

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



**Mémoire de fin d'études en vue de l'obtention du  
Diplôme de Master  
Filière : Informatique  
Option : Système Informatique et Réseaux**

Thème :

**Solution pour la substitution  
des services Cloud de type SaaS**

Présenté par :

**Amrani Khaoula**

Devant le jury :

**Mme. Boutoumi Bachira (Présidente)**

**Mme. Chikhi Imane (Examinatrice)**

**Mme. Mancer Yasmine (Promotrice)**

Année universitaire : **2019/2020**

## Dédicaces

Avant tout *Alhamdoulilah*

♥ *A mes chères Parents*

♥ *A Mon chère oncle Doua*

*Sid Ahmed*

♥ *A mes chères Sœurs*

*AIRAMS*

♥ *A moi même*

## Remerciements

*La louange appartient à الله, je le remercie de m'avoir donné la force, la volonté et le courage d'accomplir ce modeste travail.*

*Je tiens à remercier et à exprimer ma profonde gratitude à ma promotrice **Madame Mancer Yasmine**, de m'avoir fait confiance et accepter de m'encadrer au long de ce travail, pour sa disponibilité toujours en offrant ses conseils, ses motivations, pour son écoute et ses encouragements, ainsi que son orientation qui m'a afflué.*

*Je remercie autant les **membres du jury** qui me fait honneur en acceptant d'évaluer mon travail.*

*Ma reconnaissance s'adresse à mes **chères parents**, à ma **chère famille** qui ont su m'apporter, sans relâche leurs soutien durant toutes ces longues années d'études.*

*Enfin je remercie **tous** ceux qui ont coopérés de près ou de loin à ce que ce travail puisse voir le jour.*

## ملخص:

أصبحت الحوسبة السحابية الآن جزءًا من حياتنا اليومية ، على سبيل المثال ، إذا التقط المستخدمون صورة بهواتفهم الذكية ، فسيتم تحميلها تلقائيًا على سحابة Apple أو Google بحيث يمكن الوصول إليها مرة أخرى باستخدام الأجهزة الأخرى. تتضمن الحوسبة السحابية العديد من الخدمات المستأجرة عن بعد من قبل الموردين ، والتي توفر المرونة أثناء استخدامها ، بحيث لا يضطر العميل إلى القلق بشأن تثبيت برامج جديدة أو منصة جديدة للاستفادة من عروضها.

بشكل عام ، يهتم العميل بوظائف الخدمات المحددة في عملياته والتي تعمل في بيئة متقلبة تعمل فيها الخدمات السحابية ، وقد يواجه هذا الأخير عطلًا أثناء استخدامها ، مما يجعل من الضروري العثور على خدمة أخرى تؤدي نفس الشيء.

تقدم هذه الأطروحة حلاً لاستبدال الخدمات السحابية من خلال اكتشاف التشابه بين عمليات الخدمة وتصنيفها وفقًا لوظائفها ، كما تقدم أيضًا تنبؤًا بفشل الخدمة الذي يسمح للعملاء بتجنب الخدمات المعيبة.

### الكلمات الرئيسية :

الحوسبة السحابية ، الاستبدال ، الوصف ، الاختيار ، البرمجيات كخدمة (SaaS) ، جودة الخدمة QoS

## **Résumé**

Désormais le Cloud Computing fait partie de notre vie habituelle, par exemple si les utilisateurs prennent une photo par leurs smart phones, le chargement est automatique dans le Cloud Apple ou Google pour pouvoir y accéder à nouveau avec d'autres appareils. Le Cloud Computing comprend plusieurs services loués à distance par des fournisseurs, qui offrent une flexibilité durant leurs utilisations, tel que le client ne se soucie pas de l'installation d'un nouveau logiciel ou une nouvelle plateforme pour bénéficier de ses offres.

En général, un client s'intéresse aux fonctionnalités des services déterminées dans leurs opérations, qui s'exécutent dans un environnement volatile dans lequel les services Cloud fonctionnent, ces derniers peuvent rencontrer un dysfonctionnement pendant leurs utilisations, ce qui oblige à trouver un autre service effectuant la même chose.

Ce mémoire propose une solution pour la substitution des services cloud en détectant la similarité entre les opérations des services et les classifier selon leurs fonctionnalités, il présente aussi une prédiction d'échec du service qui permet aux clients d'éviter les services erronés.

### **Mots clés :**

Cloud Computing, substitution, description, sélection, Software as a Service (SaaS), qualité de service (QoS).

## **Abstract**

Cloud computing is now part of our daily life, for example if users take a photo with their smart phones, it is automatically uploaded to the Apple or Google cloud so that they can be accessed again with other devices. Cloud computing includes several services rented remotely by suppliers, which offer flexibility during their use, such that the customer does not have to worry about installing new software or new platform to benefit from its offerings.

In general, a customer is interested in the functionalities of the services determined in their operations which run in a volatile environment in which the Cloud services operate, the latter may encounter a malfunction during their use, which makes it necessary to find another service performing the same thing.

This thesis offers a solution for cloud service substitution by detecting similarity between service operations and classifying them according to their functionality, it also presents a service failure prediction that allows customers to avoid faulty services.

### **Keywords :**

Cloud Computing, substitution, description, selection, Software as a Service (SaaS), quality of service (QoS).

# Table des matières

## Introduction générale

1	Contexte	1
2	Problématique	1
3	Objectifs	1
4	Structure du mémoire	2

## Chapitre I Généralités sur le Cloud Computing

1	Introduction	3
2	Définition du Cloud Computing	3
3	Eléments constitutifs du Cloud Computing	3
3.1	Clients	4
3.2	Centre de données	4
3.3	Serveurs distribués	4
4	La virtualisation	4
4.1	Types de virtualisation	4
4.1.1	Para-Virtualisation	4
4.1.2	Virtualisation des conteneurs	5
4.1.3	Virtualisation complète	5
4.2	But de la virtualisation	5
5	Caractéristiques du Cloud Computing	5
6	Types de services dans les environnements du Cloud Computing	6
6.1	Infrastructure en tant que service	6
6.2	Plateforme en tant que service	7
6.3	Logiciel en tant que service	7
6.4	Autres types de services Clouds	8
7	Modèles de déploiement du Cloud Computing	9
7.1	Cloud Public	9
7.2	Cloud Privé	9
7.3	Cloud Communautaire	10
7.4	Cloud Hybride	10
8	Architecture de référence du Cloud Computing	10
8.1	Déploiement du service	10
8.2	Service d'orchestration	11
8.3	Gestion de service en Cloud	12
8.4	Sécurité	15
8.5	Vie privée	15
9	Acteurs du Cloud Computing	15
9.1	Consommateur de cloud	15
9.2	Fournisseur de cloud	16
9.3	Développeur de cloud	16

9.4	Courtier de cloud	16
9.5	Auditeur de cloud	16
9.6	Transporteur de cloud	16
10	Relations entre les acteurs du Cloud	16
11	Evolution du Cloud Computing	18
11.1	Inter-Cloud	18
11.2	Multi-Cloud	20
12	Avantages et Inconvénients du Cloud Computing	22
13	Conclusion	24

## **Chapitre II Etude de la substitution des services Cloud**

1	Introduction	25
2	Concepts fondamentaux des services	25
3	Définition de service web	26
3.1	Type de de service web	26
4	Caractéristiques de la qualité de service	27
4.1	Type de critère de QoS :	28
5	Règles de SLA	28
6	Description de Service	28
6.1	Description structurelle	28
6.2	Description comportementale	29
7	Découverte de service	29
8	Communication des services web :	29
9	Composition des services	29
9.1	La composition des services Web	30
9.2	Cycle de vie d'une composition des services :	30
10	Sélection de service	30
11	Substitution de service Web	31
11.1	La substitution adaptative des services	31
11.2	Adaptation du service	31
11.3	Calcul de la similarité	32
11.3.1	Analyse syntaxique	32
11.3.2	Analyse sémantique	32
12	Etude comparative entre les solutions de la substitution des services cloud de type SaaS	34
12.1	Approche de description des services :	34
12.2	Approche de sélection des services :	34
12.3	Approche de composition des services :	35
12.4	Approche de substitution des services :	36
12.4.1	Synthèse des travaux sur le calcul de similarité des services	36
12.4.2	Synthèse des travaux sur la substitution des services	38
12.4.2.1	Solution à base communauté	38
12.4.2.2	Solution à base réseau	41



12.4.2.3 Solution à base SLA-----	43
12.4.3 Critères d'évaluation :-----	45
13 Conclusion -----	47

### **Chapitre III Modélisation de la substitution des services Cloud**

1 Introduction -----	48
2 Méthodologie de travail-----	48
3 Analyse des besoins -----	49
4 Solution proposée -----	51
5 Description des services Cloud de type SaaS-----	53
5.1 Structure du fichier WSDL : -----	53
6 Sélection des services cloud de type SaaS -----	56
6.1 Exemple de MCDM : -----	58
7 Substitution des services cloud de type SaaS-----	59
7.1 Prédiction ( Substitution à base SLA) -----	59
7.2 Substitution lexicale (analyse sémantique): -----	63
7.2.1 Exemple de distance « Jaro-Winkler »-----	63
7.3 Substitution globale (à base communauté) :-----	65
7.3.1 Exemple de l'algorithme K-means -----	65
7.4 Substitution précise (à base réseau ) -----	66
7.4.1 Représentation graphique de la matrice de similarité-----	67
7.4.2 Représentation matricielle du graphe :-----	68
8 Conclusion -----	69

### **Chapitre IV Expérimentation**

1 Introduction -----	70
2 Outils de développement-----	70
3 Simulation-----	72
4 Comparaison entre les substitutions proposées-----	98
5 Evaluation de la solution proposée -----	99
6 Conclusion -----	100
Conclusion Générale-----	101

<b>Références</b> -----	102
-------------------------	-----

<b>Annexe A : Algorithme K-Means</b> -----	110
--	-----

# Table des figures

Figure 1: L'hierarchie des services cloud [89] -----	8
Figure 2: Déploiement des services cloud [90] -----	9
Figure 3: Public cloud [91] -----	10
Figure 4: Service d'orchestration [26]-----	11
Figure 5: Gestion des services cloud [26] -----	12
Figure 6: Relation entre le cloud broker et le cloud consumer [26] -----	17
Figure 7: Le rôle du cloud provider [26]-----	17
Figure 8: Le rôle du cloud auditor [26] -----	18
Figure 9: Inter-cloud [92] -----	20
Figure 10: Multi-cloud [93]-----	21
Figure 11:Diagramme de cas d'utilisation -----	50
Figure 12: Plan de travail de notre solution pour la substitution-----	52
Figure 13: La structure du fichier WSDL [94]-----	53
Figure 14: Partie abstraite du service-----	54
Figure 15: Partie concrète du service-----	55
Figure 16:SLA du Service Multiplication -----	61
Figure 17:Prédiction du Service Multiplication -----	62
Figure 18: Réseau de similarité -----	68
Figure 19: Home-----	72
Figure 20: Inscription de fournisseur-----	73
Figure 21: Connexion de fournisseur-----	74
Figure 22: Interface création des services -----	75
Figure 23: Suite de la création des services -----	75
Figure 24: Importation du fichier WSDL -----	76
Figure 25: Téléchargement réussi du fichier WSDL -----	76
Figure 26: Liste des services-----	77
Figure 27: Inscription client -----	78
Figure 28: Connexion de client-----	78
Figure 29: Chercher un service-----	79
Figure 30: Erreur de recherche -----	79
Figure 31: Résultat de la recherche -----	80

Figure 32: La recherche du meilleur service -----	81
Figure 33: Résultat de la méthode MCDM-----	82
Figure 34: Combinaison de service ABS -----	84
Figure 35: Combinaison de service Cal-----	84
Figure 36: Combinaison de Dessin -----	85
Figure 37: Combinaison de FM -----	85
Figure 38: Combinaison de IAN -----	85
Figure 39: Combinaison de Mul -----	86
Figure 40: Combinaison de QTE -----	86
Figure 41: Option de substitution -----	86
Figure 42: Menu de substitution -----	87
Figure 43: Résultat de la méthode Wu-Palmer -----	88
Figure 44: Résultat de la méthode K-means -----	90
Figure 45: Réseau de similarité -----	91
Figure 46: La matrice représentative du réseau-----	92
Figure 47: Résultat du réseau similaire -----	93
Figure 48: Règles d'associations pour le service Cal -----	94
Figure 49: Accord SLA pour le service QTE -----	95
Figure 50: Prédiction pour le service Cal -----	95
Figure 51: Prédiction de violation -----	97
Figure 52: Comparaison des substitutions -----	98
Figure 53: Algorithme K-Means[110] -----	110

# Liste des tableaux

Tableau 1: Types de services Clouds[24]-----	8
Tableau 2: Business support [26] -----	13
Tableau 3: Element provisioning[26] -----	14
Tableau 4: Exemples de multi-cloud[35,36,37] -----	22
Tableau 5: Avantages et inconvénients du cloud computing [20,31,32] -----	23
Tableau 6: Types de services web[40] -----	27
Tableau 7: Mesure de la similarité[65]-----	33
Tableau 8: Langage de de description-----	34
Tableau 9: Méthodes de sélection -----	35
Tableau 10: Types de compositions[73, 74, 75]-----	35
Tableau 11: Différentes solutions de la substitution -----	37
Tableau 12: Solution de la substitution à base communauté[65] -----	40
Tableau 13: Solution de la substitution à base réseau[76] -----	42
Tableau 14: Solution à base SLA[77] -----	44
Tableau 15: Critères de comparaison-----	45
Tableau 16: Comparaison entre les solutions de la substitution des services ---	46
Tableau 17: Acteurs principaux et leurs rôles dans notre application Web -----	49
Tableau 18: Exemple des critères non fonctionnels d'un service -----	58
Tableau 19: Application de la méthode MCDM sur la matrice normalisée -----	59
Tableau 20: Informations colléctées dû la surveillance du service « multiplication »-----	60
21: La similarité entre les différents type [95]-----	63
Tableau 22: Identifiant entre ABS et SSN-----	65
Tableau 23: Paramètre de similarité entre ABS et SSN -----	65
Tableau 24: Combinaison entre ABS et IAN -----	67
Tableau 25: Combinaison entre ABS et SSN -----	67
Tableau 26: Représentation matricielle du graphe -----	68
Tableau 27: Evaluation -----	99

## Liste des abréviations

- AHP** : Analytic Hierarchy Process
- AR** : Association Rules
- ARFF** : Attribute Relation File Format
- BPEL**: Business Process Execution Language
- CPU** : Central Processing Unit
- CSP** : Cloud Service Provider
- DCSP** : Distributed Cloud Storage Platform
- EMR**: Electronic Medical Record
- EWD** : Exponential Weighted Difference
- FTP** : File Transfer Protocol
- HTTP**: Hypertext Transfer Protocol
- IaaS**: Infrastructure as a Service
- IBM** : International Business Machine
- IDE** : Integrated Development Environment
- IP** : Internet Protocol
- JavaEE** : Java Enterprise Edition
- JDK** : Java Development Kit
- JMS**: Java Message Services
- MCDM** : Multi Criteria Decision Making
- NIST** : National Institute of Standards and Technology
- OASIS** : Organization for the Advancement of Structured Information Standards
- OWL** : Web Ontology Language
- PaaS**: Platform as a service
- QoS** : Quality of Service
- RACS** : Redundant Array of Cloud Storage
- RAID** : Recherche Assistance Intervention Dissuasion
- RDF** : Resource Description Framework
- RDFS** : RDF Schema

**REST** : Representational State Transfer

**RPC** : Remote Procedure Call

**SaaS**: Software as a Service

**SLA** : Service Level Agreement

**SLO** : Service Level Objective

**SMTP** : Simple Mail Transfer Protocol

**SOA**: Service-Oriented Architecture

**SOAP** : Simple Object Access Protocol

**SSL** : Secure Sockets Layer

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

**URI**: Uniform Resource Identifier

**USDL** : Universal Service-Semantics Description Language

**VIM**: Virtual Interface Machine

**VM**: Virtual Machine

**W3C**: World Wide Web Consortium

**WD** : Weighted Difference

**Weka** : Waikato environment for knowledge analysis

**WSDL**: Web Services Description Language

**WSFL**: Web Services Flow Language

**XLANG**: Xml Based-Language

**XML**: Extensible Markup Language

**XP** : Xtreme Programming

## 1 Contexte

Le Cloud Computing est devenu un concept de plus en plus utilisé faisant référence à l'utilisation de la mémoire, des capacités de calcul, des ordinateurs et des serveurs répartis dans le monde entier et liés par un réseau, tel internet.

Une des caractéristiques de base du Cloud Computing est d'assurer un service continu avec une élasticité rapide et sans altérations. Toutefois, les correspondances entre les clients et fournisseurs de services Cloud sont régis par des SLA (Service Level Agreement). Dans un SLA une date d'expiration du service est définie selon les accords entre le client et le fournisseur.

Pour des raisons diverses un client peut se voir son contrat expiré avant de pouvoir exploité pleinement son service. Une des raisons les plus courantes est le retard causé par le problème de connexion internet. Dans ce cas, et pour ne pas perdre le travail déjà réalisé le client se voit obligé de renouveler son SLA avec le fournisseur. Le problème est que le service Cloud peut ne plus être disponible chez ce fournisseur. La substitution peut constituer une alternative de solution pour ce problème.

## 2 Problématique

Assurer la haute disponibilité des services Cloud est la préoccupation majeure des fournisseurs. Plusieurs situations contraignent les clients à remplacer des services Cloud surtout lors de la composition. La substitution de services Cloud peut résoudre le problème des services échoués par des services garantissant les mêmes fonctionnalités.

La problématique de ce travail consiste à définir une solution pour la substitution des services Cloud de type SaaS(Software as a Service).

## 3 Objectifs

Le but de ce travail est d'assurer les objectifs suivants :

1. Etude de la substitution dans le Cloud Computing et dans les services SaaS.
2. Etude comparative des solutions existantes pour la substitution au niveau SaaS.
3. Extraction des problèmes de la substitution dans les environnements Cloud Computing de type SaaS.
4. Définir une solution pour la substitution de services Cloud de type SaaS.

La solution proposée doit garantir les points suivants :

- a. Description des services Cloud de type SaaS.
  - b. Elaboration d'un modèle de ressemblance entre les services Cloud.
  - c. Définition d'un format SLA.
  - d. Sélection des services Cloud de type SaaS.
  - e. Substitution des services Cloud échoués.
5. Validation de la solution proposée par une expérimentation à travers une application JavaEE (Entreprise Edition).

## 4 Structure du mémoire

Ce mémoire est constitué de quatre chapitres, il commence par une introduction générale qui décrit le contenu et pose la problématique de recherche, ensuite propose un enchaînement d'objectifs pour résoudre le problème mentionné.

**Chapitre I :** Décrit les concepts de base du Cloud Computing tel que les définitions ; modèles de déploiement ; modèles de service ; caractéristiques essentielles ; architectures de référence ; acteurs principaux ; obstacles et avantages du Cloud ainsi que l'évolution du Cloud Computing vers l'inter-Cloud.

**Chapitre II :** Présente un état de l'art sur la substitution des services cloud. Ce chapitre est organisé en deux parties. La première partie décrit les concepts de base relatifs aux services cloud en exposant les problématiques de description, de découverte, de sélection et de format SLA, et de substitution des services Cloud. La deuxième partie présente les travaux connexes avec des synthèses et des études comparatives.

**Chapitre III :** Introduit la conception pour la réalisation de notre solution qui répond aux problèmes précédents à propos de la description des services cloud, la méthode de sélection utilisée et enfin la substitution des services Cloud.

**Chapitre IV :** Produit une implémentation de la solution proposée et les différents concepts décrits dans le chapitre III à travers la réalisation d'un prototype.

Le mémoire s'achève par une conclusion générale qui rappelle le travail réalisé et propose des perspectives futures.



## 1 Introduction

La technologie de l'internet se développe d'une manière exponentielle depuis sa création. Au cours de la dernière décennie l'utilisation du Cloud Computing augmente rapidement, les petites et les moyennes entreprises l'utilisent pour diverses raisons, notamment parce qu'il permet un accès rapide à leurs applications et réduit le coût d'infrastructures.

Dans ce chapitre, nous allons définir le cloud computing, voir les types existants, son déploiement, ses caractéristiques, la relation entre les clouds jusqu'à l'évolution vers Inter-Cloud et Multi-Cloud.

## 2 Définition du Cloud Computing

Le cloud computing est une infrastructure dans laquelle la puissance de calcul et le stockage sont gérés par des serveurs distants auxquels les usagers se connectent via une liaison Internet sécurisée. Plusieurs définitions ont été proposées par différents auteurs :

D'après NIST (National Institute of Standards and Technology) "Le cloud computing est un modèle permettant un accès réseau omniprésent, pratique et sur demande à l'ensemble de ressources informatiques configurables via un réseau partagé (par exemple, réseaux, serveurs, stockages, applications et services) qui peuvent être rapidement approvisionnés et libérés avec un minimum d'effort de gestion ou d'interaction avec les fournisseurs de services"[1].

