1 Les avantages :	16
I.8.2 Les inconvénients :	16
I.9 Conclusion :	17
II CHAPITRE II : Description des services cloud	18

II.1	Introduction :	18
II.2	Partie 1 : généralités et définitions :	19
II.2.1	La description de service :	19
II.2.2	La découverte de service :	19
II.2.3	La composition des services :	19
II.2.4	La sélection :	19
II.2.5	Utilisation des ontologies :	20
II.2.6	Aspects de description :	20
II.3	Partie 2 : les langages de description de services WSDL et USDL	21
II.3.1	Vue d'ensemble WSDL	21
II.3.2	Vue d'ensemble USDL :	23
II.3.3	WSDL vs USDL :	27
II.4	Partie 3 : Travaux connexes	27
II.4.1	Approches de service IAAS :	28
II.4.2	Approches de services PAAS :	33
II.4.3	Approches de services SAAS:	35
II.4.4	Approches de services génériques :	37
II.5	Partie 4 : Exigences pour une description de services Cloud	44
II.6	Conclusion	48
III	CHAPITRE III : Conception	49
III.1	Introduction	49
III.2	Les nouvelles exigences proposées:	50
III.3	Conception :	52
III.3.1	Représentation ontologique du module Service :	55
III.3.2	Représentation ontologique du module Functional :	57
III.3.3	Représentation ontologique du module Technical :	58
III.3.4	Représentation ontologique du module Participant :	58
III.3.5	Représentation ontologique du module Pricing :	60
III.3.6	Représentation ontologique du module Legal :	61
III.3.7	Représentation ontologique du module Interaction :	64
III.3.8	Représentation ontologique du module Foundation :	65
III.3.9	Représentation ontologique du module Service Level:	66
III.4	Etudes de cas :	68
III.4.1	Amazon EBS (IAAS) :	68

III.5	4.2 Microsoft Azure Cloud Service(PAAS) : [59]	73
III.6	4.3 SharePoint Online Business (SAAS) :	77
III.7	Conclusion:	82
IV	Chapitre IV : Expérimentation.....	83
IV.1	Introduction :	83
IV.2	Les outils de développement	83
IV.2.1	Langage de programmation java :	83
IV.2.2	Environnement de développement Netbeans :	84
IV.2.3	Xamp server:	84
IV.3	Description de l'application :	85
IV.3.1	Partie 1 : Creation de service	85
IV.3.2	Partie 2 : Découverte de service.....	93
IV.3.3	Partie 3 : Sélection de services :	93
IV.4	Conclusion:	109
	Conclusion générale :	110
	References:	112
	Annexes.....	120

LISTES DES FIGURES

Figure 1: les caractéristiques du cloud computing [15]	7
Figure 2: les modèles de base du cloud computing[1].....	8
Figure 3: les modèles de déploiement du cloud computing [5]	12
Figure 4: les balises WSDL [26]	22
Figure 5: les modules USDL [30]	25
Figure 6: Représentation ontologique du module Service	55
Figure 7: une partie de l'ontologie pour résoudre l'exigence R1	56
Figure 8: Une partie de l'ontologie pour résoudre l'exigence R2	56
Figure 9: Une partie de l'ontologie pour résoudre l'exigence R4	57
Figure 10: Représentation ontologique du module Functional	57
Figure 11: Représentation ontologique du module Technical	58
Figure 12: Représentation ontologique du module Participant.....	59
Figure 13: Une partie de l'ontologie pour résoudre l'exigence R6	59
Figure 14: Une partie de l'ontologie pour résoudre l'exigence R7	60
Figure 15: Une partie de l'ontologie pour résoudre l'exigence R8	60
Figure 16: Représentation ontologique du module Pricing	61
Figure 17: Représentation ontologique du module Legal	62
Figure 18: Une partie de l'ontologie pour résoudre l'exigence R17	62
Figure 19: Une Partie de l'ontologie pour résoudre l'exigence R18.....	63
Figure 20: Une partie de l'ontologie pour résoudre l'exigence R14	63
Figure 21: Représentation ontologique du module Interaction.....	64
Figure 22: Représentation ontologique du module Foundation.....	65
Figure 23: Représentation ontologique du module Service Level	66
Figure 24 : Une partie de de l'ontologie pour résoudre l'exigence R15	67
Figure 25: Une partie de l'ontologie pour résoudre l'exigence R12 SLA pour IaaS .	67
Figure 26: Une partie de l'ontologie pour résoudre l'exigence R12 SLA pour PaaS	68
Figure 27: Une partie de l'ontologie pour résoudre l'exigence R12 SLA pour SaaS	68
Figure 28: Représentation ontologique du module fonctionnel d'EBS	70
Figure 29: le processus composite du service Amazon EBS	72
Figure 30: Graphe de la représentation OWL-S d'EBS	72
Figure 31: Représentation du Module service de Azure cloud service.....	74

Figure 32: Représentation du Module Participant de Azure cloud service.....	75
Figure 33: Représentation du Module Foundation de Azure cloud service.....	75
Figure 34 : Représentation du Module Functional de Azure cloud service.....	76
Figure 35: le processus composite du service Azure Cloud Service	76
Figure 36: Graphe de la représentation d' Azure Cloud Service en OWL-S.....	77
Figure 37: Représentation du Module Functional de SharePoint	79
Figure 38: Représentation du Module service de SharePoint	80
Figure 39: Représentation du Module Participant de SharePoint	80
Figure 40: Processus Composite du service Sharepoint	81
Figure 41: Graphe Principal SharePoint OWL-S.....	81
Figure 42: interface d'accueil	86
Figure 43: Liste des services	86
Figure 44: Creation de service fenetre1	87
Figure 45: Creation de service fenetre2	88
Figure 46: Creation de service fenêtre3	89
Figure 47: Creation de service fenetre4 PaaS	90
Figure 48: Creation de service fenetre4 IaaS	91
Figure 49: Creation de service fenetre4 SaaS	92
Figure 50: decouverte de service.....	93
Figure 51: Filtrage par score	100
Figure 52: filtrage par Reputation.....	102
Figure 53: filtrage par Experience.....	103
Figure 54: filtrage par confiance	104
Figure 55: filtrage par Disponibilité.....	105
Figure 56: filtrage par prix	106
Figure 57: filtrage par temps de réponse	107
Figure 58: filtrage par MTTR.....	108
Figure 59: filtrage par risque.....	109

LISTE DES TABLEAUX

Tableau 1: Comparatif des types de service	11
Tableau 2: Tableau récapitulatif des travaux connexes	42
Tableau 3: Tableau des exigences pour une bonne description.....	44
Tableau 4: les nouvelles exigences	50
Tableau 5: Mapping USDL Ontologie, [48].....	54
Tableau 6: Domaine de valeur	96
Tableau 7: les données.....	99
Tableau 8: Valeurs maximales et minimales des QoS.....	99
Tableau 9: Normalisation des QoS	99
Tableau 10: score final des services	100

LISTE DES ACRONYMES ET ABRÉVIATIONS

API	Applications Programming Interface
AWS	Amazon Web Service
BDL	Blueprint Definition Language
BSM	Business Service Management
BCL	BlueLine Constraint Language
BPAAS	Business process as a Service
BML	Blueprint Manipulation Language
BQL	Blueprint Query Language
CAAS	Communication as a Service
ClaaS	Cluster –as-a-Service
CSV	Comma Separated Values
COCOON	Cloud Computing Ontology
CloudML	Cloud Modeling language
CSDM	Cloud Service Description Model
CDDL	Common Development and Distribution License
DAAS	Database as a Service
DeAAS	Desktop as a Service
D-Cloud	Distributed Cloud
EBS	Elastic Block Storage
EMF	Eclipse Modeling Framework
EC2	Elastic Compute Cloud

FTP	File Transfert Protocol
HTTP	Hyper Text Transfer Protocol
IAAS	Infrastructure as a Service
IOS	IPhone OS
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
JVM	Java Virtual Machine
JRE	Java Runtime Environment
JSP	Java Server Pages
MDE	Model Driven Engineering
MetaMORP OSY	Meta-Modeling of Mas Object-based with Real-time Specification in Project Of complex Systems
MAS	Modeling Analysis Systems
NIST	National Institute of Standard and technology
NAAS	Network as a Service
OWL	Web Ontology Language
OWL-S	Semantic Web Ontology Language
PAAS	Plateform as a Service
PDA	Personal Digital Assistant
P2P	Peer To Peer
PHP	Hypertext Preprocessor
QOS	Quality of Service
RVWS	Resources via Web Services framework

SAAS	Software as a Service
SeAAS	Security as a Service
STAAS	Storage as a Service
SOAP	Simple Object Access Protocol
SLA	Service Level Agreement
SQL	Structured Query Language
SPL	Software Product Lines
SBA	Service-based Application
S3	Simple Storage Service
USDL	Unified Service Description Language
UDDI	Universal Description Discovery and Integration
UML	Unified Modeling Language
VPN	Virtual Private Network
VMS	Virtual Memory System
WAAS	Workplace as a Service
WSDL	Web Service Description Language
W3C	World Wide Web Consortium
WS	Web Service
WSRF	Web Service Resource Framework
WSDL-S	Semantic Web Services Description Language
WSMO	Web Services Modeling Ontology
XML	Extensible Markup Language
XSD	XML Schema Definition

XAAS

Anything as a Service

XP

eXtreme Programming

INTRODUCTION GENERALE

Malgré le fort engouement que suscite le Cloud Computing, son adoption et sa vulgarisation se heurtent à plusieurs obstacles tels que : l'hétérogénéité des architectures et des API, le grand nombre de fournisseurs et de technologies, l'absence de standard, le problème du Vendor Lock-In¹, etc.

Le nombre de fournisseurs augmente continuellement, ce qui conduit à augmenter le nombre de services Cloud. Cependant, de nombreux problèmes apparaissent aux utilisateurs et compliquent la tâche de découverte de services, car les fournisseurs utilisent des techniques distinctes pour la description de leur services (telles que les langages, les normes, les ontologies, ou les modèles, etc.).

Cette diversité de techniques conduit les fournisseurs de services à fournir différentes interfaces et d'utiliser différents protocoles pour l'accès aux services. Ce manque de représentation standard de services de Cloud empêche l'interopérabilité entre les services et renforce le problème de verrouillage des fournisseurs.

Problématique :

Les descriptions de services Cloud existantes ne couvrent que certains aspects et négligent d'autres. Le langage WSDL prend en charge l'aspect technique mais ne couvre pas les aspects business et sémantique. Les chercheurs considèrent USDL (Unified Service Description Language) comme un langage qui couvre la description de service à partir de trois aspects (technique, opérationnel et business), mais il ne prend pas en compte l'aspect sémantique et n'est pas conçu pour le domaine du Cloud Computing.

¹ Situation d'enfermement propriétaire.

La problématique de ce sujet consiste à définir une solution pour la description des services Cloud, basée sur le langage USDL, en considérant les aspects (technique, opérationnel, business et sémantique).

Objectifs :

1. Etude comparative des solutions existantes pour la description des services Cloud.
2. Définir une solution pour la description des services Cloud, basée sur le langage USDL, en considérant les aspects (technique, opérationnel, business et sémantique).
3. Validation de la solution proposée par une expérimentation.

Organisation du mémoire :

Le présent mémoire est organisé de la manière suivante :

Chapitre I : Présente un aperçu sur les concepts de base du Cloud Computing, définition caractéristiques, types de déploiement, types de services, et l'évolution du Cloud Computing vers l'Inter-Cloud.

Chapitre II : Etat de l'art sur la description des services Cloud et les solutions existantes pour la description.

Chapitre III : Modélisation de la solution proposée et ses différents concepts.

Chapitre IV : Expérimentation de la solution proposée et réalisation d'une application java qui permet la création et la sélection des services cloud.

I CHAPITRE I : Cloud Computing

I.1 Introduction :

Le cloud computing, nuage informatique en français, n'est pas un nouveau modèle technologique, mais une intégration des technologies du passé qui a donné lieu à une nouvelle façon de rendre des services ordinateurs disponibles sur internet [3]. L'idée principale derrière ce concept est de permettre aux consommateurs d'acquérir les ressources informatiques à la demande, et de payer en fonction de l'utilisation de volume [4].

Dans ce chapitre nous allons présenter les notions et concepts de base du Cloud Computing, nous commençons par un historique et des définitions, ensuite nous détaillons les caractéristiques de base, les types de services, les modèles de déploiement ainsi que le concept d'Inter-cloud et multi-cloud et pour finir nous énumérerons quelques avantages et inconvénients du cloud computing.

I.2 Historique :

Le cloud computing est rapidement passé du stade de concept à celui de phénomène largement répandu. En 2008, Nick Carr a publié « The Big Switch » [72], l'une des premières tentatives d'exploration de l'impact de l'informatique à la demande, la capacité de payer des ressources informatiques au compteur. En trois ans, ce phénomène est devenu monnaie courante, non seulement par l'intermédiaire des publicités imprimées et télévisées de Microsoft, mais aussi lorsque Amazon a hébergé (puis deshébergé) « Wikileaks » dans le cadre de son offre de services en nuage, Amazon Web Services.

La fin des années 1990 et le début des années 2000 représentent une période propice pour créer une entreprise en ligne ou investir dans une telle activité. Avec le

développement de l'architecture multi-tenant, l'omniprésence du haut débit et la mise en place de normes d'interopérabilité universelles entre les logiciels, c'est le cadre idéal pour permettre au Cloud Computing de décoller. Salesforce.com est lancé en 1999. Il est le premier site à proposer des applications d'entreprise à partir d'un simple site web standard, accessible via un navigateur Web : c'est ce qu'on appelle aujourd'hui le Cloud Computing. Amazon.com lance Amazon Web Services en 2002. Ce nouveau service permet aux utilisateurs de stocker des données et tire profit des compétences d'un très grand nombre de personnes pour de très petites tâches.

En 2006, Amazon développe ses services Cloud. Le premier à voir le jour est Elastic Compute Cloud (EC2), qui permet aux utilisateurs d'accéder à des ordinateurs et d'y exécuter leurs propres applications, le tout sur le Cloud. Le deuxième service lancé est Simple Storage Service (S3). Il permet d'introduire le modèle de paiement à l'utilisation auprès des clients et du secteur en général, modèle qui représente désormais une pratique courante. Salesforce.com lance ensuite Force.com en 2007. Cette plate-forme en tant que service (PaaS) permet aux développeurs de concevoir, de stocker et d'exécuter toutes les applications et tous les sites Web nécessaires à leurs activités sur le Cloud. Google Apps arrive en 2009 et permet à ses utilisateurs de créer et de stocker des documents entièrement sur le Cloud. Plus récemment, les entreprises de Cloud Computing ont cherché à accroître d'avantage l'intégration de leurs produits.

En 2010, salesforce.com lance sa base de données Cloud avec Database.com pour les développeurs, marquant ainsi le développement des services de Cloud Computing utilisables sur n'importe quel terminal, exécutables sur n'importe quelle plateforme et écrits dans n'importe quel langage de programmation.

I.3 Définition :

Beaucoup de définitions sur le cloud computing sont disponibles dans la littérature. Les chercheurs dans [6] ont effectué une étude rigoureuse de nombreuses définitions, les présentant sous différentes perspectives.

Il existe de nombreuses définitions du terme Cloud Computing et il y a peu de consensus sur une seule et universelle définition.

En effet, Depuis l'émergence du terme « cloud computing », organismes et universités, essaie chacun de donner sa propre définition du cloud suivant différents angles. Nous présentons ici quelques définitions du cloud computing dans la littérature.

Le NIST (National Institut of Standard and technology) défini le Cloud Computing comme suit : « le cloud computing est un modèle qui permet un accès omniprésent, pratique, très répandue, à la demande, à un réseau partagé et à un ensemble de ressources informatiques configurables (réseaux, serveurs, stockage, applications et services) qui peuvent être provisionnées et libérées avec un minimum d'administration » [9].

Le Cloud computing peut être défini aussi comme un nouveau modèle d'approvisionnement des services sur internet. C'est le domaine de recherche où un cloud est utilisé pour accéder aux données, fichiers, logiciels etc [10].

Le Cloud Computing est un environnement d'exécution élastique de ressources impliquant de multiples acteurs pour offrir un service tarifié avec un certain niveau de qualité de service [2].

Dans ce travail, nous adoptons la définition du NIST qui est l'une des définitions communément acceptées.

I.4 Les caractéristiques du Cloud Computing :

Le cloud computing est composé de quatre parties : l'application, l'informatique, Le stockage, et la connectivité. Chaque partie se compose de différents produits et sert un but différent pour les entreprises et les particuliers. Les principales caractéristiques du cloud computing d'après les définitions fournies par le NIST sont [9] :

- Self-service (service à la demande) : interaction avec le serveur et le réseau de stockage en temps, la planification et l'optimisation nécessaire est fournis sans la nécessité de l'interaction d'un technicien de service coté fournisseur.
- Large accès au réseau : l'accès aux services et fonctionnalités se fait sur internet via les points d'accès standards tels que les téléphones portables, ordinateurs portables et PDA (Personal Digital Assistant).
- mise en commun des ressources : les ressources sont fournies aux consommateurs tout en leur permettant de fixer des exigences au niveau approprié à leurs besoins spécifiques. Le fournisseur, quant à lui gère l'acheminement des ressources spécifiques desservant plus d'un consommateur en même temps.
- Élasticité rapide : Le fournisseur cloud peut modifier les ressources et les fonctionnalités sans beaucoup de retard ou sans nécessité d'interaction avec le matériel. Les ressources pour le consommateur peuvent aussi être évolutives pour presque toute la capacité en fonction des besoins.
- Service mesuré : les ressources de données sont optimisées et à effet de levier en fonction de divers paramètres, en fonction du niveau de stockage, la bande passante et le nombre de comptes d'utilisateur. C'est à dire évaluer et garantir la performance et la disponibilité selon les besoins du consommateur.

En plus des cinq caractéristiques définies par NIST, il y a d'autres caractéristiques telles que : la facilité d'utilisation, la haute disponibilité, fiabilité et tolérance aux pannes, évolutivité.

La figure 1 représente quelques caractéristiques du cloud computing :

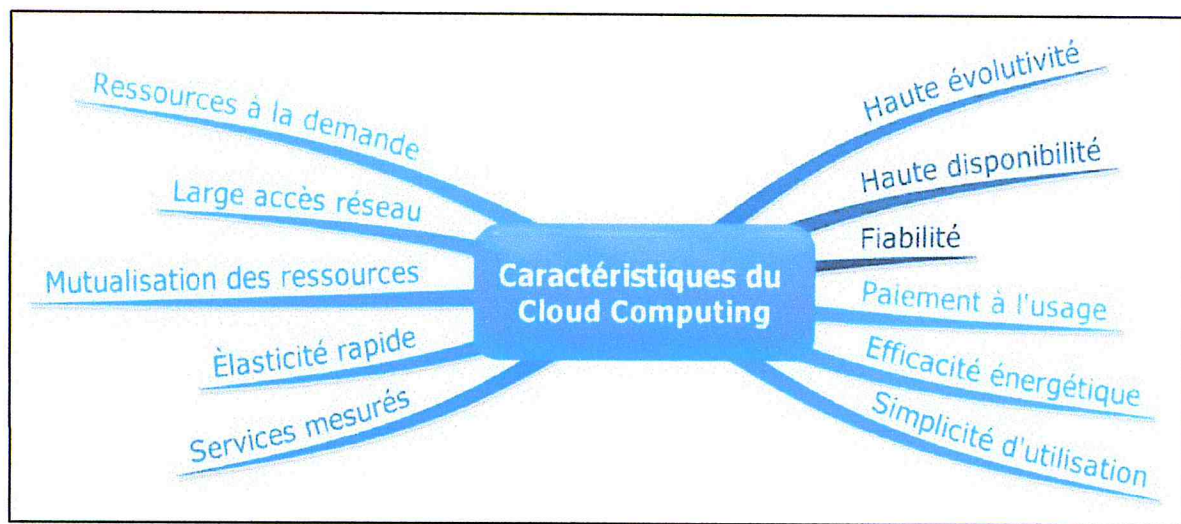


Figure 1: les caractéristiques du Cloud Computing [15]

I.5 Les types de services du Cloud Computing :

Il y a trois principaux modèles de base de service cloud qui sont : le IaaS (Infrastructure as a Service), le PaaS (Plateform as a Service), le SaaS (Software as a Service) [11].

La figure 2 présente ces différents modèles de prestation de service cloud :

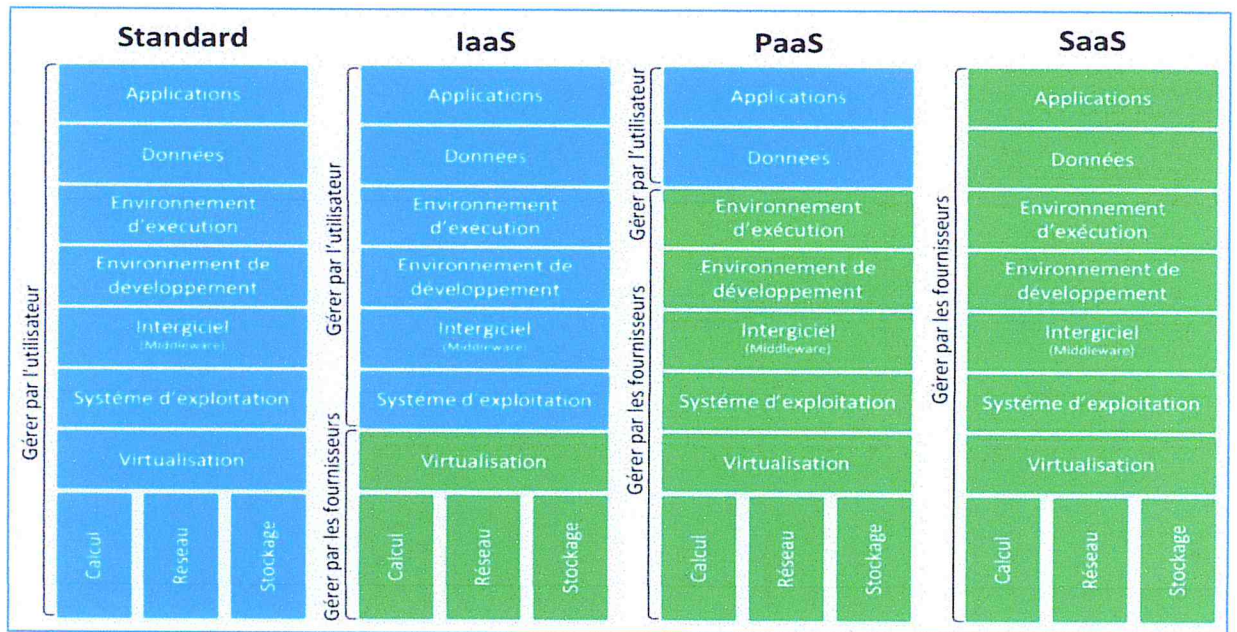


Figure 2: les modèles de base du Cloud Computing [1]

I.5.1 IaaS : Infrastructure as a Service :

L'IaaS désigne une infrastructure matérielle louée à la demande par le client (serveurs, réseaux, stockage, puissance de calcul, système d'exploitation) hébergés chez le prestataire de service.

L'entreprise maintient les applications, les runtimes, les bases de données et le logiciel serveur.

Le fournisseur cloud maintient la virtualisation, le matériel serveur, le stockage, les réseaux.

I.5.2 PaaS: Platform as a Service:

Le PaaS concerne l'utilisation par le biais internet de plateformes sur lesquelles on peut déployer ses propres applications.

L'entreprise maintient uniquement les applications.

Le fournisseur cloud maintient les runtimes, les bases de données, le logiciel serveur, la virtualisation, le matériel serveur, le stockage et les réseaux.

I.5.3 SaaS: Software as a Service:

Le SaaS concerne l'utilisation de logiciels directement sur le cloud. Le fournisseur propose des logiciels basés sur un ensemble commun de définitions de données et de code, qui sont consommés par tous les clients sous contrat à tout moment sur le principe de l'utilisateur-payeur ou sous forme d'abonnement.

Le fournisseur cloud maintient les applications, les runtimes, les bases de données, le logiciel serveur, la virtualisation, le matériel serveur, le stockage et les réseaux.

Bien que les trois types de services de cloud computing (IaaS, PaaS et SaaS) soient la base de distinction du type de service, le style de nomenclature « as a Service » a été utilisé ailleurs pour caractériser des services et ressources Cloud Computing. Chacune des nouvelles abréviations peut-être vues comme un sous-ensemble d'un ou plusieurs des trois types de base. Ainsi, on parle aujourd'hui de :

- XaaS (Anything as a Service) : en français, Le « Tout en tant que service ». Cela concerne tous les services cloud.
- DaaS (Database as a Service) : en Français, « Base de données en tant que service », offre des fonctions traditionnelles de bases de données, typiquement la définition des données, le stockage et la récupération, sur une base d'abonnement sur le Web [14].
- CaaS (Communication as a Service) : en Français, « Communication en tant que Service », est un service de communication audio et vidéo, services collaboratifs, communications unies, messagerie électronique, messagerie instantanée partage de donnée (web conférence) [15].

- STaaS : (Storage as a Service) : en français, « stockage en tant que service » concerne la location d'espace de stockage à la demande [15].
- SeaaS :(Security as a Service) : en français, « sécurité en tant que service » offre bien plus de solutions spécifiques pour remédier à des problèmes de sécurité spécifiques comme la protection contre les virus. Aussi, elle assure la gestion des identités. [14]
- WaaS : (Workplace as a Service) : en français, « workplace en tant que service ». Le Workplace as service est un environnement de travail virtuelle disponible partout quel que soit l'appareil utilisé, il permet aux utilisateurs finaux professionnels un accès sécurisé, permanent et en tout lieu à leurs applications et données d'entreprise. [14]
- BPaaS :(Business Process as a Service) : en français, « processus commercial en tant que service », consiste à externaliser une procédure d'entreprise suffisamment industrialisée pour s'adresser directement aux managers d'une organisation, sans nécessiter l'aide de professionnels de l'informatique [15]
- DeaaS :(Desktop as a Service) : en français, « Bureau en tant que Service » le DeaaS est l'externalisation d'une Virtual Desktop Infrastructure auprès d'un fournisseur de services. Généralement, le Desktop as a Service est proposé avec un abonnement payant. [15]

I.5.4 Les avantages et inconvénients des types de services :

Dans le tableau 1, nous présentons quelques avantages et inconvénients des différents types de services de base du cloud computing :

Types de service	Avantages	Inconvénients
IaaS	<ul style="list-style-type: none"> -pas d'achat ni gestion de matériels -contrôle total des systèmes - personnalisation -grande flexibilité 	<ul style="list-style-type: none"> -besoin d'un administrateur système
PaaS	<ul style="list-style-type: none"> -environnement hétérogène - pas d'installation -Ne nécessite pas une infrastructure 	<ul style="list-style-type: none"> -limitation des langages -besoin d'un administrateur système
SaaS	<ul style="list-style-type: none"> -pas d'installation -pas de mises jour -pas de licence -facilité de test de nouveaux logiciels 	<ul style="list-style-type: none"> - sécurité -dépendance totale -traitement moins rapide -logiciels limités

Tableau 1: Comparatif des types de service

I.6 Les modèles de déploiement du Cloud Computing :

Il existe 4 types de déploiement pour les cloud qui sont le cloud privé, le cloud public, le cloud communautaire et le cloud hybride [9].

La figure 3 représente les différents modèles de déploiement cloud :

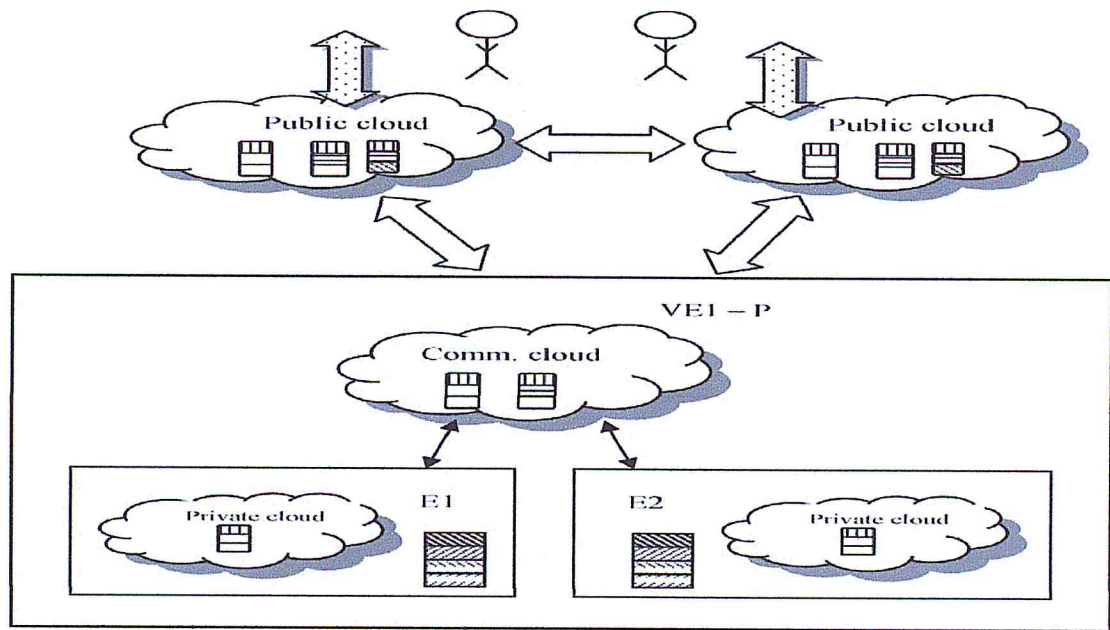


Figure 3: les modèles de déploiement du cloud computing [5]

I.6.1 Cloud privé :

Cette infrastructure de cloud computing est configurée pour une utilisation exclusive par une organisation unique comprenant plusieurs consommateurs. Il peut être administré, géré et exploité par l'organisation elle-même (cloud privé interne), un tiers (cloud privé externe), ou une combinaison d'entre eux, et il peut être localisé à l'intérieur ou à l'extérieur des locaux du client.

I.6.2 Cloud public :

Les ressources sont disponibles par internet et sont gérées par un prestataire externe c'est à dire l'utilisation de ces types de cloud est ouvert à tous. Il peut être administré, géré et exploité par une entreprise, une université, une organisation du gouvernement, ou une combinaison d'entre eux. Il existe sur les locaux du fournisseur de services cloud.

I.6.3 Cloud communautaire :

Cette infrastructure de cloud computing est configurée pour une utilisation exclusive par une communauté de consommateurs comme les organisations qui ont des préoccupations communes (par exemple, les missions, les exigences en matière de sécurité, de politique). Il peut être administré, géré et exploité par un ou plusieurs des organismes de la communauté, un tiers, ou une combinaison d'entre eux, et il peut exister sur ou à l'extérieur des locaux.

I.6.4 Cloud hybride :

Cette infrastructure de cloud computing est une composition de deux ou plusieurs des infrastructures de cloud computing (privé, communautaire ou public) qui restent des entités uniques, mais sont liés par la technologie normalisées ou propriétaires qui permet la portabilité des données et des applications (par exemple, l'éclatement des clouds pour l'équilibrage de charge entre les clouds).

I.6.5 Différence entre cloud privée et cloud public :

Dans le cas du cloud public, un grand nombre de ressources informatiques sont partagées avec de nombreuses entreprises à travers l'ensemble du réseau Internet. Si ce modèle possède de nombreux avantages en termes de réduction des coûts, de collaboration et d'agilité, pour certaines entreprises, en revanche, il soulève, parfois certaines questions sur la sécurité et la confidentialité des données.

De son côté, le cloud privé propose des ressources informatiques dont l'usage est uniquement réservé à une entreprise. L'entreprise ou client peut héberger son cloud privé soit sur site, dans son centre de données (en utilisant une virtualisation et une automatisation à grande échelle), soit hors site chez un fournisseur de services de cloud.

Le cloud privé possède la plupart des avantages du cloud public mais permet d'avantage de contrôle et de personnalisation du fait que des ressources dédiées sont à la disposition du client. Il peut offrir encore plus de flexibilité ce qui peut rendre son coût prohibitif et amoindrir les économies d'échelle pour certaines entreprises.

I.7 Inter-cloud et Multi-cloud :

La notion d'inter-cloud est encore ambiguë. En effet, il existe deux approches d'inter-cloud qui sont vraiment différentes : la fédération plus connu sous le nom inter-cloud et le multi-cloud. Les fédérations et les multi-cloud sont donc tous deux types d'inter-cloud [17].

I.7.1 Inter-Cloud :

L'inter-cloud est un réseau global "cloud de cloud" et une extension de l'internet, réseaux de réseaux ; c'est l'interconnexion de plusieurs fournisseurs d'infrastructures cloud [7].

L'IEEE (Institute of Electrical and Electronics Engineers) définit l'inter-cloud comme un concept des réseaux cloud connectés, y compris les clouds publics, privés et hybrides. Il intègre un certain nombre d'efforts technologiques mis en place pour améliorer l'interopérabilité et la portabilité entre les réseaux de cloud [17].

L'idée derrière un inter-cloud est qu'une seule fonctionnalité commune combinerait plusieurs différents cloud individuels en une seule masse transparente en termes d'opérations à la demande [16].

I.7.2 Multi-cloud :

Le multi-cloud consiste en l'utilisation de multiple environnement du cloud indépendants sans nécessité d'accord préalable entre les différents fournisseurs.

Une stratégie multi-cloud consiste à faire appel à plusieurs fournisseurs de services cloud pour profiter des avantages de chacun afin d'élaborer une solution parfaitement adaptée aux besoins de l'entreprise [18].

Le multi-cloud présente de nombreux avantages :

Il permet notamment d'améliorer la récupération en cas de désastre. Les clients peuvent aussi utiliser des services cloud spécifiques en provenance de différents fournisseurs. Il offre également l'opportunité de profiter des tarifs avantageux et de la scalabilité illimitée du cloud public pour déplacer des applications agiles vers le cloud, tout en exploitant le cloud privé pour les données les plus sensibles. Répartir les données sur plusieurs plateformes permet également de minimiser les possibilités de downtime.

Le multi-cloud permet également de prendre avantage des Data Centers cloud localisés dans différentes régions géographiques. [18]

I.7.3 Différence entre Inter-Cloud et Multi-cloud :

Une stratégie inter-cloud nécessite un accord préalable entre les différents fournisseurs cloud contrairement à une stratégie multi-cloud.

Une fédération est réalisée lorsque l'ensemble des fournisseurs de cloud interconnecte volontairement leurs infrastructures afin de permettre le partage de ressources entre eux. Multi-cloud désigne l'utilisation de plusieurs cloud indépendants par un client ou un service. Contrairement à une fédération, un environnement multi-cloud n'implique pas l'interconnexion volontaire et le partage des infrastructures des fournisseurs. Les clients ou leurs représentants sont directement responsables de la gestion du provisionnement et de l'ordonnancement des ressources.

I.8 Les avantages et inconvénients du Cloud Computing :

Malgré ses avantages, le cloud computing présente également des inconvénients. Ici, nous présentons quelques avantages et inconvénients du cloud computing.

I.8.1 Les avantages :

Le cloud computing est une solution proposant divers avantages pour les entreprises de tout type, ainsi le cloud computing permet [8], [9], [15]:

- de partager (réseaux, serveurs, stockage, service) comme ressources informatique;
- la réduction des coûts ;
- la mutualisation des ressources informatique ;
- La facturation à l'usage ;
- l'accès aux services à tout moment sur tous les supports via internet ;
- le déploiement rapide ;
- de faciliter l'adaptation des applications ;
- des accès aux ressources plus flexibles ;
- de fournir des services souples et bon marché ;
- aux petites organisations et personnes d'accueillir et d'offrir des services à l'échelle mondiale.

I.8.2 Les inconvénients :

Si le cloud computing offre des avantages plutôt satisfaisant, cela n'exclut pas qu'il ait également des obstacles et des inconvénients.

Nous citons quelques obstacles ou inconvénients :

- Problème d'interopérabilité ;
- Faible contrôlabilité ;
- Données Lock-in ;
- Confidentialité des données ;
- Dépendance totale à internet ;
- Mauvais fonctionnement avec les connexions à basse vitesse ;
- Plus l'entreprise s'agrandit, plus le coût du service cloud augmente.

I.9 Conclusion :

Le Cloud computing est considéré comme la nouvelle révolution de l'informatique à travers l'internet. Ainsi, le cloud computing offre des avantages pour les organisations et les individus. Dans ce chapitre nous avons donné un aperçu sur les concepts de base du cloud computing.

Dans le chapitre suivant nous abordons la notion de description des services cloud, les différents travaux connexes ainsi que les exigences proposées dans la littérature pour une bonne description des services cloud.

II CHAPITRE II : Description des services cloud

II.1 Introduction :

Les fournisseurs de services cloud ne cessent d'augmenter et l'utilisation de tout service nécessite sa localisation, pour ce faire, une description du service est nécessaire. L'objectif de la description est de faciliter entre autre la découverte, la sélection et la composition.

Il existe une variété de langages de modélisation, de spécification bien conçus au service des services Internet, mais aucun d'entre eux n'est capable de décrire les fonctionnalités spéciales des services cloud, du point de vue technique, sémantique, opérationnel et commercial. C'est dans cette optique que plusieurs chercheurs se sont intéressés à la description des services cloud en utilisant différentes méthodes : les ontologies, les langages, les modèles et normes à travers les 3 types de services Cloud (IaaS, PaaS et SaaS).

Dans ce chapitre nous présentons un aperçu des éléments de base d'une description. Nous commençons par définir certains concepts propres à la description, ensuite nous donnons une vue d'ensemble sur les langages de descriptions USDL (Unified Service Description Language) et WSDL (Web Service Description Language), par la suite nous présentons certains travaux connexes suivie d'une analyse critique et enfin nous présentons les exigences proposées dans la littérature pour effectuer une bonne description.

II.2 Partie 1 : généralités et définitions :

Nous présentons ici quelques définitions et concepts propre à la description de service.

II.2.1 La description de service :

Une description de service est une spécification des capacités fonctionnelles, non-fonctionnelles et les caractéristiques d'un service. Lors de la description d'un service, les propriétés de service, les capacités et les contraintes doivent tous être pris en compte.

Une description doit aussi permettre de décrire les différents acteurs, leurs relations et leurs interactions avec leurs services [19].

II.2.2 La découverte de service :

La découverte est la localisation automatique des services répondant à une requête utilisateur [20].

La découverte de service désigne le processus par lequel les utilisateurs futurs d'un service recherchent manuellement ou semi automatiquement le service correspondant à leur besoin [21].

II.2.3 La composition des services :

La composition du service fait référence au processus qui consiste à combiner les fonctionnalités de plusieurs services, simples ou eux-mêmes composés au sein d'un même processus métier dans le but de répondre à des demandes complexes qu'un seul service ne pourrait satisfaire [22].

II.2.4 La sélection :

La sélection consiste à choisir, parmi les services découverts, ceux qui répondent au mieux aux exigences de l'utilisateur sur la base des besoins fonctionnels et/ou non fonctionnels, cette découverte est fondée sur les informations contenues dans les

descriptions des services publiés. Il est possible que plus d'un service réponde aux besoins. Par conséquent, le meilleur service doit être sélectionné [22].

II.2.5 Utilisation des ontologies :

En philosophie, l'ontologie est l'étude de l'être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe, et en informatique, elle est utilisée pour modéliser et représenter les connaissances dans un domaine spécifique.

Une ontologie définit formellement les termes employés pour décrire et représenter un domaine de connaissance, elle cherche à représenter le sens des concepts et les relations qui les lient.

Au sein d'une ontologie, les termes sont regroupés sous forme de concepts (ou classes sémantiques): ils définissent un groupe d'individus (instances) possédant des propriétés similaires [23].

II.2.6 Aspects de description :

Une bonne description doit couvrir les aspects techniques, opérationnels, commerciaux et sémantiques.

Les aspects technique et opérationnel décrivent l'interface et les fonctionnalités des services. L'aspect commercial décrit les acteurs qui interagissent avec un service, la tarification, les accords de niveau de service (SLA, Service Level Agreement), obligations juridiques, etc. L'aspect sémantique se distingue par l'utilisation de l'ontologie (exemple : OWL, Web Ontology Language) [24].

II.3 Partie 2 : les langages de description de services WSDL et USDL

Parmi les langages de description de service, USDL et WSDL font partie des plus utilisés, ici nous présentons un aperçu de ces deux langages et étudions les avantages et les inconvénients de l'un par rapport à l'autre.

II.3.1 Vue d'ensemble WSDL

II.3.1.1 Définition :

Le Web Service Description Language (WSDL) est une spécification définissant comment décrire les services web dans une syntaxe XML (eXtensible Markup Language) commune.

C'est un langage au format XML décrivant les services web comme un ensemble de nœuds opérants sur des messages contenant soit des informations orientées document soit des informations orientées procédure [25].

L'objectif principal de WSDL est de décrire les services comme un ensemble d'opérations et de message abstraits relié à des protocoles et serveurs réseau.

WSDL décrit quatre pièces essentielles de données [25] :

- Informations d'interfaces décrivant toutes les fonctions accessibles au public.
- Informations de type de données pour toutes les demandes de message et les réponses de messages.
- Informations sur le protocole de liaison de transport à utiliser.
- Information sur l'adresse pour localiser les services spécifiés.

II.3.1.2 Spécification :

Une description WSDL est un document XML qui commence par la balise <definitions> Et utilise les éléments suivants dans la description des services [26] (voir annexe A)

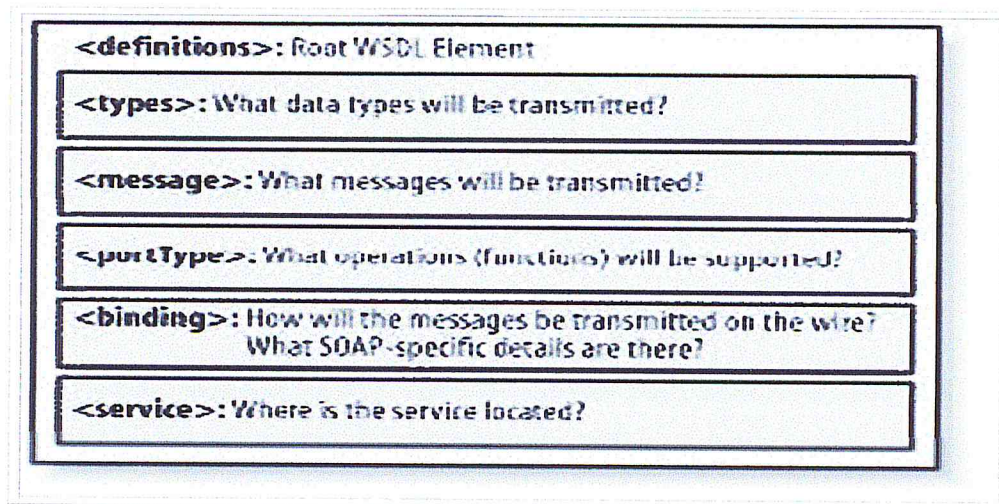


Figure 4: les balises WSDL [26]

II.3.1.3 Caractéristiques :

Le WSDL est caractérisé par deux niveaux [27] :

- Niveau abstrait :

Au niveau abstrait WSDL décrit un service web en termes de message qu'il envoie et qu'il reçoit. Les messages sont décrits indépendamment d'un format de fil spécifique en utilisant un système de types (XML schéma).

- Niveau concret :

A un niveau concret, une liaison spécifie les détails du transport et du format de fil pour une ou plusieurs interfaces. Le (end points) point final associe une adresse de réseau à une liaison. Enfin un groupe de services regroupe des extrémités qui implémentent une interface commune.

II.3.1.4 Avantages et Inconvénients :

➤ **Avantage :**

- WSDL fournit un langage commun pour décrire les services et une plateforme pour l'intégration automatique.
- WSDL est indépendant du langage et de la plateforme.

- WSDL est extensible afin de permettre la description de nœuds et de messages sans se soucier du format des messages et des protocoles réseaux utilisées lors de la communication [25].

➤ **Inconvénients :**

WSDL ne résout pas tous les problèmes ;

- Publication et découverte avec une description de haut niveau d'abstraction (UDDI Universal Description Discovery and Integration).
- Difficile de faire le lien entre l'aspect technique (WSDL) et l'aspect haut niveau (UDDI).
- Il impose de nombreuses contraintes sémantiques au-delà de la conformité structurelle à un ensemble d'informations XML.
- WSDL est assez verbeux.
- WSDL ne couvre que l'aspect technique.

II.3.2 Vue d'ensemble USDL :

II.3.2.1 Définition :

Le langage de description de service unifié (USDL) est un langage de description des services Internet à partir de trois points de vue différents (technique, commercial et opérationnel).

L'aspect technique est couvert par les modules Technique et Fondation d'USDL. L'aspect opérationnel est couverte par les modules Service, fonctionnel, et Fondation d'USDL. L'aspect commercial est décrit par les modules Participant, interaction, prix, juridique, niveau de service, et Fondation d'USDL [28]. .

II.3.2.2 Spécification :

Les spécifications USDL sont composées d'un schéma et d'une API (Integrated Programming Application). Le modèle de base utilisé par USDL est défini dans un schéma XML, car il offre une vue de plate-forme neutre des données

et permet de décrire les relations de manière simple et naturelle comparé à d'autres formalismes tels que OWL (Web Ontology Language).

Il prend en charge les types de données riches et peut facilement valider les instances en utilisant des schémas.

Le schéma USDL définit trois groupes principaux d'informations qui fournissent des descriptions qu'un consommateur peut utiliser pour découvrir, sélectionner, invoquer des services et avoir une vue sur le comportement des services au moment de l'exécution. Ces trois groupes sont les pôles d'activité, opérationnels et techniques. Tandis que le méta modèle est un modèle conceptuel, un modèle physique concret a été créé en utilisant XML Schéma [29].

II.3.2.3 Conception d'USDL :

USDL s'appuie sur, et reconnaît les normes pour les efforts techniques de description informatique pour des services tels que WSDL. Cependant, USDL ajoute des informations commerciales et opérationnelles. Pour ce faire, USDL définit des modules UML normatifs (via Eclipse Modeling Framework (EMF)) pour capturer les «données de base» d'un service. Cela inclut des modules normatifs, c'est-à-dire des modèles de classe pour les prix, les aspects juridiques, fonctionnels, les participants, l'interaction et les SLA. Par conséquent, les services manuels et informatiques peuvent être décrits avec USDL [28].

USDL est divisé en plusieurs paquets (selon la terminologie UML). Chaque paquet représente un « module » USDL « » et contient un seul modèle de classe. La figure 5 représente les différents modules USDL [24], [30], [48] :

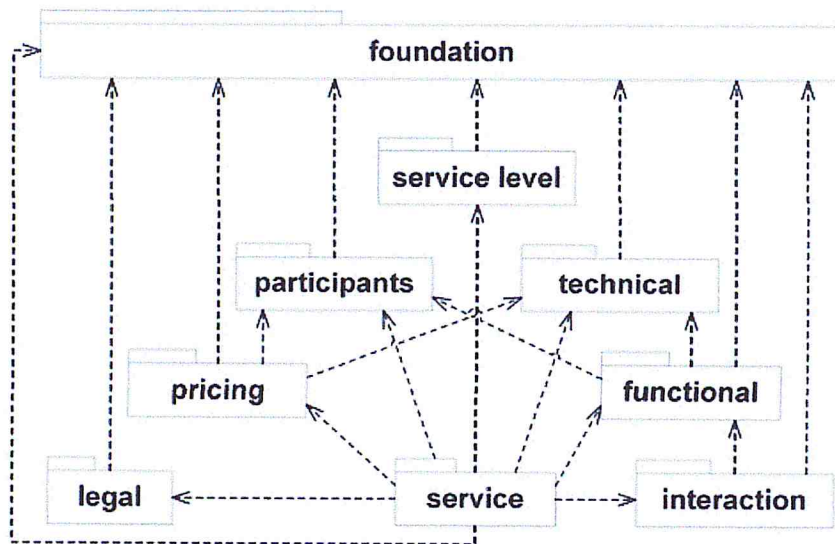


Figure 5: les modules USDL [30]

- Module de service: (service)

Ce module décrit les informations générales d'un service, comme le titre, la description, etc.

- Module fonctionnel: (functional)

Ce module contient les informations relatives aux capacités de service, les paramètres d'entrée-sortie et les contraintes.

- Module technique: (technical)

Le module technique soutient une manière commune de décrire les interfaces techniques des services. À la suite de services Web sémantique, le Module technique sert l'association sémantique entre les descriptions techniques d'interface et les éléments d'USDL.

- Module Participant: (Participants)

Ce module décrit les organisations participantes, les personnes et leur rôle dans la réalisation des services.

- Module d'interaction: (Interaction)

Ce module concerne l'aspect comportemental du service. Il décrit les acteurs et les interactions entre eux.

- Module de prix: (Pricing)

Ce module concerne la charge des services mutuellement compris par ceux qui possèdent ou offrent des services et ceux qui les consomment. USDL a une structure hiérarchique pour la tarification des services.

- Module de niveau de service: (Service Level)

Il est responsable de la qualité de service. Il spécifie le niveau d'entente de services tels que la disponibilité, la performance, le temps d'exécution de service, la sécurité, etc.

- Module juridique: (Legal)

Il contient les informations sur les conditions, les licences et les droits des utilisateurs pour les parties participantes.

- Module Fondation: (Foundation)

Ce module fournit un ensemble de propriétés communes telles que le temps, le lieu, l'organisation, etc. Ces propriétés sont utilisées par tous les modules.

II.3.2.4 Avantages et Inconvénients :

➤ Avantage :

- USDL est indépendant de la plateforme.
- USDL couvre les aspects techniques, opérationnels et commerciaux.

➤ Inconvénients :

- USDL ne couvre pas l'aspect sémantique.
- USDL ne tient pas compte de la sécurité.

II.3.3 WSDL vs USDL :

Le langage WSDL était développé pour décrire les détails techniques de la façon dont un service web peut être accédé et invoqué à distance sur le Web.

Il détaille les exigences techniques telles que les adresses Internet, ports, noms de méthodes, arguments et types de données utilisés par un service Web.

L'emphase de WSDL est sur les aspects techniques et de mise en œuvre de prestations de service. WSDL a été conçu pour être utilisé par les ordinateurs par contre USDL à un objectif différent puisqu'il doit également être utilisé dans l'internet des services par les personnes et les organisations. L'internet des services a un ensemble différent des exigences de celles remplies avec WSDL.

Le Tableau ci-dessous (tableau 3) est un comparatif entre WSDL et USDL suivant les critères suivants : standard, représentation, aspects couverts et les plateformes couverts.

	Standardisé	Représentation	Aspect	Plateformes
WSDL	Oui	XML	Technique	Multiplateformes
USDL	Non	XML	Commerciaux Opérationnels Technique	Multiplateformes

Tableau 3 : comparatif WSDL USDL

II.4 Partie 3 : Travaux connexes

Plusieurs chercheurs se sont intéressés à la normalisation de la description des services cloud pour tous les types de services en utilisant différentes approches. Certains se sont basés sur des ontologies, d'autres sur des langages ou des modèles.

Nous présentons ici quelques travaux de recherche sur la description de service cloud.

II.4.1 Approches de service IAAS :

Nous présentons ici les travaux de recherche qui concernent le type de service IaaS en fonction de la technique utilisée.

II.4.1.1 Approches fondées sur les ontologies :

➤ COCOON (Cloud Computing Ontologie):

Les chercheurs dans [31] proposent une ontologie basée OWL pour décrire les services IaaS : le Cloud Computing Ontologie.

COCOON définit les concepts fonctionnels et non-fonctionnels, les attributs et les relations de services d'infrastructure.

Les auteurs présentent également un système, CloudRecommender qui met en œuvre cette ontologie de domaine dans un modèle relationnel. Ce système utilise des expressions SQL régulières pour les requêtes utilisateur correspondant aux descriptions de service.

COCOON facilite la description des services d'infrastructure de Cloud; et par des correspondances entre les descriptions des fournisseurs facilitent la découverte de services d'infrastructure en fonction de la fonctionnalité et la qualité des paramètres de service (QoS). Cependant, l'ontologie COCOON ne couvre pas encore les couches PaaS et SaaS.

II.4.1.2 Approches fondées sur les modèles :

➤ Projet de rassemblement des services IaaS :

Dans [32], les auteurs face aux problèmes de verrouillage du fournisseur, de la fiabilité et la conformité réglementaire pour la localité de données dans IaaS, mettent en place le projet Aggregating IaaS service.

IAASA (Infrastructure-as-a-Service-Aggregator) permet de rassembler des services IaaS de différents fournisseurs en utilisant une même interface commune c'est-à-dire fournir une interface commune aux fournisseurs pour la gestion des environnements.

Ils proposent un ensemble de modèle pour décrire le Meta modèle : Modèle d'abstraction qui prend en charge (le calcul, le stockage, le réseau et la machine virtuelle) ; Modèle de services (Service de gestion des ressources qui définit les opérations de gestion du cycle de vie, telles que la création, la suppression , et les opérations fonctionnelles particulières ; Virtual Service qui définit les opérations de la machine virtuel ; Service de Clone qui définit la création et la suppression des images et des calculs instantanés de stockage ,enfin le service de sécurité qui fournit des fonctionnalités permettant de générer des informations d'identification).

Ils ont aussi utilisé le Web2Exchange qui offre un environnement d'intégration des services à base de modèles pour faciliter le déploiement de services, la découverte et les intégrations.

A travers les méthodes utilisées nous en concluons que le projet de rassemblement de service ne couvre que les aspects technique et opérationnel et ne concerne que le type de service IaaS.

➤ **Le projet réservoir :**

Dans [33], les chercheurs ont proposé un modèle de cloud computing fédéré, l'architecture et la fonctionnalité en cours d'élaboration dans le projet du réservoir. Le modèle du réservoir traite explicitement l'évolutivité limité d'un cloud unique fournisseur, le manque d'interopérabilité entre les fournisseurs cloud et le manque de prise en charge intégré de BSM (Business Service Management) dans les offres de nuage courant.

L'approche de nuage fédérée démocratise l'offre de cloud computing en permettant aux petites et moyennes entreprises et les nouveaux arrivants de devenir des fournisseurs de cloud, cependant, la fourniture de services complexes à travers un réseau fédéré de centre de données disparates est une difficulté et problème non encore résolu.

Une fédération souffre de l'absence d'une langue standard pour décrire les offres des fournisseurs ce qui rend la découverte et la sélection difficile au sein d'une fédération.

Les chercheurs se sont focalisés sur l'aspect technique et commercial, et se sont intéressés aux modèles IAAS.

➤ **Le projet contrail :**

Le projet contrail propose une approche fédérée et intégrée de cloud. Les facteurs importants sont le choix open-source et les collaborations avec d'autres projets impliqués dans la recherche Cloud.

Le but de contrail est de minimiser la charge de l'utilisateur et d'accroître l'efficacité dans l'utilisation des plateformes cloud en effectuant à la fois une intégration verticale et horizontale. L'intégration verticale fournit une plate-forme unifiée pour les différents types de ressources et l'intégration horizontale fait abstraction des modèles d'interaction des différents fournisseurs de cloud.

En plus des fonctionnalités de sélection des ressources entre plusieurs fournisseurs et la gestion cohérente des ressources, la fédération Contrail coordonne le soutien SLA des fournisseurs de cloud. Son rôle consiste à installer, coordonner et appliquer un SLA global, éventuellement négocier les SLAs avec des fournisseurs pour le compte des utilisateurs d'où la possibilité d'une Gestion verticale SLA. La fédération de soutien doit surveiller le SLA réalisée par les fournisseurs, et de réagir à des violations SLA.

Les fédérations contrail permettent aux utilisateurs d'exploiter les ressources appartenant à différents fournisseurs cloud quel que soit le type de technologie des fournisseurs avec une interface homogène et sécurisé. Cependant, les chercheurs se sont focalisés sur l'aspect technique et commercial, et se sont intéressé aux modèles IAAS et PAAS.

➤ **Les travaux de [35]:**

Les chercheurs dans [35] ont défini un méta-modèle pour une application cloud qui est capable de capturer la syntaxe et la sémantique de certaines des applications cloud. Le méta-modèle proposé montre les principaux éléments de vocabulaire et de conception cloud, l'ensemble des règles de configuration et l'interprétation sémantique.

La solution proposé couvre les aspects opérationnels, technique, sémantique et néglige l'aspect commercial, de plus elle concerne seulement le SAAS.

➤ **Les travaux de [36] :**

Les auteurs dans [36] proposent un modèle fondé sur les contraintes pour mapper une application à un ensemble de ressources de Cloud, qui doivent être utilisés pour exécuter l'application, dans un environnement multifournisseur. Les contraintes peuvent concerner le matériel, les logiciel, le stockage, la sécurité, les données, la performance, les coûts, la conformité.

Les auteurs ont mis en place un prototype simple, en utilisant le modèle de découverte de ressources en fonction des contraintes proposées. Cependant, la solution proposée couvre uniquement l'aspect technique et ne concerne que l'IAAS.