D'après [2] : "Le cloud computing est un modèle de déploiement informatique, basé sur la virtualisation, où les ressources sont en termes d'infrastructure, les applications et les données sont déployées via l'internet en tant que service distribué par un ou plusieurs fournisseurs de services . Ces services sont modulables à la demande et peuvent être tarifés sur la base d'un paiement à l'utilisation ”.

Il peut être aussi défini autrement selon [3]

"Le cloud est un type de système parallèle et distribué qui consiste à un ensemble d'ordinateurs interconnectés et virtualisés, qui sont provisionnés de manière dynamique et présentés comme une ou plusieurs ressources informatiques plus unifiées, basées sur le niveau de service, les accords établis par la négociation entre le fournisseur de services et les consommateurs ”.

## 3 Eléments constitutifs du Cloud Computing

Il existe trois principaux composants du cloud computing : les clients, les centres de données et les serveurs distribués, tel que chacun d'eux a un but spécifiques .

### 3.1 Clients

Ce sont les dispositifs qui sont utilisés par l'utilisateur final pour gérer ses ressources dans le cloud, Ces appareils sont les ordinateurs de bureau par exemple, les ordinateurs portables, les smartphones ou les iPad.

Le client léger ne doit disposer d'un processeur à haut débit et d'un grand espace de stockage de données, il doit pouvoir faire fonctionner un navigateur web tel que Google Chrome, Firefox ou autre.

Les catégories de clients pourrait être divisé en trois types (clients légers, mobiles, clients lourds) [4][3].

### 3.2 Centre de données

Les applications utilisées par les clients de Cloud Computing sont hébergées dans un grand nombre de serveurs, il peut s'agir d'un bâtiment ou d'une pièce dont il n'est pas nécessaire pour être à votre place mais doit être accessible par Internet. La VM (Virtual Machine) peut exécuter un ensemble de programme sur un seul serveur physique appelé hôte, elle permet aux différents clients d'utiliser les applications hébergés [4][5].

### 3.3 Serveurs distribués

Afin d'assurer la fiabilité et la disponibilité des serveurs, le cloud a réparti les serveurs dans différentes zones géographiques. En cas de panne dans le serveur spécifique, l'autre serveur prendra alors les mesures nécessaires, d'autre part, pour augmenter l'évolutivité un nouveau serveur est ajouté à celui qui existe déjà [6].

## 4 La virtualisation

La technologie de virtualisation est présentée depuis 40 ans mais il y a eu une limitation pour l'appliquer, cette dernière dépend du cloud computing en tant que technologie majeure [7]

La technologie de la virtualisation sépare le matériel physique sous-jacent et fournit des ressources virtualisées aux applications [8].

Un serveur typique est capable d'héberger un certain nombre d'instances de machines virtuelles, qui personnalisent le logiciel à la demande. Il s'agit donc d'un serveur virtuel à la demande basé sur le client, tel que VMware, vCloud, Amazon et autres [7] .

La virtualisation est la base de l'informatique dématérialisée, car elle permet à la mise en commun des ressources informatiques d'un groupe de serveurs qui sont des clusters l'attribution dynamique des ressources virtuelles au client selon les besoins [8].

C'est une technologie attrayante en raison de la capacité l'isolement et la personnalisation des environnements avec peu d'impact sur les performances [8, 9, 10].

### 4.1 Types de virtualisation

Il existe trois types de virtualisations : para- virtualisation, virtualisation des conteneurs et la virtualisation complète.

#### 4.1.1 Para-Virtualisation

La para-virtualisation est une technique dans laquelle l'invité est conscient qu'il opère directement sur le système d'exploitation de l'hyperviseur plutôt que le matériel sous-jacent.

Un hyperviseur de soutien est installé sur le système d'exploitation qui fonctionne sur le matériel sous-jacent. C'est-à-dire que l'hyperviseur agira comme l'hôte sur lequel les systèmes d'exploitation invités sont chargés tandis que les Systèmes d'exploitation invités fera les appels de système nécessaires aux hyperviseurs pour l'utilisation des ressources matérielles [11].

#### 4.1.2 Virtualisation des conteneurs

C'est une technique dans laquelle chaque noyau de système d'exploitation est modifié pour charger plusieurs systèmes d'exploitation invités.

Ici les systèmes d'exploitation sont emballés sous forme de conteneurs dont chaque conteneur a sa propre adresse IP (Internet Protocol), sa propre mémoire, son accès à la racine, et il sera autorisé à être chargé un par un [12].

#### 4.1.3 Virtualisation complète

Dans la virtualisation complète l'hyperviseur est responsable du chargement des systèmes d'exploitation. Et chaque système d'exploitation invité fonctionnera comme s'ils fonctionnaient directement sur le matériel sous-jacent.

C'est-à-dire que chaque système d'exploitation invité obtiendra toutes les fonctionnalités de matériel sous-jacent. Ici, l'hyperviseur interagit directement avec la mémoire et l'espace disque sous-jacents, il isolera les activités d'un système d'exploitation invité de l'autre, ainsi qu'il prenne une console de gestion de la machine virtuelle à partir laquelle les machines virtuelles peuvent être facilement gérées [13].

### 4.2 But de la virtualisation

La virtualisation augmente l'utilisation des ressources par conséquent et d'après [14] nous devrions l'appliquer pour les raisons suivantes :

➤ **Isolement entre les utilisateurs**

Un utilisateur devrait être isolé d'autres utilisateurs afin qu'il ne puisse pas obtenir leurs informations, autrement dit l'utilisation interdit l'accès aux données d'autres personnes.

➤ **Partage des ressources**

Une ressource importante peut être fragmentée en plusieurs ressources virtuelles qui seront utiliser par différents utilisateurs à l'aide de la virtualisation.

➤ **Ressources dynamiques**

La réaffectation de ressources informatiques est très difficile, mais si elle est virtualisée, les ressources seront facilement réattribués.

## 5 Caractéristiques du Cloud Computing

Selon NIST il existe cinq caractéristiques essentielles du cloud computing qui sont les suivantes [15] :

### ➤ Libre-service à la demande

Le cloud offre un large éventail de ressources qui permettent à l'utilisateur de configurer et d'accéder à l'information en fonction du besoins [16].

Le libre-service signifie que la mise à disposition des ressources se fera automatiquement et sans aucune interaction humaine[17, 3] .Le terme "à la demande" signifie que les clients peuvent accéder immédiatement aux ressources demandées.

### ➤ Accès au réseau étendu

Les ressources de l'informatique dématérialisées sont utilisées par de nombreuses applications clientes avec un type de plateforme différent, qui sont accessibles via le réseau [18, 19, 3].

### ➤ Mise en commun des ressources

Plusieurs clients collectent et utilisent les ressources du prestataire, à l'aide d'un type multi-locataires avec différentes ressources qui sont réaffectées de manière dynamique selon l'ordre du client[19, 3].

### ➤ Elasticité

Le Cloud Computing dispose d'un nombre illimité de ressources fournit par le prestataire aux clients à tout moment et en quantité, ces ressources peuvent être augmentées automatiquement lorsque la charge de travail des applications augmente et vice versa[17, 3].

L'élasticité c'est la capacité à mettre les ressources à l'échelle à la fois en haut et en bas selon les besoins [20].

### ➤ Service mesuré

C'est le mécanisme de mesurer combien un client à utiliser une ressource, bien que ces ressources sont partagées et mises en commun par de nombreux clients. Le taux d'embauche est différent d'un fournisseur de cloud à l'autre [18, 3]. Les services en cloud sont contrôlés et surveillés par le fournisseur de services en ligne. Cela est crucial pour la facturation, contrôle d'accès, optimisation des ressources, planification des capacités ainsi que d'autres tâches[20].

## 6 Types de services dans les environnements du Cloud Computing

Le Cloud Computing est défini essentiellement par trois services qui sont les plus connus :

### 6.1 Infrastructure en tant que service

L'IaaS (Infrastructure as a Service) est basée sur des interfaces, qui offre le service de traitement des ressources ou de stockage par exemple Les services web Amazon [21].

Au lieu de vendre des infrastructures matérielles brutes, les fournisseurs proposent généralement une infrastructure virtualisée sous forme de service[22].

Selon [20] IaaS est également connu sous le nom de Hardware as a Service ,l'infrastructure fournie peut être augmentée ou réduite en fonction des éventuels besoins des ressources qui sont généralement facturé sur la base d'un calcul d'utilité, donc les fournisseurs facturent en fonction de la quantité de ressources consommées, comme Amazon EC2 (Elastic Cloud Compute), Amazon S3 (Simple Storage Service).

Il se compose de :

- Accords sur les niveaux de service SLA , Il s'agit d'un accord entre le prestataire et le client, pour garantir le niveau de performance du système.
- Matériel informatique représente les composants dont les ressources seront louées.
- Réseau englobe les pare-feux, les routeurs, l'équilibrage des charges, etc.
- Connectivité à l'internet qui permet aux clients d'accéder au matériel de leurs propres organisations.

## 6.2 Plateforme en tant que service

Le PaaS (Platform as a Service) fournit des offres destinés aux développeurs de logiciels qui peuvent écrire des demandes selon leurs spécifications sans se soucier de la plateforme sous-jacente d'IaaS [21]. Les offres PaaS peuvent couvrir toutes les phases du logiciel ou peut être spécialisés dans un domaine spécifique comme la gestion du contenu [21].

On peut citer comme exemple Google App qui permet d'exécuter des applications sur le moteur de Google et la plateforme Force.com de Salesforce. Le PaaS d'un cloud repose sur l'interface normalisée de la couche IaaS qui virtualise l'accès aux ressources disponibles et offre des interfaces normalisées avec un développement pour la couche SaaS [22].

Il est également connu sous le nom de "cloudware". D'après [20] PaaS offre comme service la conception ,le développement, le test, ainsi que le déploiement et l'hébergement d'applications en garantissant les ressources nécessaires.

Parmi ces avantages on peut citer :

- La possibilité de développer en collaboration même si l'équipes de développement est géographiquement isolée.
- La sécurité, l'extensibilité .

Par contre la principale raison de la chute du PaaS est que les plateformes fournis par différents fournisseurs ne sont généralement pas compatibles, donc elles manque d'interopérabilité et de portabilité entre les fournisseurs ce qui empêche les clients de passer d'un vendeur à l'autre.

## 6.3 Logiciel en tant que service

L'offre SaaS (Software as a Service) n'a généralement ni connaissance ni contrôle sur l'infrastructure sous-jacente [23], qu'il s'agisse de la plate-forme logicielle sur laquelle l'offre SaaS est basée sur le PaaS ou l'IaaS. Cependant, ces couches sont nécessaires et peut être externalisée pour le fournisseur de SaaS .

Par exemple, une application SaaS peut être développé sur une plateforme existante et fonctionnant sur une infrastructure d'une tierce partie. L'obtention de Paas ainsi que d'IaaS est attrayant pour les fournisseurs de SaaS elle permet également de lui soulager de lourds coûts de licence[22]

Parmis ces avantages :

- La fiabilité, si un serveur tombe en panne, un autre serveur se mettre en charge, en utilisant Le protocole SSL (Secure Socket Layer) qui permet au client d'accéder aux applications en toute sécurité.
- En raison de l'augmentation de la bande passante, l'organisation peut accéder aux applications avec moins de latences et à grande vitesse[20].

La figure 1 ci-dessous montre clairement la hiérarchie des services cloud :

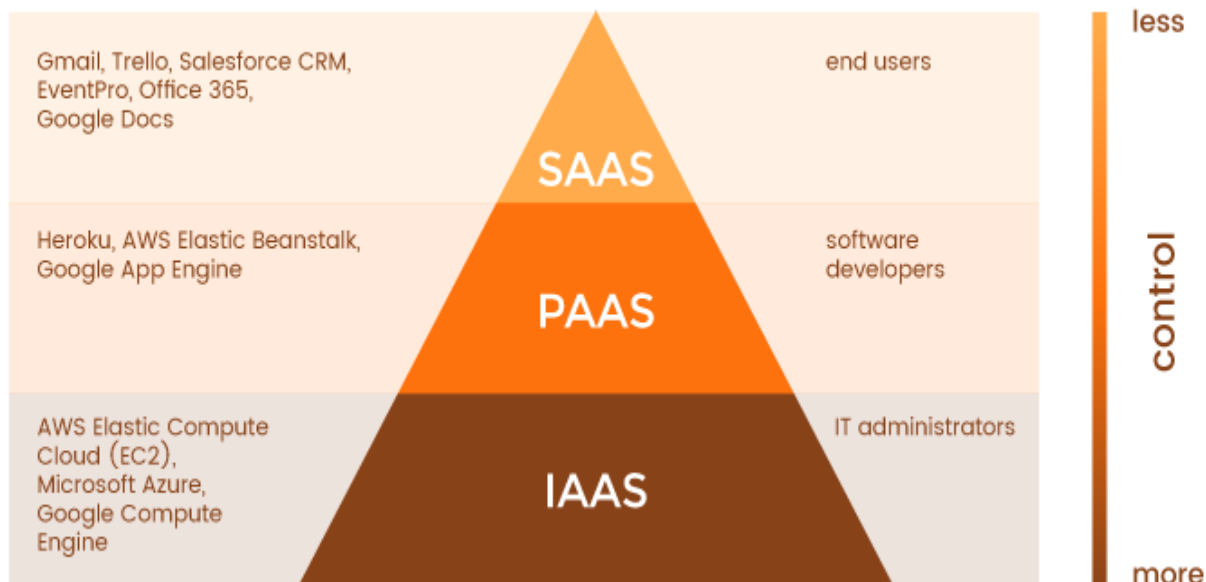


Figure 1: L'hierarchie des services cloud [89]

### 6.4 Autres types de services Clouds

Nous avons vu déjà trois types de services cloud, bien qu'il existe actuellement d'autres offres de type **x as a service** où **x** peut prendre plusieurs formes, en citant quelques unes depuis [24], ci-dessous dans le tableau 1 :

X as a service	Nom complé du service
APaaS	Application Platform as a Service
BaaS	Backend as a Service
BaaS	Backup as a Service
CaaS	Communication as a Service
MaaS	Management as a Service
DaaS	Data as a Service
DaaS	Desktop as a Service
DBaaS	Database as a Service
DCaaS	Data Center as a Service
STaaS	Storage as a Service
SECaaS	Security as a Service
NaaS	Network as a Service

Tableau 1: Types de services Clouds[24]

## 7 Modèles de déploiement du Cloud Computing

Il existe quatre types de cloud computing comme indique la figure 2, chacun d'eux présente des risques de sécurité et une gestion informatique différente

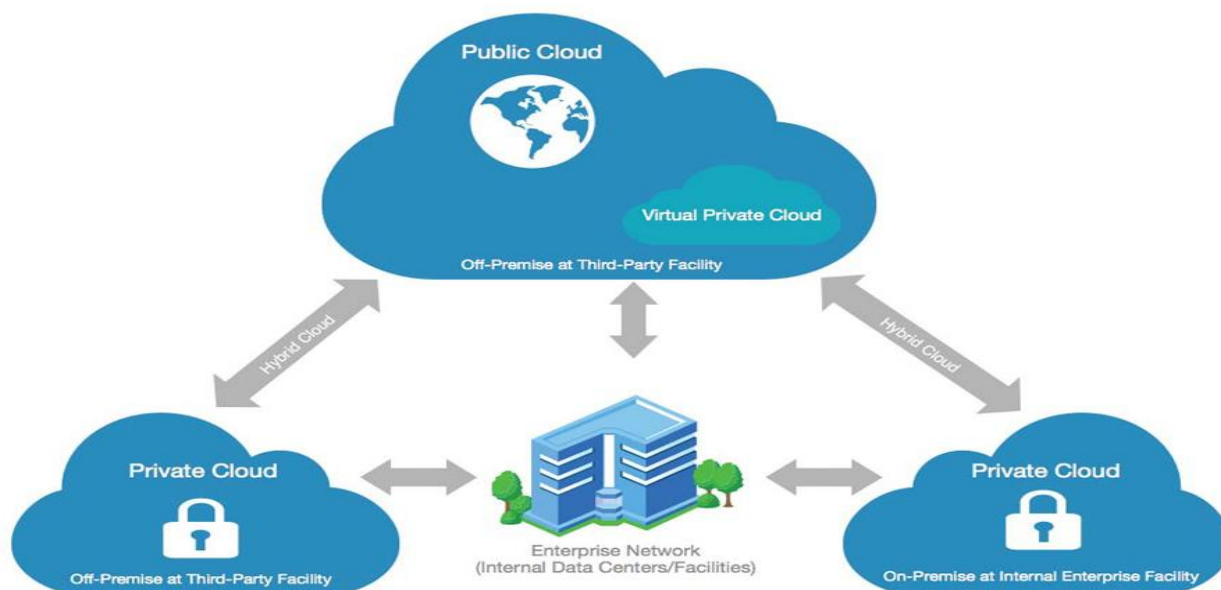


Figure 2: Déploiement des services cloud [90]

### 7.1 Cloud Public

Le cloud publique est également connu sous cloud externe ou cloud multi-locataires, il peut être détenu et gérer par un organisme gouvernemental ou par une combinaison de ces deux entités[16]. Le cloud publique signifie que les données des utilisateurs sont publiquement visibles, ces services sont à la disposition des clients depuis le fournisseur via l'internet, qui peuvent être gratuit ou assez peu cher .

Les fournisseurs généralement produisent un mécanisme de contrôle d'accès pour leurs utilisateurs, par exemple Google et Amazon, qui exposent leurs services aux entreprises et aux consommateurs via Internet [25].

### 7.2 Cloud Privé

Le cloud privé est également connu sous le cloud interne ou cloud dans les locaux qui est géré et exploité par une seule organisation ou un groupe, il offre un accès limité à ses ressources et ses services aux consommateurs qui appartiennent à la même organisation que celle qui possède le cloud [16].

Le cloud privé Selon [21] est géré au sein de l'organisation sans les restrictions du réseau comme la bande passante, les risques de sécurité et les exigences légales. En outre, les services de cloud privé offrent au fournisseur et à l'utilisateur un contrôle de l'infrastructure en cloud ,améliorant la sécurité et la résilience.

Il est également élastique et basé sur des services mais contrairement aux données et aux processus du cloud public, elle n'est pas nécessairement gérée et hébergée par l'organisation qui l'utilise et peut être physiquement situé hors des locaux mais garde toujours le contrôle total de l'infrastructure .

### 7.3 Cloud Communautaire

Le cloud communautaire s'agit d'un environnement de cloud à usage spécifique qui est partagé et géré par un certain nombre d'organisation apparentée participant à un domaine commun ou à un marché vertical[16].

Il est contrôlé et utilisé par un groupe d'organisations qui ont des intérêts communs, par exemple des exigences de sécurité ou autre. Les membres de la communauté partagent l'accès aux données et aux applications dans le cloud[20].

### 7.4 Cloud Hybride

Il s'agit d'une combinaison d'un cloud publique et privé, d'après [25] dans ce modèle, les utilisateurs externalisent généralement les informations et les traitements non critiques pour l'entreprise vers un cloud publique, tout en conservant les services et les données critiques pour les entreprises dans leurs contrôle.

Cela peut rendre plus complexe la distribution des applications dans différents environnements, notamment la surveillance des infrastructures interne et externe concernées, la sécurité et la vie privée, et peut donc ne pas convenir les applications nécessitant des bases de données complexes ou synchronisées . Il offre les avantages de plusieurs modèles de déploiement. Il permet à l'entreprise de gérer la charge de travail d'une façon permanente dans cloud privé[16].

## 8 Architecture de référence du Cloud Computing

Nous adoptant l'architecture de référence de NIST [26] dans le cloud qui se compose de plusieurs éléments expliqués en détail dans les sections suivantes :

### 8.1 Déploiement du service

Une infrastructure cloud peut être exploitée dans le cloud publique, privé ou communautaire. Les différences dépendent de la manière dont les ressources informatiques sont attribuées à un client .