II.4.1.3 Approches fondées sur les langages :

➤ **CloudML (Cloud Modeling Language) :**

Les auteurs dans [38] ont proposé le CloudML un langage basé sur XML utilisé pour des environnements D-Cloud (Cloud Distribuée).

Le D-Cloud, peut être considérée comme un paradigme qui est en mesure d'exploiter le potentiel de partage des ressources à travers des frontières géographiques et de fournir l'allocation liée à la latence des ressources aux tiers développeurs.

CloudML est un langage de modélisation qui permet de représenter différents niveaux d'abstraction, telles que les ressources informatiques et réseau, services de fournisseur, et les besoins des développeurs.

CloudML fournit une méthode à base de profils pour décrire les services du fournisseur, qui sont décrites par un schéma XML dont l'élément racine est le Type de service. Ce type a un attribut de version et une séquence de profils nœud et de liaison des éléments.

Avec CloudML il est possible d'utiliser des solutions et des API présentes dans l'écosystème XML pour garantir la correction de la syntaxe, la requête de documents, et d'autres installations ; par contre CloudML ne permet pas de décrire les règles d'échelle et n'est utilisé que pour les types de service IaaS ,aussi le CloudML ne permet de décrire que les aspects technique et opérationnel.

➤ **Cloud# :**

Les auteurs dans [39] proposent un langage de spécification, cloud# pour modéliser l'organisation du cloud de l'infrastructure en tant que service. Ils ont décrit une application cloud# en construisant une architecture : architecture du cloud trust, dans laquelle les modèles cloud# sont combinés avec attestation à distance pour fournir des services de confiance.

Le langage cloud# a plusieurs fonctionnalités syntaxiques uniques de modélisation cloud. D'abord, cloud# à la structure pour exprimer des unités de calcul, qui représentent d'autres machines virtuelles ou le nuage lui-même. Ensuite, cloud# peut exprimer différents niveaux de calcul de privilèges. Encore, cloud# permet un contrôle bidirectionnel programmable et le transfert de données entre les

unités de calcul de différents privilèges. et pour finir, cloud# peut modéliser les ressources physiques avec souplesse.

Cependant, Cette solution permet seulement la description de type IAAS .Aussi, les auteurs ne se focalisent que sur l'aspect technique. Ils négligent l'aspect commercial, sémantique et opérationnel.

II.4.2 Approches de services PAAS :

Nous présentons ici les travaux de recherche qui concernent le type de service PaaS en fonction de la technique utilisée.

II.4.2.1 Approches fondées sur les modèles :

➤ Le projet 4CAAST :

Le projet 4CaaS [40], vise à développer une solution PaaS permettant une définition, un marketing, un déploiement et une gestion des services et des applications basés sur le Cloud.

Les innovations principales 4CaaS concernent Blueprints et le CloudMarketplace. Cloud Blueprinting est une solution puissante qui fournit aux développeurs de nouvelle génération des méthodes et des outils leur permettant d'agrèger, configurer et déployer des charges applicatives basées sur des services virtuels sur des machines virtuelles et des pools de ressources sur le Cloud [41] .

L'approche Cloud Blueprinting présente une série de modèles Blueprints utilisés pour résumer et décrire les composants des applications basées sur le Cloud Blueprinting.

Pour mieux gérer les modèles Blueprints, le framework Blueprint entrelace plusieurs composants interdépendants [42] : un langage BDL (Blueprint Definition Language) un langage BlueLine Constraint Language (BCL) ; un langage BML (Blueprint Manipulation Language) et un langage Blueprint Query Language (BQL).

Aussi, En utilisant les plans 4CaaS, les fournisseurs d'applications peuvent choisir parmi différentes plates-formes et services pour exécuter leurs applications, y compris différents composants, de middleware et d'applications.

Le concept innovant du Blueprint 4CaaS, associé à la fonctionnalité marketplace et à la gestion intégrée des logiciels, services et ressources, permet la réification de nombreux modèles d'utilisation et modèles économiques.

En plus de provisionner automatiquement les ressources ,4CaaS ajoute la prise en charge des environnements hétérogènes, ce qui, faciliterait la commutation entre les composants alternatifs pour éviter les verrouillages de fournisseurs. Cependant 4CaaS couvre les aspects technique, opérationnel commercial et ne concerne que le PaaS.

➤ **SPL (Software Product Lines): les travaux de [43]**

Les chercheurs Dans [43] proposent SPL (Software Product Lines); les SPL sont dédiés pour automatiser la configuration et la dérivation par exemple, la composition et/ou la génération de produits logiciel.

L'ingénierie SPL repose sur 4 grands aspects :

- Environnements Cloud comme modèles de fonction (la description, la gestion et la mise en œuvre des points communs et variabilités existant entre les membres de la même famille de produits logiciels) ;
- Le modèle de connaissance du cloud (il est utilisé pour définir des concepts de base, par exemple le langage, qui définit la langue dans laquelle l'application à déployer a été développée) ;
- Les fichiers de configuration et les scripts d'exécution en tant qu'actifs (Pour obtenir les configurations de fichiers et des scripts Shell, les acteurs se sont appuyés sur des modèles et des techniques de génération de code, ainsi que des mécanismes à la fusion)

- Menaces validité (Les principales préoccupations dans cette approche qui peut constituer une menace pour la validité sont les modèles sur lesquelles elle repose).

L'approche SPL repose sur une combinaison de logiciels et un modèle de domaine, ce qui permet au développeur d'automatiquement sélectionner un environnement de cloud qui correspond à un ensemble d'exigences et d'obtenir les fichiers de description et les scripts exécutables pour configurer l'environnement de cloud associés.

Ce modèle ne couvre que les aspects technique et opérationnel et les types de services PaaS et SaaS.

II.4.3 Approches de services SAAS:

Nous présentons ici les travaux de recherche qui concernent le type de service SaaS en fonction de la technique utilisée.

II.4.3.1 Approches fondées sur les ontologies :

➤ Les travaux de [44] et [45] :

Les auteurs dans [44], [45] ont développé un ensemble d'ontologies OWL pour saisir et exprimer les aspects commerciaux des offres des fournisseurs et des demandes des clients. Le but de leur travail est de soutenir un processus sémantique avancé d'appariement entre l'offre et la demande, proposer un modèle sémantique pour décrire les caractéristiques liées aux affaires des ressources du cloud. Une procédure de jumelage a également été conçue pour rechercher sémantiquement le domaine des offres et fournir au client une liste des offres les plus rentables. Enfin, un prototype du cadre a été mis en place. Des tests intensifs ont été effectués afin d'évaluer la viabilité du modèle proposé, et une étude de cas simple a été discutée.

Les ontologies Application, Support, mOSAIC et SLA sont des sous-domaines partageables pour les perspectives. La requête est le seul sous-domaine spécifique au client. L'offre et le marché sont deux sous-domaines spécifiques au fournisseur.

Une des forces de cette solution est qu'elle ait subi des tests intensifs pour évaluer sa viabilité et les auteurs ont utilisé un prototype pour une étude de cas. Cependant, La solution proposée couvre l'aspect commercial, sémantique Cela dit elle néglige l'aspect technique et ne concerne que le SAAS et PAAS.

II.4.3.2 Approches fondées sur les langages :

➤ Approche P2P :

Dans [19], Les auteurs ont entrepris une étude préliminaire sur la découverte de services cloud en adoptant un paradigme P2P non structuré. Ils ont mis au point un mécanisme pour le routage des demandes de service en couplant un certain nombre de composants: la réplication d'un saut, sémantique, réorganisation de la topologie et des nœuds.

Ils ont commencé en utilisant WSDL-S, Dans WSDL-S, l'expressivité du WSDL a été augmentée avec la sémantique en adoptant des concepts similaires à ceux de OWL-S. ils ont ensuite progressivement amélioré WSDL-S au moyen de l'extensibilité fournie par WSDL lui-même selon les exigences spécifiques pour les descriptions de services Cloud.

Ils partent du principe que tous les fournisseurs de services et demandeurs Cloud sont prêts pour partager leurs services Cloud (y compris les éventuels resquilleurs) et volontairement former un réseau P2P. À l'arrivée, chaque pair échange ses informations sur les services Cloud hébergés avec pairs voisins, c'est-à-dire, la réplication à un saut est effectuée en dehors.

L'avantage avec le P2P est qu'il ait été testé à travers plusieurs simulation .il couvre les aspects technique et sémantique au niveau du types de service SaaS

II.4.4 Approches de services génériques :

Nous présentons ici les travaux de recherche qui concernent les trois types de services (IaaS, SaaS, PaaS) en fonction de la technique utilisée.

II.4.4.1 Approches fondées sur les ontologies :

➤ Les travaux de [24] et [48]:

Les auteurs dans [24] et [48] ont proposé une description à base d'USDL. Ils ont représenté la description du cloud sémantiquement par une ontologie WSMO (Web Service Modeling Ontology). En effet ils ont utilisé WSMO pour représenter les services cloud sémantiquement et ils ont fait le mapping entre USDL et WSMO.

L'applicabilité de la solution proposée a été évalué en instanciant un service de type IAAS (Amazon EC2), PAAS (Google App Engine) et SAAS (Salesforce CRM).

La solution proposée traite donc tous les aspects (technique, opérationnel, commercial, sémantique) et tous les types de services (IAAS, SAAS, PAAS). Cependant les auteurs n'ont pas traité les problèmes de sécurité et ils n'ont pas défini de SLA entre les différentes couches.

II.4.4.2 Approches fondées sur les modèles :

➤ Cloud Service Description Model (CSDM) :

Les auteurs dans [49] proposent un modèle de description de service Cloud sémantique unifié, CSDM. Le CSDM proposé a été étendu à partir de la structure de base de l'USDL en définissant un module supplémentaire, le module de transaction, qui modélise le système de notation des services de cloud à partir de plusieurs aspects, tels que le risque, la confiance et la réputation. Les chercheurs ont également conçu un système d'annotation basé sur OWL pour enrichir l'expressivité sémantique de ce modèle.

la solution proposée atteint principalement cinq objectifs spécifiques au paradigme du Cloud, à savoir: rendre les services Cloud mesurables pour faciliter les

transactions de qualité de service; permettre la description de divers services Cloud avec différents modèles de livraison ou de déploiement; minimiser la complexité de la description des services Cloud intégrés; capturer les relations et les connexions des modules, identifier les interactions entre les composants, et permettre l'évaluation des services par rapport aux qualités basées sur les interactions; et prendre en charge à la fois les expressions syntaxiques et sémantiques de la description de service.

La solution proposée traite donc tous les aspects (technique, opérationnel, commercial, sémantique) et tous les types de services (IAAS, SAAS, PAAS).

➤ **Blueprinting :**

La solution (Blueprinting) proposée dans [50] est le résultat d'un ensemble de test sur certaines lacunes des offres de service actuel et certaines approches de gestion afin de trouver une solution globale sur la base de concept et des techniques de cloud Blueprinting.

Suite à ces tests les auteurs observent l'absence d'une approche générique pour l'intégration des offres SaaS à travers de multiples fournisseurs ; les offres PaaS sont limitées par les capacités qui sont disponibles par le fournisseur PaaS qui ne sont pas facilement extensible ; Au niveau de l'IaaS, les configurations croisées des machines virtuelles comprenant un service cloud spécifique ne sont actuellement possibles que dans les offres IaaS d'un seul fournisseur.

Compte tenu des lacunes des approches existantes dans le cloud computing indiqué ci-dessus, ils présentent alors le Blueprint concept, modèle Blueprint.

Le concept Blueprint est une description abstraite et uniforme des offres de services cloud inter-couches, un modèle Blueprint pour décrire les plans, et un cycle de vie Blueprint qui explique comment les plans sont utilisés pendant toutes les phases d'ingénierie d'un SBA (Service-based Application).

La solution Blueprinting concerne toutes les couches du cloud computing à savoir IaaS, PaaS, SaaS et couvre l'aspect technique et commercial.

➤ **MetaMORP(h) :**

Dans ce travail, les auteurs dans [51] décrivent une méthodologie MetaMORP(h) OSY, (Meta-Modeling of Mas Object-based with Real-time specification in Project Of complex Systems) basée sur des modèles multi-agents qui permet la description, la composition et la vérification des besoins des services basés sur le Cloud. La méthodologie utilise un profil de modélisation capable de décrire les services comme agents dans un environnement multi-agent et il est basé sur le modèle (Model Drivent Engineering) MDE technologie. La méthodologie proposée comprend un processus de vérification pour les exigences qui exploite les méthodes formelles pendant tout le cycle de vie des services.

MetaMORP(h) OSY, est basée sur la modélisation formelle et l'analyse des systèmes MAS. Les composants du Cloud pour chaque service sont modélisés en utilisant un profil UML étendu et conforme aux modèles MAS. Le modèle principal est ensuite analysé au moyen de modèles formels obtenus à partir du modèle UML avec des algorithmes de transformation de modèles.

Dans MetaMORP (h) OSY, la définition des propriétés est également assurée par la définition d'une ontologie commune qui décrit des composants et services.

Le modèle proposé couvre les aspects sémantiques et techniques, et concernent tous les types de service.

➤ **Les travaux de [46]:**

Dans [46], les auteurs ont présenté certains moyens sur la façon dont la publication, la découverte, la sélection et l'utilisation des ressources dans le cloud sont exposées. Les ressources dans le cloud computing ne se limitent pas au stockage des données et des serveurs, ils peuvent aussi être des systèmes complets, surtout distribués clusters. En outre, les clusters ne sont pas facilement découverts, utilisés ou même publiés et exposés. Ils ont alors, proposé et mis en œuvre une technologie CaaS (Cluster –as-a-Service) qui fait usage de l'information d'attribut

dynamique, cela grâce à l'utilisation du Framework RVWS (Ressources Via Web Services framework).

Le RVWS a été rendu possible grâce à la combinaison d'attributs dynamique, service web avec état WSRF (Web Service Resource Framework), le connecteur de ressources et le courtage.

Les caractéristiques de l'attribut dynamique couvrent des aspects non fonctionnels des services Web et de leurs ressources, Le processus sur la façon dont les attributs dynamiques des ressources sont présentés aux clients est fait par l'intermédiaire d'un document WSDL dynamique.

Comme le Framework RVWS publie des attributs dynamiques aux documents WSDL des services Web, deux formes de documents WSDL sont considérés: sans état et avec état. Un document WSDL sans état est un document WSDL qui a été créé à l'aide du schéma WSDL [47]. Un document WSDL avec état est un document WSDL sans état auquel des informations d'attribut dynamiques ont été ajouté via le Framework RVWS.

La technologie CaaS fonctionne en plus du Framework RVWS et se concentre sur la découverte et la sélection des clusters requis.

Ce travail combine un ensemble de modèle permettant la découverte, la sélection et l'utilisation des ressources d'une manière général c'est à dire tous les niveaux de couche sont concerné (IaaS, PaaS, SaaS), et ne concernent que l'aspect technique.

II.4.4.3 Approches fondées sur les langages :

➤ Les travaux de [52] :

Les chercheurs dans [52] identifient les informations nécessaires aux organisations pour la sélection des services cloud en utilisant le langage de description de service unifié (USDL) comme solution. Ils affinent alors chaque module de langue vers le domaine des services de cloud computing en intégrant les caractéristiques de service de cloud requis par les clients. Ainsi, cela aide les clients à structurer leur sélection de services ainsi que les fournisseurs de cloud dans la création de descriptions de services en fonction des besoins du client.

La solution proposée concerne tous les types de service cloud à savoir IaaS, PaaS, SaaS et couvre l'aspect technique, opérationnel et commercial.

❖ **Discussion :**

Après une étude des travaux de recherche sur la description des services cloud, nous regroupons ces différents travaux en fonction de la technique utilisée, le modèle de prestation, les domaines et les aspects couverts (tableau 2).

Travaux de recherche	Technique utilisée	Représentation proposée	Modèle de prestation	Domaine couvert	Aspect couvert
[31]	Ontologie OWL	Ontologie COCOON	IAAS	Description, Découverte	Technique, Semantique
[32]	Modèle	Meta-modèle	IAAS	Description, Découverte	Technique, Opérationnel
[33]	Modèle	Projet Réservoir	IAAS	Découverte, sélection	Technique, commercial
[34]	Modèle	Projet Contrail	IAAS, PAAS	Sélection	Technique, commercial
[35]	Modèle	Meta-modèle	SAAS	Découverte, sélection	Opérationnel, Technique, sémantique
[36]	Modèle	modèle de contraintes	IAAS	Description	Technique
[39]	Langage	Cloud #	IAAS	Description	Technique
[38]	Langage	CloudML	IAAS	Description,	Technique, opérationnel

[40]	modèle	Projet 4caast	PAAS	Description,	Technique, Commercial, Opérationnel
[44], [45]	Ontologie OWL	Ontologie d'affaire	SAAS, PAAS	Description	Commercial, sémantique
[43]	modèle	SPL	SAAS, PAAS	Composition, Description	Technique, opérationnel
[19]	Langage	P2P	SAAS	Description, Découverte,	Technique, Sémantique
[46]	modèle	CaaS	IAAS, SAAS, PAAS	Découverte, sélection	Technique,
50	modèle	Modèle de plan directeur	IAAS, SAAS, PAAS	Description	Technique, commercial
[51]	modèle	Metamorph	IAAS, SAAS, PAAS	Description, Composition	Technique, Sémantique
[52]	Langage	USDL	IAAS, SAAS, PAAS	Description, Sélection	Technique, opérationnel, Commercial
[24], [48]	Ontologie Et langue	WSMO, USDL	IAAS, SAAS, PAAS	Description	Technique, opérationnel sémantique, commercial
[49]	modèle et Langage	CSDM	IAAS, SAAS, PAAS	Description, Découverte	Technique, opérationnel sémantique, commercial

Tableau 2:Tableau récapitulatif des travaux connexes

D'après le tableau 2, nous pouvons remarquer que :

- ✓ six des travaux que nous avons étudié concernent uniquement les services IAAS.
- ✓ Deux travaux se focalisent sur les services de type SAAS
- ✓ Seul un des travaux se focalise sur les services de type PAAS,
- ✓ Deux travaux concernent les types de services PAAS et SAAS,
- ✓ Un seul se focalise sur les types de service (PAAS ; IAAS),
- ✓ Six ont travaillé sur les trois types de services (IAAS ; PAAS;SAAS),

Nous remarquons que la plupart des chercheurs ont travaillé sur le modèle de prestation IAAS. Les techniques les plus utilisées demeurent les modèles et langages et les travaux couvrent surtout la description.

Nous concluons également que la plupart des travaux étudiés se focalisent sur l'aspect technique, sur dix-huit travaux de recherche seuls deux travaux traitent tous les aspects à savoir les travaux de [24-48] et [49].

Cependant, aucune de ces solutions n'a été adoptée par la communauté ainsi le problème de standardisation demeure non résolu.

II.5 Partie 4 : Exigences pour une description de services Cloud

Les chercheurs dans [48], [30], [53] et [37] ont proposé un certain nombre de règles nécessaires pour une description générique de service à base d'USDL représenté dans le tableau ci-dessous :

Exigences	Explications
R1	Soutenir différents modèles de prestation de services
R2	Prise en charge de différents modèles de déploiement de service
R3	Soutenir l'aspect sémantique
R4	Mesure des attributs de service
R5	Soutenir la visibilité et le mécanisme de contrôle
R6	Décrire différents acteurs
R7	Définir un profil d'acteur dynamique
R8	Bonne définition de la requête de l'utilisateur
R9	Adaptation des changements de contexte
R10	Définir les contraintes d'environnement
R11	Couvrir les problèmes de sécurité
R12	Bien définir le SLA entre les trois couches de cloud
R13	Soutenir la réutilisabilité du service par la composition du service

Tableau 3:Tableau des exigences pour une bonne description

Détails des exigences:

- R1 : Soutenir différents modèles de prestation de services

Au départ USDL n'a pas été conçu pour décrire les services Cloud, mais plutôt pour décrire les services Internet. [53] et [30] ont adopté l'USDL pour représenter les services Cloud et les redéfinir en tenant compte des différents modèles de prestation dans le Cloud Computing pour cela, il a ajouté un attribut `delivery-model` dans la classe `Service` du module `Service`.

- R2 : Prise en charge de différents modèles de déploiement de service

Les services cloud sont déployés en fonction de différents modèles de déploiement (public, privé, communautaire et hybride). Cependant, l'USDL est incapable de saisir la différence entre ces modèles. [30] et [37] ont traité cette limitation en ajoutant un attribut `deployment-model` dans la classe `Service` du module `Service`.

- R3 : Soutenir l'aspect sémantique

L'aspect sémantique améliore le résultat de la découverte de service et il en va de même pour la composition du service. Il aide à assurer un meilleur résultat des services qui répondent aux besoins des utilisateurs. Il permet d'automatiser les tâches de utilisation des services, tels que: publication, découverte et composition. Il se distingue par l'utilisation de l'ontologie (OWL, WSMO, etc.), l'USDL est considéré comme un langage syntaxique et ne peut pas soutenir l'aspect sémantique. [30] a résolu ce problème en utilisant une ontologie OWL pour annoter le modèle CSDM.

R4 : Mesure des attributs de service

Les mécanismes d'évaluation sont très importants dans l'environnement Cloud Computing, car ils aident les consommateurs Cloud à choisir et à sélectionner des services. Aussi, l'évaluation des services permettent au fournisseur de services de comparer ses services à ceux qui sont disponibles sur le marché. Les mécanismes de mesure sont également utilisés pour détecter les violations SLA. Parmi les œuvres existantes, nous mentionnons que [30] a adressé cette limitation en ajoutant un nouveau module à l'USDL appelé le module de Transaction. Ce module capture les concepts utilisés pour mesurer et évaluer les facteurs pertinents pour les transactions de service.

- R5 : Soutenir la visibilité et le mécanisme de contrôle

Les modèles de déploiement offrent différents niveaux de visibilité et de contrôle pour les consommateurs. En d'autres termes, les ressources Cloud sont accessibles uniquement aux consommateurs qui ont le droit d'accéder au service. [30] a ajouté

deux Attributs le contrôle et la visibilité à la classe Resource du module Fondation afin de résoudre cette limitation. Ces attributs ne montrent que les ressources disponibles à un consommateur approprié en fonction de ses besoins. La nouvelle description du service Cloud devrait tenir compte de ces caractéristiques.

- R6 : Décrire différents acteurs

Selon [52], le Cloud Computing comprend cinq acteurs: consommateur, fournisseur, broker, auditeur et opérateur. Ainsi, la description du service Cloud devrait décrire ces différents acteurs, leurs relations et leurs interactions avec leurs services. [30] a ajouté au modèle CSDM un nouvel acteur nommé Broker via la classe Broker qui gère les relations dans le module Participant. Le broker est un acteur responsable de la composition de services.

- R7 : Définir un profil d'acteur dynamique

Les consommateurs ont différents niveaux de besoins, de compétences et de préférences qui diffèrent d'un consommateur à un autre. Chaque acteur a un profil le décrivant. Ce dernier peut être statique ou dynamique. Le profil statique présente les informations définies une fois pour toutes telles que le nom du consommateur, la date de naissance, etc. Ce profil est décrit en USDL par la classe ContactProfile du module Fondation. Cependant, un profil dynamique contient des informations telles que des objectifs, des préférences / désirs, des compétences, des contraintes, etc. [30] intègre des compétences dans le module Participant pour décrire les compétences techniques requises pour que le consommateur puisse utiliser les services Cloud. Ce profil dynamique améliore la découverte de services grâce à la sélection des meilleurs services qui répondent aux demandes des consommateurs.

- R8 : Bonne définition de l'utilisateur

La requête de l'utilisateur est l'élément principal de la découverte de service. Il peut être exprimé par l'utilisateur sous une forme textuelle ou dans un formulaire rempli afin de définir les fonctionnalités de service à appeler. Meilleure est la définition de la demande, meilleur est le résultat de la découverte de service. USDL ne répond pas à la demande de l'utilisateur final et il n'y a pas de travail de recherche

qui a abordé ce problème. La description du service Cloud proposé devrait décrire les demandes des consommateurs pour faciliter la découverte du service. [48] a résolu ce problème en ajoutant la classe ConsumerRequest dans le module Participant.

- R9 : Adaptation des changements de contexte

Le service existe dans un environnement dynamique qui peut être modifié en raison de divers facteurs tels que: service défaillance, dégradation de la qualité du service, changement des besoins des utilisateurs ou indisponibilité des services en raison d'un problème de réseau. Ces facteurs externes peuvent être déclenchés de manière imprévue. USDL est incapable de décrire les changements dynamiques dans un environnement. [48] a résolu ce problème en ajoutant la classe contexte dans le module foundation.

- R10 : Définir les contraintes d'environnement

Le service Cloud existe dans un environnement variable. Son utilisation est limitée par plusieurs contraintes environnementales, ce qui peut être des contraintes de temps ou d'espace. [48] a résolu ce problème en ajoutant la classe location dans le module foundation.

- R11 : Couvrir les problèmes de sécurité

La sécurité est un aspect non fonctionnel qui est essentiel pour déterminer la qualité du service dans le Cloud. USDL traite la sécurité dans le module service level afin d'assurer le niveau de sécurité pour chaque service [48].

- R12 : Bien définir le SLA entre les trois couches de cloud

Après la sélection des services, il est nécessaire de définir différents éléments contractuels (définition de service fournie, garanties et pénalités). Les clauses de pénalité SLA sont déclenchées et exécutées en cas de violation. Le modèle de description de service devrait: bien définir le contrat SLA; bien décrire les pénalités; et bien définir un contrat SLA entre différentes couches Cloud (SaaS, PaaS et IaaS) afin d'établir l'interopérabilité entre les différents services des trois couches et, par

conséquent, l'interopérabilité entre les fournisseurs Cloud, ce qui permet d'éviter le problème de verrouillage du fournisseur [48].

- R13 : Soutenir la réutilisabilité du service par la composition du service

Les développeurs sont en mesure de réutiliser les composants logiciels existants afin d'éviter les conflits de développement et gagner du temps. Cette tâche est caractérisée en composant des services existants pour obtenir un nouveau service. Dans le Cloud Computing, un broker est l'entité responsable de la gestion de la composition du service. Comme solution, [30] a ajouté la classe Broker dans le module Participant.

II.6 Conclusion

Dans ce chapitre nous avons donné un aperçu des éléments de base d'une description, suivi d'une étude comparative entre les langages de descriptions à USDL et WSDL qui nous a permis de voir l'avantage d'USDL par rapport à WSDL.

Nous avons aussi présenté certains travaux connexes sur la description de service cloud avec une analyse critique des différents travaux pour savoir ce qui existait déjà dans la littérature, ce qui manquait afin de donner une meilleure proposition, pour finir nous avons présenté certaines exigences des descriptions décrites par quelques chercheurs.

Dans le chapitre suivant, nous présenterons notre proposition de description de service cloud à base d'USDL couvrants les aspects techniques, commerciaux, sémantiques et opérationnels pour tous les types de service (IaaS, PaaS, SaaS).

III.1 Introduction

Après avoir étudié les solutions existantes pour la description des services Cloud dans le chapitre précédent, nous allons dans ce chapitre, présenter notre solution pour une description générique de service cloud à base d'USDL, l'objectif de ce chapitre est de prouver que nous pouvons décrire tous les aspects d'un service cloud en se basant sur les modules USDL par la conception d'une ontologie regroupant tous les concepts du cloud à l'aide des différents modules d'USDL. Nous avons utilisé OWL_S (Semantic Web Ontology Language) et l'outil « Protégé » pour proposer notre ontologie.

Pour commencer, nous détaillerons des exigences nouvelles que nous avons rajouté pour une meilleure description des services, qui sera suivi de la représentation des différents modules USDL dans notre ontologie, pour finir nous allons appliquer notre proposition sur trois études de cas : un service de type IAAS (Amazon Elastic Block Store, EBS), PAAS (Azure Cloud Service) puis SAAS (SharePoint Online).

❖ Démarche de travail :

La démarche de travail que nous avons suivi afin de réaliser notre projet correspond à l'XP-eXtreme Programming. L'eXtreme Programming (XP) est une méthode agile de gestion de projet informatique adaptée aux équipes réduites avec des besoins changeants [74].

Tout en mettant l'accent sur les bonnes pratiques de programmation, XP préconise un déroulement par itération courte et géré collectivement, avec une implication constante du client. Il en découle une redéfinition de la relation entre client et fournisseur, avec de surprenants résultats en termes de qualité de code, de délais et de satisfaction de la demande du client [73].

Les principes de l'eXtreme Programming sont basés sur :
 La Revue du code en permanence, faire les tests systématiquement avant chaque mise en œuvre, conception faite tout au long du projet, choisir toujours la solution la plus simple, évaluation des métaphores, Intégration des modifications, cycle de développement très rapides [75].

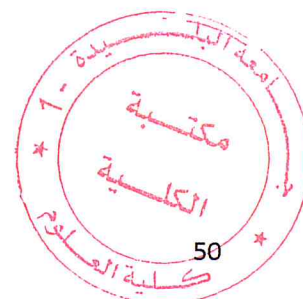
III.2 Les nouvelles exigences proposées:

Différents auteurs (voir chapitre II) ont proposés des exigences à respecter pour une bonne description de service de cloud computing.

Le tableau suivant représente les exigences supplémentaires que nous proposons pour une meilleure description de service cloud.

Exigences	Explication
R14	Une description cloud doit spécifier les responsabilités des deux parties (fournisseur et consommateur de service)
R15	Une description cloud doit spécifier la disponibilité du service fournie
R16	Une description cloud doit permettre au moins de faire une découverte
R17	Une description cloud doit pouvoir spécifier toutes les informations relatives au contrat entre le fournisseur et le consommateur de service.
R18	Une description cloud doit pouvoir spécifier le point de contact entre le fournisseur et le consommateur de service

Tableau 4: les nouvelles exigences



➤ **Détails des exigences :**

- R14 : Une description cloud doit spécifier les responsabilités des deux parties (fournisseur et consommateur de service).

Les services informatiques peuvent facilement subir des contrecoups, il est nécessaire de spécifier les responsabilités des deux partis en cas de problème afin d'éviter toute ambiguïté.

R15 : Une description cloud doit spécifier la disponibilité du service fournie.

L'engagement de disponibilité est l'un des premiers paramètres que le client regarde en comparant les offres. La disponibilité est exprimée en trois, quatre ou cinq neuf. Toute description doit permettre de savoir la disponibilité du service que le fournisseur s'engage à offrir.

R16 : Une description cloud doit permettre au moins de faire une découverte.

La finalité d'une description consiste à être utilisée dans un processus découverte, de sélection ou de composition, elle doit au moins permettre une découverte ou sans quoi elle n'est d'aucune utilité.

R17 : Une description cloud doit pouvoir spécifier toutes les informations relatives au contrat entre le fournisseur et le consommateur de service.

Le contrat est l'outil qui permet de circonscrire les droits et les obligations des prestataires de cloud computing, il doit contenir toutes les informations utiles de façon simple et facile à lire pour permettre aux clients de bien le comprendre avant tout engagement.

R18 : Une description cloud doit pouvoir spécifier le point de contact entre le fournisseur et le consommateur de service.

Les fournisseurs de services cloud offrent des services pour un nombre important de consommateurs, cependant ces différents clients peuvent vouloir contacter leur fournisseur à tout moment en cas de problème ou en cas de recherche d'information. Il serait intéressant qu'une description puisse fournir un contact qui permettra au client de pouvoir communiquer avec son fournisseur.

III.3 Conception :

Dans notre conception, nous abordons la limitation d'aspect sémantique R3 vu dans l'état de l'art (voir chapitre II). Pour cela, nous utiliserons une ontologie pour décrire les services de cloud computing et leurs caractéristiques sous une forme sémantique.

Nous avons utilisé l'éditeur OWL-S (Semantic Web Ontology Language) et Protégé pour la conception notre proposition.

➤ Protégé:

Protégé est un éditeur d'ontologies employé par les développeurs et des experts de domaine pour développer des systèmes basés sur les connaissances (Ontologies). Protégé n'est pas un outil spécialement dédié à OWL, mais un éditeur hautement extensible, capable de manipuler des formats très divers. Le support d'OWL, comme de nombreux autres formats, sont possible dans protégé grâce à un plugin dédié.

Protégé permet aussi de créer ou d'importer des ontologies écrites 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 [55].

➤ OWL-S:

OWL-S est une ontologie de concepts de services qui présente les ontologies pour décrire, d'une part, les concepts dans le domaine des services et d'autre part, les concepts génériques pour décrire les services eux-mêmes et la façon dont ils se rapportent aux ontologies de domaine (via les entrées et les sorties, les conditions préalables et les effets, etc.).

OWL-S organise une description de service en quatre domaines conceptuels: le modèle de processus, le profil, la mise à terre (échouage) et le service [56] :

❖ Le modèle de processus :

Le modèle de processus décrit comment un service effectue ses tâches. Il comprend des informations à propos des entrées, des sorties, les conditions préalables. Il se caractérise par trois types de processus : les processus composites, les processus simples et les processus atomiques.

Les processus atomiques sont essentiellement des boîtes noires des différentes fonctionnalités.

Les processus simples sont des descriptions de processus abstraits qui peuvent se rapporter à d'autres processus composites ou atomiques.

Les processus composites sont constitués d'un ensemble de processus atomiques.

❖ Le profil :

Le profil fournit une description générale du service, destiné à être publié et partagé pour faciliter la découverte de services. Les profils peuvent inclure les propriétés fonctionnelles (entrées, sorties, conditions et résultats) et les propriétés non fonctionnelles de service.

❖ **Une mise à terre (échouage) :**

L'échouage spécifie comment un service est appelé, en détaillant comment les processus atomiques dans la carte du modèle de processus d'un service sur un protocole de messagerie concrète. OWL-S prévoit différents types d'échouages à utiliser, mais le seul type développé à ce jour est la mise à la terre WSDL, qui permet à tout service web d'être marqué comme un service web sémantique en utilisant OWL-S.

❖ **Un service :**

Il lie simplement les autres parties ensemble dans une unité qui peut être publiée et invoqué. Il est important de comprendre que les différentes parties d'un service peuvent être réutilisées et connectées de différentes manières.

Le tableau suivant présente le mapping entre USDL et OWL_S (Tableau 2):

USDL	Ontologie
Module	Concept
Classe	Sous-concept
Dépendances entre modules	Relation entre concepts
Dépendances entre classes	Relation entre classes
Propriétés	Attributs
Données	Instance

Tableau 5: Mapping USDL Ontologie, [48]

USDL est constitué de neuf modules (Service, functional, technical, Interaction, Legal, Pricing, Participant, ServiceLevel, et Foundation). Nous allons donc définir neuf principaux concepts dans notre ontologie à savoir (ServiceModule, FunctionalModule, TechnicalModule, LegalModule, ServiceLevelModule, PricingModule, InteractionModule, ParticipantModule et FoundationModule).

III.3.1 Représentation ontologique du module Service :

Nous représentons le module Service par le concept ServiceModule et ces différentes classes par des sous concepts. Pour résoudre l'exigence R1, nous avons rajouté le sous concept CloudDeliveryModel pour représenter les différents modèles de livraisons (SaaS, PaaS, IaaS). Pour résoudre l'exigence R2 nous avons rajouté le sous concept CloudDeploymentModel pour représenter les différents modèles de déploiements (publique, hybride, privé, communautaire). Pour résoudre l'exigence R4, nous avons rajouté le sous concept CloudServiceEvaluation qui permet d'évaluer le service cloud à travers les sous concepts réputation, confiance, expérience, estimation des clients, et risque. Pour résoudre l'exigence R13, nous avons rajouté le sous concept CompositeService. La figure suivante est la représentation ontologique du module service :

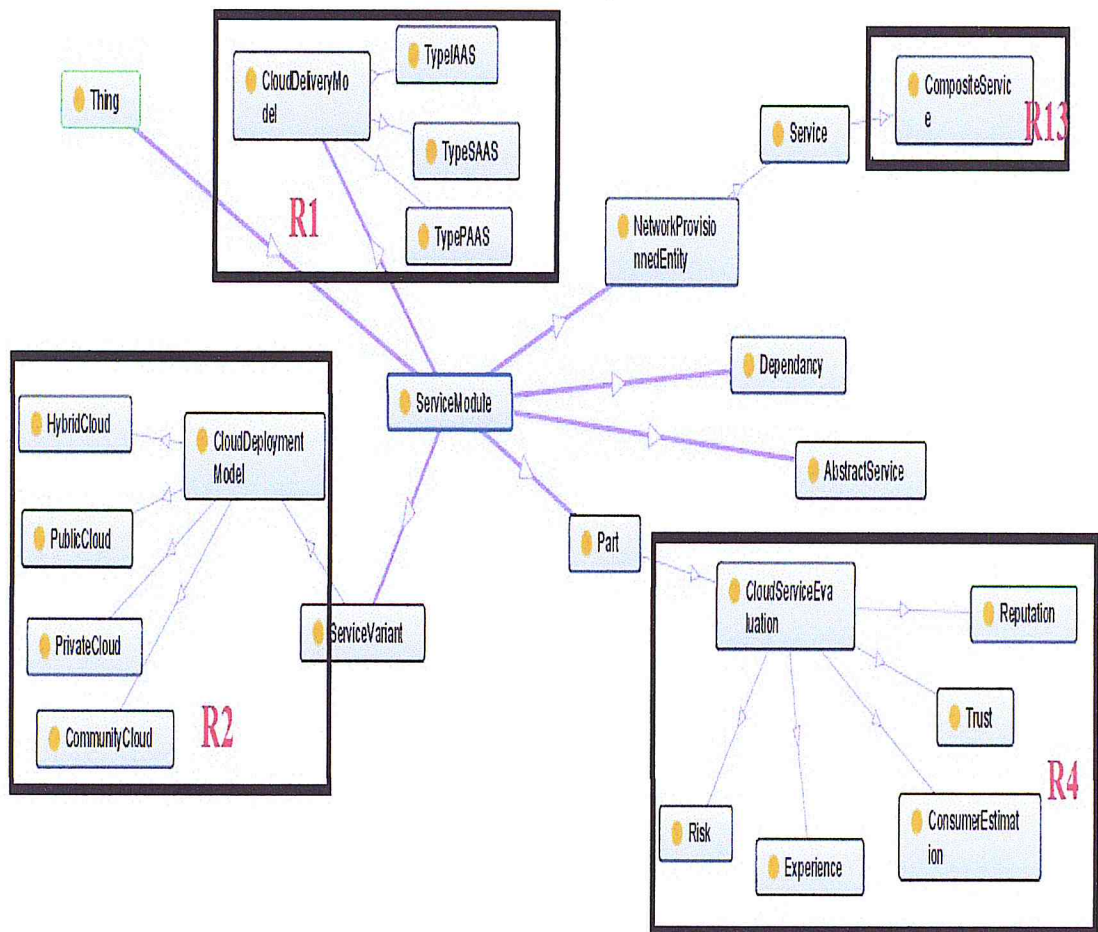


Figure 6: Représentation ontologique du module Service

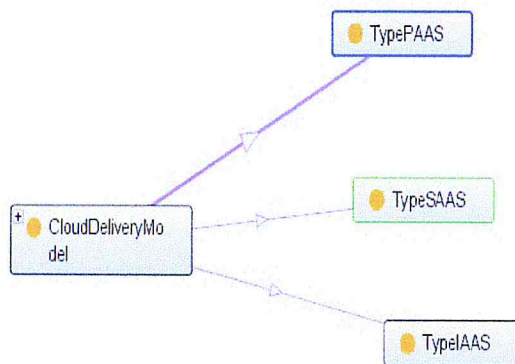


Figure 7: une partie de l'ontologie pour résoudre l'exigence R1

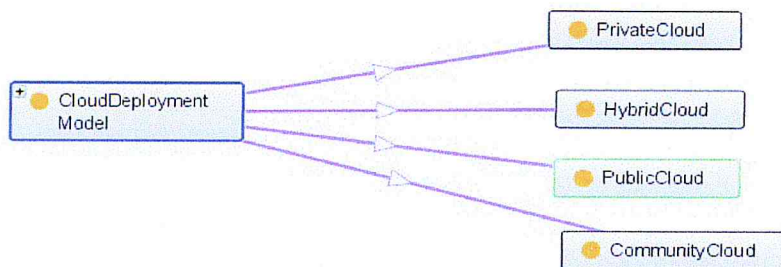


Figure 8: Une partie de l'ontologie pour résoudre l'exigence R2

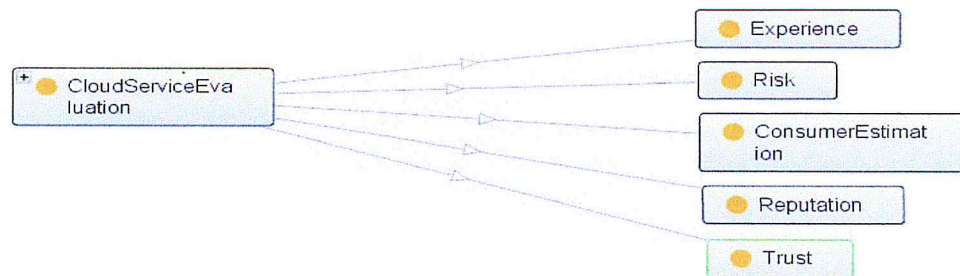


Figure 9: Une partie de l'ontologie pour résoudre l'exigence R4

III.3.2 Représentation ontologique du module Fonctional :

Nous avons représenté le module fonctional par le concept FunctionalModule et ses différentes classes par les sous concepts parameter, Function, **Fault** et FunctionalOption. La figure suivante est la représentation ontologique du module fonctional :

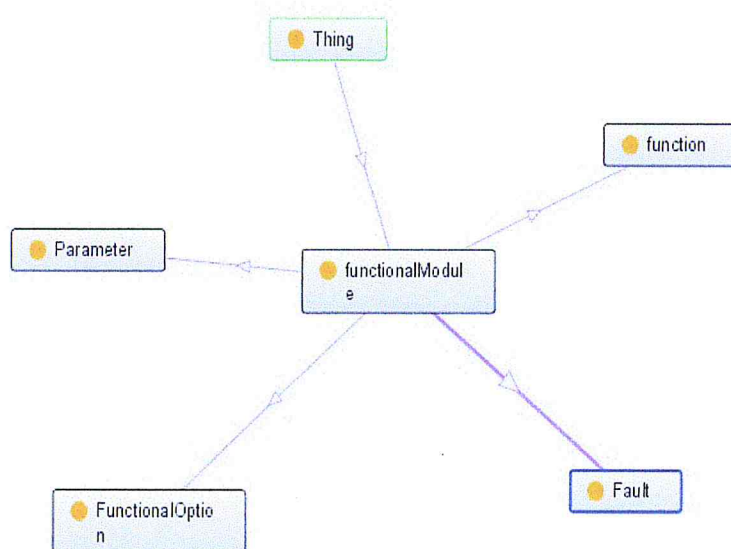


Figure 10: Représentation ontologique du module Fonctional

III.3.3 Représentation ontologique du module Technical :

Le module technical est représenté par le concept TechnicalModule dans notre ontologie et ses différentes classes par les sous concepts. La figure suivante est la représentation ontologique du module technical :

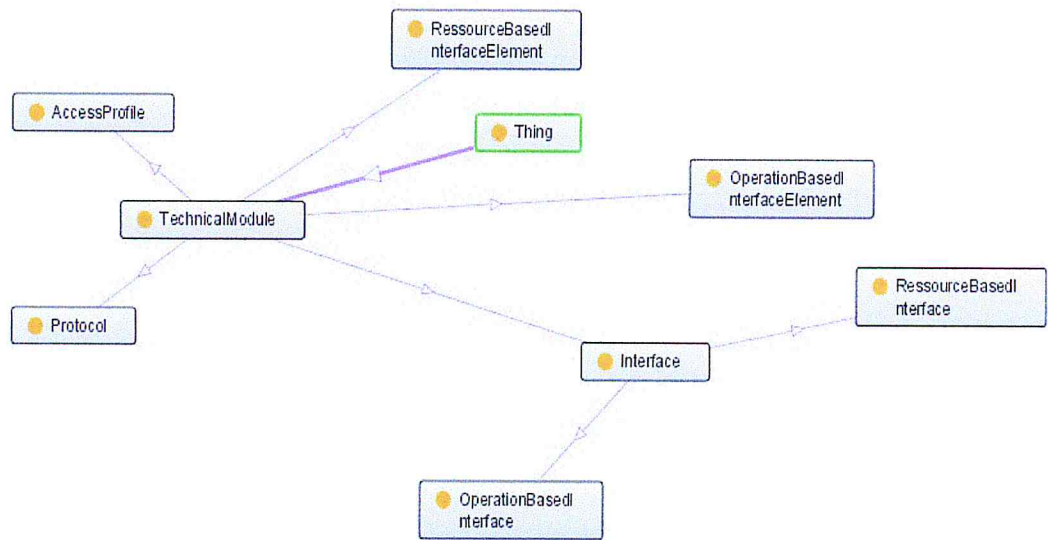


Figure 11: Représentation ontologique du module Technical

III.3.4 Représentation ontologique du module Participant :

Le module participant est représenté par le concept ParticipantModule et ses différentes classes par les sous concepts. Pour répondre à l'exigence R6 nous avons rajouté les sous concepts Creator, Publisher, Contributor, Auditor Carrier qui représentent les différents acteurs. Pour résoudre l'exigence R8, nous avons rajouté le sous concept ConsumerRequest (pour exprimer la requête du client) et pour résoudre l'exigence R7, nous avons rajouté le sous concept consumerProfile (Pour exprimer le profil du client). La figure suivante est la représentation ontologique de module Participant :

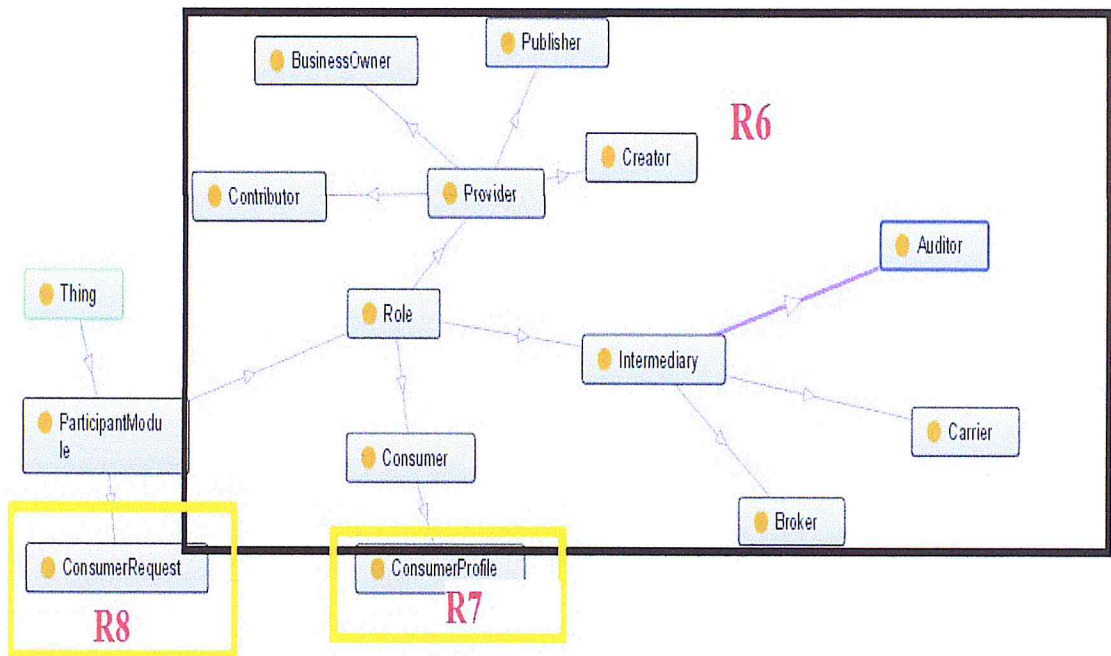


Figure 12: Représentation ontologique du module Participant

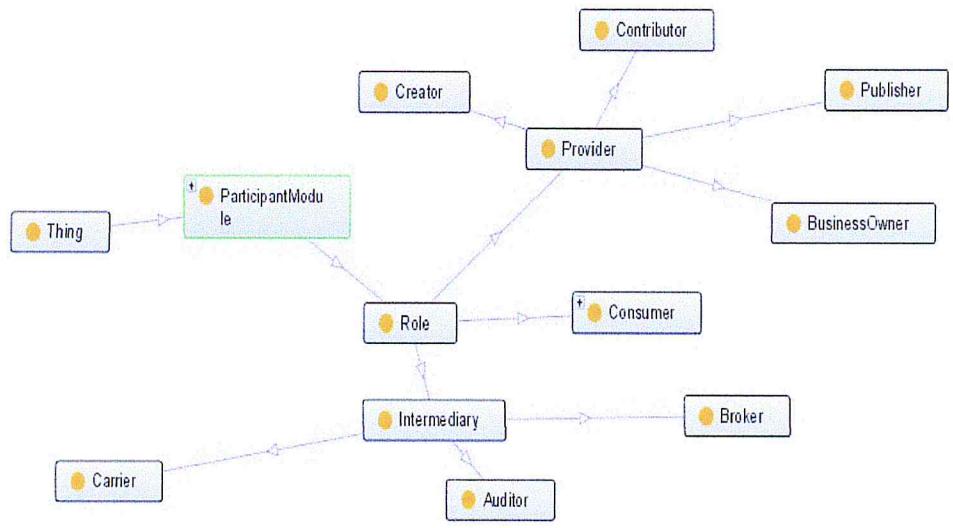


Figure 13: Une partie de l'ontologie pour résoudre l'exigence R6

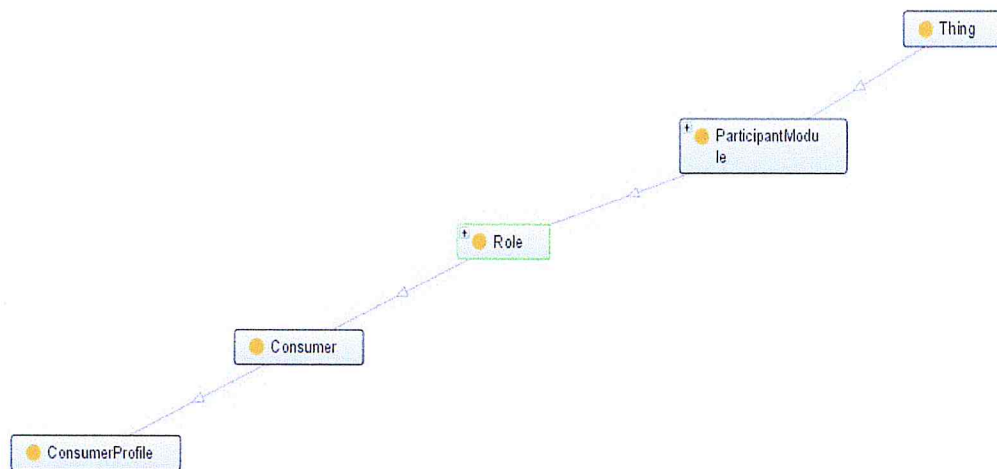


Figure 14: Une partie de l'ontologie pour résoudre l'exigence R7

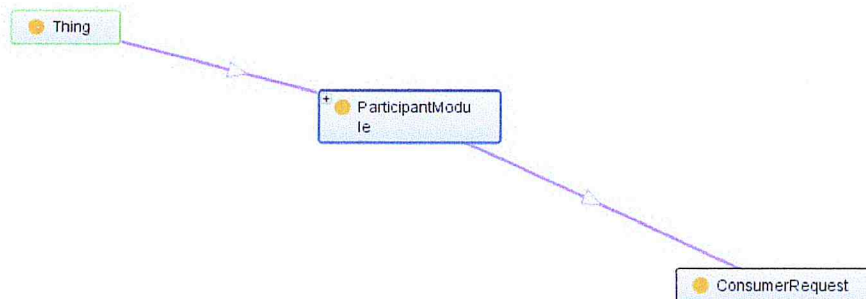


Figure 15: Une partie de l'ontologie pour résoudre l'exigence R8

III.3.5 Représentation ontologique du module Pricing :

Nous avons représenté le module prix par le concept PricingModule dans notre ontologie et ses différentes classes par les sous concepts. La figure suivante est la représentation ontologique du module Prix:

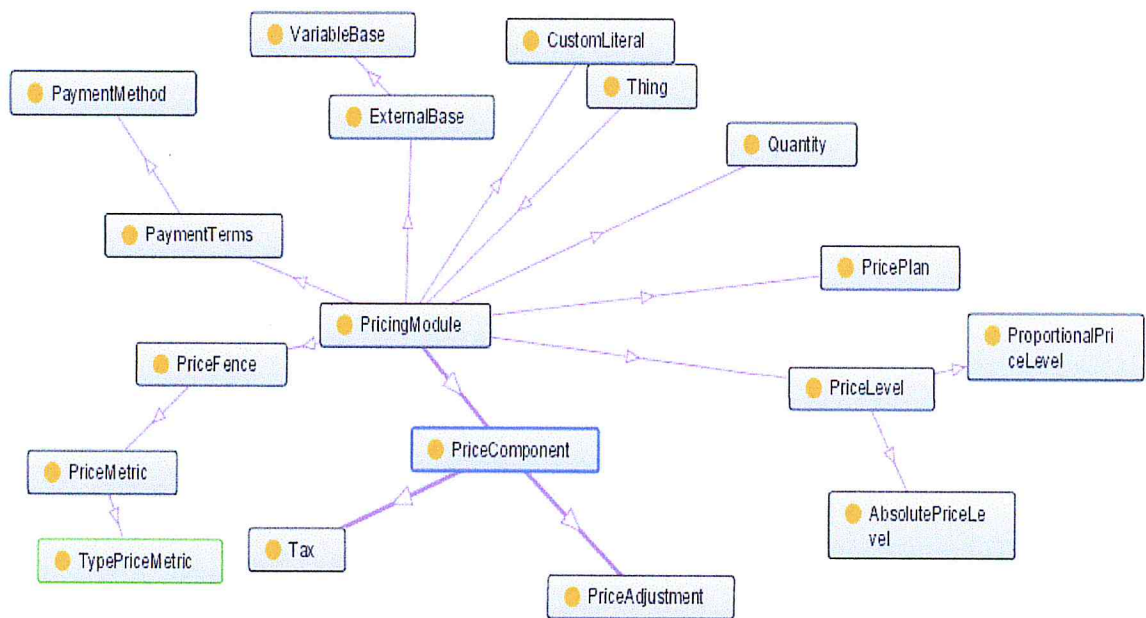


Figure 16: Représentation ontologique du module Pricing

III.3.6 Représentation ontologique du module Legal :

Nous représentons le module legal par le concept LegalModule dans notre ontologie et ses différentes classes par les sous concepts. Pour résoudre l'exigence R14, nous avons rajouté le sous concept ResponsibilityPerimeter (pour spécifier les responsabilités des deux partis), et SinglePointOfContact pour résoudre l'exigence R18 (pour spécifier un point de contact entre le client et le fournisseur). Pour répondre à l'exigence R17, nous avons rajouté en plus les sous concepts ContractLenght (Pour spécifier la durée du contrat), ConditionOfIntervention (pour spécifier les conditions d'interventions), Compensation (pour exprimer les conditions de dédommagement), RenewOption (pour exprimer les options de renouvellement du contrat), Penalty (pour exprimer les sanctions), UsedTerms (pour exprimer les conditions d'utilisations). La figure suivante est la représentation ontologique du module juridique :