La figure 3 suivante montre une simple vue d'un cloud publique avec ses consommateurs :



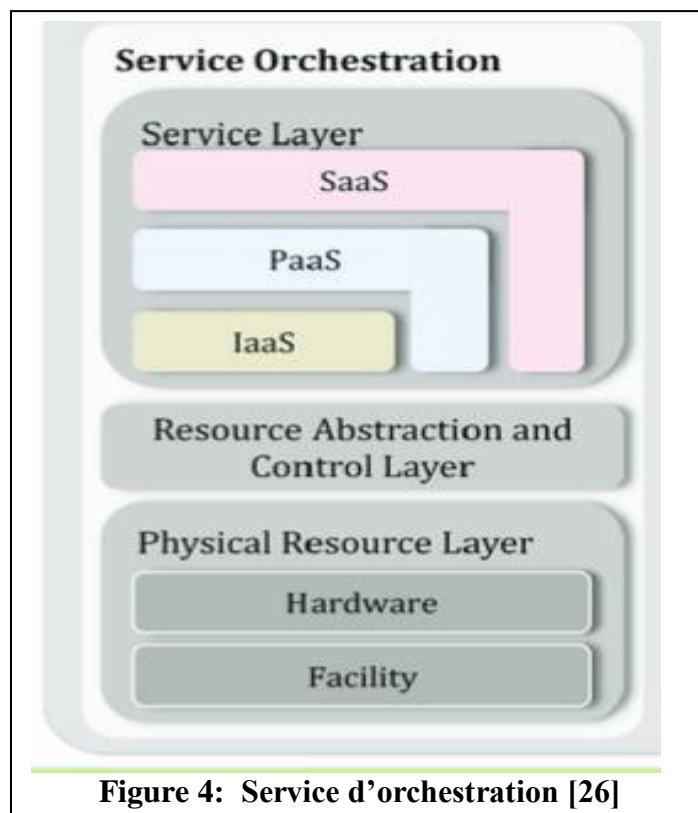
Figure 3: Public cloud [91]



## 8.2 Service d'orchestration

Service d'orchestration fait référence à la composition de composants de système pour prendre en charge les activités de coordination et de gestion des ressources informatiques afin de fournir des services aux consommateurs de cloud

La figure 4 montre un diagramme de pile générique de cette composition sous-jacente :



Un modèle à trois couches est utilisé dans cette représentation, qui regroupe les trois types de système pour fournir les services des fournisseurs, la couche supérieure est la couche de service, c'est là où les fournisseurs de cloud permettant aux utilisateurs d'accéder aux services informatiques. Il est possible, bien que non nécessaire, que les applications SaaS puissent être construites sur des composants PaaS et des composants PaaS peuvent être construits sur des composants IaaS.

Les relations de dépendance optionnelles entre les composants SaaS, PaaS et IaaS sont représentées graphiquement en tant que composants empilés les uns sur les autres ; tandis que la prise des composants représente que chacun des composants de service peut être autonome.

Par exemple, une application SaaS peut être implémentée et hébergée sur une machine virtuelle d'un IaaS ou peut être implémentée directement au-dessus des ressources du Cloud.

La couche intermédiaire du modèle est l'abstraction des ressources avec la couche de contrôle. Cette couche contient les composants utilisés par les fournisseurs de cloud pour fournir et gérer l'accès aux ressources physiques via l'abstraction logicielle, par exemple les logiciels des éléments tels que des hyperviseurs, des machines virtuelles ou un stockage de données virtuel.

L'abstraction des ressources doit assurer une utilisation efficace, sécurisée et fiable des ressources physiques sous-jacentes. Bien que la technologie des machines virtuelles soit couramment utilisée sur cette couche. L'aspect contrôle de cette couche fait référence aux composants logiciels responsables de l'allocation des ressources, du contrôle d'accès et de la surveillance.

C'est la structure logicielle qui relie les nombreuses ressources physiques sous-jacentes et leurs abstractions logicielles dans le but de regrouper des ressources, de fournir l'allocation dynamique et le service mesuré.

La couche la plus basse de la pile est la couche des ressources physiques, cette couche comprend des ressources matérielles, telles que des ordinateurs (CPU « Central Processing Unit » ainsi que la mémoire), des réseaux (les routeurs, le pare-feu, les commutateurs, et les interfaces), composants de stockage (disques durs) et autres éléments d'infrastructure informatique physique.

### 8.3 Gestion de service en Cloud

La gestion des services en Cloud inclut toutes les fonctions liées au service nécessaires à la gestion et l'exploitation des services demandés ou proposés par les consommateurs de Cloud Computing. Comme illustré dans la figure 5, la gestion des services en cloud peut être décrite par rapport au support technique ou aux exigences de portabilité et d'interopérabilité.

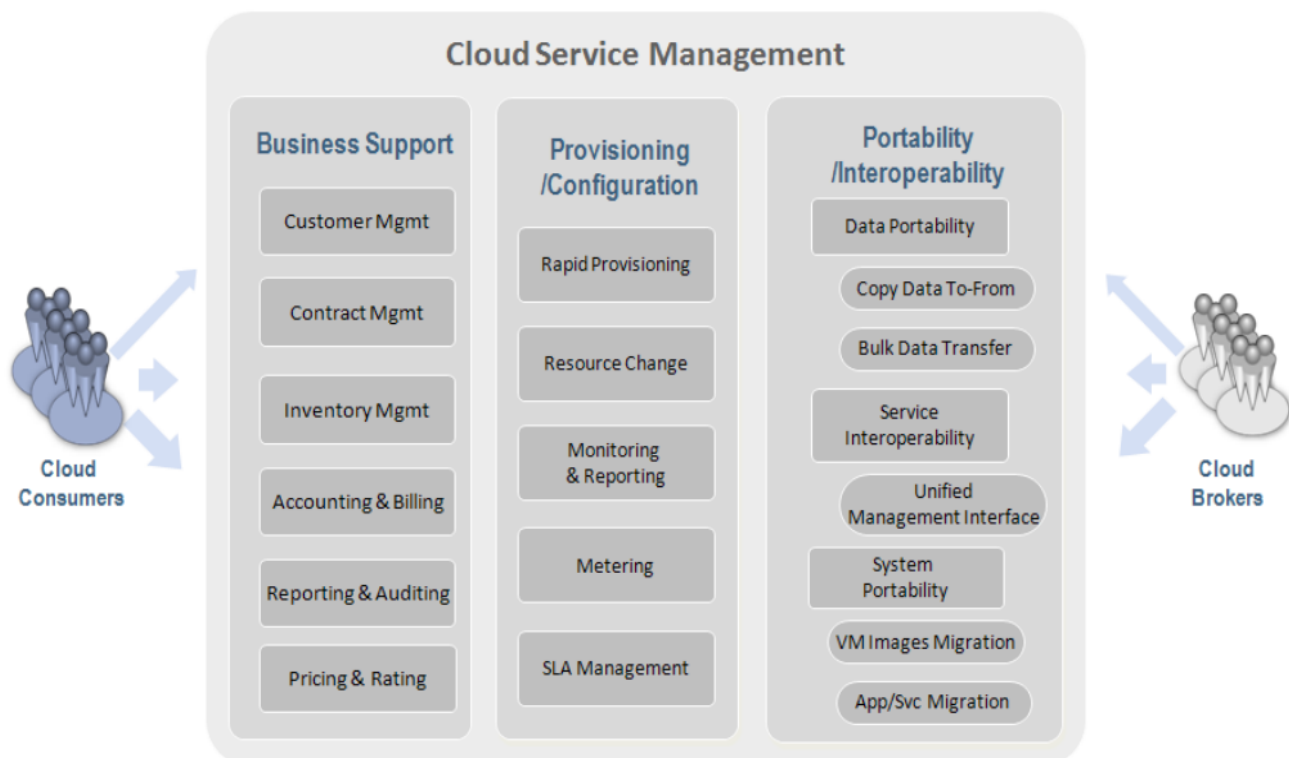


Figure 5: Gestion des services cloud [26]

### A.Soutien aux entreprises « Business support » :

Englobe les services liés aux entreprises traitant avec les clients. Le tableau 2 explique nettement les composants de l'élément Business Support de gestion des services cloud :

Eléments	Définitions
<b>Gestion de la clientèle « Customer Management »</b>	C'est la gestion des comptes clients qui signifie l'ouverture, la fermeture et la clôture de comptes, gestion des relations clients en fournissant des points de contact et en résolvant les questions et les problèmes des clients.
<b>La gestion des contrats « Contract Management »</b>	L'établissement, la négociation, la cloture et la résiliation de contrat.
<b>Gestion de l'inventaire « Inventory Management »</b>	La configuration et la gestion des catalogues de services
<b>Compatibilité et facturation « Accounting and Billing »</b>	La gestion des informations de facturation des clients, l'envoi des relevés de facturation, le traitement de paiements reçus suivi des factures.
<b>Repport et audit «Reporting and Auditing »</b>	La surveillance des opérations des utilisateurs accompagner à la génération des rapports.

**Tableau 2: Business support [26]**

### B. Mise à disposition et compatibilité de configuration « Provisioning and Configuration Comptabilise » :

Des charges anticipées avec la possibilité de la configuration . Le tableau 3 explique les composants de l'élément « Provisioning and Configuration » de gestion des services cloud :

<b>Eléments</b>	<b>Définitions</b>
<b>Provisionnement rapide « Rapid provisioning »</b>	Le déploiement automatique de systèmes cloud en fonction de la demande : de service, de ressources ou de capacités.
<b>Changement des ressources « Resource changing »</b>	L'ajustement de la configuration, l'affectation des ressources pour les réparations, les mises à niveau et pour rejoindre de nouveaux nœuds dans le cloud.
<b>Surveillance et rapport « Monitoring and Reporting »</b>	La surveillance des opérations, des événements du cloud avec la génération de rapports de performance.
<b>Mesure « Metering »</b>	Fournir une capacité de mesure à un niveau d'abstraction approprié au type de service par exemple : comptes de stockage, de traitement ou de bande passante.
<b>SLA</b>	Englobant la définition de contrat de niveau de service avec la qualité de service, la surveillance et la mise en oeuvre des SLA conformément aux politiques définies.

**Tableau 3: Element provisioning[26]**

### C. Portabilité et interopérabilité

Fournit des économies de coûts dans l'infrastructure technologique et une mises à niveau logicielles. Le tableau 4 explique les composants de l'élément Portabilité et interopérabilité de gestion des services cloud :

#### ➤ La portabilité

Les clients souhaitent savoir s'ils peuvent transférer leurs données ou applications dans plusieurs environnements de cloud à faible coût avec un minimum de perturbations. Elle permet la migration d'une instance de machine virtuelle complètement arrêtée ou d'une image de machine à partir d'un fournisseur à un autre fournisseur [27].

### ➤ L'interopérabilité

C'est la capacité d'utiliser leurs données et leurs services sur plusieurs fournisseurs de cloud avec une interface de gestion unifiée.

## 8.4 Sécurité

Il est essentiel de reconnaître que la sécurité est un aspect transversal de l'architecture qui couvre tous les couches du modèle de référence, allant de la sécurité physique à la sécurité des applications.

Par conséquent, la sécurité en matière d'architecture de cloud computing ne relève pas uniquement de la compétence des fournisseurs de cloud, mais aussi sur d'autres acteurs concernés.

Les systèmes basés sur le cloud doivent encore gérer les exigences de sécurité telles que l'authentification, l'autorisation, et la disponibilité, la confidentialité, la gestion de l'identité, l'intégrité, la surveillance de la sécurité, la réponse aux incidents et enfin la gestion des politiques de sécurité.

## 8.5 Vie privée

L'informatique en cloud offre une solution souple pour le partage des ressources, des logiciels et des informations ou autre, elle pose également des défis supplémentaires en matière de protection de la vie privée pour les consommateurs qui l'utilise, c'est donc pourquoi les fournisseurs de services cloud doivent protéger la collecte, le traitement et la communication des données de manière sûr, appropriée et cohérente [28].

## 9 Acteurs du Cloud Computing

Dans le cloud computing existe six principaux acteurs selon [29], tel que chaque acteur a ses propres activités, exigences, et responsabilités.

### 9.1 Consommateur de cloud

Les consommateurs jouent un rôle important dans les entreprises alors que leurs demandes et les attentes sont précieuses dans les industries des technologies de l'information par conséquent elles soutiendront en permanence le développement des services cloud. Les consommateurs peuvent être classés en quatre groupes différents :

- **Consommateur d'IaaS :**

Les clients qui ont besoin d'un service dans les cloud concernant les capacités d'infrastructure telles que la fourniture et les ressources informatiques fondamentales.

- **Consommateurs de PaaS :**

C'est une personne ou organisation qui attend une plate-forme utilisée pour développer un logiciel ou une application sans même se soucier d'installer les outils de développement, de programmation et de sécurité sur le système d'utilisateur.

- **Consommateur de SaaS :**

Est généralement lié à tous les consommateurs qui utilisent une application sur Internet pour leurs activités quotidiennes.

- **Consommateur XaaS :**

Les consommateurs qui ont leurs propres définitions du service, ils espèrent un cloud qui peut gérer et satisfaire leurs demandes.

### 9.2 Fournisseur de cloud

Les fournisseurs de services en cloud ont des composantes et des activités beaucoup plus compliquées, car ils fournissent des services au consommateur, et ils possèdent aussi une infrastructure pour maintenir leurs qualités, leurs capacités et la performance. Mais tout ça ne suffit pas, ils devraient se concentrer sur la gestion à l'intérieur d'un environnement informatique en cloud, qui doit être pris au sérieux.

Les fournisseurs de cloud computing doivent comprendre l'exigence fondamentale d'un service cloud, y compris le modèle de service, son déploiement, la vie privée et la sécurité qui doivent être conçus de manière appropriée, car sans eux, il risque de devenir un énorme échec pour cette révolution. En outre, le maintien, la surveillance, l'exploitation et la gestion sont les principales activités pour les fournisseurs [30].

### 9.3 Développeur de cloud

C'est une personne ou une organisation qui est responsable de développer des services dans le cloud. Il ne peut pas être toujours connecté aux autres acteurs d'une manière directe, car les développeurs de cloud computing ont des capacités et des aptitudes à développer, à mettre en œuvre des services, qu'ils le déploient par la suite dans le système du fournisseur de cloud computing.

### 9.4 Courtier de cloud

Ils jouent le rôle d'agent ou de négociateur entre les acteurs du cloud, lorsque les acteurs ne veulent pas négocier ou faire face directement entre eux.

### 9.5 Auditeur de cloud

Il est chargé de procéder à l'évaluation des performances, la sécurité et la confidentialité des services. L'audit est utilisé pour vérifier la compatibilité avec les normes relatives aux clouds dans le but d'améliorer la qualité des services.

### 9.6 Transporteur de cloud

Représente un transport physique pour que les acteurs du Cloud puissent atteindre leur cible depuis Internet avec accès efficace, il peut avoir besoin d'un transporteur agent, Il ne s'agit pas d'un simple agent ordinaire, mais d'un expert en cloud computing, pour avoir la possibilité de configurer et de définir le service dans le système de consommation.

## 10 Relations entre les acteurs du Cloud

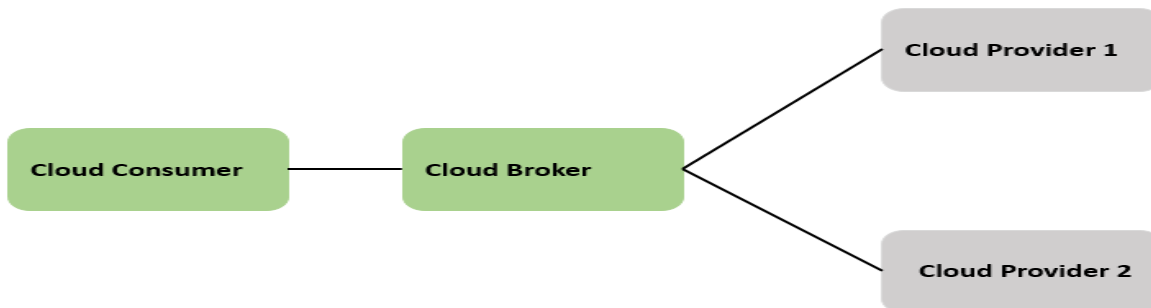
Nous avons mentionné précédemment qu'un Cloud Consumer « consommateur de cloud » est une Personne ou une organisation qui utilise les services de fournisseurs dans le cloud, qui sont fournis par le Cloud Provider « fournisseur de cloud ».

Une évaluation indépendante des services est effectuée à l'aide d'un Cloud Auditor « auditeur de cloud », par contre le Cloud Broker « courtier de cloud » gère l'utilisation et négocie les relations entre les fournisseurs et les consommateurs. Un autre cloud est utilisé comme intermédiaire pour assurer la connectivité et le transport des services c'est le Cloud Carrier « transporteur de cloud ».

La relation entre les différents types d'acteurs du cloud est comme suit [26] :

Un Cloud Consumer peut demander directement a un Cloud Broker des services qui peut créer un nouveau service en combinant de multiples services ou en améliorant un service existant.

Dans cette figure (figure 6), les fournisseurs sont invisibles pour le consommateur, ce dernier contacte le Cloud Broker au lieu le Cloud Provider .



**Figure 6: Relation entre le cloud broker et le cloud consumer [26]**

Les Cloud carriers assurent le transport, comme l'illustre la figure 7 , un Cloud Provider participe a deux SLAs uniques, l'un avec le Cloud Consumer (par exemple, SLA2) et l'autre avec un Cloud Provider (par exemple, SLA1).

Un Cloud Provider arrange des SLA avec le Cloud Carrier qui demande des services dédiés et chiffrés pour garantir que les services cloud sont consommés a un niveau cohérent selon les obligations contractuelles avec les Cloud Consumer.

Dans ce cas, le Cloud Provider peut spécifier son exigences en matière de capacité et de fonctionnalité dans SLA2 afin de satisfaire les exigences de SLA1.



**Figure 7: Le rôle du cloud provider [26]**

Pour un service de cloud computing, un Cloud Auditor effectue des évaluations du fonctionnement, de sécurité même de la mise en oeuvre. L'audit implique des interactions à la fois avec le consommateur et le fournisseur de services dans le cloud, comme l'indique la figure 8.

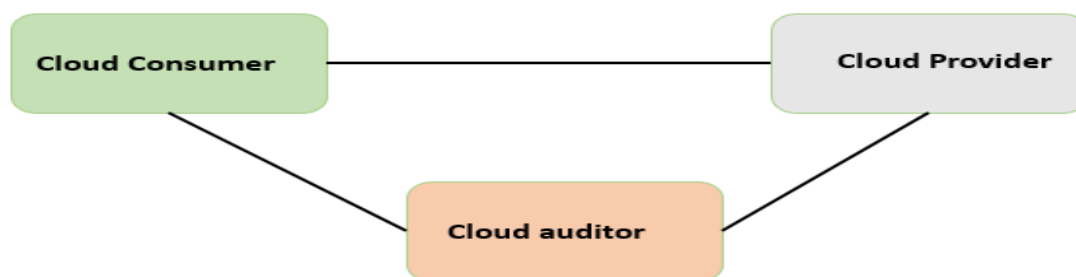


Figure 8: Le rôle du cloud auditor [26]

## 11 Evolution du Cloud Computing

Le Cloud Computing peut être considéré comme une innovation de différentes manières, son concept n'est pas nouveau, l'idée était proposée par le pionnier de l'informatique John McCarthy dans les années 60s, lorsqu'il a proposé le concept de "calcul d'utilité". Il a prédit que le calcul pourrait un jour être organisé comme un service public.

L'informatique utilitaire, traite le cloud en tant qu'utilité plutôt qu'un produit spécifique, donc il est perçu comme un emballage de ressources informatiques telles que le calcul, le stockage [20, 2].

Malgré les multiples services offerts par le cloud, certains besoins n'y étaient pas satisfait, notamment l'intégration entre deux services situés sur des infrastructures différentes, où l'exploitation d'un service cloud sur plusieurs cloud, ce qui a encouragé l'apparition des nouvelles approches tel que l'Inter-Cloud [33, 34] et le Multi-Cloud [35, 36, 37] :

### 11.1 Inter-Cloud

Les clouds sont largement utilisés par l'industrie et la recherche communautaire ; ils offrent un accès abordable avec une puissance d'installation pour le publique. Les technologies l'évalue comme un moyen commun de fournir des services d'infrastructure via la virtualisation, pour l'utilisation efficace des ressources physiques, qui nécessite des capacités d'approvisionnement à la demande de pouvoir réaliser une commercialisation adéquate.

Les technologies du cloud apportent les applications et les services d'infrastructure ,la mobilité et l'indépendance de la plate-forme physique et matérielle par rapport aux applications de calcul distribué et de mise en réseau.

La réponse pour traiter les questions de fiabilité et la limitation des clouds isolés à domaine unique, est de tirer parti d'une multitude de fournisseurs de services en cloud, autrement dit c'est de former l'Intercloud, qui devrait devenir une couche de plus en plus importante dans le modèle de cloud computing au sens large, qui se trouve au-dessus de la couche de clouds a domaine unique.

La couche Intercloud offre un environnement unique pour la mise en place des services fiables. Il est très utiles pour la tolérance aux fautes comme l'un des aspects clés de la fiabilité ainsi que l'amélioration de la confidentialité, de l'intégrité et de la cohérence.



Les services d'infrastructure en cloud fournis peuvent impliquer les ressources multi-fournisseurs et multi-domaines, y compris l'intégration avec les services et les infrastructures existants.

Ainsi, les clouds représentent une nouvelle étape dans l'évolution de développement des technologies de l'informatique et de la communication en introduisant un nouveau type de services, et une nouvelle couche d'abstraction pour les services d'infrastructure pour obtenir la mobilité des applications distribuées via la virtualisation.

Un nouveau système de stockage RAID (Recherche, assistance, intervention, dissuasion) s'appuie sur les techniques traditionnelles de disponibilité des données, il offre une meilleure protection contre la dépendance à un seul fournisseur de stockage dans le cloud.

La sécurité d'un cloud toujours pose le problème, comme les données sont envoyées et traitées dans un environnement qui n'est pas sous le contrôle de l'utilisateur ou du propriétaire des données, et peut être compromise soit par les clouds ou par d'autres utilisateurs partageant la même ressource.

Les données ou les informations doivent être sécurisées en téléchargement, en traitement, en stockage, en diffusion ou même en visualisation. Lorsqu'il est prévu de construire une infrastructure de sécurité pour l'environnement et l'infrastructure inter-cloud il faut tenir en compte les éléments suivants :

- ✓ Protection des données stockées
- ✓ Gestion de la confiance et des clés dans le provisionnement à la demande
- ✓ La virtualisation et la dynamique de l'infrastructure de contrôle d'accès
- ✓ La gestion des SLA, y compris la négociation initiale des SLA
- ✓ Session spéciale pour le transfert de données qui devrait également
- ✓ soutenir le partitionnement des données, l'activation de l'exécution
- ✓ Les mécanismes de synchronisation des sessions qui devraient protéger l'intégrité de la durée d'exécution à distance.

La figure 9 illustre un exemple d'Inter-cloud.



Figure 9: Inter-cloud [92]

## 11.2 Multi-Cloud

Ces dernières années, le service de stockage dans le cloud est devenu un point de croissance des bénéfices en offrant un coût comparativement faible, une plate-forme évolutive et indépendante de la position pour les données des clients.

Comme l'environnement de l'informatique en cloud est construit sur la base des architectures et interfaces ouvertes, il a la capacité d'intégrer plusieurs services de cloud interne ou externe ou les deux pour assurer une interopérabilité élevée.

Nous appelons ces environnements distribués dans le cloud en tant que multi-Cloud (ou cloud hybride). Souvent, en utilisant la gestion de VIM (Virtual Interface Machine), un multicloud permet aux clients d'accéder facilement à ses ressources à distance via des interfaces telles que les services web fournis par Amazon EC2.

Il existe différents outils et technologies pour le multi-cloud, tels que Platform VM Orchestrator, VMware vSphere, et Ovirt. Ces outils aident les fournisseurs de services à construire une DCSP (Distributed Cloud Storage Platform) pour gérer les données des clients. Toutefois, si une plate-forme est aussi importante est vulnérable aux attaques de sécurité, il apporterait des pertes pour les clients.

La pratique générale des fournisseurs de services dans les clouds consiste à publier divers niveaux de services en cloud basés sur leurs différentes performances ainsi que les prix sur leurs sites web. Les utilisateurs sont ensuite invités à sélectionner un ou plusieurs pour les tâches requises.

Cette méthode comporte quelques inconvénients de la sélection des utilisateurs comme la passivité et les statuts en cours des services sélectionnés qui sont en train de changer. Cependant,

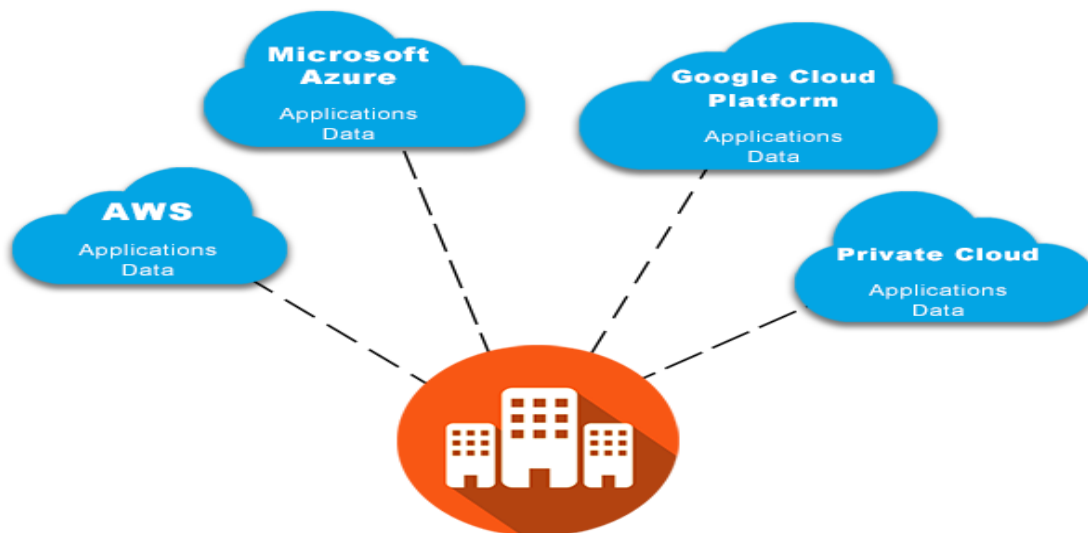
pour les utilisateurs, de nombreux services ou applications en ligne nécessitent une plus grande stabilité ainsi que des prix plus bas en termes de services sélectionnés.

Pour que le service sélectionné s'adapte automatiquement à la diversité du service en cloud des courtiers seront adopter pour gérer les ressources ces services et pour aider les utilisateurs d'accomplir le travail de sélection du service en utilisant une méthode adaptative.

Les courtiers peuvent ressembler à des agents, qui sont le concept dans le domaine de la distribution artificielle mais ils peuvent résoudre des tâches réparties dans différents domaines, et possèdent une certaine autonomie et une caractéristiques que les services en cloud n'ont pas.

Les agents peuvent effectuer des sélections optimisées pour des demandes de service en fonction des statuts actuels et de toute les changements apportés aux services qu'ils gèrent.

La figure 10 montre un exemple de stockage en multi-cloud. les CSP (Cloud Service Provider)développent de nouvelles technologies pour améliorer les capacités du cloud. Les mashups de clouds sont une tendance récente sert à combiner plusieurs services ou une application unique, éventuellement avec les données sur le site (côte client).



**Figure 10: Multi-cloud [93]**

Ce service de composition permet aux CSP d'offrir de nouvelles fonctionnalités plus sophistiquées à leurs clients à des coûts de développement moins élevés.

Voici des exemples dans le tableau 4 de mashups de cloud associe aux technologies pour les soutenir :

Exemple 1	Exemple 2	Exemple 3
<p>Le Mashup Center d'IBM(International Business Machines) : est une plateforme de création rapide, le partage et la découverte d'éléments constitutifs d'applications réutilisables avec des outils permettant de les assembler dans de nouvelles applications Web.</p>	<p>Force.com pour le Google App Engine, un ensemble de bibliothèques qui permettent le développement d'applications Web et commerciales en utilisant les ressources de Salesforce.com et de Google cloud..</p>	<p>Appirio Cloud Storage, un service de stockage dans le cloud qui permet aux clients de Salesforce.com de stocker des informations sur des comptes.</p>

**Tableau 4: Exemples de multi-cloud[35,36,37]**

Les agences gouvernementales s'efforcent de mettre en place des systèmes d'information sur les soins de santé intéropérables qui favorisent l'échange électronique de données entre plusieurs organisations , par exemple,EMR qui est en train d'émerger.

Aujourd'hui, les mashups de cloud nécessitent des accords préétablis entre les fournisseurs ainsi que l'utilisation d'outils propriétaires personnalisés qui combinent les services par le biais des techniques d'intégration contrôlées et contraignantes .

## 12 Avantages et Inconvénients du Cloud Computing

Le cloud computing offre de nombreux avantages aux utilisateurs ,il facilite le travail bien qu'il a aussi des problèmes majeurs par exemple la perte des données.

Nous mentionnons ci dessous dans le tableau 5 les principaux avantages ainsi que les obstacles les plus courants. Parmi les meilleurs avantages de cloud computing et selon [20, 31, 32] on trouve :

Avantages	Inconvénients
La réduction des coûts ou de complexité soit pour l'utilisation des ordinateurs ou des réseaux.	La disponibilité du service: le prestataire de services ait de multiples centres de données repartis dans le monde entier, il peut avoir des infrastructures logicielles ainsi que des systèmes comptables communs par conséquent l'entreprise peut même faire faillite.
Les utilisateurs du cloud ne doivent pas investir dans l'infrastructure informatique : l'achat du matériel, ou des licences de logiciels	Verrouillage des données : la portabilité des applications d'un cloud vers l'autre peut être très coûteuse.
La fiabilité car il permet l'accès aux applications, aux documents ou autre partout dans le monde via Internet	La sécurité ; les clients ne peuvent pas protéger l'accès à leurs données puisque elles sont stockées à l'extérieur de leurs locaux, donc ils ne connaissent même pas son emplacement.
L'évolutivité signifie que le cloud computing offre une capacité de traitement et de stockage illimitée.	L'équilibrage de la charge : concerne la gestion dynamique et efficace des ressources afin de répondre aux besoins des abonnés .
Amélioration des efforts visant à renforcer la protection de la vie privée et de la sécurité	Bugs dans les systèmes distribués à grande échelle, c'est la capacité de supprimer les erreurs dans un cloud computing, qui se considère comme un défi difficile à relever .
L'acquisition et l'amélioration flexibles des technologies de l'information qui peuvent permettre des ajustements aux procédures en fonction de la sensibilité de la données.	La consommation d'énergie : une utilisation de matériel à faible consommation d'énergie avec des données informatiques à grande échelle.

**Tableau 5: Avantages et inconvénients du cloud computing [20,31,32]**

## 13 Conclusion

Le Cloud Computing correspond à l'accès à des services informatiques par exemple serveurs, stockage, mise en réseau, logiciels via internet. Les grandes entreprises du secteur des technologies de l'information développent massivement le Cloud Computing, allant des services Cloud à la location de logiciel.

Dans ce chapitre, nous avons parlé des généralités du cloud computing qui incluent la définition des Clouds, leurs déploiements, les principaux services (SaaS, PaaS, IaaS) , les relations entre eux et enfin l'évolution du cloud vers l'Inter-Cloud et le Multi-cloud.

Dans le chapitre suivant, nous allons présenter un état de l'art concernant la substitution des services dans le cloud computing et discuter les solutions existantes.

## 1 Introduction

Plusieurs paradigmes de développement de logiciel ont été proposés pour satisfaire le besoin de la réutilisation, par exemple le paradigme SOA (Service Oriented Architecture) qui est l'un des modèles les plus prometteurs qui fournit l'utilisation des services cloud de type SaaS, étant donné qu'ils présentent plusieurs avantages tels que l'accessibilité, la rapidité et la simplicité.

Après avoir abordé une vue globale sur les cloud computing dans le chapitre précédent, maintenant nous nous intéressons à la substitution des services Web puisque la substitution des services cloud n'a pas été traitée d'une façon efficace c'est pourquoi on se focalise dans ce chapitre sur les services web .

Tout d'abord nous présentons les concepts clés du cloud computing ainsi que la substitution des services cloud en cas d'un dysfonctionnement, ensuite nous allons discuter les différents travaux et solutions proposées dans ce domaine .

## 2 Concepts fondamentaux des services

Nous introduisons les différents concepts liés aux services Web comme suit selon [38] :

- **Interfaces et échanges de messages**

Un consommateur a toujours besoin d'émettre une requête vers le fournisseur celui-ci renvoie une réponse.

L'interface structurelle spécifie la liste des opérations offertes par le service , ce dernier contient le triplet [**nom de l'opération, message de requête, message de réponse**] qui est appelé signature de l'opérations qui sont accessibles via un ou plusieurs protocoles de communications (http « Hypertext Transfer Protocol » , FTP « File Transfer Protocol » , SMTP « Simple Mail Transfer Protocol » , etc.) d'une manière indépendantes de la technologie.

Par contre le protocole et la localisation des services dépend d'une technologie réseau.

La description des interface structurelle de service web utilise le langage standard WSDL (Web Service Description Langage)

- **Compatibilité d'interfaces**

L'interface d'un consommateur doit respecter les interfaces d'un service aussi bien sur le plan structurel et comportementale . Sinon, les interactions ne peuvent pas aboutir entre les deux services. On distingue des interfaces compatibles si la structure de chacune de leurs messages sont les mêmes ainsi qu'ils contiennent la même séquence des opérations.

- **RPC (Remote Procedure Call)**

Un RPC c'est protocole selon [49] regarde le serveur comme un ensemble de procédures tel que le client appelle ces procédures pour accomplir une tâche. Un RPC est une modélisation centralisée.

- **REST (Representational State Transfer)**

Un REST selon [50] fournit un ensemble de contraintes architecturales et permet l'extensibilité des composantes d'interactions, la généralité des interfaces, la réduction de la latence en fournissant le déploiement indépendant de composants.

- **Broker :**

Cette technologie a été étudiée dans le contexte des services dédiés aux entreprises à forte intensité de connaissances. Le « Broker » joue un rôle d'éventail entre le client et le fournisseur autrement dire c'est le lien qui relie les deux par exemple si un client cherche un service, il exprime ses besoins via une requête, le broker reçoit cette demande et essaye de trouver le bon service parmi les services créés par le fournisseur ensuite il retourne le résultat au client [96].

### 3 Définition de service web

On utilise quelques définitions depuis la littérature pour décrire le service web : Selon [39] les SOAs sont un ensemble de composants qui sont appelés dont les descriptions d'interfaces peuvent être éditées et découvertes.

De nos jours, les Services Web fournissent les technologies les plus adaptées pour rendre possible la création d'architectures orientées services.

Un service Web est un composant logiciel (programme informatique) identifié par une URI (Uniform Resource Identifier) , dont les interfaces publiques sont définies et appelées en XML (Extensible Markup Language), cette dernière peut être découverte par d'autres systèmes logiciels.

Les services Web peuvent interagir entre eux par leurs définitions, en utilisant des messages XML portés par les protocoles d'internet.

#### 3.1 Type de de service web

Il existe plusieurs services tels que le service logique, le service web concret et le groupe de services, nous définissons les trois services comme suit dépend de [40] classés dans le tableau 6:



Nom de service	Définition
<b>Service logique</b>	Les services logiques sont des abstractions d'un ensemble de services chacun peut remplir certaines fonctions en masquant les détails techniques et encapsulant les informations sémantiques. Par exemple une requête de la reconnaissance faciale.
<b>Service concret</b>	Les services concrets sont des services physiques, utilisés pour désigner un service web concret, qui est défini comme $sc=(Bi, Op, Cnf, Impl)$ tel que : - Bi : Basic Information of the concrète web service. - Op : Operations of web service . - Cnf : Non-Functional attributes of the Concrete web service. - Impl : the Implementation file of the concrete web service. Par exemple c'est la recherche des services qui font la reconnaissance faciale.
<b>Services Cluster</b>	C'est un groupe de services qui doit rassembler les services web satisfaisant les mêmes besoins par le biais d'un interface. Un cluster de services contient un service logique et un ensemble de services concrets qui sont des équivalences. Par exemple c'est l'ensemble des services qui sont sélectionner ou composer pour faire la reconnaissance faciale .

Tableau 6: Types de services web[40]

## 4 Caractéristiques de la qualité de service

La qualité du service est un sous-ensemble de caractéristiques non fonctionnelles, qui est différente selon les points de vue de [43, 44, 45] . En général, un critère non fonctionnel se divise en types de caractéristiques quantitatives et qualitatives selon [46, 47]. On cite quelque uns ci-dessous :

- **Temps de réponse**

Le temps nécessaire pour qu'un service accomplit sa tâche (entre le moment de la demande de l'utilisateur et le moment de la réponse reçue).

- **Fiabilité du service**

C'est la continuation de fonctionnement correcte de manière cohérente dans la période spécifiée et sous certaines conditions.

- **Réputation :**

Une réputation dépend de l'acceptation du service par les utilisateurs.

- **Disponibilité du service**

C'est la disposition du fonctionnement du service, elle est liée à la défaillance du système.

- **Coût de service**  
Signifie le prix d'un service, généralement calculé comme suit :  
Coût total = coût d'exécution du service + (réseau transport / transaction) coût.

- **Pertinence :**  
C'est le niveau d'accord entre les exigences du client et les fournisseurs du service.

#### 4.1 Type de critère de QoS :

Il existe deux types :

- **Les critères positifs :** Représente les critères qui favorisent la maximisation de leurs valeurs comme la sécurité, la réputation, la disponibilité.
- **Les critères négatifs :** Représente les critères qui favorisent la minimisation de leurs valeurs comme le temps de réponse et le coût.

- Il existe une méthode de normalisation pour éviter d'avoir le problème d'évaluation des résultats, par exemple un score max est favorable dans les critères positifs par contre il est défavorable en critères négatifs, si on n'utilise pas cette méthode le choix de la classification du score sera impossible.

## 5 Règles de SLA

SLA s'agit d'un format de document juridique, définit les objectifs attendus du fournisseur, il se compose de plusieurs composants, selon [57] on mentionne les éléments suivants :

- **Contrat :** Le contexte d'accord consiste à des informations générales telles que les parties mentionnées et le cycle de vie ect. Ces informations comprennent l'adresse et le profil du producteur de services ainsi que du consommateur.
- **Conditions de service :** C'est un élément clé de SLA, composé de la référence de service, qui est la disponibilité d'URL des services déterminée, et des informations sur les paramètres .
- **Les principales exigences des SLA :** Représente le format qui doit être une définition d'un service claire, de sorte que le consommateur doit comprendre le fonctionnement de service, elle doit aussi fournir un niveau d'efficacité de service, des méthodes de contrôle des paramètres de service, une pénalité en cas de non-respect des services.

## 6 Description de Service

Un service web permet une communication des applications entre un client et un serveur via internet en effectuant l'appel par RPC (Remote Procedure Call) qui signifie l'appel à distance. Les capacités d'un service web peut être décrite comme suit selon [58] :

### 6.1 Description structurelle

Signifie l'interface qui est un standard du W3C (World Wide Web Consortium), qui permet de définir la structure abstraite et concrète d'un service, basé sur un document XML qui décrit la signature des opérations offertes par le service (nom d'opérations, noms et types des paramètres d'entrées/sorties).

## 6.2 Description comportementale

Décrit l'interaction, l'ordre d'invocation des opérations d'un ou plusieurs services. Un service peut avoir plusieurs descriptions comportementales et chaque description est une vue sur un comportement possible du service dans une communication donnée. Nous distinguons deux types de modèles comportementaux : l'orchestration (centralisée) et la chorégraphie (décentralisée).

## 7 Découverte de service

Il existe plusieurs méthodes de découverte parmi ces dernière on cite l'UDDI (Universal Description, Discovery and Integration) , qui est une spécification d'annuaires de services Web , cette norme W3C propose un ensemble de structures à publier par les fournisseurs de services sous forme XML.

Nous notons que l'accès à l'UDDI est réalisé par des services Web. Il existe des d'annuaires UDDI publiques et privés selon [48]:

- **Annuaire UDDI public** Est implémenté sous forme d'un réseau de nœuds UDDI, ces nœuds sont synchronisés, chacun d'eux est fourni par une entreprise donnée (telles que IBM, Microsoft..), le service dans ce cas propage automatiquement ses informations aux différents nœuds UDDI. L'accès à l'ensemble des informations des registres peut se faire gratuitement à n'importe quel nœud UDDI .
- **Annuaire UDDI privé** N'est accessible que sur l'intranet d'une entreprise particulière (ou d'un groupe d'entreprises).

## 8 Communication des services web :

Cette couche spécifie les protocoles d'échanges de documents XML entre le service web et ses clients, elle caractérise aussi le mode d'échange (s'il est bloquant ou non).

Le protocole SOAP (Simple Object Access Protocol) est adopté comme un standard pour la messagerie entre les services web, il permet un échange de message indépendant des plateformes, c'est un produit de Microsoft et IBM. Sa première version été acceptée par le W3C en 2000. Il est constitué d'une enveloppe XML, et d'une entête d'un protocole de transport tels que :HTTP, FTP, SMTP, JMS (Java Messagerie Services) [48].

## 9 Composition des services

Plusieurs langages de compositions de services ont été proposés dans la littérature. Le langage BPEL (Business Process Execution Language) est l'un des plus importants formalismes, car c'est un langage qui s'articule sur deux formalismes prédécesseurs : le WSFL(Web Services Flow Language) d' IBM et le XLANG(Xml Based-Langage) de Microsoft. Le but de BPEL est de créer une fonctionnalité complexe qui réutilise les services existants.

Une composition consiste à combiner les fonctionnalités de plusieurs services au sein d'un même processus dans le but de répondre à des demandes complexes qu'un seul service ne pourrait pas satisfaire.

La composition de services est considérée comme l'une des motivations les plus importantes du paradigme SOA, plusieurs définitions ont été proposées pour le concept de composition.

Nous allons définir la composition des services web, et donner son cycle de vie [57] :

### 9.1 La composition des services Web

Est un moyen efficace pour créer, exécuter, et maintenir des services qui dépendent à d'autres services. La composition des services peut être modélisée comme un problème multidimensionnel à choix multiples. Un autre type de problème de composition conscient de la QoS (Quality of Service) est de déterminer un processus d'entreprise qui optimise le processus de bout en bout des critères de qualité de service.