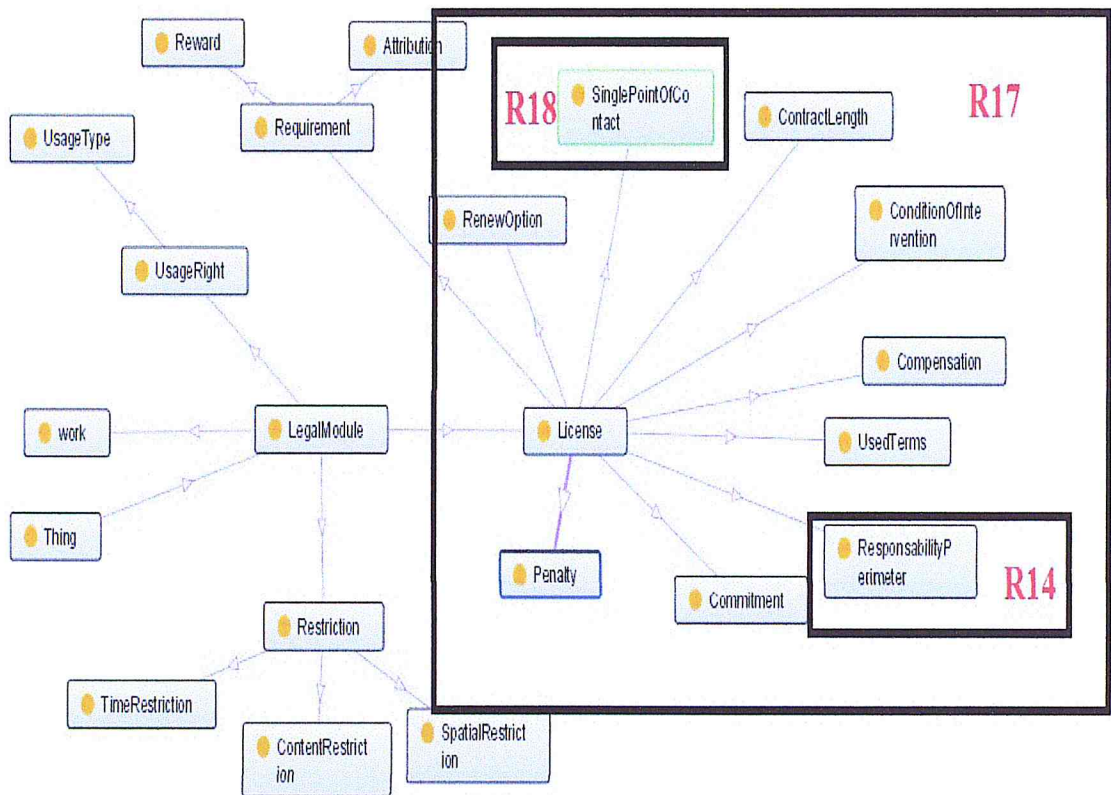


Figure 17: Représentation ontologique du module Legal

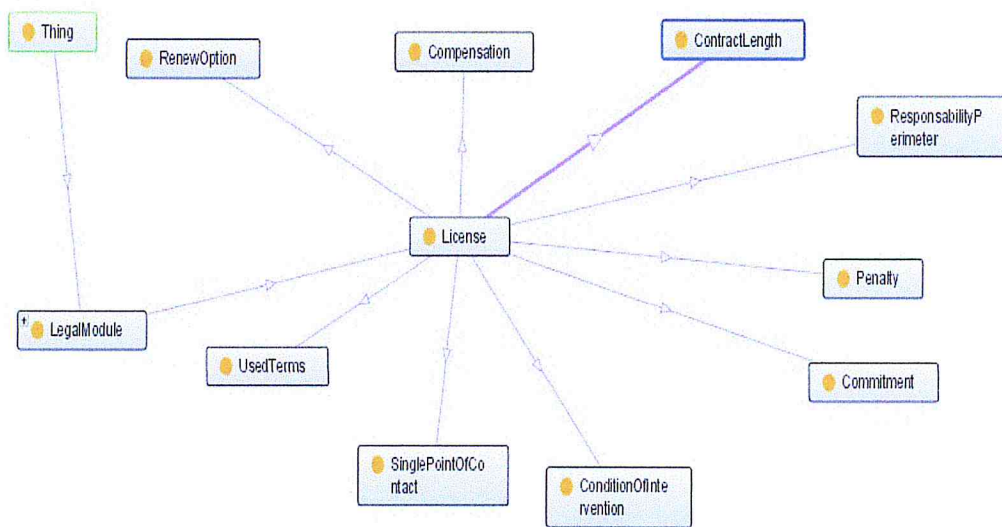


Figure 18: Une partie de l'ontologie pour résoudre l'exigence R17

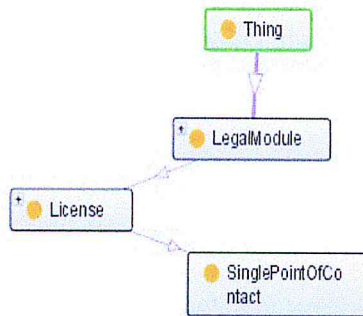


Figure 19: Une Partie de l'ontologie pour résoudre l'exigence R18

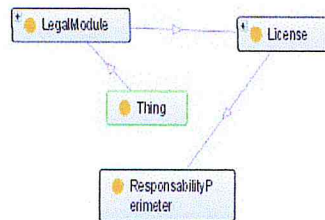


Figure 20: Une partie de l'ontologie pour résoudre l'exigence R14

III.3.7 Représentation ontologique du module Interaction :

Nous avons représenté le module interaction par le concept InteractionModule dans notre ontologie. La figure suivante est la représentation ontologique du module Interaction:

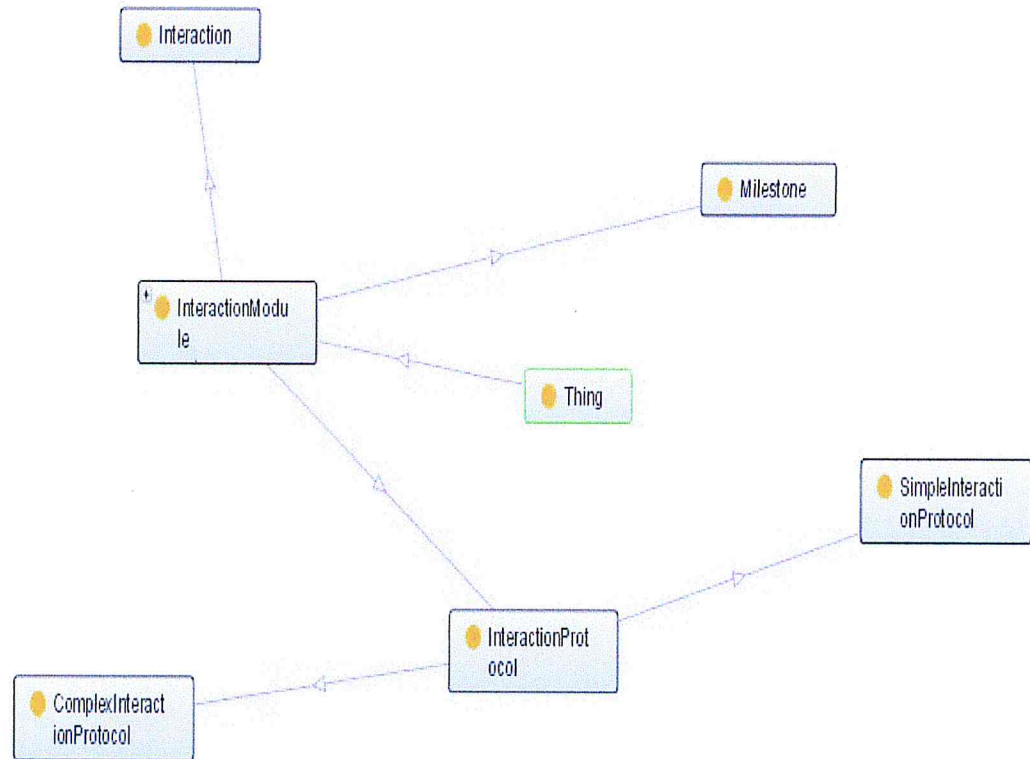


Figure 21: Représentation ontologique du module Interaction

III.3.8 Représentation ontologique du module Foundation :

Nous représentons le module foundation par le concept foundationModule dans notre ontologie et ses différentes classes par les sous concepts. La figure suivante est la représentation ontologique du module foundation :

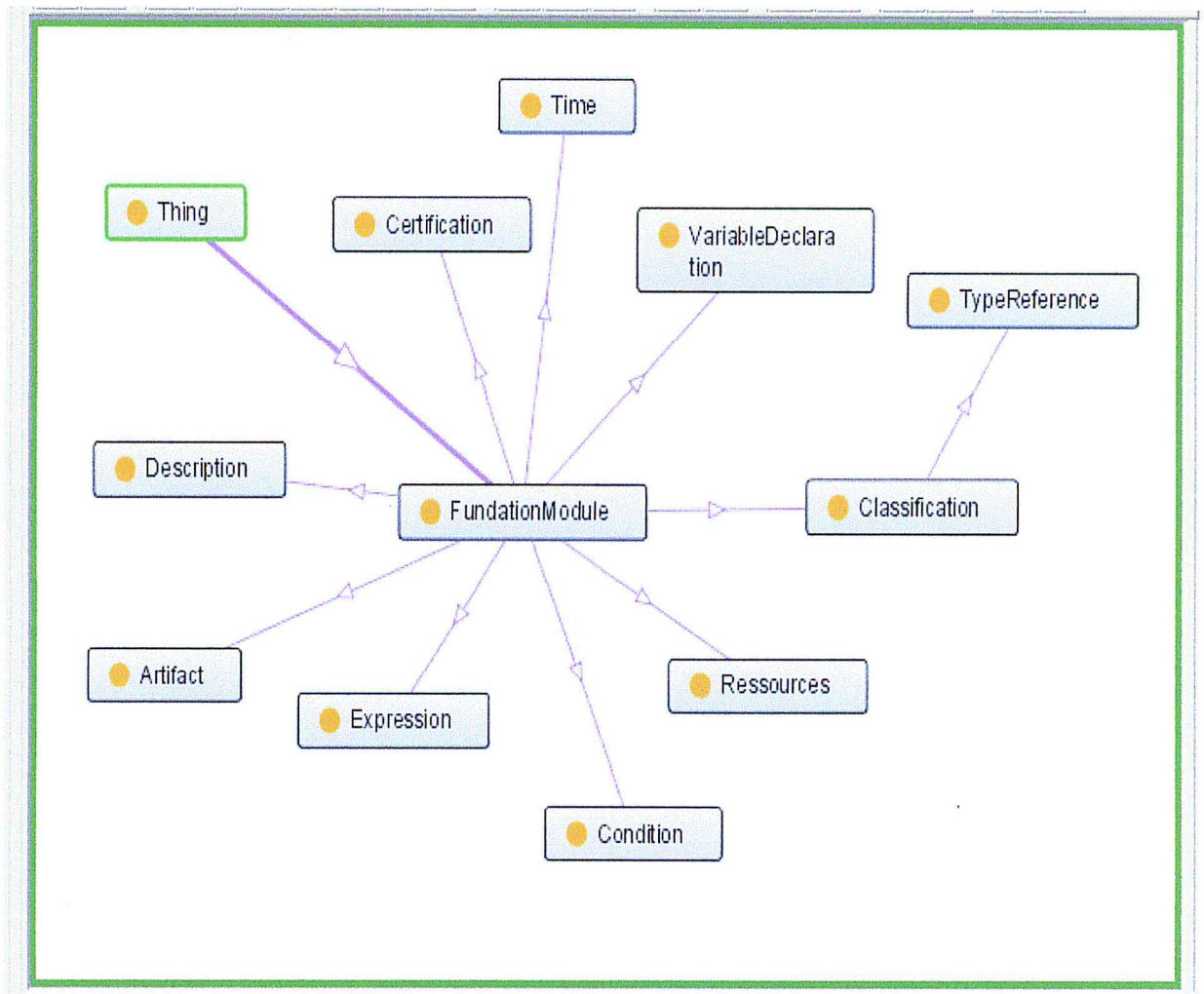


Figure 22: Représentation ontologique du module Foundation

III.3.9 Représentation ontologique du module Service Level:

D'abord, nous présentons un aperçu des éléments d'informations contenus dans les SLA en fonction du modèle de déploiement (SAAS, PAAS ou IAAS), (Voir annexe B).

Nous représentons le module Service Level par le concept ServiceLevelModule dans notre ontologie. Nous avons rajouté le sous concepts Disponibilité pour exprimer la disponibilité du service. Nous avons également rajouté les sous concepts GenericSLA, SLA_SAAS, SLA_PAAS et SLA_IAAS pour décrire les différents éléments contenus dans les SLA des 3 types de service.

La figure suivante est la représentation ontologique du module Service Level :

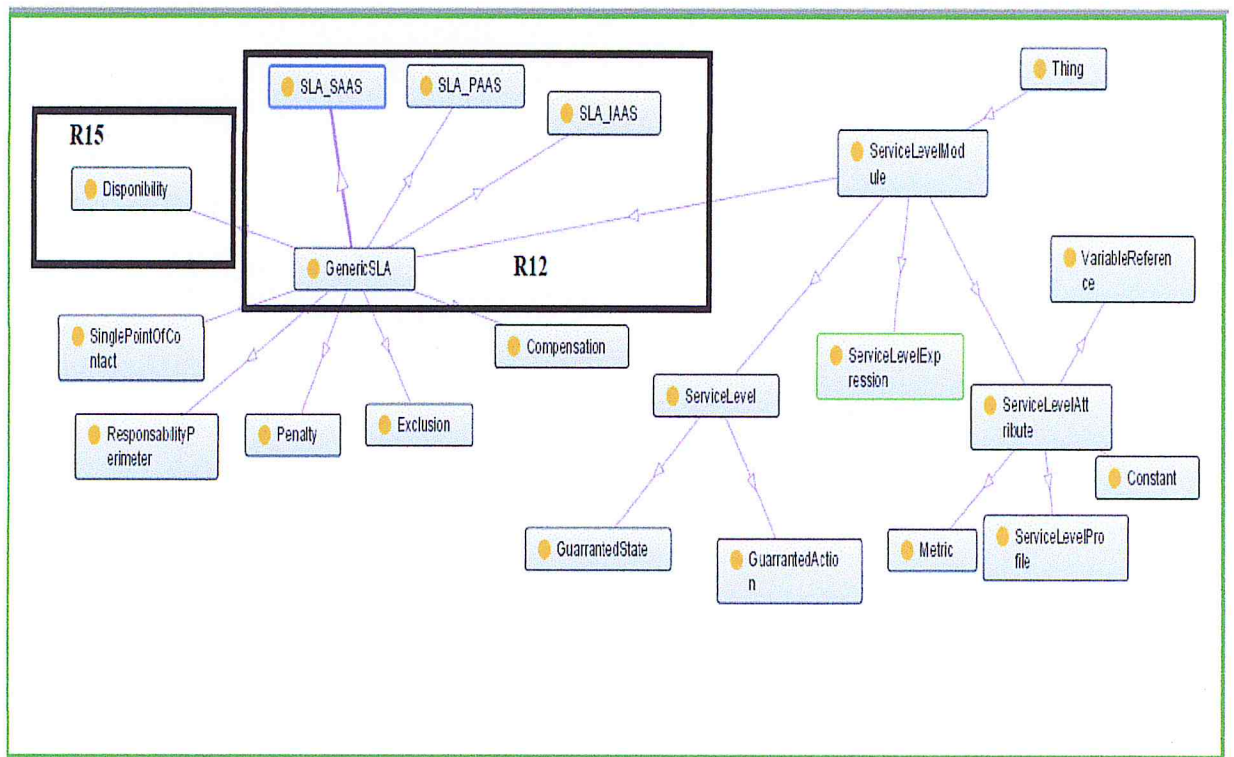


Figure 23: Représentation ontologique du module Service Level

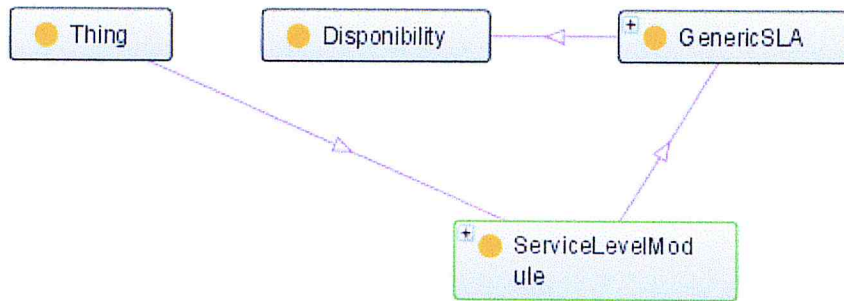


Figure 24 : Une partie de de l'ontologie pour résoudre l'exigence R15

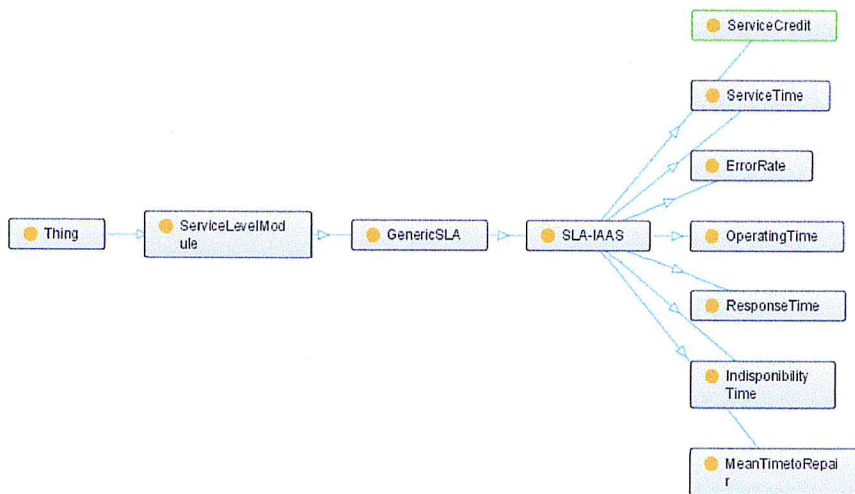
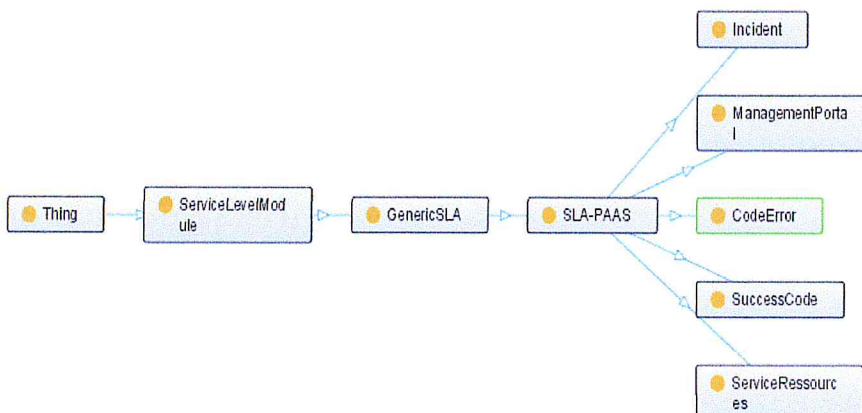


Figure 25: Une partie de l'ontologie pour résoudre l'exigence R12 SLA pour IaaS



dispositif de stockage pour les données nécessitant des mises à jour fréquentes et granulaires (Bases de données NoSQL et relationnelles à fort taux d'E/S).

Amazon EBS est un service de type IAAS déployé comme cloud public, permettant de [61], [64], [65], [67], [66], [67], [68]:

Créer des volumes de stockage et de les associer à des instances Amazon EC2 avec la fonctionnalité **Create**.

Placer les volumes EBS dans une zone de disponibilité spécifique dans laquelle ils sont automatiquement répliqués avec la fonctionnalité de **disponibilité**.

D'enregistrer des instantanés des volumes à un instant précis dans Amazon S3 avec la fonctionnalité **Snapshot Recording**.

Augmentez les capacités, ajustez les performances et modifiez le type de volumes EBS actifs de manière dynamique avec la fonctionnalité **Elastic Volumes**.

Amazon EBS Encryption offre un chiffrement aisé des volumes de données, des volumes de démarrage et des instantanés EBS, sans qu'il soit nécessaire de créer ni de gérer d'infrastructure de gestion sécurisée des clés. Identity and Access Management (**IAM**) permet de contrôler l'accès des volumes Amazon EBS.

Les instances optimisées pour EBS permettent aux instances EC2 d'utiliser pleinement les opérations d'E/S par seconde dimensionnées sur un volume EBS avec la fonctionnalité **Instant-Optimization**.

Nous représentons toutes ces fonctionnalités dans **FunctionnalModulofEBS** dans la figure (23) :

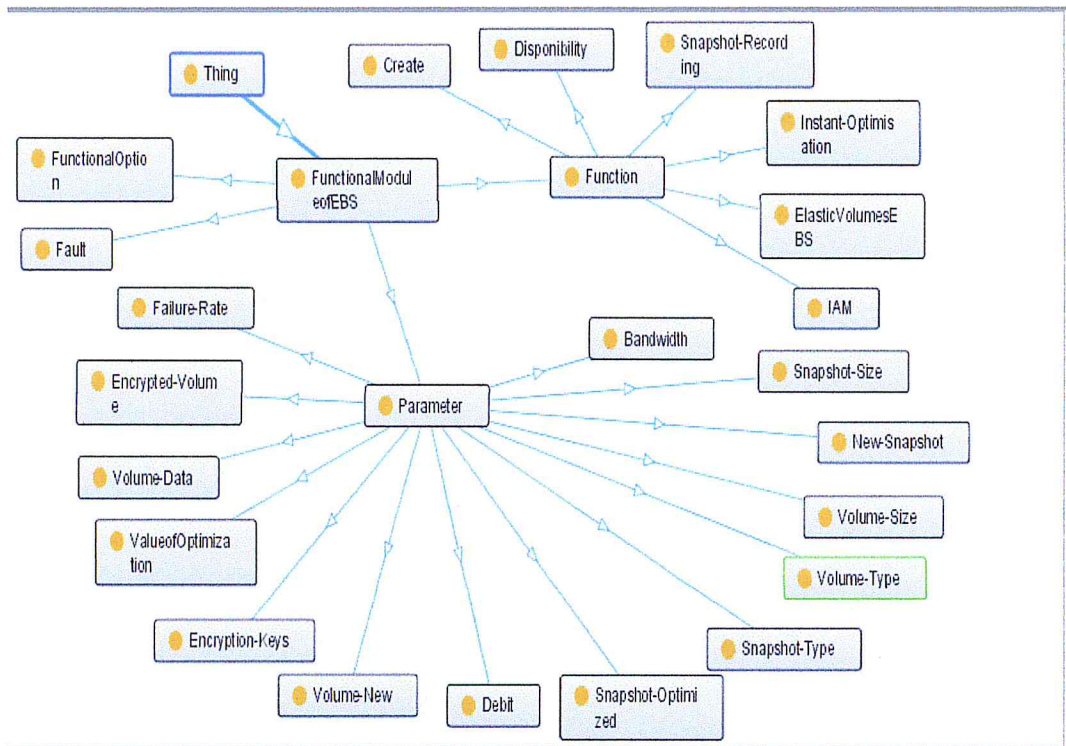


Figure 28: Représentation ontologique du module fonctionnel d'EBS

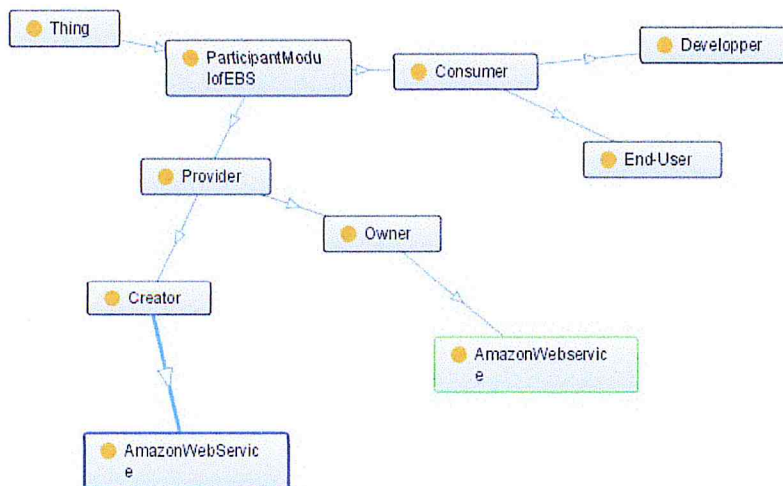


Figure 24 : Représentation Ontologique du module participant d'EBS

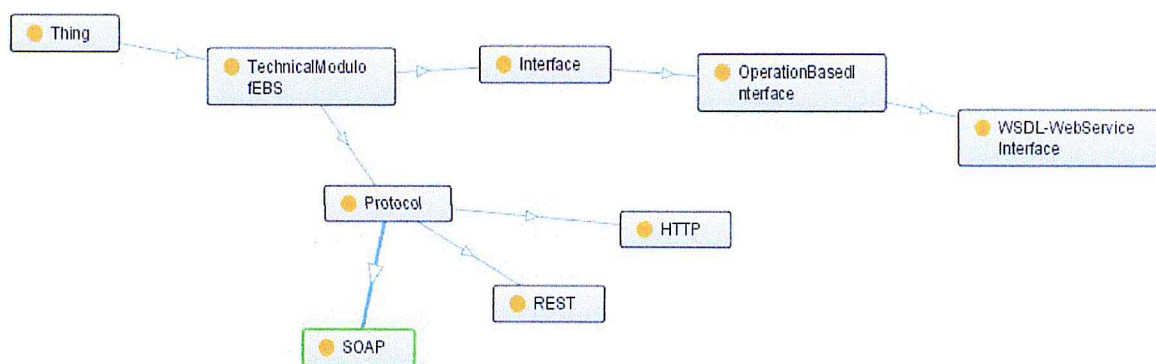


Figure 25 : Représentation ontologique du module technique d’Amazon EBS

Après une représentation ontologique des différents modules d’EBS nous utilisons les fonctionnalités et les différents paramètres du module fonctionnel pour la représentation d’OWL-S

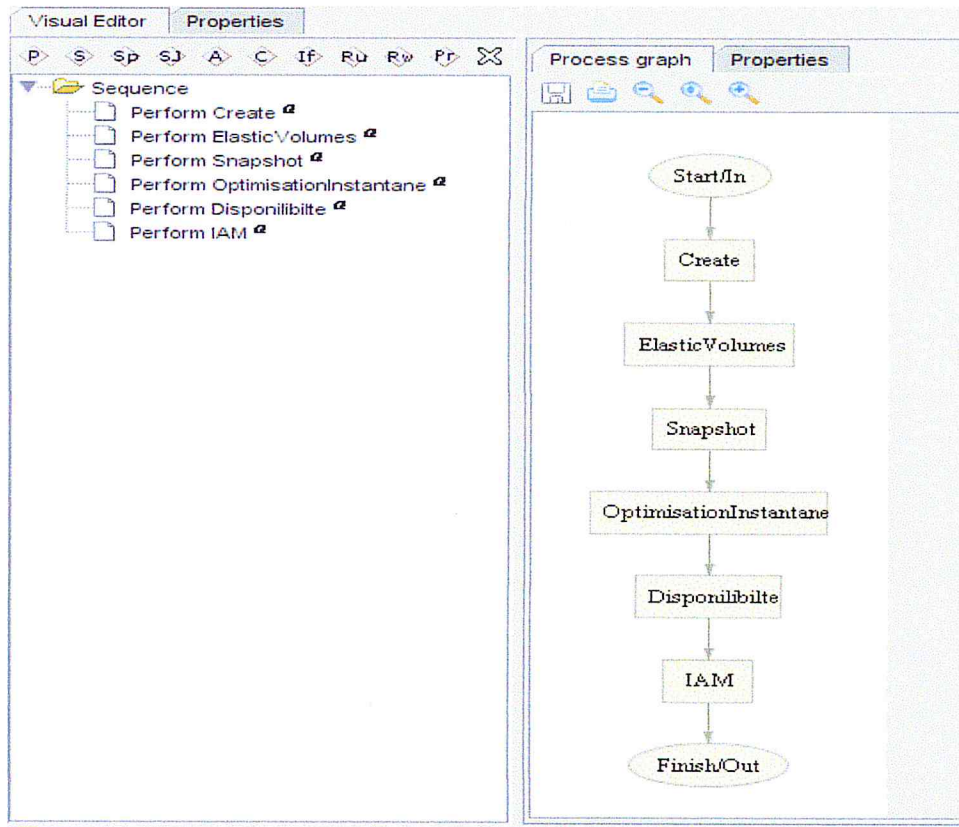


Figure 29: le processus composite du service Amazon EBS

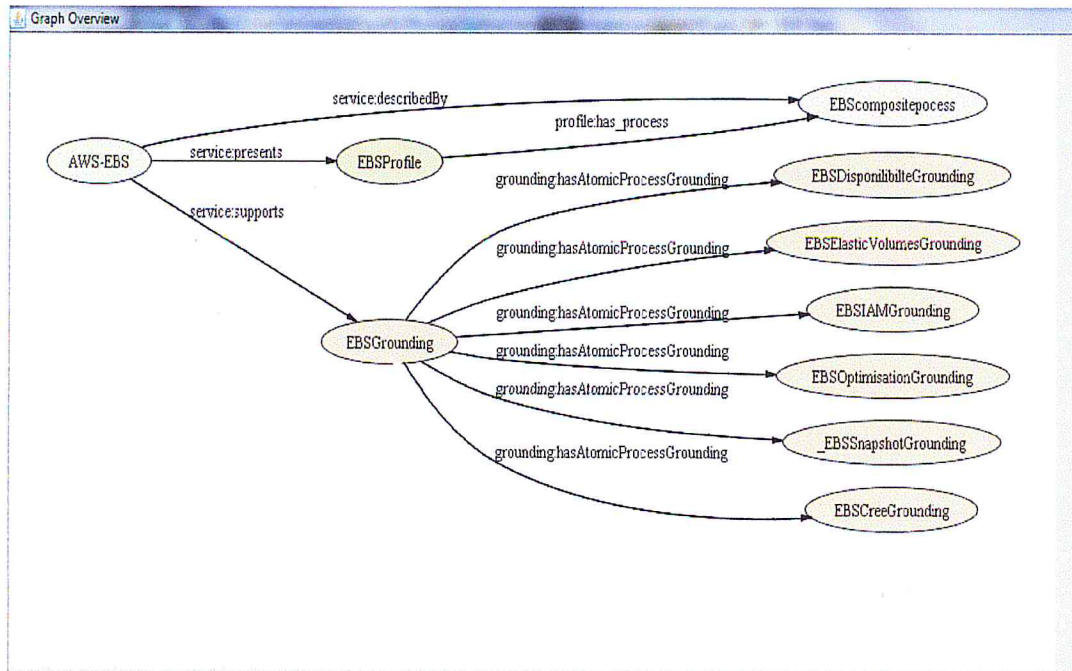


Figure 30: Graphe de la représentation OWL-S d'EBS

III.5 4.2 Microsoft Azure Cloud Service(PAAS) : [59]

Annoncée en octobre 2008, La plate-forme Azure fournit des fonctionnalités permettant de créer et de gérer des applications allant du Web client aux scénarios d'entreprise. Azure est le composant clé de la plate-forme Azure. Azure est l'environnement de développement, d'hébergement de services et de gestion de services pour la plate-forme Azure.

Les utilisateurs et les développeurs Windows ont la possibilité d'avoir une plate-forme en tant que service (PaaS) sous la forme de Microsoft Windows Azure.

Nous représentons le service PaaS de Azure par ServiceModuleOfAzure. Nous décrivons Le type de service en tant qu'instance PaaS de sous-concept PaaS. Azure est déployé sur un Cloud privé.

Les fonctionnalités de Azure Cloud Service sont représentées par certaines instances du sous-concept Function telles que: DevelopApplication (Développement), DeployApplication (Déploiement), TestApplication (Test), LookAtApplication (Surveillance).

Azure est destiné aux consommateurs, et en particulier aux développeurs représentés par une instance de Developer du sous-concept Consumer.

Azure est fourni par Microsoft qui est représenté par l'instance Microsoft du sous-concept Creator.

Azure fournit aux consommateurs des ressources telles que Service cloud, Role worker Role web.

Service cloud : désigne un ensemble de ressources informatiques utilisées par des role web et les role worker.

Role Worker : est un composant des services cloud exécuté dans l'environnement d'exécution Azure utile pour le développement généralisé, et peut effectuer des opérations de traitement en arrière-plan pour un Role web.

Role Web : est un composant des services cloud Computing exécuté dans l'environnement d'exécution Azure.

Une représentation d'Azure Cloud Service est donnée dans les figures 31, 32, 33, 34, 35 et 36.

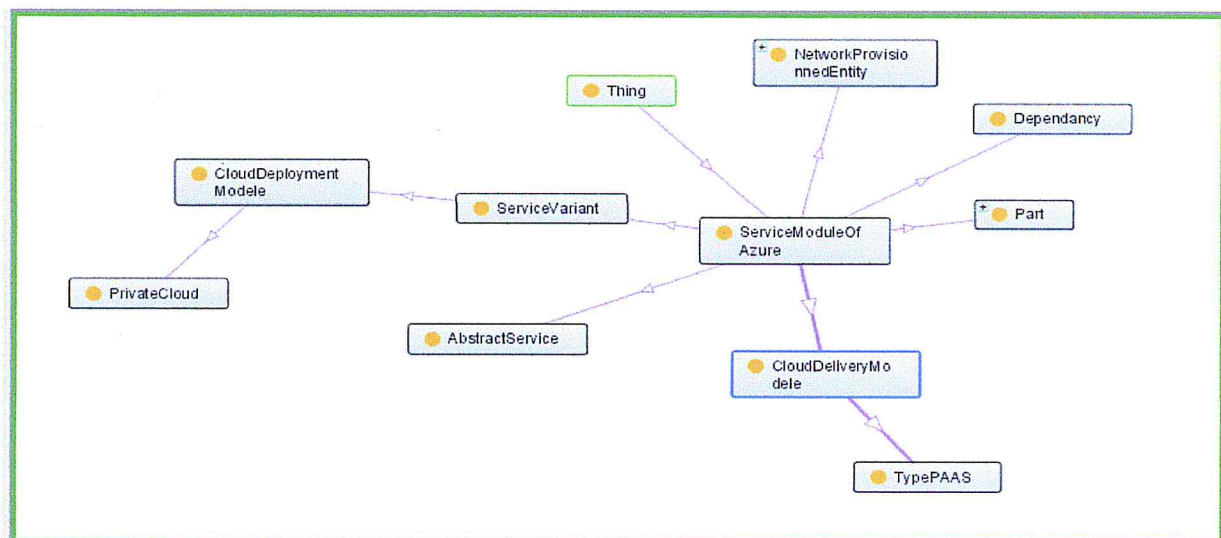


Figure 31: Représentation du Module service de Azure cloud service

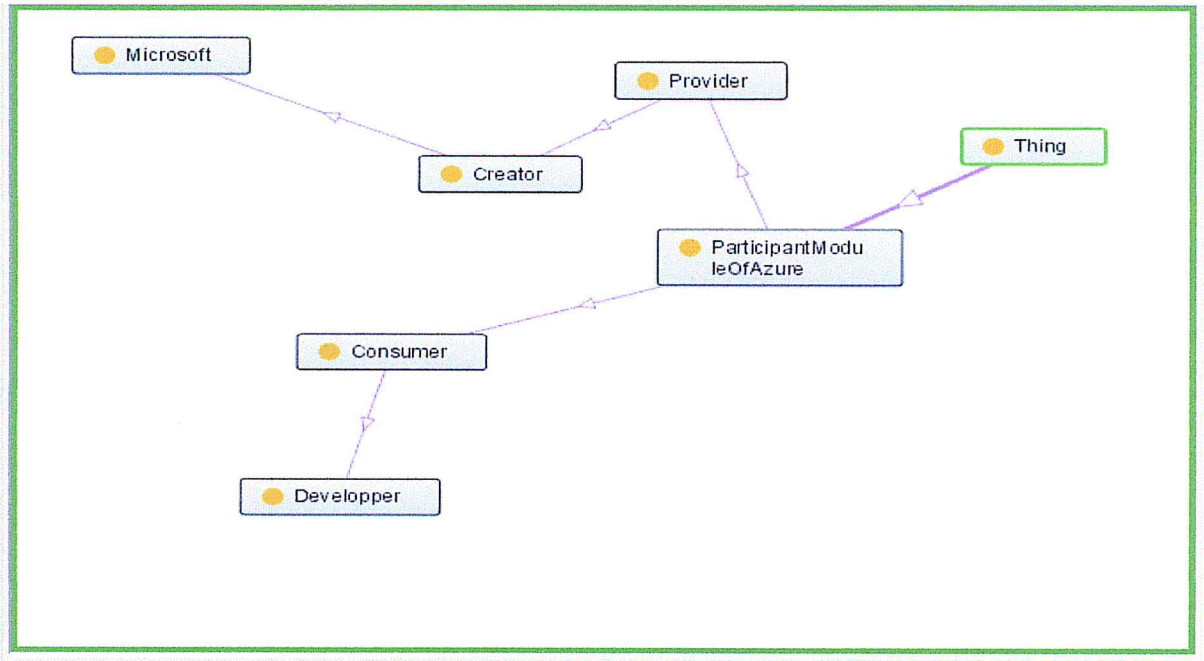


Figure 32: Représentation du Module Participant de Azure cloud service

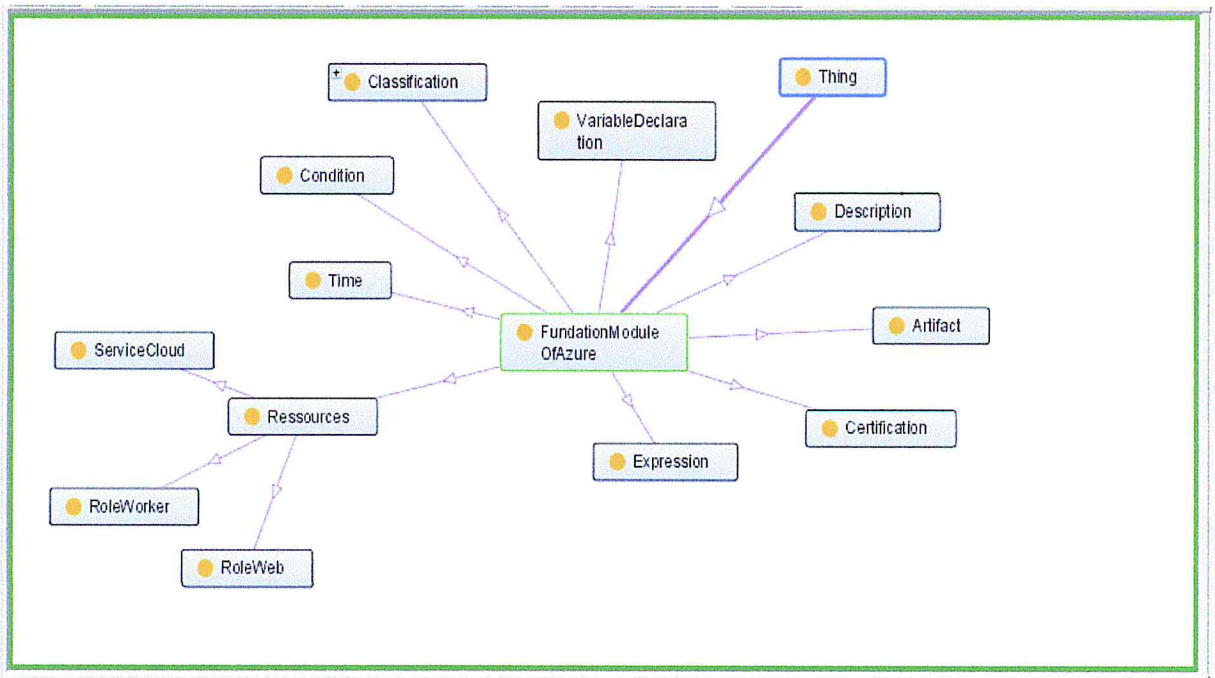


Figure 33: Représentation du Module Foundation de Azure cloud service

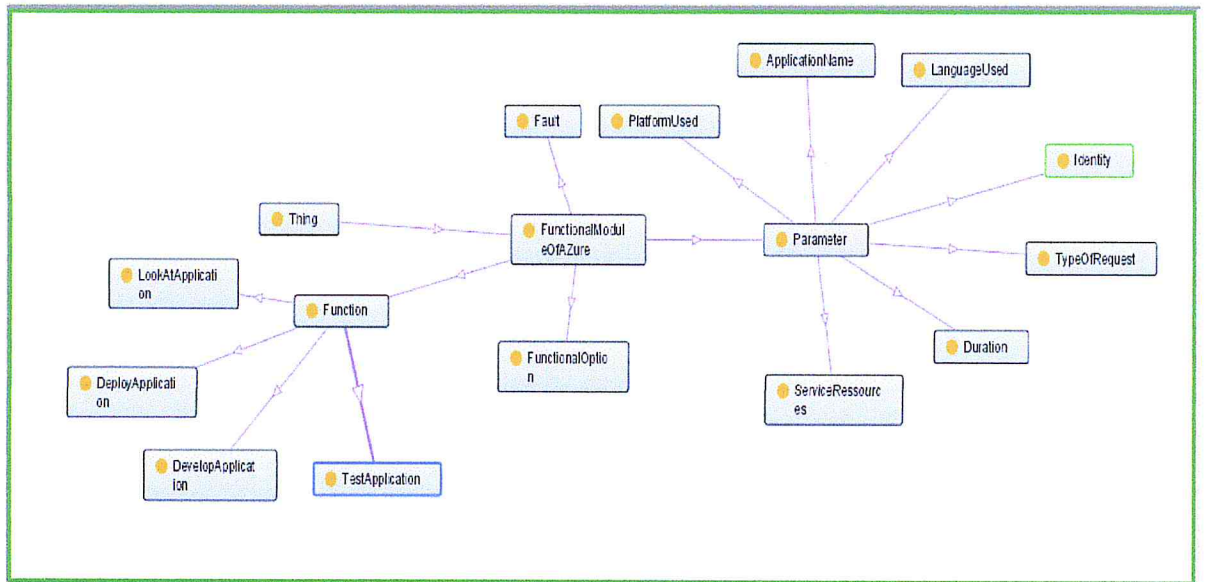


Figure 34 : Représentation du Module Fonctionnel de Azure cloud service

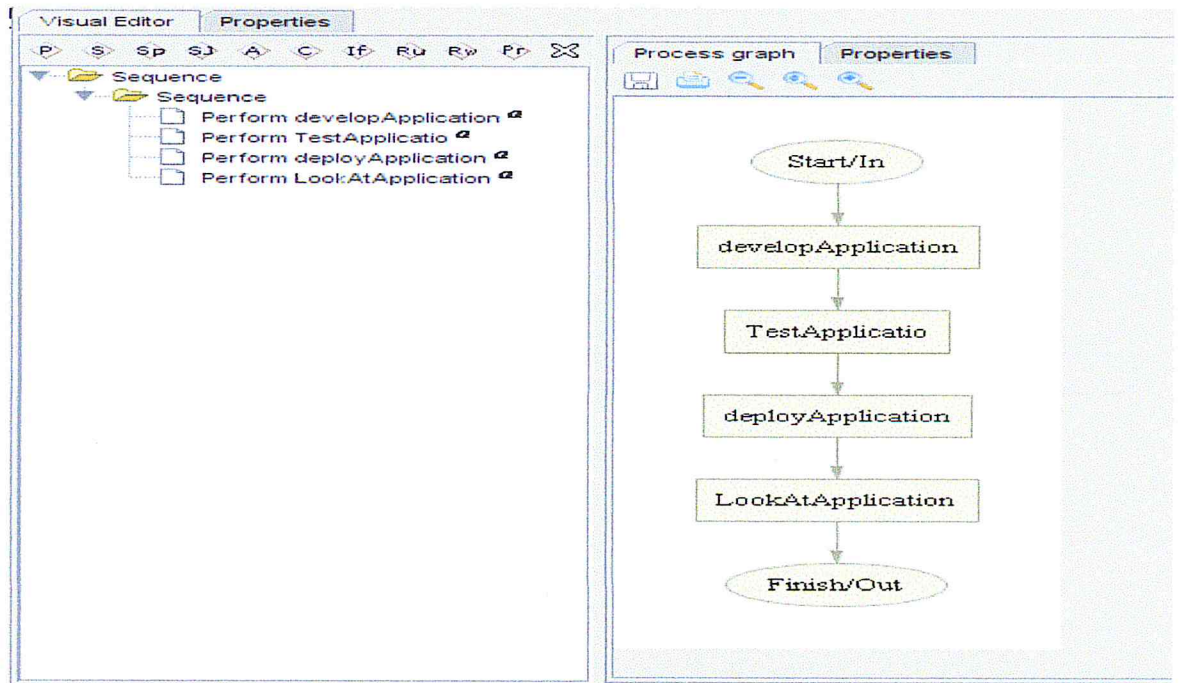


Figure 35: le processus composite du service Azure Cloud Service

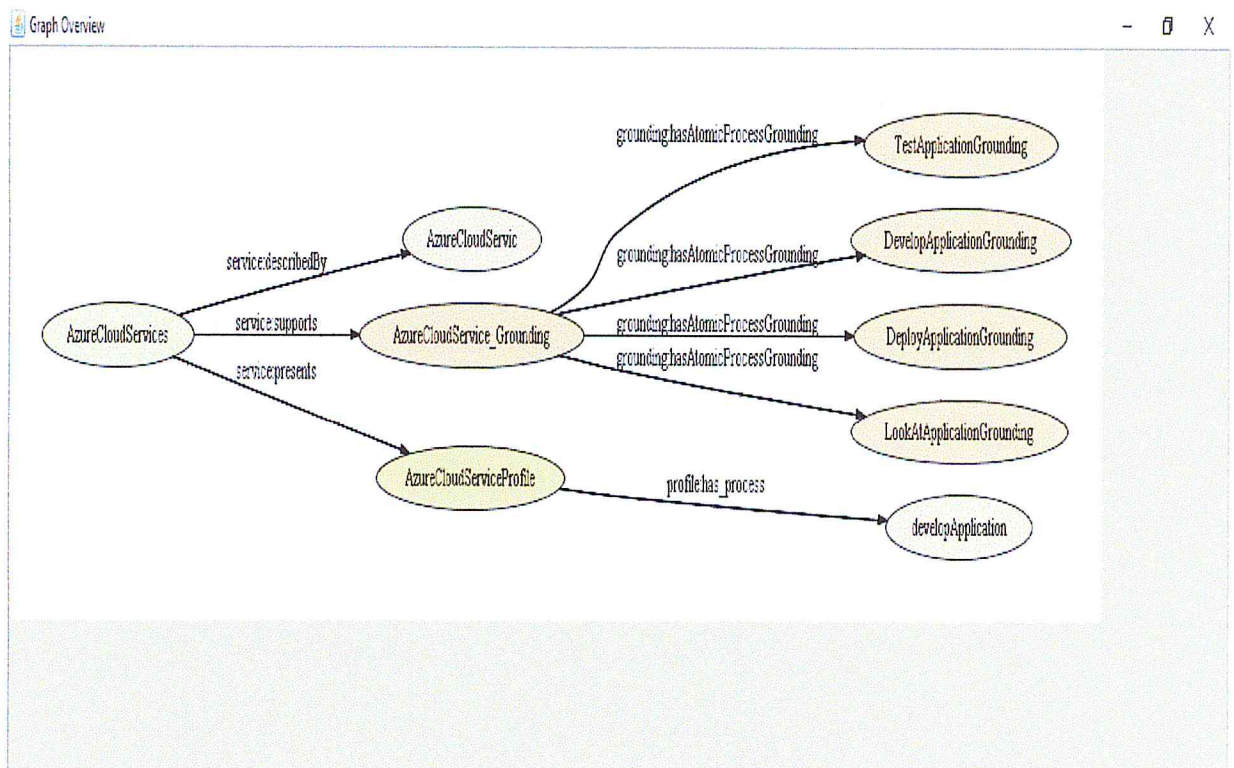


Figure 36: Graphe de la représentation d' Azure Cloud Service en OWL-S

III.6 4.3 SharePoint Online Business (SAAS) :

SharePoint Online est un service dans le cloud, hébergé par Microsoft dédié aux entreprises de toutes tailles. Au lieu d'installer et de déployer SharePoint Server en local, il suffit maintenant que les entreprises s'abonnent à une offre Office 365 ou au service SharePoint Online autonome. Ils peuvent ainsi partager (des fichiers, des données, des actualités et des ressources) et collaborer entre collègues, partenaires et clients. Avec SharePoint, les utilisateurs peuvent accéder à des sites internes, à des documents et à d'autres informations de n'importe où, au bureau, à la maison ou à partir d'un appareil mobile [62], [63].

SharePoint est un service de type SAAS, déployé comme cloud privé, nous représentons ces caractéristiques par l'instance ServiceModuleofSharePoint du concept ServiceModule.

SharePoint offre différentes fonctionnalités (création de collection de site ; partage de fichier ; modification de fichier ; recherche d'informations ; suppression de fichier) ; qui sont des instances du sous-concept Function dans FunctionalModuleofSharePoint dans notre représentation.

Les figures suivantes (37, 38, 39, 40 et 41) donnent une représentation de SharePoint :

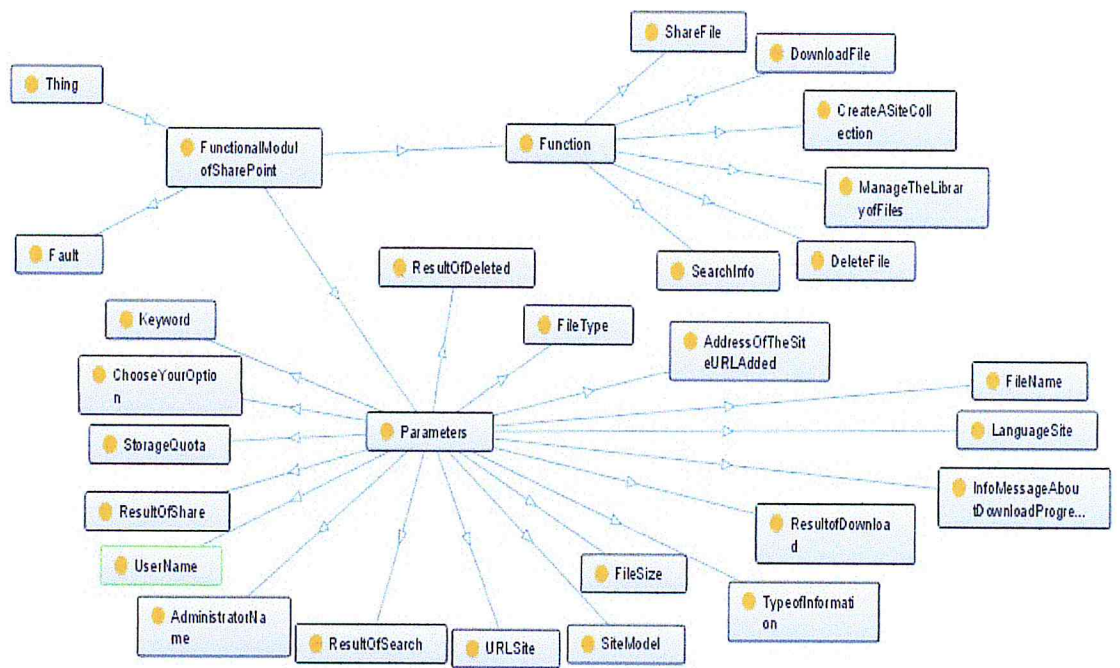


Figure 37: Représentation du Module Fonctionnel de SharePoint

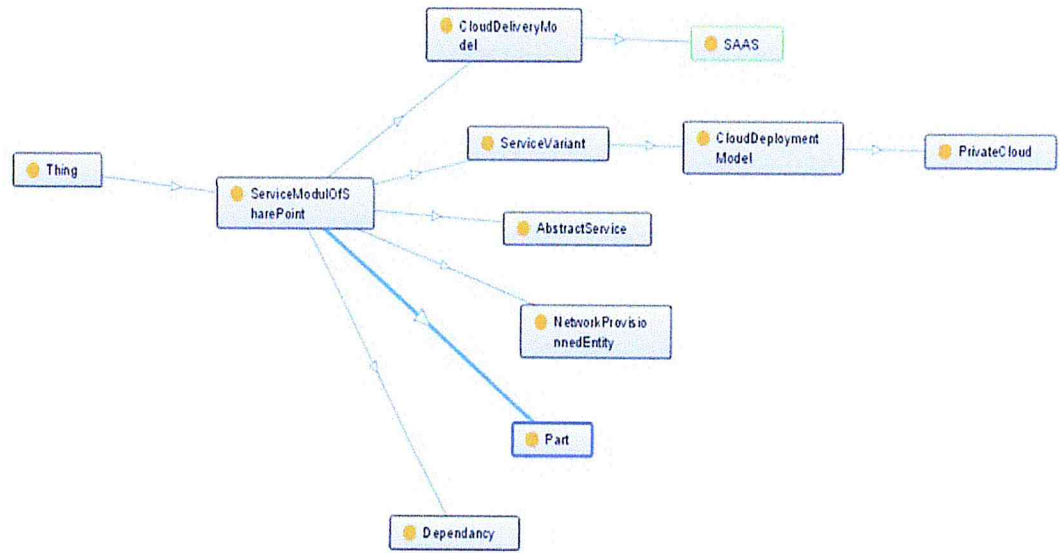


Figure 38: Représentation du Module service de SharePoint

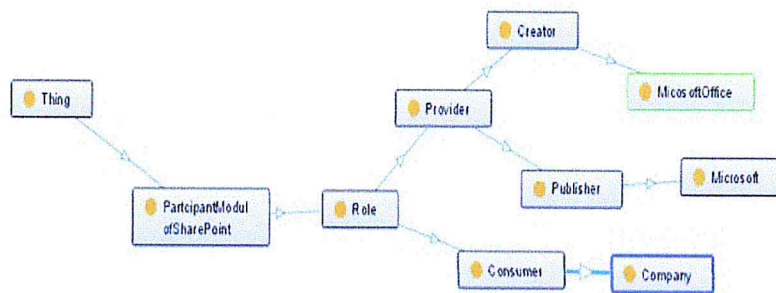


Figure 39: Représentation du Module Participant de SharePoint

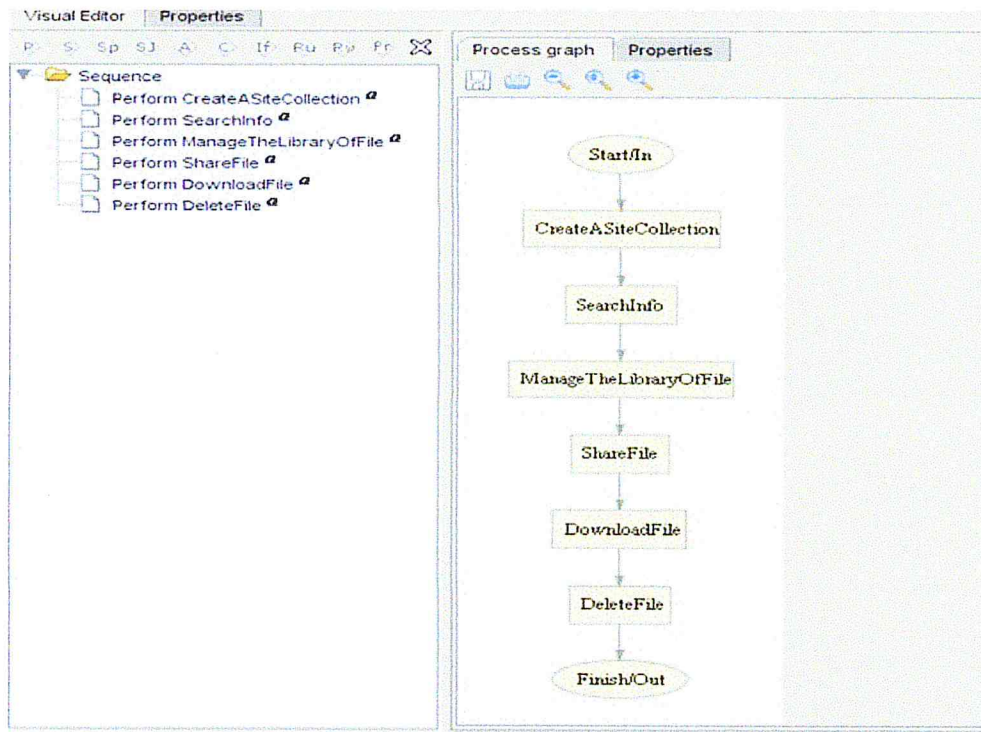


Figure 40: Processus Composite du service Sharepoint

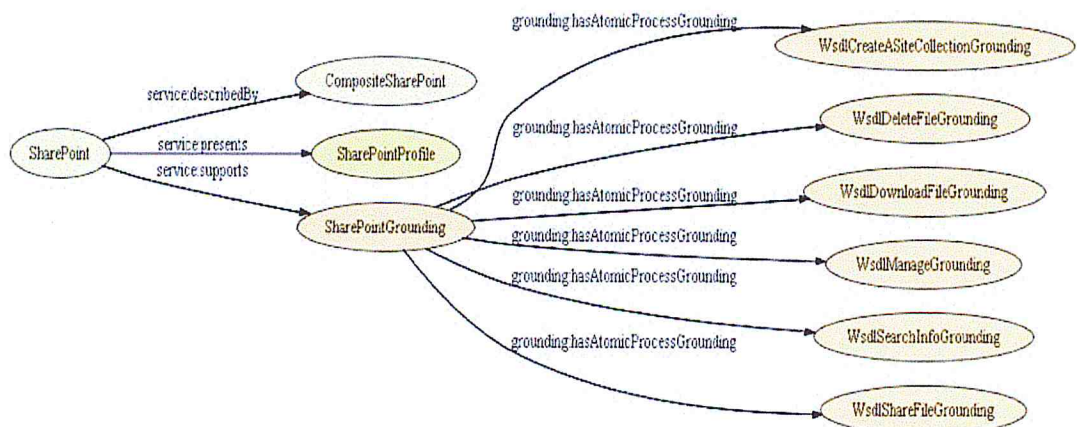


Figure 41: Graphe Principal SharePoint OWL-S

III.7 Conclusion:

Dans ce chapitre, nous avons présenté notre solution pour la description générique des services Cloud.