Dans [62], les auteurs modélisent les exigences fonctionnelles et non fonctionnelles en contraintes entières et proposent une technique de programmation en nombres entiers pour résoudre les problèmes.

### 9.2 Cycle de vie d'une composition des services :

La composition contient plusieurs activités durant son cycle de vie, nous allons présenter les différents cycle de vide d'une composition des services selon [64]:

#### 1. La phase de définition :

Le consommateur précise, la spécification de ses besoins avec la priorité de service composée, ces exigences construit un modèle abstrait contenant un ensemble d'informations (contrôle, flux des données, QoS).

#### 2. La phase de sélection :

Permet de trouver les services qui correspondent aux besoins définies ; la découverte des services est basé sur les description des services publiés, par conséquent le meilleur service doit être sélectionné.

#### 3. La phase de déploiement :

Le service composé est déployé pour qu'il soit instancier afin d'être utilisé par les utilisateurs finaux, ce qui donne un service composé exécutable.

#### 4. La phase d'exécution :

Un mécanisme d'exécution est chargé de créer et d'exécuter l'instance de service composé tout en assurant la performance .

## 10 Sélection de service

La sélection des services selon [48] est l'une des problématiques les plus importantes de l'architecture orientée service, qui consiste à choisir un seule service parmi plusieurs services qui fonctionnent de la même façon. Pour la présenter on considère l'exemple suivant :

On suppose qu'il y a un utilisateur qui veut planifier un voyage, pour cela il a besoin de consommer trois types de services :

Une réservation de billet d'avion, une réservation d'hôtel et une location de voiture. On doit sélectionner un seul service de chaque catégorie, en utilisant les critères de QoS qui possède la réputation, la fiabilité, le coût, le temps d'exécution. . .

L'utilisateur exige des contraintes sur chaque critère de QoS, par exemple le coût des trois services, ne doit pas dépasser une certaine limite. Donc il est claire que les solutions exactes de ce problème ont une complexité exponentielle par conséquent nous ne pouvons plus garantir les exigences en temps réelles, ou le passage à l'échelle (scalability).

## 11 Substitution de service Web

L'informatique orientée services [52, 53, 54] offre une approche puissante pour l'assemblage de systèmes distribués complexes à partir de composants logiciels développés indépendamment dans de nombreuses applications.

Parmi les substitutions de candidats qui offrent la fonctionnalité souhaitée, l'utilisateur peut préférer certaines substitutions à d'autres basées sur les attributs non fonctionnels du service, par exemple la sécurité. La substitution de service basée sur les propriétés fonctionnelles des composants était abordée par de nombreux auteurs, elle exige que le pouvoir exprime ses préférences par rapport aux propriétés des services.

Ces préférences peuvent être qualitatives ou quantitatives : **Les préférences qualitatives** s'affirment sur la base de la qualité relative de deux alternatives, tandis que **les préférences quantitatives** oblige l'utilisateur à quantifier de combien il en préfère une alternative à une autre, tel que **les préférences qualitatives** sur de multiples attributs peuvent être difficiles à obtenir des utilisateurs .

### 11.1 La substitution adaptative des services

La Substitution des services se produit également en cas de défaillance du service, de dégradation de la qualité ou des exigences. Selon [56] Idéalement, le service concret devrait être sémantiquement équivalent au service.

Cependant, ce n'est pas toujours le cas en réalité puisque les services disponibles sont généralement fournis par des vendeurs tiers indépendants. En conséquence, des inadéquations (mismatches) existent toujours, soit au niveau de l'interface de service, soit au niveau du protocole.

L'interface de service définit l'ensemble des opérations qu'un service fournit ainsi que les formats de messages et les types de données. Le protocole de service précise l'ordre dans lequel les opérations d'un service peuvent être invoquées.

Avec l'aide des invocations de services abstraits sont adaptées à des mises en œuvre concrètes, en faisant une substitution satisfaisante.

### 11.2 Adaptation du service

La réussite du développement et du déploiement d'applications logicielles orientées services repose sur la résolution efficace des problèmes interdépendants.

Un consommateur recherche une composition qui satisfasse les exigences fonctionnelles et de préférence contient des attributs qualitatifs non fonctionnels souhaités tels que la sécurité et le coût. Le processus d'adaptation d'un service est comme suit :

**(a) La composition des services** : la composition des services à partir d'un ensemble de composants dans un dépôt qui satisfait aux exigences données .

**(b) La substitution** : identifier les alternatives appropriées pour remplacer les services de composants défaillants.

(c) **Adaptation** : Modification des services composites existants en réponse à des changements dans la composition fonctionnelle/non fonctionnelle.

### 11.3 Calcul de la similarité

L'étude de la similarité des services cloud consiste à étudier plusieurs critères tous dépendent de la méthode de calcul utilisée.

Deux types d'analyses peuvent être considérées: analyse syntaxique et analyse sémantique. Ces dernières renvoient des mesures de similarité entre  $[0,1]$  : plus la mesure tend vers 1 plus les éléments sont considérés similaires. Nous allons la définition de [65] pour détailler l'analyse de similarité :

#### 11.3.1 Analyse syntaxique

Cette analyse se base sur la structure des chaînes à comparer, plus elles partagent de caractères en commun, plus elles sont similaires. Cette analyse peut être divisée en deux sous-catégories : **les méthodes basées sur des chaînes de caractères** (analyse la structure), **les méthodes basées sur les tokens** .

#### 11.3.2 Analyse sémantique

L'étude sémantique consiste à attribuer à un couple de mots une métrique basée sur la ressemblance au niveau sémantique. Plusieurs mesures de similarité sémantique existent, la plupart reposent sur une base de donnée lexicale comme WordNet .

Cette analyse peut être divisée sous trois catégories : **Calcul de similarité par le nombre d'arc** (choisir le chemin le plus court), **Calcul de similarité par le contenu informatif** (traduit la pertinence d'un concept dans le corpus en tenant compte de la fréquence de l'apparition des mots), **Calcul de similarité hybride** (combinaison entre la première et la deuxième catégorie).

Le tableau 7 résume les catégories du calcul de la similarité :

Analyse Syntaxique			Analyse sémantique					
Chaînes de Caractères		Tokens	Nombres d'arcs		Contenu informatif		Hybride	
Jaro	Jaro Winkler	X	Mesure de Rada et al	Wu et Palmer	Resnik	Lin	Jiang & Conrath (Jiac)	Leacock et Chodorow (Lec)
<p>-s et t deux chaînes de caractères.</p> <p>-m le nombre des caractères communs dans les deux chaînes.</p> <p>-t' le nombre des caractères communs dans des positions différentes.</p> <p>La distance de Jaro entre les chaînes et est définie par :</p> $d_j = \frac{1}{3} \left( \frac{m}{ s_1 } + \frac{m}{ s_2 } + \frac{m-t}{m} \right)$	<p>-p un coefficient de préfixe qui favorise les chaînes commençant par un préfixe de longueur <math>\leq 4</math>.</p> $d_w = d_j + (\ell p(1 - d_j))$ <p>-d<sub>j</sub> est la distance de Jaro entre s1 et s2</p> <p>-l est la longueur du préfixe commun (maximum 4 caractères)</p> <p>-p est un coefficient de préfixe commun. Winkler propose p = 0.1</p>	<p>Analyse les textes longs comme les documents</p>	<p>Calcule la similarité en se basant sur les liens hiérarchiques.</p> <p>-un objet A est similaire à un objet B qu'à un objet C, si la distance de A à B au sein du graphe est plus courte que celle de A à C.</p>	<p>Mesure la profondeur de deux concepts donnés dans la taxonomie WordNet, ainsi que de leur plus bas ancêtre commun et les combine pour obtenir un score de similarité</p>	<p>Traduit la pertinence d'un concept dans le corpus en tenant compte de la fréquence de son apparition dans le corpus ainsi que de la fréquence d'apparition des concepts qu'il subsume. On dit qu'un concept C1 subsume un concept C2 si C2 est plus spécifique que C1.</p>	<p>La mesure proposée par Lin est une normalisation de celle de Resnik. La normalisation est faite en factorisant par le contenu de l'information.</p>	<p>Combine entre les techniques basées sur les arcs et les techniques basées sur les nœuds qui consistent à compter les arcs afin d'améliorer les résultats par des calculs basés sur les nœuds.</p>	<p>Présentée par Lecqui qui combine entre la méthode de comptage des arcs et la méthode du contenu informationnel.</p>

Tableau 7: Mesure de la similarité[65]

## 12 Etude comparative entre les solutions de la substitution des services cloud de type SaaS

Dans cette section, nous allons présenter une études comparatives de la substitution des services Cloud de type SaaS en comparant les différents approches de description, composition, sélection, substitution des services dans le Cloud.

Notre comparaison est basée sur les critères d'évaluation discutés précédemment pour pouvoir récapituler les comparaisons effectuées dans un tableau finale.

### 12.1 Approche de description des services :

Plusieurs langues de description ont été proposées pour la description du service Web et pour les services clouds. Le W3C et l'OASIS (Organization for the Advancement of Structured Information Standards) sont les deux principaux organismes de normalisation dans ce domaine. On résume quelques uns dans le teableau 8 :

Langues de description	Définition
WSDL [66]	Se concentre sur une description syntaxique des interfaces de services Web et manque de fournir les informations sémantiques.
L'ontologie(OWL ) [67]	Décrit des informations sémantiques sur les services qui est faite explicite par référence à un vocabulaire structuré. OWL(Web Ontology Langage) Est une norme du W3C.
USDL (Universal Service-Semantics Description Language) [68]	Un langage pour décrire formellement la sémantique des services Web ,il est basé sur le langage OWL.

Tableau 8: Langage de de description

### 12.2 Approche de sélection des services :

La sélection d'un service Cloud est une étape très importante et pour cela elle doit être précise et correcte. Nous citant ci-dessous dans le tableau les algorithmes les plus utilisés mentionnés dans le tableau 9 :



Nom de l'algorithme	Définition
<b>Sel CSP(Selection of Cloud Service Provider)</b>	Basée sur le calcul de la fiabilité, pour trouver les risques des services causés par les fournisseurs en respectant les contrats de SLA [69]
<b>MCDM (Multi-Criteria Decision making)</b>	Utilise un nombre finis d'attributs en fonction des besoins du consommateur [70]
<b>Algorithme CMFactory</b>	Permet de sélectionner les services Cloud en fonction des exigences fonctionnelles et non fonctionnelles identifiées par l'utilisateur [71]
<b>AHP (Analytic Hierarchy Process)</b>	La sélection de produits SaaS est modélisée selon une hiérarchie composée par plusieurs critères comme le coût, la qualité, la flexibilité, l'utilisabilité [72]

Tableau 9: Méthodes de sélection

### 12.3 Approche de composition des services :

La composition des services cloud est un moyen pour satisfaire le client ; plusieurs compositions ont été proposées, on note les plus connues selon [73, 74, 75] cités dans le tableau 10 :

Composition	Définition
<b>Composition manuelle</b>	L'utilisateur sélectionne et exécute le service en les surveillant pendant son exécution.
<b>Composition automatiques</b>	L'utilisateur spécifie le langage de description ensuite le service approprié est sélectionné avec transparence.
<b>Composition semi-automatique</b>	Offre une automatisation de certaines parties de la composition et donne la chance à l'utilisateur de contrôler le reste.
<b>Composition statique</b>	Le choix des services participants se fait durant la compilation.
<b>Composition dynamique</b>	Le choix des services participants se fait durant l'exécution.

Tableau 10: Types de compositions [73, 74, 75]

## 12.4 Approche de substitution des services :

Un client s'intéresse à un service, en général, pour ses fonctionnalités définies par les opérations, donc il suffit de chercher l'opération pour trouver les services souhaités. Le client exprime ces besoins par une requête ensuite une liste des services cloud correspondant à ses besoins est retournée.

Avant d'aborder les solutions de la substitution il faut tout d'abord commencer par les travaux existants sur la similarité entre les services cloud.

### 12.4.1 Synthèse des travaux sur le calcul de similarité des services

Il existe plusieurs travaux sur le calcul de la similarité des services cloud on cite quelques uns dans le tableau 11 suivant :

Références	Solution de substitution	Remarque
[97]	Appelée « WSSim » Le degré de similarité entre opérations doit être supérieur ou égal à 0,7, et le degré de similitude entre leurs messages (entrées et sorties) est supérieur ou égal à 0,75.	La solution est valable si le client cherche à substituer les services avec les mêmes entrées et les mêmes sorties.
[98]	Une opération qui produit la même chose qu'une autre opération, en général, signifie qu'elles font la même chose.	
[99]	L'étude de similarité a été basée sur l'étude de similitude de leurs opérations, en utilisant une technique de cartographie ..	
[100]	Appelée «Woogle». À l'intérieur, il existe un algorithme de clustering pour regrouper les paramètres de service dans un concept similaire.	La solution est basée sur les services alors que le client peut chercher une opération au lieu de chercher un service.
[101]	L'étude de similarité est basée sur la mesure de la similarité entre les descriptions (documentation) de divers éléments.	La solution est basée sur la documentation des services, mais la majorité des services n'étaient pas documentés.
[101]	L'étude de similarité est basée sur la comparaison sémantique d'éléments extraits de fichiers WSDL tels que les noms de paramètres et noms d'opérations, en utilisant la méthode Wu.	La solution est basée sur la sémantique, les auteurs considèrent quelques types et ignorent d'autres.

Tableau 11: Différentes solutions de la substitution

### 12.4.2 Synthèse des travaux sur la substitution des services

La substitution consiste à remplacer le système défaillant par un autre opérationnel. Chacun des clients a des besoins différents donc il est difficile de définir formellement la substitution. Parmi plusieurs solutions proposées ; nous allons se concentrer sur trois solutions seulement :

#### 12.4.2.1 Solution à base communauté

Un service peut contenir plusieurs opérations, chacune prend un ensemble de paramètres d'entrées et produit un ensemble de paramètres de sorties. Une relation entre les entrées peut être ignorée si deux opérations appartiennent au même domaine d'étude, et se focaliser sur les paramètres de sorties qui représente la fonctionnalité des opérations.

Alors pour substituer un service échoué par un autre qui fonctionne il suffit seulement de substituer les opérations fonctionnellement équivalente, selon [65] le processus de substitution à base communauté est comme suit :

Cette approche consiste à regrouper dans un même cluster les opérations offrant le même fonctionnement. Les étapes de substitution consistent à déterminer les opérations fonctionnellement équivalentes, de trouver les relations entre les opérations, et de former les communautés comme le montre le tableau 12 .

Un regroupement des opérations nécessite l'utilisation d'un algorithme de classification, il existe deux méthodes classificatoires : Supervisée et non Supervisée.

- **La classification supervisée :** Un expert crée des classes d'appartenance pour chacune des donnée , alors on peut affecter une nouvelle donnée à la classe la plus conforme parmi les classes existantes.
- **La classification non supervisée :** La création des classes qui regroupent les données se fait selon les catégories des données et non pas à l'avance, cette technique utilise deux types de partitionnement:
  - ✓ **Hiérarchique :**  
Permet de partitionner un jeu de données de manière hiérarchique, l'idée est d'assembler les deux clusters (catégories) les plus proches à chaque fois, contient deux façons :
    - Ascendant : l'algorithme essaie de fusionner deux clusters en un seul.
    - Descendant : l'algorithme essaie de découper un gros cluster en deux clusters le plus petits possible.
  - ✓ **Non hiérarchique :**  
Fourni un partitionnement en décomposant tous les individus en différents ensembles sous un nombre fixés.

La solution à base communauté consiste à utiliser une méthode non supervisée car le regroupement des opérations n'est pas définis à l'avance, cette technique utilise un partitionnement non hiérarchique, le choix est orienté vers l'algorithme le plus rapide K-means.

✓ **L'algorithme K-means (L'algorithme se trouve dans « l'Annexe A » )**

On suppose qu'il existe K classes différentes, Pour chacune il faut designer « K » centres «  $c_1, \dots, c_K$  », ces derniers sont choisis par l'utilisateur ou bien désignés aléatoirement.

Les étapes suivantes sont effectuées itérativement :

- ✓ On cherche quel est le centre de classe le plus proche aux individus.
- ✓ On définit les classes  $C_i = \{\text{ensemble des points les plus proches du centre } c_i\}$ .
- ✓ Dans chacune des classes  $C_i$ , on définit un nouveau centre de classe.
- ✓ Il existe deux critères d'arrêt de l'algorithme : soit l'utilisateur fixe un nombre limite d'itérations ou bien jusqu'à la formation des classes finales (entre deux itérations les positions des données restent inchangées).

<p align="center"><b>Déterminer les opérations fonctionnellement équivalentes</b></p>	<p align="center"><b>Construire les Communautés similaires</b></p>
<p>-Extraction du Nom de service + nom d'opération+ nom de message de sortie. -<b>Concaténer</b> les trois identificateurs.</p>	<p align="center"><b>Calcule de distance entre les opérations :</b></p> <p>-La distance entre deux opérations est définie par leur degré de similitude.</p>
<p>-Utiliser la méthode de <b>Wu palmer</b> pour mesurer la similarité si les deux identificateurs existe dans le <b>WordNet</b> sinon utiliser la méthode de <b>Jaro-Winkler</b>.</p>	<p align="center"><b>Le Centre du cluster :</b></p> <p>-L'algorithme K-means consiste à déterminer pour chaque cluster un centre dont l'opération (opération) sera affecté au groupe dont la distance entre l'opération et le centre est minimale.</p>

Tableau 12: Solution de la substitution à base communauté[65]

### 12.4.2.2 Solution à base réseau

Une deuxième approche est proposée selon [76] qui consiste à utiliser les même deux premières étapes mais se diffère à la dernière, c'est-à-dire au lieu de construire les cluster de communauté, il faut former un réseau similaire basé sur un représentation graphique et matricielle.

Les étapes de substitution consistent à déterminer les opérations fonctionnellement équivalentes, de trouver les relations entre les opérations, et de former le réseau similaire comme le montre le tableau 13 suivant :

Déterminer les opérations fonctionnellement équivalentes	Relations entre les opérations	Former le réseau similaire
<p>-Extraction du Nom de service + nom d'opération+ nom de message de sortie. -<b>Concaténer</b> les trois identificateurs.</p> <p>-Utiliser la méthode de <b>Wu palmer</b> pour mesurer la similarité si les deux identificateurs existe dans le <b>WordNet</b> sinon utiliser la méthode de <b>Jaro-Winkler</b>.</p>	<p><b>Egalité :</b></p> <p>-Si les deux opérations ont le même nombre de paramètres. -Toutes les valeurs de similarité sont supérieur ou égal au seuil (<math>\geq 0.7</math>)</p>	<p><b>Le graphe qui représente les relations :</b> Une opération = nœud ; Une relation= arc</p> <p><b>-Egalité :</b> Deux nœuds avec deux arcs inversés</p> <p><b>-Inclusion :</b> Deux nœuds avec un seul arc (L'opération qui a plus de parametres est considérée comme nœud initial qui tend vers l'autre opération ).</p>
<p>-Appliquer la méthode <b>Hangroise</b> pour calculer le score max qui représente le degré de la similarité entre deux identificateurs.</p>	<p><b>Inclusion :</b></p> <p>-Si le nombre de paramètres est différent. - Toutes les valeurs de similarité sont supérieur ou égal au seuil (<math>\geq 0.7</math>). - Nous avons deux cas:</p> <p>L'opération dont le nombre de paramètres est inférieur ne peut pas remplacer celle avec plus de paramètres mais le contraire est possible.</p>	<p><b>La matrice qui modélise le graphe</b></p> <ul style="list-style-type: none"> <li>- Un nœud initial= 1</li> <li>- Autre nœud= 0</li> </ul>

Tableau 13: Solution de la substitution à base réseau[76]



### 12.4.2.3 Solution à base SLA

Dans CSBA (Cloud Service Based Application), un client peut louer un service cloud d'un ou de plusieurs fournisseurs pour créer sa candidature. En règle générale, un SLA comprend un ensemble d'objectifs de niveau de service(SLO) chacun représente une contrainte sur le système.

Les SLA définissent généralement des pénalités, en termes pécuniaires que le fournisseur doit accorder au client si une QoS est violée, desfois la violation peut menacer la disponibilité du service. Les systèmes doivent s'adapter de manière autonome aux changements soit en utilisant le système de surveillance réactif ou proactif :

- **L'approche réactive** : réagit après l'apparition de la violation.
- **L'approche proactive** : est plus préventif, réagit avant l' apparition d'une violation dans le but de l'éviter.

Tous d'abord un client exprime sa requête ensuite une instance du service voulu est créée, à ce moment le surveiller de SLA est appelé pour vérifier les violations de QoS.

Une méthode minière est utilisée pour construire un modèle de prédiction depuis des journaux d'exécution pour réagir d'une manière proactive.

SLA est écrite en langage naturel pour décrire le comportement d'un service. Les violations viennent dû aux différents échecs, il est important alors de définir le type d'échec pour pouvoir réagir correctement.

Nous présentons l'approche d'apprentissage et de prédiction des violations dans l'environnements cloud [77] expliqué dans le tableau [14] .

✓ L'approche proposée contient deux phases complémentaires :

Une phase d'apprentissage où le système découvre les violations pour la première fois et autre phase de prédiction qui est utilisé pour détecter et empêcher les violations

1

2

Phase d'apprentissage	Phase de prédiction
<p>L'apprentissage utilise un historique des journaux de violations SLA enregistrées dans la base de données. Cette phase découvre les éléments les plus fréquents dans l'ensemble et construit les règles d'associations (affecter pour chaque événement un ensemble de données suspects) en utilisant <b>l'algorithme J48</b> (représente les attributs hiérarchiquement pour l'évaluation d'appartenance de chacun à la classe la plus proche) pour extraire et raffiner les règles d'associations ou association rules(AR).</p>	<p>La prédiction utilise les règles d'associations trouvées dans la première étape pour construire une configuration qui peut prédire les violations de SLA</p>
<p><b>Extraction et raffinement des AR :</b> L'algorithme utilisé prend en entrée les journaux de violations et en résulte les règles d'associations. Il utilise les données correctes seulement, comme suit :</p> <ol style="list-style-type: none"> <li>1. Exécuter les journaux extraire les éléments les plus fréquents</li> <li>2. Trouver les règles d'associations possibles</li> <li>3. Enregistrer les règles correctes.</li> </ol>	<p><b>Prédiction des AR prédictive :</b> Prédire une violation SLA revient à prédire les valeurs des SLO (Service Level Objective) basés sur la surveillance comme suit :</p> <ol style="list-style-type: none"> <li>1. Comparer les AR en différents temps</li> <li>2. Supposer des prédictions</li> <li>3. Enregistrer les règles de prédictes correctes seulement.</li> </ol>

Tableau 14: Solution à base SLA[77]

### 12.4.3 Critères d'évaluation :

Dans le but de comparer les différents approches on utilise les critères de comparaison suivants résumés dans le tableau 15:

<b>Critères</b>	<b>Définition</b>
<b>Description</b>	Décrit les critères fonctionnelles et non fonctionnelles d'un service dans un fichier WSDL ou USDL
<b>Composition</b>	C'est le regroupement de deux ou plusieurs services dans un seul service hybride, afin de répondre au besoin de l'utilisateur.
<b>Sélection</b>	Permet de choisir le service le plus approprié qui satisfait les besoins d'utilisateur.
<b>Comparaison</b>	Analyser syntaxiquement ou sémantiquement les opérations fonctionnellement équivalentes, pour trouver la similarité entre eux.
<b>Classification</b>	Regroupe les opérations des services qui ont le même fonctionnement en utilisant des communautés ou un réseau.
<b>SLA</b>	Détecte la violation du contrat, c'est-à-dire de vérifier si le service fonctionne comme il est indiqué dans le SLA, afin de trouver les services échoués pour les substituer.
<b>Type de services</b>	Vérifier le type des services qui peut être : soit un service web ou un service cloud.

**Tableau 15: Critères de comparaison**

Le tableau 16 suivant résume l'ensemble des comparaisons déjà traitée :

Auteurs / Critères	Description		Composition	Sélection	Compararaison		Classification		SLA	Type de services	
	WSDL	USDL			Syntaxique	Sémantique	Communauté	Réseau		Web	Cloud
R.Sara [65]	√	×	×	×	√	√	√	√	×	√	×
R.Sara [76]	√	×	×	×	√	√	×	√	×	√	×
A.Meskini [77]	×	×	×	×	×	√	×	×	√	×	√
Chantal Cherifi [79]	√	×	×	×	×	√	×	√	×	√	×
BoutaharJaouad [80]	√	×	×	×	√	√	×	×	×	√	×
Boudjemaaboudaa [81]	√	×	√	×	×	√	×	×	×	√	×
Dong, X [82]	√	×	√	×	√	√	√	×	×	√	×
Rachad.T [83]	×	×	×	×	√	√	×	×	×	√	×
VishkaeiMahsa Jamal [84]	×	×	×	×	√	√	√	×	×	√	×
Huang [85]	×	×	×	×	√	√	√	×	×	×	×
OkbaTibermacine [86]	√	×	×	×	×	√	×	×	×	√	×
YehiaTaher [87]	×	×	√	×	√	√	√	×	×	√	×
Y. Taher [88]	√	×	×	×	√	√	√	×	×	√	×

Tableau 16: Comparaison entre les solutions de la substitution des services

### ➤ Discussion

Comme on peut le voir la majorité des solutions ont été proposées pour les services web et non pas pour les services cloud, ces dernières utilisent le WSDL comme une description du service, on remarque aussi que la partie de sélection était négligée.

Toutes les solutions comprennent l'analyse sémantique, par contre l'analyse syntaxique est peu utilisée, malgré l'utilisation de ces mesures de similarités la minorité des solutions qui traitent le problème de la composition des services, on constate aussi que la plus part des solutions se base sur la classification selon les communautés, par rapport à la formation des réseaux, ainsi que la considération des SLA est presque éliminée.

## 13 Conclusion

Un service cloud est un programme informatique accessible via internet, qui fournit différent services à plusieurs consommateurs tout dépend de leurs besoins.

Dans ce chapitre, nous avons présenté un état de l'art sur la substitution des services web ainsi que sur les services cloud.

Le chapitre contient deux parties, la première partie décrit les notions de base de la substitution, et la deuxième comprend des synthèses sur les travaux réalisés.

Dans le chapitre suivant, nous allons présenter une conception de notre solution pour la substitution des services cloud de types SaaS, en décrivant les méthodes et les algorithmes utilisés.

## 1 Introduction

Le cloud computing est dynamique en raison du concept de migration de machine virtuelle, qui a été rendu possible par la virtualisation qui est une des technologies clés du cloud computing.

Après avoir entamer les concepts de base du cloud computing, et comparer les solutions existantes dans le chapitre précédent, nous allons introduire dans ce chapitre notre solution pour la substitution des services cloud de type SaaS.

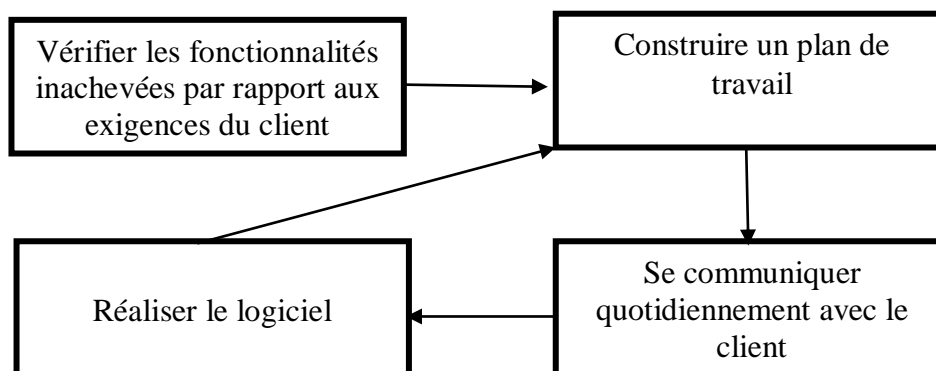
La solution proposée consiste à traiter les problèmes de description des services cloud, de sélectionner un service SaaS et enfin substituer les services échoués.

## 2 Méthodologie de travail

Dans notre projet nous avons utilisé une méthodologie de eXtreme Programming(XP), il s'agit d'une méthode agile de gestion de projet conçu pour améliorer la qualité du logiciel et sa capacité à s'adapter correctement aux besoins changeants du client.

Il existe plusieurs catégories de planification, de gestion, de conception, de codage et de test dans le XP, il est défini par ses règles pas ses pratiques telle que les «**règles d'engagement**» qui dictent l'environnement dans lequel le développement logiciel peut avoir lieu efficacement, et les «**règles du jeu**» qui définissent les activités et les règles minute par minute dans le cadre des règles d'engagement [108].

Le schéma suivant explique la méthode XP :



Le processus qui comprend la construction d'un plan de travail, la communication avec le client, et la réalisation du logiciel, se répète itérativement jusqu'à satisfaire le client.

### 3 Analyse des besoins

On utilise dans ce système trois acteurs principaux, qui sont :

**Fournisseur (provider):** l'acteur qui produit les services

**Client:** l'acteur qui consomme les services.

**Admin :** l'acteur qui gère le système.

Le tableau 17 montre le rôle de chacun d'eux :

Acteurs principaux	Rôle
<b>Fournisseur</b>	Se connecter. Créer compte. Créer service. Gérer services. Consulter les services.
<b>Client</b>	Se connecter. Créer compte Demander un service. Choisir les services. Exiger les QoS.
<b>Admin</b>	Substituer les services. Prédire les services violés.

**Tableau 17: Acteurs principaux et leurs rôles dans notre application Web**

Le diagramme de cas d'utilisation décrit dans la **figure 11** affiche l'interaction entre les acteurs et notre système :

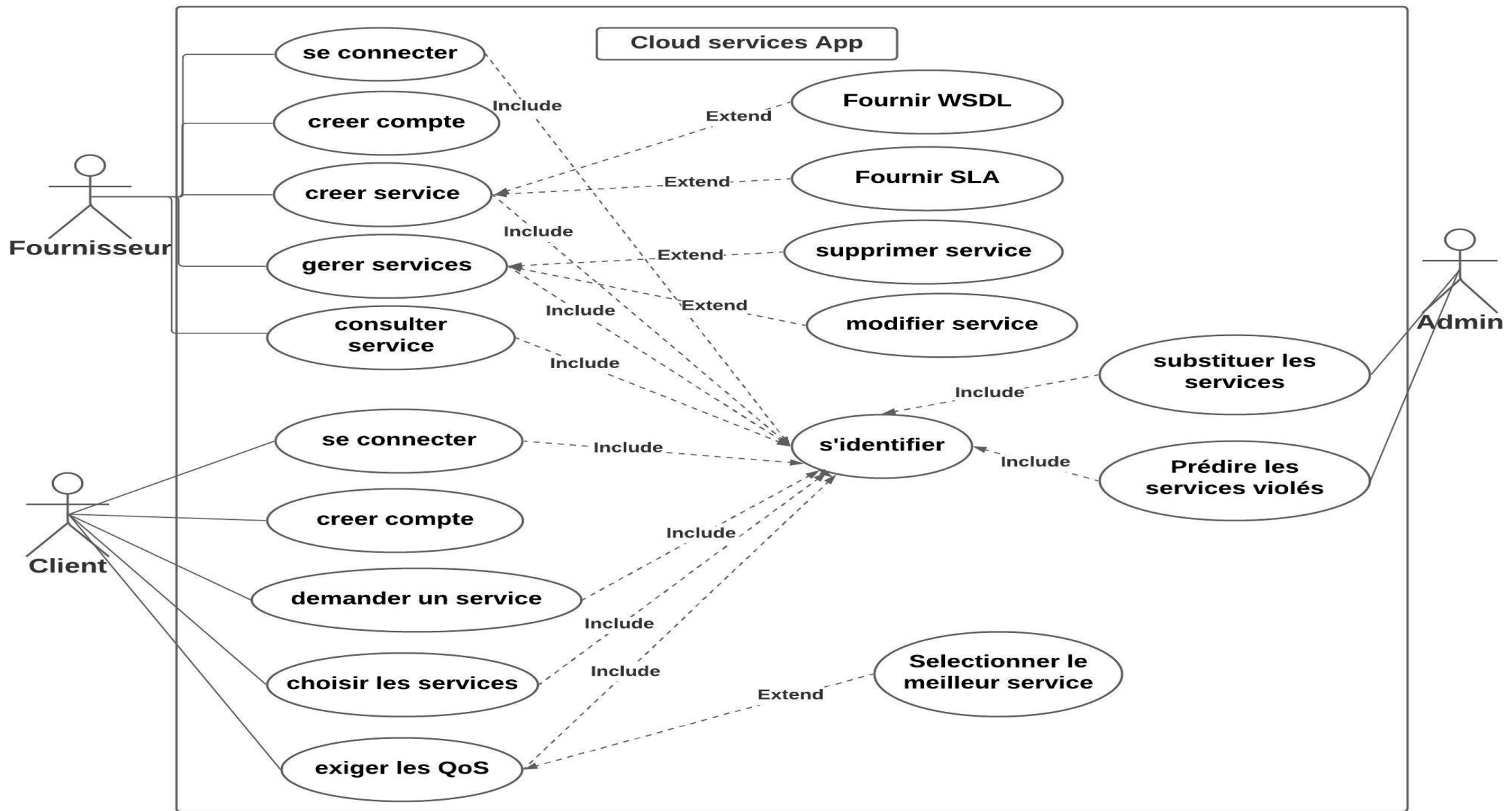


Figure 11: Diagramme de cas d'utilisation



## 4 Solution proposée

Nous avons proposé une solution qui résout le problème de la substitution des services cloud de type SaaS. Notre solution aide le client à savoir les services échoués par une prédiction et propose une substitution qui comprend trois options: substitution lexicale, substitution globale, substitution précise.

Afin de rendre notre client satisfait nous avons pensé à une substitution des services aux choix dans le but de le mettre plus à l'aise, donc il peut sélectionner librement l'option qui le convaincra mieux en cas d'échec du service, comme il peut aussi consulter la page « prédiction » qui donne des prévisions du dysfonctionnement des services.

### 1. Prédiction « basée sur SLA » :

Comprend une phase d'apprentissage basée sur la collection des informations, et une autre phase de prédiction en cas de violation des exigences fonctionnelles mentionnées dans le SLA .

Ces prédictions aident à connaître quels services sont en panne pour pouvoir alerter le client à ne pas choisir ces derniers.

### 2. Substitution lexicale :

Utilise la similarité sémantique de Wu-Palmer qui dépend de la base de données lexical « WordNet » tel que deux mots sont considérés similaires si ils ont le même sens, dans notre cas les mots représentent la fonctionnalité des opérations des services.

### 3. Substitution globale « basée sur les communautés » :

Utilise la similarité syntaxique de Jaro-Winkler qui se base sur la distance de Jaro, nous utilisons cette méthode pour trouver la ressemblance entre les différents paramètres des opérations des services, ensuite on applique la classification K-means pour rassembler les opérations qui fonctionnent de la même manière dans des groupes.

### 4. Substitution précise « basée sur le réseau » :

Utilise la même mesure de similarité syntaxique « Jaro Winkler », ensuite on forme un réseau sous forme d'un graphe (création d'une ontologie) qui regroupe les opérations similaires.

Par rapport à ce qui existe dans la littérature nous avons ajouté les points suivants :

- a) Comparer le résultat de la distance de Jaro par rapport à  $0.7^1$ , puisque c'est le seuil qui détermine la similarité entre les différents paramètres d'opérations, cette comparaison aide à éliminer dès le début les cas dissimilaires, c'est-à-dire ceux qui ne seront pas utiles pour la substitution .
- b) Utiliser la prédiction pour les critères fonctionnelles, et non pas les critères non fonctionnelles de la QoS décrits dans le SLA, tel que chaque prédiction d'échec propose une substitution de service comme action .
- c) Combiner les quatre solutions (basée sur le lexique, basée sur les communautés, basée sur le réseau, et la solution de la prédiction basée sur le SLA) .

Le plan de travail de cette proposition est détaillé dans l'organigramme de la **figure 12** :

---

<sup>1</sup> 0.7 représente le seuil de la similarité dans la littérature

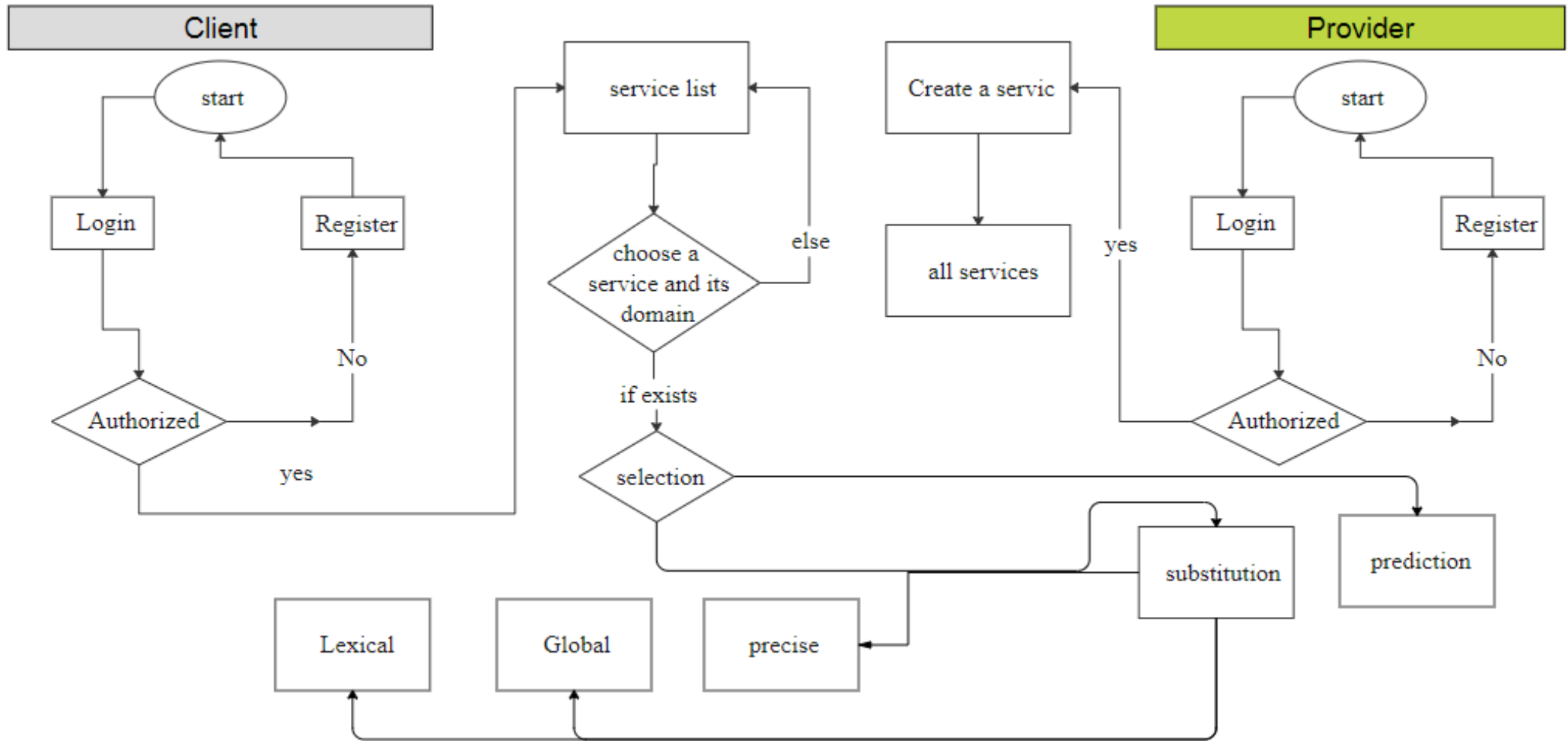


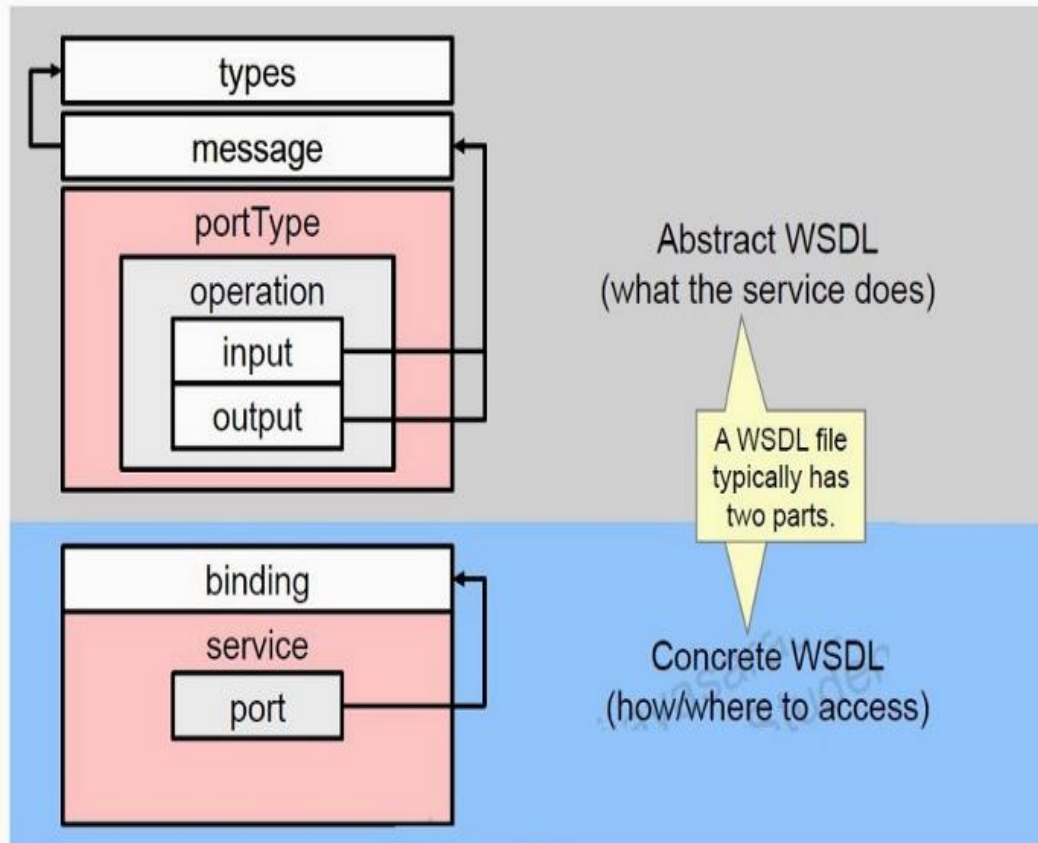
Figure 12: Plan de travail de notre solution pour la substitution

## 5 Description des services Cloud de type SaaS

Un cloud peut contenir plusieurs services, pour ce faire il faut utiliser une description pour chacun des services. Dans notre système nous utilisons une description WSDL pour les services cloud de type SaaS .