Nous nous sommes basées sur les différents modules d'USDL pour proposer une ontologie qui regroupe tous les concepts et éléments qui interviennent dans une description de services cloud. Nous avons ensuite décrit un exemple de service de chaque type (SaaS, PaaS et IaaS) à l'aide de notre proposition, les fonctionnalités ont été décrites avec OWL_S.

Dans le chapitre suivant, nous allons utiliser notre proposition pour décrire des services via une application java que nous avons développé, ensuite nous allons utiliser ces services dans un processus de découverte et de sélection.

IV.1 Introduction :

Dans le chapitre précédent de ce mémoire, nous avons proposé une solution de description générique de service cloud à base d'USDL par l'utilisation d'ontologie. Afin d'illustrer les différentes idées et concepts inclus dans la solution proposée, nous allons l'utiliser pour une expérimentation dans une application java.

Pour cela, dans ce chapitre nous allons présenter, dans un premier temps, les outils de développement utilisés, ensuite notre application avec un jeu de test.

IV.2 Les outils de développement

Pour la réalisation de notre application, nous avons utilisé le langage de programmation Java, l'éditeur Netbeans et Xamp Server.

IV.2.1 Langage de programmation java :

Java est à la fois un langage de programmation orienté objet et un environnement d'exécution informatique portable créé par James Gosling et Patrick Naughton, (employés de Sun Microsystems) avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld [69]. Java permet de développer des applications autonomes et des applications client-serveur.

Les applications Java peuvent être exécutées sur tous les systèmes d'exploitation dans lesquels a été développée une plate-forme Java, dont le nom technique est JRE (Java Runtime Environment - Environnement d'exécution Java) tel que : Unix,

Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java.

Le JRE est constituée d'une JVM (Java Virtual Machine - Machine Virtuelle Java), le programme qui interprète le code Java est convertit en code natif, mais le JRE est surtout constitué d'une bibliothèque standard à partir de laquelle tous les programmes en Java doivent être développés [69].

IV.2.2 Environnement de développement Netbeans :

Netbeans est un environnement de développement intégré (IDE en anglais Integrated Development Environment), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2 (). En plus de Java, Netbeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML (Hyper Text Markup Language). Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web. Compilé en java, Netbeans est disponible sous windows, Linux, Solaris, Mac OS X et Open VMS. [70]

IV.2.3 Xamp server:

XAMPP signifie Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) et Perl (P). C'est une distribution Apache simple et légère qui permet aux développeurs de créer un serveur Web local à des fins de test et de déploiement. Le paquet comprend le serveur Web Apache, MySQL, PHP, Perl, un serveur FTP et phpMyAdmin [71].

XAMPP a été conçu pour être le moyen le plus simple d'installer et d'exécuter un serveur de développement. Outre Apache, MySQL et PHP, XAMPP comprend d'autres outils très utiles tels que l'outil d'administration de la base de données phpMyAdmin, le serveur FTP FileZilla, le serveur de messagerie Mercury, le langage de programmation Perl et le serveur JSP Tomcat.

XAMPP est également multiplateforme, ce qui signifie qu'il fonctionne également sous Linux, Mac et Windows. Étant donné que la plupart des déploiements de serveurs Web utilisent les mêmes composants que XAMPP, la transition entre un serveur de test local et un serveur actif est extrêmement simple.

IV.3 Description de l'application :

Notre application a trois principales fonctionnalités, la première partie consiste à la création de service dans la base de données, la deuxième consiste à l'utilisation des descriptions stockées dans la base de données pour la découverte de service et la troisième consiste à l'utilisation de ces descriptions dans un processus de sélection de service.

Nous présentons ici les différentes fonctionnalités de l'application.

IV.3.1 Partie 1 : Creation de service

Dès le lancement de notre application, la figure (42) apparait en premier aux utilisateurs, elle est constituée de 3 boutons : Creation of Service, Research Of Service et List Of Service.

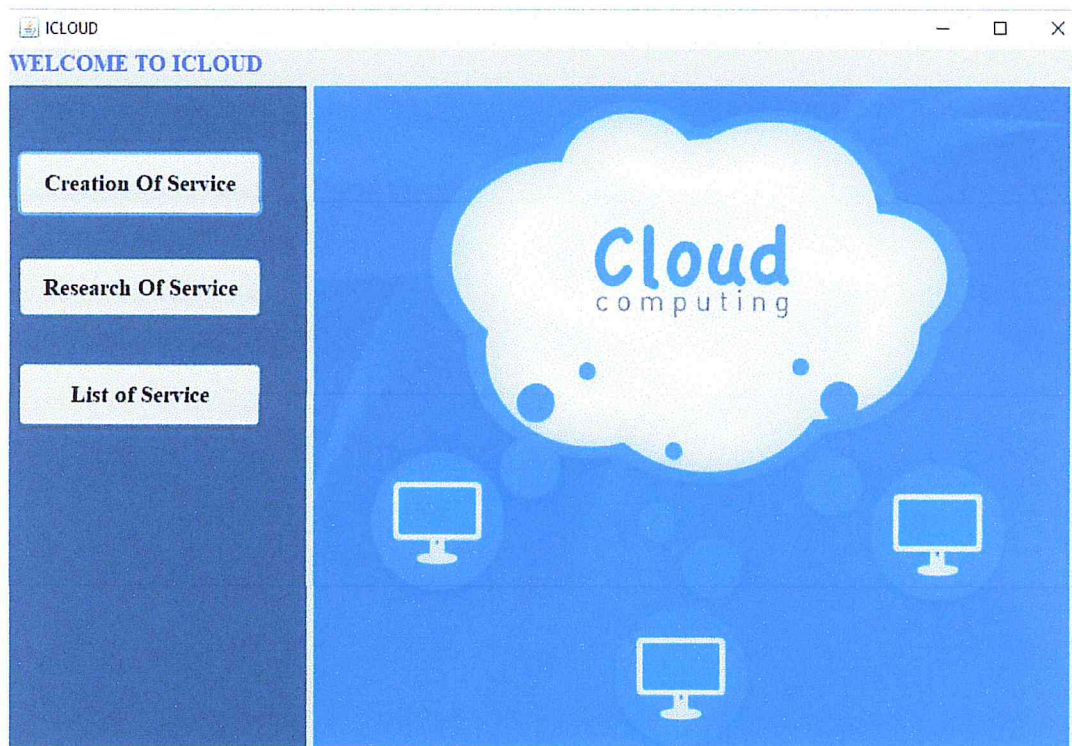


Figure 42: interface d'accueil

Le bouton « List Of Service » permet d'accéder à l'interface permettant d'afficher la liste des services (figure 43).

ServiceNa...	CloudDeliv...	CloudDepl...	FunctonM...	Provider	Reputation	Experience	Disponibility	MeanTime...	Response...	Risk
SharePoint	SAAS	PRIVATE	GESTION	Microsoft 3...	17	20	98	97	95	30
AMAZON S3	PAAS	PUBLIC	STORAGE	Amazon W...	14	10	99	30	40	40
Salesforce...	SAAS	PRIVATE	GESTION	Salesforce	17	20	98	97	98	17
Amazone ...	IAAS	PUBLIC	STORAGE	AWS	12	06	96	60	50	20
AZURE CL...	PAAS	PRIVATE	DEPLOYM...	MICROSO...	15	30	96	40	50	10
GOOGLE ...	PAAS	PRIVATE	DEPLOYM...	GOOGLE	16	15	98	50	40	12
MaaS360	SAAS	PRIVATE	GESTION	IBM	18	40	96	40	30	18
GOOGLE...	IAAS	HYBRID	STORAGE	GOOGLE	14	10	99	50	60	30
ZENDESK	SAAS	PRIVATE	GESTION	ZENDESK	16	05	94	60	50	12
DROPBOX	SAAS	PRIVATE	STORAGE	DROPBOX	18	05	98	40	30	10
STORDATA	IAAS	PRIVATE	STORAGE	STORDATA	16	10	96	40	30	12

Figure 43: Liste des services

Le bouton "Creation of service" permet d'accéder aux différentes interfaces pour l'enregistrement d'un service. Dans les fenêtres suivantes, l'utilisateur peut saisir toutes les informations relatives au service, les participants, le prix, le niveau de service (selon que le service soit un PaaS, un SaaS ou un IaaS), etc.

Service		Participant	
ServiceName*:	ICLOUD	Provider*:	APPLE
CloudDeliveryMode*:	IAAS	Creator:	
CloudDeploymentModele*:	PRIVATE	Publisher:	
Reputation*:	14	Contributor:	
Trust*:	13	Broker:	
Risk*:	12	Carrier:	
Experience*:	50	Auditor:	
ConsumerEstimation*:	1000		
Disponibility*:	80		
Mean Time To Repair*:	40		
ResponseTime*:	30		

Figure 44: Creation de service fenetre1

Dans cette fenêtre, l'utilisateur peut saisir les informations du service (le nom, le type de déploiement, le type de livraison, la réputation, la confiance, le risque, la disponibilité etc) et des différents participants (fournisseur, créateur, auditeur etc).

Le bouton « next » ouvre la fenêtre suivante pour continuer l'enregistrement du service.

The screenshot shows a window titled "Creation Of Service" with a light blue header. Below the header, the form is split into two columns. The left column, labeled "Legal", contains six text input fields: "Licence:", "UseTerms", "SinglePointOfContact*", "Compensation:", "Penalty:", and "ResponsibilityPerimeter*". The right column, labeled "Price", contains seven text input fields: "Price*", "PaymentMethods:", "PriceFence:", "PriceMetric:", "Tax:", "Quantity:", and "PriceAdjustment:". At the bottom right of the form area, there is a blue button labeled "Next".

Figure 45: Creation de service fenetre2

Dans cette fenêtre, l'utilisateur peut saisir les informations liées au contrat (licence, droits d'utilisation, périmètre de responsabilité, point de contact etc) et au prix (coût, méthodes de paiement, taxes etc).

Le bouton « next » ouvre la fenêtre suivante pour continuer l'enregistrement du service.

The screenshot shows a window titled "ICLOUD" with the subtitle "Creation Of Service". The window is divided into three main sections:

- Technical/Interaction:** Contains three input fields labeled "Protocol*", "SimpleInteractionProtocol:", and "ComplexInteractionProtocol:".
- Fondation:** Contains three input fields labeled "Certification:", "Description:", and "Expression:".
- Functionnal:** Contains three input fields labeled "Function main*:", "Parameters:", "Function 2:", and "Function 3:".

A blue "Next" button is located at the bottom right of the form.

Figure 46: Creation de service fenêtre3

Dans cette fenêtre, l'utilisateur peut saisir les informations techniques (protocole), des informations sur l'interaction(les protocoles), sur la fondation (certification, description etc) et les fonctionnalités du service (fonctionnalité principale et fonctionnalités secondaires).

Le bouton « next » ouvre la fenêtre suivante pour continuer l'enregistrement du service.

Design Preview [SLA_PAAS]

Creation Of Service

SLA_PAAS

ManagementPortal:

CodeErro:

SuccessCode:

Incident:

ServiceRessources:

Disponibility:

SAVE

Figure 47: Creation de service fenetre4 PaaS

Cette fenêtre apparaît si le service à enregistrer est un PaaS, ici l'utilisateur peut saisir les informations concernant les SLA particuliers du service (portail de gestion, incident etc).Le bouton « Save » enregistre le service dans le base de donnée.

Creation Of Service

SLA_IAAS

ErrorRate :

OperatingTime:

ServiceTime:

IndisponibilityTi...

SAVE

Figure 48: Creation de service fenetre4 IaaS

Cette fenêtre apparaît si le service à enregistrer est un IaaS, ici l'utilisateur peut saisir les informations concernant les SLA particuliers du service (crédit de service, temps de service etc).

Le bouton « Save » enregistre le service dans la base de données.

Design Preview [SLA_SAAS]

Creation Of Service

SLA_SAAS

Reporting:

ResponseTime:

MeanTimeToRepair:

Disponibility:

SAVE

Figure 49: Creation de service fenetre4 SaaS

Cette fenêtre apparaît si le service à enregistrer est un SaaS, ici l'utilisateur peut saisir les informations concernant les SLA particuliers du service (temps moyen de réparation, journalisation etc).Le bouton « Save » enregistre le service dans le base de donnée.

IV.3.2 Partie 2 : Découverte de service

La découverte est la localisation automatique des services répondant à une requête utilisateur. Dans notre application, l'utilisateur a la possibilité de rechercher un service en fonction de sa fonctionnalité principale.

Cette fonctionnalité est assurée via la fenêtre « Research Of Service » présentée sur la figure suivante :

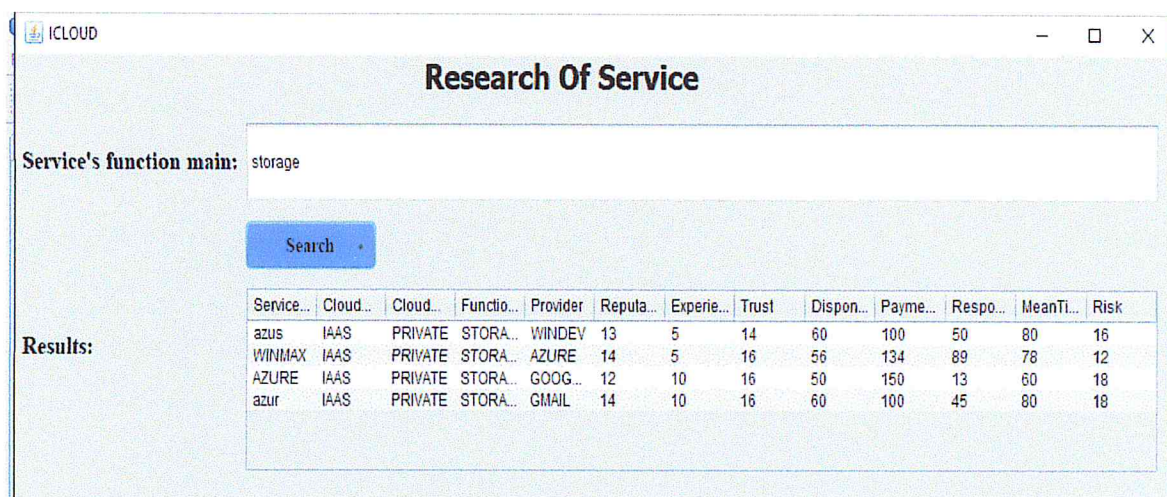


Figure 50:decouverte de service

IV.3.3 Partie 3 : Sélection de services :

La sélection consiste à choisir, parmi les services découverts, ceux qui répondent au mieux aux exigences de l'utilisateur sur la base des besoins fonctionnels et/ou non fonctionnels, cette découverte est fondée sur les informations contenues dans les descriptions des services.

Plusieurs services peuvent répondre aux mêmes besoins fonctionnels, dans ce cas le meilleur service sera retourné en fonctions des caractéristiques non fonctionnelles du service.

Dans notre proposition, nous évaluons la qualité de service QoS (Quality Of Service) à travers les paramètres suivants : la réputation, la confiance, l'expérience, la disponibilité, le risque, le temps de réponse, le temps minimal de réparation, le coût.

Nous avons identifié ces différents paramètres QoS dans notre méthode car ces paramètres ont un grand impact sur l'assistance aux demandeurs pour des sélections raisonnables. En outre, ces paramètres sont les qualités fondamentales pour remplir les objectifs du service. Ceux-ci peuvent être divisés en deux groupes :

Le premier groupe des attributs QoS positifs qui ont besoin être maximisé tel que disponibilité, confiance, expérience et réputation.

Le deuxième groupe des valeurs d'attribut QoS négatifs ont besoin d'être minimisées, c'est-à-dire plus la valeur du temps de réponse ou le coût est inférieur meilleur est la qualité.

❖ **Les attributs positifs:**

- Reputation : la réputation d'un service est une mesure de sa fidélité. Elle dépend principalement des expériences de l'utilisateur d'employer ce service.
- Trust : La confiance qu'on donne à un service est en général en fonction de sa réputation.
- Experience : L'expérience qu'a le fournisseur dans le domaine.
- Disponibility : La disponibilité d'un service est la probabilité que le service est accessible.

❖ **Les attributs négatifs:**

- Price: le Coût fait référence à la taxe qu'un demandeur de service doit payer au fournisseur de service pour l'appel de service.

- Response Time: Le temps de réponse correspond au délai prévu entre le moment où un demandeur de service envoie une demande de service et le moment où le résultat est obtenu.
- Mean Time To Repair: le temps moyen de réparation représente le temps nécessaire pour remettre le système en état de marche.
- Risk : le risque représente la probabilité d'occurrence d'un péril probable ou d'un obstacle.

Pour pouvoir retourner le service qui répond aux besoins de l'utilisateur, nous allons utiliser deux méthodes de sélection: le calcul de score, où on calculera le score de chaque service qui a les mêmes fonctionnalités principales ; la priorité de l'utilisateur où on comparera le QoS des services qui ont les mêmes fonctionnalités principales selon la priorité de l'utilisateur.

a. Calcul du score :

Le score représente le nombre de point obtenu par chaque service en fonction des qualités de services. Il est calculé par une fonction objective que nous allons détailler dans les lignes qui suivent.

Soit $Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, Q_8$ les qualités de services respectivement (Reputation, confiance, expérience, disponibilité, Coût, temps de réponse, temps minimal de réparation, risque).

Soit S : le score du service ; et $f(S)$: une fonction objective qui calcule le score S en fonction des qualités de service Q_1, \dots, Q_8 .

Q^{max} et Q^{min} Représentent le maximum actuel et la valeur minimale de l'indice d'évaluation respectivement. Leur valeur va changer dynamiquement avec l'ajout ou retrait d'un service.

Les valeurs sont normalisées entre $[0,1]$.

$X_j(S)$ et $Y_k(S)$, representent respectivement les valeurs normalisées des QoS aux attributs positifs et negatifs.

Le score de chaque service est calculé en additionnant les valeurs normalisées de ces qualités de services Q_1, \dots, Q_8 .

Soit les équations suivantes :

$$X_j(S) = \begin{cases} \frac{Q_j - Q_j^{min}}{Q_j^{max} - Q_j^{min}} \\ 1 \end{cases} ; \quad Q_j^{max} - Q_j^{min} \neq 0 \quad j=1, 2, 3, 4$$

Equation (1)

$$Y_k(S) = \begin{cases} \frac{Q_k^{max} - Q_k}{Q_k^{max} - Q_k^{min}} \\ 1 \end{cases} ; \quad Q_k^{max} - Q_k^{min} \neq 0 \quad k=5, 6, 7, 8$$

Equation (2)

$$f(S) = \sum_{j=1}^4 X_j(S) + \sum_{k=5}^8 Y_k(S)$$

Equation(3)

On suppose que les domaines de valeurs de critères de QoS sont définit comme suit :

Critères	Domaine de valeurs
Réputation	0...20
Trust	0...20
Experience	0...50
Disponibility	0...100
Price	0...1000
ResponseTime	0...100
MeanTimeToRepair	0...100
Risk	0...100

Tableau 6: Domaine de valeur

❖ **Algorithme :**

Les étapes de l'algorithme :

Entrée : Les informations des qualités de service du jeu de services

Sortie : Les informations du service ayant le meilleur score dans jeu de services

Étape 1 : Obtenir les informations de qualité de service de chaque ensemble de services.

Étape 2 : Obtenir la valeur maximale et minimale, la valeur de chaque indice dans l'ensemble des services

Étape 3 : Pour chaque enregistrement de service, les valeurs des qualités de service sont normalisées selon l'équation (1) ou (2).

Étape 4 : calculer le score final de chaque service après la normalisation avec l'équation (3).

Étape 5 : comparer les scores et retourner le ou les services qui ont le meilleur score.

Fonction (QoS) (S : Service, Q : Qualité)

{

X1=X2=X3=X4=0 ; //on initialise les score à 0

Y5=Y6= Y7=Y8=0 ; //

Q max : la qualité maximal de chaque type de qualité de service, Q min : la qualité minimal de chaque type de qualité de service.

Pour i=1 jusqu'à n faire

$S_i = (Q1S_i, Q2S_i, Q3S_i, Q4S_i, Q5S_i, Q6S_i, Q7S_i, Q8S_i)$ // on applique les huit qualités de service sur chaque service.

$X1 = Q1 - Q_{min} / Q_{max} - Q_{min}$; // normaliser la Réputation.

$X2 = Q2 - Q_{min} / Q_{max} - Q_{min}$; // normaliser la confiance.

$X3 = Q3 - Q_{min} / Q_{max} - Q_{min}$; // normaliser l'expérience.

$X4 = Q4 - Q_{min} / Q_{max} - Q_{min}$; // normaliser la disponibilité.

$Y5 = Q_{max} - Q5 / Q_{max} - Q_{min}$; // normaliser le coût.

$Y6 = Q_{max} - Q6 / Q_{max} - Q_{min}$; // normaliser le temps de réponse.

$Y7 = Q_{max} - Q7 / Q_{max} - Q_{min}$; // normaliser le temps minimal de réparation.

$Y8 = Q_{max} - Q8 / Q_{max} - Q_{min}$; // normaliser le risque.

$SC = X1 + X2 + X3 + X4 + Y5 + Y6 + Y7 + Y8$;

Fin Pour ;

$j = n$;

Répéter

Si $SC_j < SC_{j-1}$ alors $SC = SC_{j-1}$;

Sinon $SC = SC_j$; // on prend la valeur maximal

Jusqu'à $j=2$;

// On sélectionne le service qui a la plus grande valeur de qualité SC maximal

}

❖ Jeu de test

Soit à sélectionner le meilleur service entre les services suivants S1, S2 et S3.

On suppose que les trois services ont la même fonctionnalité principale : « Storage » et on donne les informations des services dans le tableau 7 :

Service	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
S1	14	16	10	99	150	30	40	40
S2	12	17	6	96	100	50	60	20
S3	14	14	10	99	130	60	50	30

Tableau 7: les données

Etape1 : on détermine les valeurs maximales et minimales de chaque QoS.

Ces résultats sont donnés dans le tableau 8 :

QoS	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Max	14	17	10	99	150	60	60	40
Min	12	14	6	96	100	30	40	20

Tableau 8: Valeurs maximales et minimales des QoS

Etape2 : On calcule la valeur normalisée de chaque QoS a l'aide de l'équation (1) pour les attributs positifs et (2) pour les attributs négatifs.

Ces résultats sont reportés dans le tableau 9:

Service	X1	X2	X3	X4	Y5	Y6	Y7	Y8
S1	1	0,66	1	1	0	1	1	0
S2	0	1	0	0	1	0,33	0	1
S3	1	0	1	1	0,4	0	0,5	0,5

Tableau 9:Normalisation des QoS

Etape 3 : On calcule le score de chaque service à l'aide de l'équation (3).

Les résultats obtenus sont répertoriés dans le tableau 10 :

Service	Score
S1	5,66
S2	3,33
S3	4,4

Tableau 10:score final des services

D'après l'algorithme utilisé, le meilleur service est S1. C'est donc S1 le service qui sera sélectionné.

Dans notre application, cette fonctionnalité est assurée via la fenêtre « Research Of Service ». Pour sélectionner le service en fonction du score, l'utilisateur doit cliquer sur le bouton « Standard Filtrage » (voir figure 51)

The screenshot shows a web application window titled "Research Of Service". At the top, there is a search bar with the text "Service's function main: STORAGE" and a "Search" button. Below the search bar, there is a table of results. The table has columns: Service..., Cloud..., Cloud..., Functio..., Provider, Reputa..., Experie..., Trust, Dispon..., Payme..., Respo..., MeanTi..., Risk. The results are:

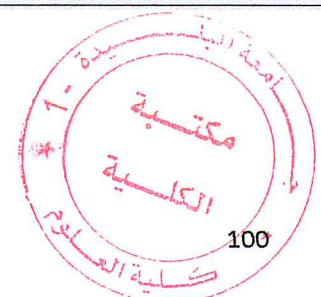
Service...	Cloud...	Cloud...	Functio...	Provider	Reputa...	Experie...	Trust	Dispon...	Payme...	Respo...	MeanTi...	Risk
AMAZ...	PAAS	PUBLIC	STORA...	Amazo...	14	10	16	99	150	40	30	40
Amazo...	IAAS	PUBLIC	STORA...	AWS	12	06	17	96	100	50	60	20
GOOG...	IAAS	HYBRID	STORA...	GOOG...	14	10	14	99	130	60	50	30

Below the table, there are radio buttons for filtering: "Standard Filtrage" (selected), "Filtrage By Disponibility", "FiltrageByExperience", "Filtrage By Reputation", "Filtrage By Risk", "Filtrage By MITR", "FiltrageByResponse", "Filtrage By Cost", and "Filtrage By Trust".

At the bottom, there is another table with columns: Service..., Cloud..., Cloud..., Functio..., Reputa..., Experie..., Trust, Dispon..., Payme..., Respo..., MeanTi..., Risk, Score. The results are:

Service...	Cloud...	Cloud...	Functio...	Reputa...	Experie...	Trust	Dispon...	Payme...	Respo...	MeanTi...	Risk	Score
AMAZ...	PAAS	PUBLIC	STORA...	14	10	16	99	150	40	30	40	5.6666...
Amazo...	IAAS	PUBLIC	STORA...	12	06	17	96	100	50	60	20	3.5
GOOG...	IAAS	HYBRID	STORA...	14	10	14	99	130	60	50	30	4.2333...

Figure 51:Filtrage par score



b. Priorité de l'utilisateur

La priorité de l'utilisateur désigne la qualité de service qui lui est plus importante en termes de comparaison.

Dans notre application si plusieurs services répondent aux mêmes exigences (fonctionnalité principale) alors nous donnons la main à l'utilisateur pour désigner la qualité de service qui lui est prioritaire pour faire la sélection. Ainsi l'application retourne le ou les services qui répondent le plus.

Notre application fait la sélection en fonction de huit QoS donc l'utilisateur a la possibilité de choisir une de ces QoS dans le processus de sélection.

Cette fonctionnalité est réalisée dans la fenêtre « Research Of Service » (voir figures 52, 53,54, 55, 56, 57,58 et 59).

Le bouton « FiltrageByReputation » sélectionne les services qui ont la même fonctionnalité principale et retourne le ou les services qui ont la plus grande réputation (voir figure 52).