### 5.1 Structure du fichier WSDL :

Le fichier WSDL selon [65] est constitué des deux parties concrète et abstraite comme l'indique la **figure 13** :



**Figure 13: La structure du fichier WSDL [94]**

**Section abstraite :** Définit les éléments de l'interface du service cloud

- 1.type de services :Il peut être simple ou complexe afin de décrire les données utilisées entre le fournisseur et le client.
- 2.Message : Décrit les paramètres utilisés par les opérations du service cloud.
- 3.Port type : Contient les différentes opérations du service cloud

**Section concrète :** Décrit comment un client peut accéder à un service cloud

- 1.Binding : Définit les protocoles de communication ainsi que l'URL pour localiser le service cloud.
- 2.Service : C'est une collection des ports dont le but de montrer comment accéder au service cloud .

Soit un service Cal qui fait une addition, la **figure 14** montre la partie abstraite par contre la **figure 15** montre la partie concrète :

```
<wsdl:types>
  <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://c.b.a">
    <xs:element name="add">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="n1" type="xs:int" />
          <xs:element minOccurs="0" name="n2" type="xs:int" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="addResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="return" type="xs:int" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</wsdl:types>
<wsdl:message name="addRequest">
  <wsdl:part name="parameters" element="ns:add" />
</wsdl:message>
<wsdl:message name="response">
  <wsdl:part name="parameters" element="ns:response" />
</wsdl:message>
<wsdl:portType name="CalculatorPortType">
  <wsdl:operation name="add">
    <wsdl:input message="ns:addRequest" wsaw:Action="urn:add" />
    <wsdl:output message="ns:response" wsaw:Action="urn:addResponse" />
  </wsdl:operation>
</wsdl:portType>
```

Figure 14: Partie abstraite du service

```
</wsdl:binding>
<wsdl:binding name="CalculatorSoap12Binding" type="ns:CalculatorPortType">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <wsdl:operation name="add">
    <soap12:operation soapAction="urn:add" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="CalculatorHttpBinding" type="ns:CalculatorPortType">
  <http:binding verb="POST" />
  <wsdl:operation name="add">
    <http:operation location="add" />
    <wsdl:input>
      <mime:content type="text/xml" part="parameters" />
    </wsdl:input>
    <wsdl:output>
      <mime:content type="text/xml" part="parameters" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="Cal">
```

Figure 15: Partie concrète du service

## 6 Sélection des services cloud de type SaaS

La variation des QoS ne peut pas être limitée, puisque chacun des clients a différents besoins. Afin de choisir le meilleur service parmi les services publiés qui ont les mêmes critères fonctionnels, nous allons laisser le choix des critères non fonctionnels d'un service au consommateur pour l'aider à sélectionner le service le plus approprié.

La sélection d'un service parmi des services disponibles nécessite une prise de décision.

Pour résoudre ce problème on adopte la méthode MCDM (Multiple-criteria decision-making), puisque elle utilise la QoS, et elle est plus flexible, car elle donne la main au client pour exiger ces critères de la QoS, de cette façon il peut trouver le service qui répond le mieux à ses besoins.

## ➤ Méthode MCDM

La prise de décision consiste à formuler le problème de décision . Nous présentons le problème de sélection des services cloud dans un contexte généralisé sous un modèle mathématique [78] :

### 1. Service

Soit  $S_1 = \{S1, S2, S3, \dots, Sl\}$  l'ensemble des services publiés par les fournisseurs tel que  $l \geq 2$ .

### 2. Critère de performance du fournisseur

Soit  $C = \{c1, c2, c3, \dots, cm\}$  l'ensemble des critères de service utiles pour la sélection, tel que  $m \geq 2$ .

### 3. Fonctions de mesurent de performance

Soit  $F = \{f1, f2, f3, \dots, fm\}$   $f_i$  représente la fonction d'évaluation de chaque critère  $C_i$ , tel que  $m \geq 2$ .

### 4. Descripteur de service

Soit  $D_i = [d1, d2, d3, \dots, di]$  un vecteur qui décrit un service  $S_i$  sous critère  $C_i$ , tel que  $di = f_i(S_i)$ .

### 5. Matrice de décision

Les vecteurs de description de service ( $D_i$ ) sont combinés dans une matrice de décision  $A$  de taille  $l * n$ , Tel que  $a_{i,j}$  est l'évaluation de service  $i$  et le critère  $j$  donné par  $a_{i,j} = f_k(S_k)$ .

### 6. Critères de performance du client

Soit  $R = [r1, r2, r3, \dots, rm]$  un vecteur de QoS recommandés par le client tel que  $r_i$  est la valeur minimale des critères .  
La sélection de service se fait en comparant les vecteurs  $D$  et  $R$  afin de choisir les meilleurs services.

### 7. Poids de priorité

$W = [w1, w2, w3, \dots, wm]$  représente un vecteur de poids des critères qui prend la valeur 1 par défaut.  
La sélection implique une comparaison entre le vecteur  $R$  et le vecteurs  $D$ . Il existe deux méthodes :

1

#### Weighted Difference (WD)

$$\begin{pmatrix} (a_{1,1} - r_1)w_1 + (a_{1,2} - r_2)w_2 + \dots + (a_{1,n} - r_n)w_n \\ (a_{2,1} - r_1)w_1 + (a_{2,2} - r_2)w_2 + \dots + (a_{2,n} - r_n)w_n \\ \vdots \\ (a_{l,1} - r_1)w_1 + (a_{l,2} - r_2)w_2 + \dots + (a_{l,n} - r_n)w_n \end{pmatrix}$$

2

#### Exponential Weighted Difference (EWD)

$$\begin{pmatrix} e^{-(a_{1,1}-r_1)w_1} + e^{-(a_{1,2}-r_2)w_2} + \dots + e^{-(a_{1,n}-r_n)w_n} \\ e^{-(a_{2,1}-r_1)w_1} + e^{-(a_{2,2}-r_2)w_2} + \dots + e^{-(a_{2,n}-r_n)w_n} \\ \vdots \\ e^{-(a_{l,1}-r_1)w_1} + e^{-(a_{l,2}-r_2)w_2} + \dots + e^{-(a_{l,n}-r_n)w_n} \end{pmatrix}$$

### 6.1 Exemple de MCDM :

Nous allons illustrer la méthode MCDM par l'exemple ci-dessous en utilisant le tableau 18. Dans le but de choisir le meilleur produit, chaque valeur représente une valeur de performance. Les attributs sont repartis en deux catégories :

Critère bénéfique ( les valeur les plus grande sont souhaitées par exemple le stockage) par contre les critère non bénéfique (les valeurs les plus petites sont souhaitées par exemple le prix).

Dans notre cas nous nous intéressons aux critères non fonctionnelles du service . Afin de rendre tous les critères comparables on utilise une matrice normalisée.

Attribut/ critère	Prix	Stockage	Temps de réponse	Persistance
Service 1	0.8	0.5	0.75	1
Service 2	1	0.5	0.5	1
Service 3	0.667	1	1	0.7
Service 4	0.727	1	0.5	0.7
Service 5	0.889	0.5	1	1

**Tableau 18: Exemple des critères non fonctionnels d'un service**

La pondération aux critères qui représente l'importance de chaque critère, ici on va attribuer le même poids pour tous les critères, c'est-à-dire qu'ils ont une importance égale .

Nous soustrayons les critères recommandés par le client des critères exigés par le fournisseurs, ensuite nous multiplions le poids qui convient par le résultat , et on obtient une nouvelle matrice. Nous additionnons les résultats de chaque ligne et nous prenons les maximum de toutes les lignes, ce dernier représente le meilleur service comme le montre le tableau 19.



Attribut/ critère	Prix (0.25)	Stockage(0.25)	Temps de réponse(0.25)	Persistance(0.25)	Résultat
Service 1	$(0.8-0.2) \times 0.25$	$(0.5-0.4) \times 0.25$	$(0.6-0.2) \times 0.25$	$(1-0.4) \times 0.25$	= 0.425
Service 2	$(1-0.4) \times 0.25$	$(0.9-0.2) \times 0.25$	$(0.7-0.1) \times 0.25$	$(1-0.2) \times 0.25$	=0.675
Service 3	$(0.667-0.2) \times 0.25$	$(1-0.3) \times 0.25$	$(1-0.2) \times 0.25$	$(0.7-0.3) \times 0.25$	=0.591
Service 4	$(0.727-0.3) \times 0.25$	$(1-0.2) \times 0.25$	$(0.6-0.2) \times 0.25$	$(0.7-0.2) \times 0.25$	=0.531
Service 5	$(0.889-0.3) \times 0.25$	$(0.5-0.2) \times 0.25$	$(1-0.3) \times 0.25$	$(1-0.2) \times 0.25$	=0.597

Tableau 19: Application de la méthode MCDM sur la matrice normalisée

Le max= **0.675** donc le meilleur produit est le **Service 2**

## 7 Substitution des services cloud de type SaaS

Un fournisseur peut créer et publier plusieurs services dans le cloud computing , dans le but de louer ces services aux clients un contrat SLA est utilisé.

Un service loué doit fonctionner d'une façon pertinente pour satisfaire le client, d'où l'idée de substituer les services échoués pour garantir leurs fonctionnements.

Nous adoptant les solutions suivantes :

### 7.1 Prédiction ( Substitution à base SLA)

Un accord SLA est un contrat entre le client et le fournisseur qui décrit les objectifs de la QoS souvent en CSBS (Cloud Service Based System) . En général il comprend un ensemble de niveaux d'objectifs de services (SLO), chacun représente une contrainte sur le système, donc ce contrat définit des pénalités en quelle que sorte si une QoS est violée.

Notre but est d'éviter les violations de ces critères, c'est-à-dire de réagir avant l'apparition de la violation.

Nous illustrons la substitution à base SLA par l'exemple suivant :

Supposant que nous avons un service de type SaaS qui fait la multiplication des nombres entiers seulement ; on veut savoir quand est ce que ce service peut être en panne ou violé, pour éviter de l'utiliser dans ces cas, pour ce faire on doit passer par deux phases :

#### 1. Phase d'apprentissage :

Après le lancement du service l'administrateur reste en surveillance, pour collecter et enregistrer toutes les cas possibles, donc particulièrement cette étape dépend du temps.

Soit des informations collectées au niveau de la surveillance du service« multiplication » résumé dans le tableau 20:

Provider	X	Y	X*Y>0	violation
P1	-3	8	No	viol
P1	-4	8	No	viol
P1	-5	9	No	viol
P1	6	-4	No	viol
P1	-5	3	No	viol
P1	-4	2	No	viol
P3	3	5	Yes	noviol
P4	2	4	Yes	noviol
P5	-2	5	No	viol
P9	1	5	Yes	noviol

Tableau 20: Informations collectées dû la surveillance du service « multiplication »

Ces données sont décritent dans un **fichier.arff** comme **Règles d'associations** .

- **Fichier.arff**

Il est utilisé dans les projet des Maching Learning ou l'apprentissage automatique. C'est un fichier ASCII « American Standard Code for Information Interchange » sous l'extension ARFF (Attribute-Relation File Format), qui décrit un ensemble d'attributs.

Nous présentant le fichier.arff associé à notre exemple ci-dessous :

```
@relation Multiplication

@attribute provider{p1,p2,p3,p4,p5,p6,p7,p8,p9 }
@attribute x NUMERIC
@attribute y NUMERIC
@attribute sommx*y>0 {yes,no}
@attribute violation{viol,noviol}

@data
p1,-3,8,no,viol
p1,-4,8,no,viol
p1,-5,9,no,viol
p1,6,-4,no,viol
p1,-5,3,no,viol
p1,-4,2,no,viol
p3,3,5,yes,noviol
p4,2,4,yes,noviol
p5,-2,5,no,viol
p9,1,5,yes,noviol
```

Ce fichier contient trois parties :

@relation : décrit le nom du service

@attribute : les attributs possibles pour une fonctionnalité du service

@data : les informations collectées

## 2. Phase de prédiction :

Chaque service contient un contrat SLA, qui comporte des accords entre le client et le fournisseur.

On présente les objectifs du service « Multiplication » dans la figure 16

### 5. Service Agreement

The following are the responsibility of the Service Provider in the ongoing support of this Agreement.

#### a. Service Scope

The following Services are covered by this Agreement;

1. X( 1 to 100)
2. Y( 1 to 100)
3.  $X*Y>0$  yes

#### b. Customer Requirements

Customer responsibilities and/or requirements in support of this Agreement include:

- Payment for all support costs at the agreed interval.

#### c. Service Provider Requirements

**Service Provider** responsibilities and/or requirements in support of this Agreement include:

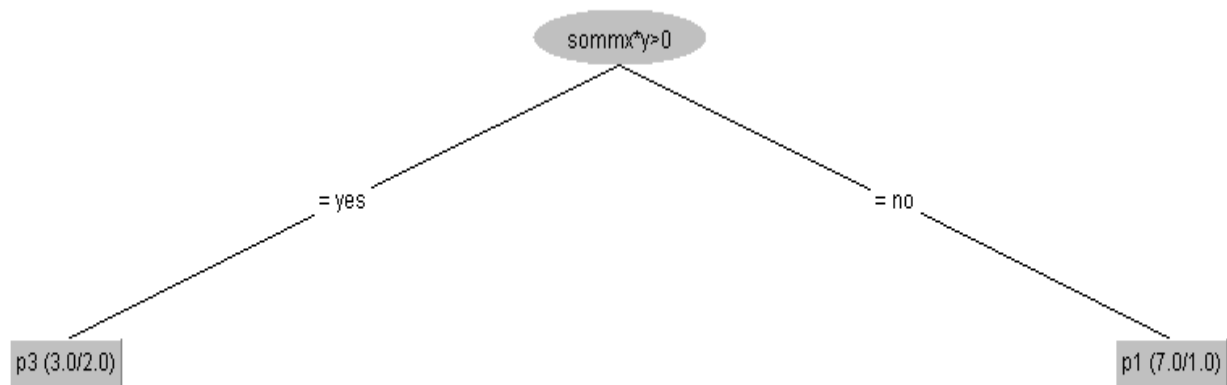
---

**Figure 16:SLA du Service Multiplication**

Si on trouve qu'il y a des valeur qui ne respectent pas l'intervalle de X ou de Y ou bien le «  $X*Y>0$  est No » alors il s'agit d'une violation

On présente ci-dessous l'arbre de prédiction, tel que chaque violation nécessite une substitution de service.

On utilise le Multiplication.arff comme une entrée dans l'algorithme J48 pour connaître les prédictions possibles (figure 17) :



**Figure 17:Prédiction du Service Multiplication**

La condition dans le SLA est :  $X(1 \text{ à } 100) \text{ AND } Y(1 \text{ à } 100) \text{ AND } \ll X*Y>0 \text{ est Yes} \gg$   
Depuis l'arbre de prédiction on remarque qu'il y a eu une violation par le Fournisseur1 (P1).

➤ **Résultat :**

Le service Multiplication doit être remplacé si le client a choisi le service du P1.

## 7.2 Substitution lexicale (analyse sémantique):

Une substitution dépend de la sémantique des opérations, par exemple soit deux opérations qui font l'addition, le paramètre de la première est « add » tandis que le paramètre de la deuxième est « sum », malgré les deux identifiants des paramètres sont différents mais ils donnent le même sens, donc ils sont considérés similaires et leurs services sont substituables.

- **Analyse syntaxique :**

Un client généralement s'intéresse aux fonctionnalités d'un service, qui sont définies par des opérations. Pour faciliter la recherche des services, on regroupe les opérations qui font la même chose (produisent la même sortie) .

Une fonctionnalité d'une opération est définie par sa sortie. Tous d'abord il faut extraire du fichier WSDL (nom du service ; nom d'opération, nom de la sortie). Ensuite on concatène ces données pour former l'identifiant d'une sortie d'un service particulier.

La comparaison des deux sorties des services différents, implique la comparaison de ses paramètres de similarité, qui représente la moyenne entre les identifiants et les types, comme il est indiqué dans la formule suivante :

$$\text{Parameters Similarité (Parameter1; Parameter2)} = \frac{[\text{identifiant Similarité (Id1; Id2)} + \text{Types Similarité (T1; T2)}] / 2}$$

✓ **Identifiant similarité :** On calcule la similarité entre deux identifiants via la distance de Jaro-Winkler ( dans notre cas on compare le résultat de la distance de Jaro si c'était supérieur à 0.7 « seuil » avant d'entamer le calcul de la distance de Jaro-Winkler, pour se concentrer seulement aux résultats qui donne beaucoup plus de ressemblance)

✓ **Types similarité :** la similarité entre les différents types est résumée dans le tableau 21, on doit toujours prendre la valeur minimale entre deux valeurs de types d'identifiants.

	Integer	Real	String	Date	Boolean
Integer	1	0.5	0.3	0.1	0.1
Real	1	1	0.1	0	0.1
String	0.7	0.7	1	0.8	0.3
Date	0.1	0	0.1	1	0
Boolean	0.1	0	0.1	0	1

21: La similarité entre les différents type [95]

Par exemple si le premier paramètre est de type string et le deuxième est de type integer nous aurons deux valeurs : soit 0.7 soit 0.3 donc on choisit la valeur la plus petite qui est 0.3.

Ces paramètres sont utilisés pour la solution de la substitution globale ainsi que pour la substitution précise.

### 7.2.1 Exemple de distance « Jaro-Winkler »

Soit deux chaînes de caractères s1 : AMRANI et s2 : ARMANI. On met un 1 si les caractères sont similaires et de même position ou d'une position approximative sinon un 0.

	A	M	R	A	N	I
A	1	0	0	0	0	0
R	0	0	1	0	0	0
M	0	1	0	0	0	0
A	0	0	0	1	0	0
N	0	0	0	0	1	0
I	0	0	0	0	0	1

Soit la distance de jaro comme suit :

$$d_j = \frac{1}{3} \left( \frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m - t}{m} \right)$$

- ✓ M= le nombre de 1 dans la matrice = 6
- ✓ |s1|=|s2|=6
- ✓ T : nombre de transposition entre les deux chaines qui ne doit pas dépasser (max(|s1|,|s2|)/2 -1) sinon t=0
- ✓ Ici nous avons (T et H) deux positions différentes , on compare 2 avec le résultat de (max(6,6)/2 -1)= 6/2 -1= 2 donc 2 ne dépasse pas le résultat alors t= 2/2=1
- ✓ On calcule la distance de jaro : 1/3(6/6+6/6+(6-1/6))=0.944
- ✓ Comparer le résultat par rapport à 0.7, ici 0.944 est supérieur au seuil donc on continue à calculer la distance de Jaro-Winkler :

Soit la distance de Jaro Winkler comme suit :

$$d_w = d_j + (lp(1 - d_j))$$

d<sub>w</sub> : distance de jaro winkler

d<sub>j</sub> : distance de jaro

l : longueur de préfixe en commun ( nombre de 1 dans la diagonal) avec (l<=4)

p : coeficient de préfixe égale 0.1

ici l=1 comme il est montré dans la matrice :

	A	M	R	A	N	I
A	1	0	0	0	0	0
R	0	0	1	0	0	0
M	0	1	0	0	0	0
A	0	0	0	1	0	0
N	0	0	0	0	1	0
I	0	0	0	0	0	1

Donc la distance de jaro = 0.944+(1 x 0.1 ( 1-0.944))= 0.9496

• **Application de la similarité syntaxique :**

Nous allons illustrer la similarité syntaxique par l'exemple des deux services sum et add , on représente dans une matrice les paramètres de chaque service (Tableau 22) .

	sum.P1	sum.P2	sum.P3
add.P1	X1	X3	X5
add.P2	X2	X4	X6

Tableau 22: Identifiant entre ABS et SSN

Soit  $x_i$  la valeur de similarité entre les identifiants de  $sum.P_i$  et  $add.P_i$  calculer par la distance de Jaro-Winkler.