ICLOUD

Research Of Service

Service's function main:

Results:

Service...	Cloud...	Cloud...	Funcio...	Provider	Reputa...	Experie...	Trust	Dispon...	Payme...	Respo...	MeanTi...	Risk
azus	IAAS	PRIVATE	STORA...	WINDEV	13	5	14	60	100	50	80	16
WINMAX	IAAS	PRIVATE	STORA...	AZURE	14	5	16	56	134	89	78	12
AZURE	IAAS	PRIVATE	STORA...	GOOG...	12	10	16	50	150	13	60	18
azur	IAAS	PRIVATE	STORA...	GMAIL	14	10	16	60	100	45	80	18

Filtrage:

Standard Filtrage
 Filtrage By Disponibility
 FiltrageByExperience
 Filtrage By Reputation

Filtrage By Risk
 Filtrage By MITR
 FiltrageByResponse
 Filtrage By Cost
 Filtrage By Trust

Service...	CloudD...	CloudD...	Funcio...	Reputati...	Experie...	Trust	Disponi...	Paymen...	Respon...	MeanTi...	Risk
WINMAX	IAAS	PRIVATE	STORA...	14	5	16	56	134	89	78	12
azur	IAAS	PRIVATE	STORA...	14	10	16	60	100	45	80	18

Figure 52: filtrage par Reputation

Le bouton « FiltrageByExperience » sélectionne les services qui ont la même fonctionnalité principale par rapport à l'expérience et retourne le ou les services qui ont la plus grande expérience (voir figure 53)

ICLOUD

Research Of Service

Service's function main:

Results:

Service...	Cloud...	Cloud...	Funcio...	Provider	Reputa...	Experie...	Trust	Dispon...	Payme...	Respo...	MeanTi...	Risk
azus	IAAS	PRIVATE	STORA...	WINDEV	13	5	14	60	100	50	80	16
WINMAX	IAAS	PRIVATE	STORA...	AZURE	14	5	16	56	134	89	78	12
AZURE	IAAS	PRIVATE	STORA...	GOOG...	12	10	16	50	150	13	60	18
azur	IAAS	PRIVATE	STORA...	GMAIL	14	10	16	60	100	45	80	18

Filtrage:

Standard Filtrage
 Filtrage By Disponibility
 FiltrageByExperience
 Filtrage By Reputation

Filtrage By Risk
 Filtrage By MITR
 FiltrageByResponse
 Filtrage By Cost
 Filtrage By Trust

Service...	CloudD...	CloudD...	Funcio...	Reputati...	Experie...	Trust	Disponi...	Paymen...	Respon...	MeanTi...	Risk
AZURE	IAAS	PRIVATE	STORA...	12	10	16	50	150	13	60	18
azur	IAAS	PRIVATE	STORA...	14	10	16	60	100	45	80	18

Figure 53:filtrage par Experience

Le bouton « FiltrageByTrust » sélectionne les services qui ont la même fonctionnalité principale et retourne le ou les services qui ont la plus grande confiance (voir figure 54)

ICLOUD

Research Of Service

Service's function main: storage

Search

Results:

Service...	Cloud...	Cloud...	Funcio...	Provider	Reputa...	Experie...	Trust	Dispon...	Payme...	Respon...	MeanTi...	Risk
azus	IAAS	PRIVATE	STORA...	WINDEV	13	5	14	60	100	50	80	16
WINMAX	IAAS	PRIVATE	STORA...	AZURE	14	5	16	56	134	89	78	12
AZURE	IAAS	PRIVATE	STORA...	GOOG...	12	10	16	50	150	13	60	18
azur	IAAS	PRIVATE	STORA...	GMAIL	14	10	16	60	100	45	80	18

Filtrage:

Standard Filtrage
 Filtrage By Disponibility
 FiltrageByExperience
 Filtrage By Reputation

Filtrage By Risk
 Filtrage By MITR
 FiltrageByResponse
 Filtrage By Cost
 Filtrage By Trust

Service...	CloudD...	CloudD...	Funcio...	Reputa...	Experie...	Trust	Disponi...	Paymen...	Respon...	MeanTi...	Risk
WINMAX	IAAS	PRIVATE	STORA...	14	5	16	56	134	89	78	12
AZURE	IAAS	PRIVATE	STORA...	12	10	16	50	150	13	60	18
azur	IAAS	PRIVATE	STORA...	14	10	16	60	100	45	80	18

Figure 54: filtrage par confiance

Le bouton « FiltrageByDisponibility » sélectionne les services qui ont la même fonctionnalité principale et retourne le ou les services qui ont la plus grande disponibilité (voir figure 55)

ICLOUD

Research Of Service

Service's function main:

Results:

Service...	Cloud...	Cloud...	Funcio...	Provider	Reputa...	Experie...	Trust	Dispon...	Payme...	Respon...	MeanTi...	Risk
azus	IAAS	PRIVATE	STORA...	WINDEV	13	5	14	60	100	50	80	15
WINMAX	IAAS	PRIVATE	STORA...	AZURE	14	5	16	56	134	89	78	12
AZURE	IAAS	PRIVATE	STORA...	GOOG...	12	10	16	50	150	13	60	18
azur	IAAS	PRIVATE	STORA...	GMAIL	14	10	16	60	100	45	80	18

Filtrage:

Standard Filtrage
 Filtrage By Disponibility
 FiltrageByExperience
 Filtrage By Reputation

Filtrage By Risk
 Filtrage By MTTR
 FiltrageByResponse
 Filtrage By Cost
 Filtrage By Trust

Service...	CloudD...	CloudD...	Funcio...	Reputati...	Experie...	Trust	Disponi...	Paymen...	Respon...	MeanTi...	Risk
azus	IAAS	PRIVATE	STORA...	13	5	14	60	100	50	80	16
azur	IAAS	PRIVATE	STORA...	14	10	16	60	100	45	80	18

Figure 55:filtrage par Disponibilité

Le bouton « FiltrageByCost » sélectionne les services qui ont la même fonctionnalité principale par rapport au prix du service et retourne le ou les services qui ont le plus petit prix (voir figure 56)

ICLOUD

Research Of Service

Service's function main:

Results:

Service...	Cloud...	CloudD...	Funcio...	Provider	Reputa...	Experie...	Trust	Dispon...	Payme...	Respo...	MeanTi...	Risk
azus	IAAS	PRIVATE	STORA...	WINDEV	13	5	14	60	100	50	80	16
WINIMAX	IAAS	PRIVATE	STORA...	AZURE	14	5	16	56	134	89	78	12
AZURE	IAAS	PRIVATE	STORA...	GOOG...	12	10	16	50	150	13	60	18
azur	IAAS	PRIVATE	STORA...	GMAL	14	10	16	60	100	45	80	18

Filtrage:

Standard Filtrage
 Filtrage By Disponibility
 FiltrageByExperience
 Filtrage By Reputation
 Filtrage By Risk
 Filtrage By MITR
 FiltrageByResponse
 Filtrage By Cost
 Filtrage By Trust

Service...	CloudD...	CloudD...	Funcio...	Reputali...	Experie...	Trust	Disponi...	Paymen...	Respon...	MeanTi...	Risk
azus	IAAS	PRIVATE	STORA...	13	5	14	60	100	50	80	16
azur	IAAS	PRIVATE	STORA...	14	10	16	60	100	45	80	18

Figure 56:filtrage par prix

Le bouton « FiltrageByResponse » sélectionne les services qui ont la même fonctionnalité principale et retourne le ou les services qui ont le plus petit temps de réponse (voir figure 57).

ICLOUD

Research Of Service

Service's function main:

Results:

Service...	Cloud...	Cloud...	Funcio...	Provider	Reputa...	Experie...	Trust	Dispon...	Payme...	Respo...	MeanTi...	Risk
azus	IAAS	PRIVATE	STORA...	WINDEV	13	5	14	60	100	50	80	16
WINIMAX	IAAS	PRIVATE	STORA...	AZURE	14	5	16	56	134	89	78	12
AZURE	IAAS	PRIVATE	STORA...	GOOG...	12	10	16	50	150	13	60	18
azur	IAAS	PRIVATE	STORA...	GMAIL	14	10	16	60	100	45	80	18

Filtrage:

Standard Filtrage
 Filtrage By Disponibility
 FiltrageByExperience
 Filtrage By Reputation
 Filtrage By Risk
 Filtrage By MTTR
 FiltrageByResponse
 Filtrage By Cost
 Filtrage By Trust

Service...	CloudD...	CloudD...	Funcio...	Reputali...	Experie...	Trust	Disponi...	Paymen...	Respon...	MeanTi...	Risk
AZURE	IAAS	PRIVATE	STORA...	12	10	16	50	150	13	60	18

Figure 58: filtrage par MTTR

Le bouton « FiltrageByRisk » sélectionne les services qui ont la même fonctionnalité principale et retourne le ou les services qui ont moins de risque (voir figure 59).

ICLOUD

Research Of Service

Service's function main: storage

Results:

Service...	Cloud...	Cloud...	Funcio...	Provider	Reputa...	Experie...	Trust	Dispon...	Payme...	Respo...	MeanTi...	Risk
azus	IAAS	PRIVATE	STORA...	WINDEV	13	5	14	60	100	50	80	16
WINMAX	IAAS	PRIVATE	STORA...	AZURE	14	5	16	56	134	89	78	12
AZURE	IAAS	PRIVATE	STORA...	GOOG...	12	10	16	50	150	13	60	18
azur	IAAS	PRIVATE	STORA...	GMAIL	14	10	16	60	100	45	80	18

Filtrage:

Standard Filtrage
 Filtrage By Disponibility
 FiltrageByExperience
 Filtrage By Reputation

Filtrage By Risk
 Filtrage By MITR
 FiltrageByResponse
 Filtrage By Cost
 Filtrage By Trust

Service...	CloudD...	CloudD...	Funcio...	Reputati...	Experie...	Trust	Disponi...	Paymen...	Respon...	MeanTi...	Risk
WINMAX	IAAS	PRIVATE	STORA...	14	5	16	56	134	89	78	12

Figure 59:filtrage par risque

IV.4 Conclusion:

Dans ce chapitre, nous avons illustré une expérimentation de notre solution. Nous avons essayé de mettre en œuvre l'ensemble des idées et concepts qui caractérisent le modèle proposé.

Notre solution a été implémentée en langage JAVA dans l'environnement de développement Netbeans.

Conclusion générale :

Le cloud computing offrent différents options (infrastructure, Plateformes, logiciels) en tant que services, il permet d'augmenter l'évolutivité des applications et de réduire le coût de l'infrastructure dans l'entreprise. Malgré ces potentiels bénéfiques, les processus de découverte, de sélection et de composition des services cloud posent d'énormes problèmes aux consommateurs, cela est dû au manque de standard pour la description des services cloud.

Le but de notre travail est de proposer une solution de description générique des services Cloud. Pour atteindre cet objectif nous avons commencé dans un premier temps par étudier les notions de base du Cloud Computing, ses modèles de déploiement, ses caractéristiques, ses types de services, ses avantages et inconvénients, les notions d'inter-cloud et multi-cloud afin de nous familiariser avec le cloud computing.

Nous avons fait par la suite une étude comparative entre les langages de description WSDL et USDL, notre choix du langage a été porter sur USDL en raison de sa richesse par rapport à WSDL (USDL permet de décrire les aspects techniques, opérationnels et commerciaux tandis que WSDL ne permet que la description technique des services). Nous avons ensuite exploré la notion de description à travers une étude bibliographique des différents travaux connexes en relation avec notre problématique, suivie d'une analyse critique. Cette exploration nous a permis de bien comprendre les enjeux de la description, et les défis.

Après avoir présente certaines exigences nécessaires pour une bonne description définis par différents auteurs, nous avons proposé de nouvelles exigences pour une meilleure description des services cloud. Le manque de l'aspect sémantique de l'USDL a fait que nous avons utilisé les ontologies, pour cela nous

avons utilisé le langage d'ontologie OWL-S. Pour illustrer notre proposition nous avons décrit trois types de services cloud réels.

Pour l'expérimentation de la solution proposée, nous avons créé une application java sur Netbeans. Notre application permet entre autre la découverte, la création et la sélection des services Cloud.

Comme perspective de notre travail, notre description peut être utilisée dans un processus de portabilité ou d'interopérabilité entre plusieurs fournisseurs de services cloud.

References:

- [1] T.Gopikiran, S. Satyanarayana, and A. Srinivasa. Cloud Computing : IAAS, (2013) vol(2), pages (775-780).p
- [2] L. Schubert, K. Jefery and B .Neidecker-Lutz. The Future of Cloud Computing - Opportunities for European Cloud Computing Beyond 2010, Expert Group report, (2010).
- [3] Q. Zhang, L. Cheng, and R. Boutaba, Cloud Computing: State of The Art and Research Challenges, Journal of Internet Services and Applications, (2010), Vol (11).
- [4] M. Armbrust, A .Fox, R .Griffith, A . Joseph , R. Katz , A. Konwinski , G .Lee , D .Patterson, A . Rabkin, I . Stoica and M . Zaharia . Above the Clouds: A Berkeley View of Cloud Computing, Technical Report, (2009).
- [5] I. Mezgar and U. Rauschecker .The challenge of networked enterprises for cloud computing interoperability, Computer in Industrie, Elsevier (2014), vol(65), no.4 , pages(657-674).
- [6] Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner. The break in the clouds: towards a cloud definition, ACM SIGCOMM Computer Communication Review, ACM (2008), vol(39), no.1, pages (50- 55).
- [7] B.Kezia-Rani, B. Padmaja-Rani and A.Vinaya-Babu. Cloud Computing and Inter-Clouds - Types, Topologies and Research Issues, Proceida Computer Science, Elsevier, (2015), vol (50), pages (24-29).
- [8] D. Ardagna. Cloud and Multi-Cloud Computing : Current Challenges and Future Applications, Proceedings of the Seventh International Workshop on Principles of Engineering Service-Oriented and Cloud Systems, IEEE Press, (2015),pages(1-2).

- [9] P. Mell and T. Grance. The NIST Definition of Cloud Computing, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, (2011).
- [10] S. Kaur, S. Sumesh and K. Gurminder. Cloud Computing Interoperability: Introduction, Concerns and Challenges, International Journal of Advanced Research in Computer Science, (2017), vol (8), no5.
- [11] M. Alzubaidi. Cloud Computing Antecedents, Challenges, and Directions, Proceedings of the International Conference on Internet of things and Cloud Computing, ACM, (2016).
- [12] A. Juan-Ferrer. Inter-cloud Research: vision for 2020, Proceeding Computer Science, Elsevier, (2016), vol (97), pages (140-143).
- [13] E. Nogueira, A. Moreira, D. Lucedio, V. Garcia and R. Fortes. Issues on developing interoperable cloud applications: definitions, concepts, approaches, requirements, characteristics and evaluation models, journal of Software Engineering Research and Development, (2016).
- [14] P. Ram-Prasad. Load balancing in Cloud Computing systems, National Institute of Technology, (2011).
- [15] Z. Kartit, H. Kamal and M. El-Marraki. Improvement of algorithm for updating firewall policies, Theoretical and Applied Information Technology, (2016), vol (66), no1.
- [16] Techopedia.com. <http://www.techopedia.com/definition/7756/intercloud-Definition>, dernière consultation : [03/09/2018].
- [17] sdxcentral. <http://www.sdxcentral.com/sdn/networkvirtualization/definitions/what-is-intercloud/>, dernière consultation : [03/09/2018].
- [18] Lebigdata.fr <http://www.lebigdata.fr/multi-cloud-3012>, dernière consultation : [03/09/2018].

[19] J. Zhou , N. Abdullah, and Z. Shi. A hybrid P2P approach to service discovery in the Cloud. In International Journal of Information Technology and Computer Science (IJITCS), MECS, (2011), vol. (3), no1, pages (1-9).

[20] U. Keller, R. Lara, H . Lausen, A. Polleres, and D. Fensel. Automatic location of services. In the 2nd European SemanticWeb Conference (ESWC), (2005), pages(1-16)

[21] K. Bentlemsan, D. Bennoua and D. Tamzalit. A hybrid re-composition based on components and web services. In international journal of computers and applications, (2018), pages (1-14).

[22] L. Zeng and B. Benatallah. Quality driven web service composition. Proceeding of the 12th international conference on World Wide Web, (2003), pages (411-421).

[23] V. Giudicelli. OWL language application to the formalization of description concepts ontology with the Protected Editor. (2011)

[24] S. Ghazouani and Y. Slimani . Towards a standardized cloud service description based on USDL. The Journal of Systems & Software, (2017).

[25] E. Christensen and F. Curbera . Web Services Description Language (WSDL) version 1.1 (2001).

[26] Web Services Description Language (WSDL) version 1.1. , W3C <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>, dernière consultation: [18/08/2018]

[27] Web Services Description Language (WSDL) Version 2.0 Part 1: W3C. <http://www.w3.org/TR/2006/CR-wsdl20-20060327>, [dernière consultation: 18/08/2018]

[28] WSDL binding. W3C Working Draft 15 February (2005); <http://www.w3.org/TR/2005/WD-ws-adr-wdl-20050215.derniere> consultation: [12/09/2018].

[29] Jorge Cardoso. Modeling Services with α – USDL. (2012), pages (15-50).

[30] Daniel Oberle, Alistair Barros, Uwe Kylau, Steffen Heinz. A unified description language for human to automated services, Elsevier, (2012), pages (155-181).

[31] M. Zhang, R. Ranjan, and A. Haller. An ontology based system for Cloud infrastructure services' discovery. In 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), Pittsburgh, PA, IEEE, (2012), pages (524-530).

[32] Bu Sung Lee, Shixing Yan, Ding Ma, Guopeng Zhao. Aggregating IaaS Service. In 2011 Annual SRII Global Conference, San Jose, CA, IEEE, (2011), pages (335-338).

[33] B. Rochwerger, D. Breitgand and E. Levy. The reservoir model and architecture for open federated Cloud Computing. In IBM Journal of Research and Development (2009), vol (53), no. 4, pages (1-4).

[34] E. Carlini, M. Coppola and P. Dazzi. Cloud federations in contrail. In European conference on Parallel Processing, Berlin Heidelberg, Springer, (2011), pages (159–168).

[35] M. Hamdaqa, T. Livogiannis and L. Tahvildari. A reference model for developing Cloud applications. In 1st International Conference on Cloud Computing and Services Science, SciTePress, (2011), pages (98-103).

[36] Y.L .Sun, T. Harmer and A. Stewart . Mapping application requirements to Cloud resources. In European Conference on Parallel Processing, Berlin Heidelberg, Springer, (2011), pages (104-112).

[37] S. Gudenkauf, M. Josefiok, and A. Goring. A reference architecture for Cloud service offers. In 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC), Vancouver, BC, IEEE, (2013), pages (227-236).

[38]G. Goncalves, P. Endo and M. Santos. CloudML: An Integrated Language for Resource, Service and Request Description for D-Clouds. In IEEE 3rd International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, (2011), pages (399-406).

[39] D. Liu and J. Zic. Cloud#: A specification language for modeling Cloud. In 2011 IEEE International Conference on Cloud Computing , Washington, DC, IEEE, (2011), pages (533-540).

[40] S. Garcia-Gomez, M. Escriche-Vicente, and M. Jiménez- Ganan. 4CaaS: Comprehensive management of Cloud services through a PaaS. In 2012 10th IEEE International Symposium on Parallel and Distributed Processing with Applications, Leganes, Madrid, Spain, IEEE, (2012), pages (494-499).

[41] P. Papazoglou and W. Heuvel . Cloud Blueprinting . In IEEE Internet Computing (2011) vol (15) no.6 pages (74-79).

[42] A. Elgammal, O. Türetken, W. Heuvel and P. Papazoglou. Root-Cause Analysis of Design-Time Compliance Violations on the Basis of Property Patterns, ICSOC, (2010) , Pages (17-31).

[43] C. Quinton, D. Romero, D and L. Duchien. Automated selection and configuration of Cloud environments using software product lines principles. In

IEEE 7th International Conference on Cloud Computing, Anchorage, USA, IEEE, (2014), pages (144-151).

[44] G. Modica, G. Petralia and O. Tomarchio. Business ontology to enable semantic matchmaking in open Cloud markets. In 8th International Conference on Semantics, Knowledge and Grids (SKG), Beijing, China, IEEE, (2012), pages (96-103).

[45] G. Modica and O. Tomarchio. Matching the business perspectives of providers and customers in future Cloud markets. In Cluster Computing, New York, Springer, (2014), vol (18), no. 1, pages (457-475).

[46] A. Goscinski and M. Brock. Toward dynamic and attribute based publication, discovery and selection for Cloud Computing. In Future Generation Computer Systems, (2010), vol (26), no. 7, pages (947-970).

[47] E. Christensen . Web Services Description Language (WSDL), Version 1.1, (2001).

[48] S. Ghazouani and Y. Slimani. A survey on cloud service description. Journal of Network and Computer Applications, (2017).

[49] L. Sun, J. Ma and H. Wang. Cloud Service Description Model: An Extension of USDL for Cloud Services. In IEEE Transactions on Services Computing, (2015).

[50] D. Nguyen, F. Lelli and P. Papazoglou . Blueprinting Approach in Support of Cloud Computing. In Future Internet, Molecular Diversity Preservation International, (2012) vol (4), no. 1, pages (322-346).

[51] F. Amato and F. Moscato . A Modeling Profile for Availability Analysis of Composite Cloud Services. In 9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), IEEE, (2014), pages (406-413).

[52] P .Hoberg, J .Wollerheim, H . Krcmar . Service Description for cloud Service – the customer’s perspective. Proceedings of Conlife Academic Conference, Cologne, (2012)

[53] A.Tahamtan, S. Beheshti and A. Anjomshoaa . A Cloud repository and discovery framework based on a unified business and Cloud service ontology. In 8th IEEE World Congress on Services, Honolulu, Hawaii, IEEE, (2012), pages (203-210).

[55] Protégé, <https://protege.stanford.edu>, dernière consultation : [12/09/2018].

[56] OWL-S, <https://www.w3.org/Submission/2004/SUBM-OWL-S20041122/>, dernière consultation : [12/09/2018].

[57] Itpro.fr, <https://www.itpro.fr/le-contrat-d-engagement-de-service-et-ses-indicateurs>, dernière consultation : [10/08/2018].

[58] Cisco.com, https://www.cisco.com/Cisco_IaaS_SLA, dernière consultation : [12/08/2018].

[59] Sla Microsoft Azure, http://azure.microsoft.com/fr-fr/support/legal/sla/cloud-service/v1_5. Dernière consultation : [08/09/2018].

[60] Lemagit.fr, <https://Lemagit.fr/conseil/SLA-des-SaaS-ce-que-doivent-inclure-les-accords>, dernière consultation : [11/09/2018].

[61] Amazon EBS. <https://aws.amazon.com/fr/ebs> Dernière consultation : [10/09/2018].

[62] Microsoft SharePoint Online <https://support.office.com/fr-fr/article/qu-est-ce-que-sharepoint> Dernière consultation : [10/09/2018].

[63] Microsoft Sharepoint <https://www.megabyte.be/logiciel-de-gestion/microsoft-sharepoint> Dernière consultation : [10/09/2018].

[64] Amazon EBS details VolumeTypes. <https://aws.amazon.com/fr/ebs/details/#VolumeTypes>, Dernière consultation : [10/09/2018].

[65] Amazon EBS detail Availability and Durability <https://aws.amazon.com/fr/ebs/details/#AvailabilityandDurability> , Dernière consultation: [10/09/2018].

[66] Amazon EBS details EncryptionandAccessManagement, <https://aws.amazon.com/fr/ebs/details/#EncryptionandAccessManagement> , Dernière consultation: [10/09/2018].

[67] Amazon EBS detail Snapshot. <https://aws.amazon.com/fr/ebs/details/#snapshots> , Dernière consultation: [10/09/2018].

[68] Amazon EBS detail ElasticVolumes. <https://aws.amazon.com/fr/ebs/details/#elasticvolumes> Dernière consultation : [10/09/2018].

[69] langage java définition et historique. http://www.comoria.com/3516/Java_%28langage%29. [Dernière consultation : 20/08/2018].

[70] TheServerSide. “Netbeans”. <http://www.theserverside.com/definition/NetBeans>
Dernière consultation : [20/08/2018].

[71] Xamp server. https://filehippo.com/fr/download_xampp/ Dernière consultation : [20/08/2018].

[72] K. Iman-Bamba .Les fondements du cloud computing (2014).

[73] Techno-Science.net “eXtreme Programming” .<http://www.techno-science.net/?onglet=glossaire&definition=723>.derniere consultation : [12/09/2018]

[74] Piloter.org. “XP Extreme Programming, les méthodes agiles”.
<https://www.piloter.org/projet/methode/xp.htm> . Dernière consultation: [12/09/2018]

[75] R. Medina . L' eXtreme Programming (2005).

Annexes

ANNEXE A : BALISES WSDL

- Types: Contient la définition de types utilisant un système de typage (comme XSD).

L'élément <types> décrit tous les types de données utilisés entre le client et le serveur. Ces types sont l'équivalent en structures C++ ou Java à des classes qui ne contiennent que des données et pas de méthodes. WSDL n'est pas liée exclusivement à un système de typage, mais il utilise le XML schéma de la spécification W3C.

- Message: Décrit les noms et types d'un ensemble de champs à transmettre

Chaque élément enfant <part> de l'élément <message> correspond à un paramètre et possède un attribut de nom et de type, tout comme un paramètre de fonction à un nom et un type. Les paramètres d'entrée sont définis dans un élément <message> unique et séparé des paramètres de sortie, qui se trouvent dans leur propre élément <message>.

Le nom d'un élément <message> de sortie se termine par Response.

- Opération: une description abstraite d'une action supportée par les services.

L'élément <operation> est analogue à un appel de méthode en Java. La différence est que seulement trois messages sont autorisés dans une opération.

Input Message : définit les données que le service Web s'attend à recevoir.

Output Message : définit les données que le service Web prévoit d'envoyer en réponse.

Fault Message : définit les messages d'erreurs qui peuvent être retournés par le service Web.

- Port Type: Décrit un ensemble d'opérations. Chaque opération à zéro ou un message en Entrée, zéro ou plusieurs messages de sortie ou de fautes.

L'élément <portType> contient l'ensemble des opérations que peut effectuer un service Web. Cependant, il ne fournit pas d'informations sur la façon de se connecter directement à ce service. Il prévoit un point d'arrêt où un client peut obtenir des informations sur tous les traitements offerts par un service Web .

- Binding: Spécifie une liaison d'un <port type> à un protocole concret (SOAP1.1, HTTP1.1, ...). Un port type peut avoir plusieurs liaisons.

L'élément <binding> permet d'obtenir les informations nécessaires pour connecter physiquement un service Web. Il décrit les spécifications concrètes de la manière dont le service sera implémenté : protocole de communication et format des données pour les opérations et messages définis par un portType particulier.

L'élément <binding> a deux objectifs. Tout d'abord, il sert de lien entre les éléments abstraits et les éléments concrets dans le WSDL. Ensuite, il fournit un conteneur pour des informations telles que le protocole et l'adresse du service Web.

- Port: Spécifie un point d'entrée comme la combinaison d'un <binding> et d'une adresse réseau.

L'élément port contient deux attributs : l'attribut name et l'attribut binding. L'attribut name donne un nom unique parmi tous les ports définis dans le document WSDL.

L'attribut binding fait référence à l'élément défini dans la section binding du document WSDL. Un port ne doit pas définir plus d'une adresse. Un port ne doit pas définir d'autres informations de liaisons autres que celles de l'adresse.

- Service: Une collection de points d'entrée relatifs [26]

L'élément <service> définit les ports soutenus par le service Web. Il contient aussi un élément <documentation> qui fournit la documentation lisible par l'homme. Pour chaque liaison supportée, un élément port est désigné. L'élément <service> n'est donc qu'une simple collection de ports.

ANNEXE B : les informations contenus dans les SLA

➤ Les informations contenues dans les SLA génériques

Un SLA sert à la fois de modèle et de garantie pour le cloud Computing. Dans un certain sens, le contrat de niveau de service fixe des attentes pour les deux parties et sert de feuille de route pour le changement du service de cloud computing, à la fois les changements attendus et les surprises.

Afin de développer de manière cohérente un SLA efficace, une liste de critères importants doit être établie tels que la disponibilité, les performances, la sécurité et confidentialité des données, les attentes de reprise après sinistre, la localisation des données, l'accès aux données, la portabilité des données, le processus d'identification des problèmes et de résolution des attentes, le processus de médiation des différends [57].

➤ Les informations contenues dans les SLA IAAS

Les contrats de niveau de service IaaS contiennent généralement [57] [58] [61]:

Le Temps d'indisponibilité du service, période pendant laquelle le système était indisponible alors qu'il devait être en état de disponibilité ;

Les Crédits de service qui constituent un moyen de recours pour tout problème de performance ou de disponibilité des services ;

Le temps de service, temps pendant lequel les prestations convenues par contrat sont fournies ;

Le taux d'erreur désignant le nombre total d'erreurs du serveur interne retourné divisé par le nombre total de demandes pendant une période fixée par le fournisseur ;

La durée de fonctionnement, temps pendant lequel le système est disponible en principe.

➤ Les informations contenues dans les SLA PAAS

Les contrats de niveau de service PaaS contiennent généralement [57] [59]:

Le portail de gestion qui permet de gérer les applications ;

Les ressources du service qui concernent les différentes ressources que le service PaaS est capable de fournir aux consommateurs ;

Le code de succès, pour indiquer la réussite d'une opération et le code d'erreur, pour indiquer des erreurs éventuelles. Chaque SLA doit pouvoir donner également une estimation des cas d'incidents du service.

➤ **Les informations contenues dans les SLA SAAS**

Les contrats de niveau de service SaaS contiennent généralement [57] [60], [62], [63] :

Les pénalités : Si le fournisseur SaaS n'atteint pas les exigences de disponibilité ou les garanties, des pénalités claires et spécifiques doivent être définies.

Escalade : Une clause d'escalade doit être incluse dans tous les contrats. Des processus clairs doivent être mis en place pour résoudre les problèmes contractuels, en particulier ceux liés à l'adhésion au SLA.

Rapport : Les clauses de SLA devraient également stipuler que les fournisseurs SaaS doivent fournir des rapports mensuels et /ou hebdomadaires sur les indicateurs clés de disponibilité, de continuité et de performance. Des réunions SLA régulières devraient être organisées pour examiner ces informations.

Résiliation : Chaque contrat d'abonnement SaaS Cloud doit inclure une disposition permettant à l'entreprise de se résilier en cas d'incapacité grave ou continue à satisfaire aux exigences de niveau de service établies. Il ne devrait y avoir aucune pénalité anticipée liée à la résiliation pour non-respect des SLA et les obligations du fournisseur en cas de résiliation doivent être clairement énoncées.