Soit 1 la similarité entre les types des deux services (Les deux sont de types String).

On applique la formule précédente pour obtenir la similarité totale :

Paramètre<sub>sim</sub>(p1,p2)=  $[x_i+1]/2$

On obtient une nouvelle matrice (tableau 23), tel que param<sub>i</sub> un paramètre de similarité entre  $sum.P_i$  et  $add.P_i$

	sum.P1	sum.P2	sum.P3
add.P1	Param1	Param3	Param5
add.P2	Param2	Param4	Param6

Tableau 23: Paramètre de similarité entre ABS et SSN

### 7.3 Substitution globale (à base communauté) :

Il s'agit de regrouper les opérations similaires dans une même communauté en se basant sur les paramètres de similarité trouvés « paramètres de similarités », ces derniers sont présentés comme entrée dans l'algorithme K-means, qui produit l'ensemble des clusters contenant les opérations similaires.

#### 7.3.1 Exemple de l'algorithme K-means

Soit l'ensemble de donnée : { 2 4 -10 12 3 20 30 11 }

On fixe un nombre  $k=3$ ,

On choisi aléatoirement trois centres parmi l'ensemble soit centreop= { 2 4 30 }

On calcule la distance (toujours positive) entre la donnée et le centre ( distance= centreop - élément de l'ensemble).

La donnée la plus petite est affectée au communauté (cluster), ensuite on calcule les nouveaux centres ( calculer la moyennes des élément du cluster) et on compare si les centre ne changent pas on s'arrête sinon on refait itérativement les étapes mentionnées.

##### Itération 1

Distance:

cluster 0 { 0 2 12 10 1 18 28 9 }

cluster 1 { 2 0 14 8 1 16 26 7 }

cluster 2 { 28 26 40 18 27 10 0 19 }

0 1 0 1 0 2 2 1

-----  
cluster 0: { 2 -10 3 }     $2-10+3/3=$  (entier)  $-5/3=-1$

cluster 1: { 4 12 11 }     $27/3= 9$

cluster 2: { 20 30 }     $50/2 25$

-----  
donnée = { 2 4 -10 12 3 20 30 11 }  $k=3$ , nouveau centreop = { -1 9 25 }

##### Itération 2

Distance:

cluster 0: { 3 5 9 13 4 21 31 12 }  
 cluster 1: { 7 5 19 3 6 11 21 2 }  
 cluster 2: { 23 21 35 13 22 5 5 14 }  
           0 0 0 1 0 2 2 1

-----  
 cluster 0: { 2 4 -10 3 }     $9-10/4 = -1/4 = 0$   
 cluster 1: { 12 11 }       $23/2 = 11.5 = 11$   
 cluster 2 : { 20 30 }       $50/2 = 25$

-----  
 donnée = { 2 4 -10 12 3 20 30 11 } k=3, nouveau centreop = {0 11 25}

### Itération 3

Distance:

cluster 0 { 2 4 10 12 3 20 30 11 }  
 cluster 1 { 9 7 21 1 8 9 19 0 }  
 cluster 2 { 23 21 35 13 22 5 5 14 }  
           0 0 0 1 0 2 2 1

-----  
 cluster 0: { 2 4 -10 3 }    0  
 cluster 1: { 12 11 }      11  
 cluster 2: { 20 30 }      25

-----  
 nouveau centreop= {0 11 25}

Le centre est stable donc on doit s'arrêter

Cluster final !

-----  
 cluster 0: { 2 4 -10 3 }  
 cluster 1: { 12 11 }  
 cluster 2: { 20 30 }  
 -----

## 7.4 Substitution précise (à base réseau )

Cette substitution dépend sur la formation du réseau, et l'interprétation de sa matrice pour trouver les services remplaçants. Chaque paramètre de similarité d'un certain service doit avoir sa correspondance avec un autre service.

Pour mieux comprendre, on prend la combinaison d'un service add avec le service sum mentionnée précédemment montrée dans le tableau 23.

Tous d'abord on cherche à trouver une correspondance de sum.P1 avec un seul paramètre de add, on remarque que la première colonne contient deux paramètres de add.

On doit comparer la valeur param1 par rapport à 0.7, si on trouve qu'elle est supérieure ou égale à 0.7 et toutes les autres ligne de sa colonne ne le sont pas, cela signifie que sum.P1 est affecté à add.P1 sinon il ne sont pas.

Les résultats trouvés ont deux interprétations :

- ✓ **Cas d'égalité** : si les deux services ont le même nombre de paramètres, et chaque paramètre est affecté à un autre paramètre par exemple : Le service add a deux paramètres et le service soustra



a deux paramètres, donc il suffit de vérifier si chacun des paramètres peuvent être affecté à l'autre (tableau 24)

	soustra.P1	soustra.P2
add.P1	Param7	Param9
add.P2	Param8	Param10

Tableau 24: Combinaison entre ABS et IAN

➤ Discussion

**soustra.P1** : Si  $\text{param7} \geq 0.7$  et  $\text{param8} < 0.7$  alors soustra.P1 est affecté à add.P1 ou  
Si  $\text{param7} < 0.7$  et  $\text{param8} \geq 0.7$  alors soustra.P1 est affecté à add.P2.

**soustra.P2** : Si  $\text{param9} \geq 0.7$  et  $\text{param10} < 0.7$  alors soustra.P2 est affecté à add.P1 ou  
Si  $\text{param9} < 0.7$  et  $\text{param10} \geq 0.7$  alors soustra.P2 est affecté à add.P2 .

- ✓ **Cas d'inclusion** : si les deux services ont un nombre de paramètres différents, et il existe au moins une affectation du paramètre du premier service vers le deuxième, par exemple : Le service add a deux paramètres, et sum a trois paramètres , donc il suffit de trouver au moins une seule affectation pour le paramètre du premier service (tableau 25) .

	sum.P1	sum.P2	sum.P3
add.P1	Param11	Param13	Param15
add.P2	Param12	Param14	Param16

Tableau 25: Combinaison entre ABS et SSN

➤ Discussion

Il suffit de trouver au moins une affectation pour sum.P1 ou sum.P2 ou sum.P3.  
On suppose qu'il y a eu une affectation en sum.P1 c'est-à-dire :

**sum.P1** :  $\text{Param11} \geq 0.7$  et  $\text{param12} < 0.7$  Alors sum.P1 est affecté à add.P1  
ou  $\text{param11} < 0.7$  et  $\text{param12} \geq 0.7$  Alors sum.P1 est affecté à add.P2

Et même si il n'y en a aucune correspondance pour sum.P2 ou sum.P3, il existe une relation d'inclusion.

#### 7.4.1 Représentation graphique de la matrice de similarité

Les relations entre les valeurs similaires sont représentées par un graphe tel que :

Un nœud représente une opération, et un arc signifie une direction possible « une affectation » ;

Une opération réfère un seul service, on considère 4 opérations des 4 services « op1, op2, op3, op4 »

Soit un réseau de similarité entre quatre opérations des différents services (figure 18)

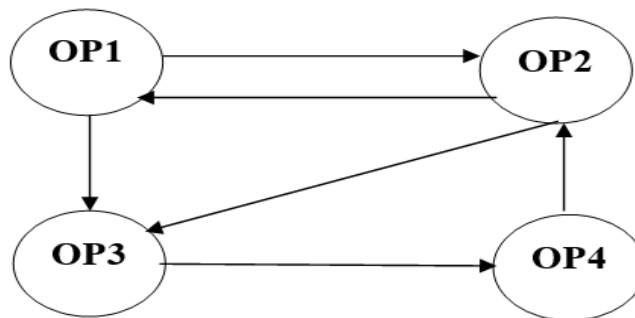


Figure 18: Réseau de similarité

#### 7.4.2 Représentation matricielle du graphe :

Soit un graphe ci-dessous dans la figure 18 qui représente les relations précédentes, ensuite nous allons l'interpréter sous forme d'une matrice tel que un nœud représente une opération et un arc signifie une relation :

Le tableau 26 montre la matrice du graphe précédent tel que 1 signifie une direction possible 0 sinon :

Opération	OP1	OP2	OP3	OP4
OP1	/	1	1	0
OP2	1	/	1	0
OP3	0	0	/	1
OP4	0	1	0	/

Tableau 26: Représentation matricielle du graphe

On peut conclure les opérations similaires, ainsi que les services substituables comme suit :

OP1 est similaire à OP2 donc le service2 substitue le service1

OP2 est similaire à OP1 et OP4 donc le service1 et le service4 substituent le service2

OP3 est similaire à OP1 et OP2 donc le service1 et le service2 substituent le service3

OP4 est similaire à OP3 donc le service3 substitue le service4

## 8 Conclusion

Afin de garantir la haute disponibilité des opérations des services cloud, une substitution des services est utilisée.

Dans ce chapitre, nous avons présenté la conception de notre solution de substitution des services Cloud de type SaaS.

La solution proposée contient la phase de découverte des SaaS, la sélection des meilleurs services ainsi que la substitution des services Cloud échoués pour répondre aux besoins des clients.

Le processus de sélection se base sur la méthode MCDM tandis que la substitution utilise une mesure syntaxique faite par la méthode de Jaro-Winkler , et une mesure sémantique via la méthode Wu-palmer ensuite des représentations graphique et matricielles sont réalisées afin de garantir la disponibilité des services à tout moment .

L'implémentation de notre solution est décrite dans le chapitre suivant.

## 1 Introduction

Après avoir vu les différentes solutions de la substitution nous allons enfin décrire notre application qui se base sur les concepts mentionnés dans les chapitres précédents.

Dans ce chapitre nous allons voir les outils utilisés dans la réalisation du prototype illustrée par des exemples.

## 2 Outils de développement

Afin de réaliser de notre application, nous avons utilisé plusieurs outils de développement :

- **Eclipse**

Eclipse est un IDE (Integrated Development Environment) fondé par IBM en 2001, principalement écrit en Java. Il produit des outils pour la réalisation de logiciels, englobant les activités de programmation notamment d'environnement de développement intégré .Eclipse fonctionne sous Microsoft Windows, macOS, Linux et Solaris [102].

- **Java Development Kit (JDK)**

JDK est développé par Oracle, désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java sous Microsoft Windows, macOS, Linux. Il existe plusieurs éditions de JDK, selon la plate-forme Java [103]

- ✓ JSE pour la Java 2 Standard Edition également désignée J2SE ;
- ✓ JEE, sigle de Java Enterprise Edition également désignée J2EE ;
- ✓ JME 'Micro Edition', destinée au marché mobiles ; ect

- **MySQL**

MySQL est un système de gestion de bases de données relationnelles (SGBDR) Son nom vient du prénom de la fille du cocréateur Michael Widenius ; SQL fait référence au Structured Query Language ( le langage de requête utilisé ). Développé par MtSQLAB, Sun Microsystems et Oracle Corporation en 1995. Il fonctionne sous Linux, Microsoft Windows, macOS, FreeBSD et Solaris [104]

- **Protégé**

Protégé est un système auteur pour la création d'ontologies (une base de connaissance qui décrit un domaine) selon [105] .

Il a été créé à l'université Stanford ,il est très populaire dans le domaine du Web sémantique et au niveau de la recherche en informatique. Protégé est développé en Java, il peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF(Ressource Description Framework), RDFS (RDF Schema), OWL.

- **Weka**

«Waikato environment for knowledge analysis », est une suite de logiciels d'apprentissage automatique écrite en Java, il contient une collection d'outils de visualisation et d'algorithmes pour l'analyse des données et la modélisation prédictive. Weka fonctionne sous Microsoft Windows et Multi-plateformes[106].

- **WordNet :**

WordNet est une base de données lexicale qui classifie et met en relation de diverses manières le contenu sémantique et lexical de la langue anglaise, la version anglaise est la plus complète à ce jour parmi toutes les autres versions, elle est disponible gratuitement accessible à partir des interfaces pour plusieurs langages de programmation (Java, Perl, PHP, Prolog, Python...) [107] .

- **Apache-Tomcat :**

Un serveur gratuit d'applications Java du projet Jakarta de la fondation Apache, il permet d'exécuter des applications Web développées avec les technologies Java (Servlets, JSP...).Nous avons utilisé son 7ème version [109].

### 3 Simulation

Nous nous retrouvons dans la partie de simulation, où nous allons montrer des figures représentatives sur notre application web, qui comporte principalement deux protagonistes « fournisseur et client », donc il y a des interfaces dédiées seulement aux fournisseurs, d'autres pour le client seulement ainsi qu'une interface partagée « l'accueil ».

✓ **Accueil « Home » :**

L'interface «Home» considérée comme la première page après l'exécution de l'application web, elle comporte l'option de la création d'un nouveau compte pour le fournisseur et pour le client. (**figure 19**)



**Figure 19: Home**

✓ **Interface pour fournisseur :**


Le fournisseur a le droit de s'inscrire, si il n'avait de compte (**figure20**) sinon il peut se connecter directement (**figure21**).

Provider Client

WELCOME!

Home

---

Provider 

Sign Up

First Name:

Last Name:


User Name:

Password:

[Already have an account?](#)

---

**Figure 20: Inscription de fournisseur**



**WELCOME!**

Provider Sign In

Provider Name:

Password:

Submit

**Figure 21: Connexion de fournisseur**

Le fournisseur peut créer un ou plusieurs services depuis son compte, en remplissant tous les champs du formulaire (**figure 22, 23**) . Cette interface donne la possibilité de visiter les services existants et l'option de se déconnecter.





Create your service 



Service domain

Service name

Evolutivity

Relevance

Security

Cost

**Figure 22: Interface création des services**


Security

Cost

Response time

Fiability

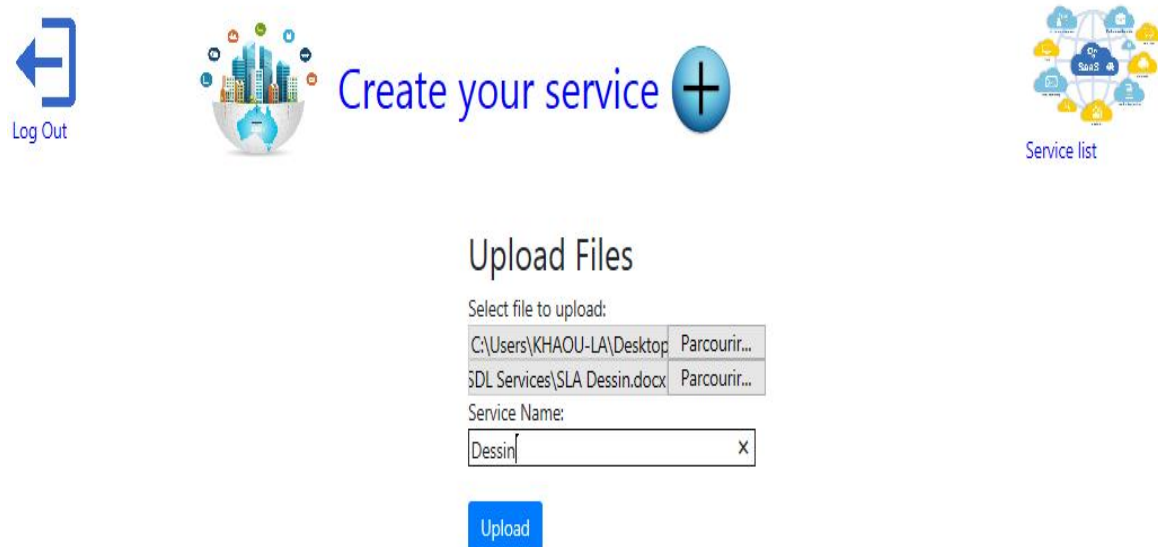
Availability

 Upload Files

**Figure 23: Suite de la création des services**

Le fournisseur peut importer le fichier WSDL du service créée, ainsi que son SLA qui contient les critères fonctionnelles du service en cliquant sur « Upload File », montré dans la figure précédente, ensuite il choisit les fichiers appropriés depuis son ordinateur, et décrit le service par son nom (**figure 24**).

Une fois cliqué sur « Upload » le fichier s'est enregistré dans la base de données et un message est affiché (**figure 25**)



The screenshot shows a web interface for creating a service. At the top, there are three navigation icons: 'Log Out' (a blue arrow pointing left), 'Create your service' (a globe with a plus sign), and 'Service list' (a circular network diagram). Below these is the 'Upload Files' section, which includes a 'Select file to upload:' label, two file selection buttons with paths like 'C:\Users\KHAOU-LA\Desktop' and 'SDL Services\SLA Dessin.docx', a 'Service Name:' label, and a text input field containing 'Dessin'. A blue 'Upload' button is positioned below the form.

**Figure 24: Importation du fichier WSDL**

Upload has been done successfully!  
[Continue Upload](#)

**Figure 25: Téléchargement réussi du fichier WSDL**

Les services créés peuvent être consultés, modifiés ou supprimés par les fournisseurs (chacun depuis son compte). **(figure 26)**.

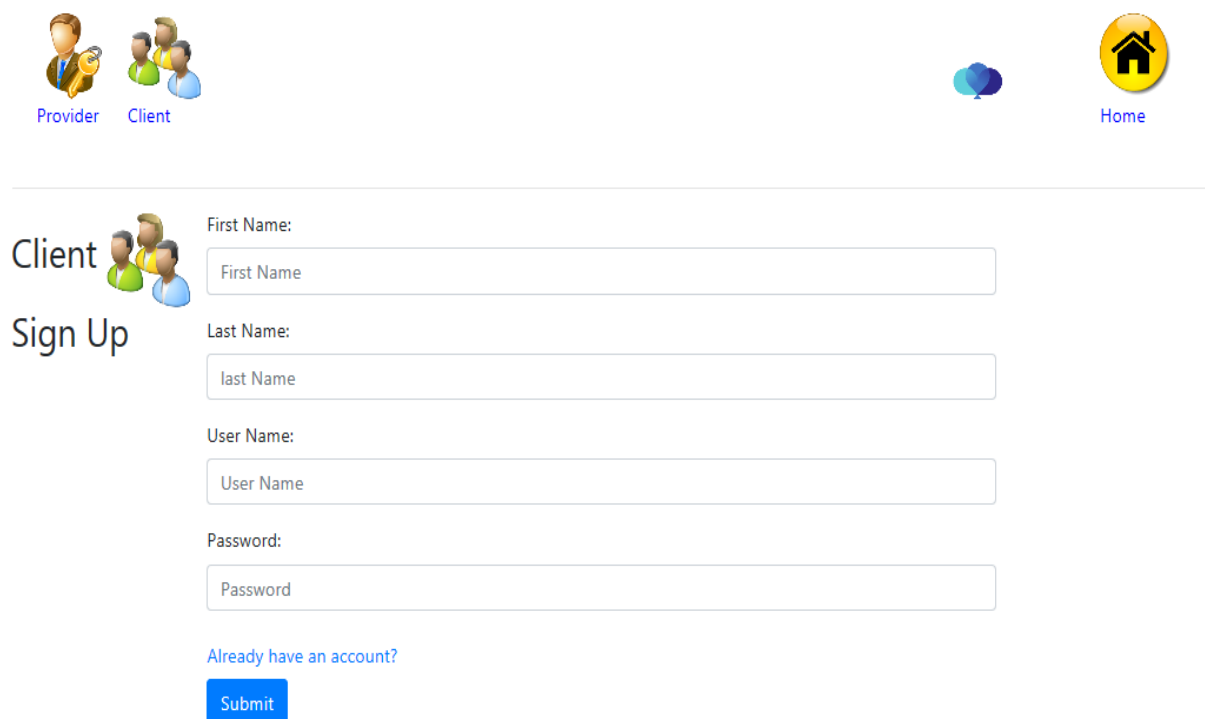


ID	Domain	Name	Evolutivity	Relevance	Security	Cost	Response time	Fiability	Availability	Actions
31	Art	Dessin	0.5	0.5	0.4	0.4	0.2	0.2	0.2	Edit Delete
32	Math	Mul	0.3	0.3	0.2	0.6	0.7	0.6	0.6	Edit Delete
25	Commerce	ABS	0.4	0.5	0.5	0.6	0.2	0.2	0.3	Edit Delete
30	Math	Cal	0.4	0.5	0.5	0.6	0.6	0.3	0.2	Edit Delete
29	Art	FM	0.4	0.3	0.3	0.9	0.5	0.6	0.6	Edit Delete
28	Art	SSN	0.3	0.5	0.5	0.5	0.6	0.7	0.7	Edit Delete
27	Commerce	IAN	0.4	0.5	0.6	0.4	0.4	0.4	0.3	Edit Delete
26	Commerce	QTE	0.1	0.1	0.3	0.4	0.5	0.4	0.5	Edit Delete

**Figure 26: Liste des services**

✓ **Interface pour client**

Un consommateur peut créer un nouveau compte (**figure 27**) sinon il se connecte directement à son compte (**figure 28**) .



Provider Client

Home

Client Sign Up

First Name:

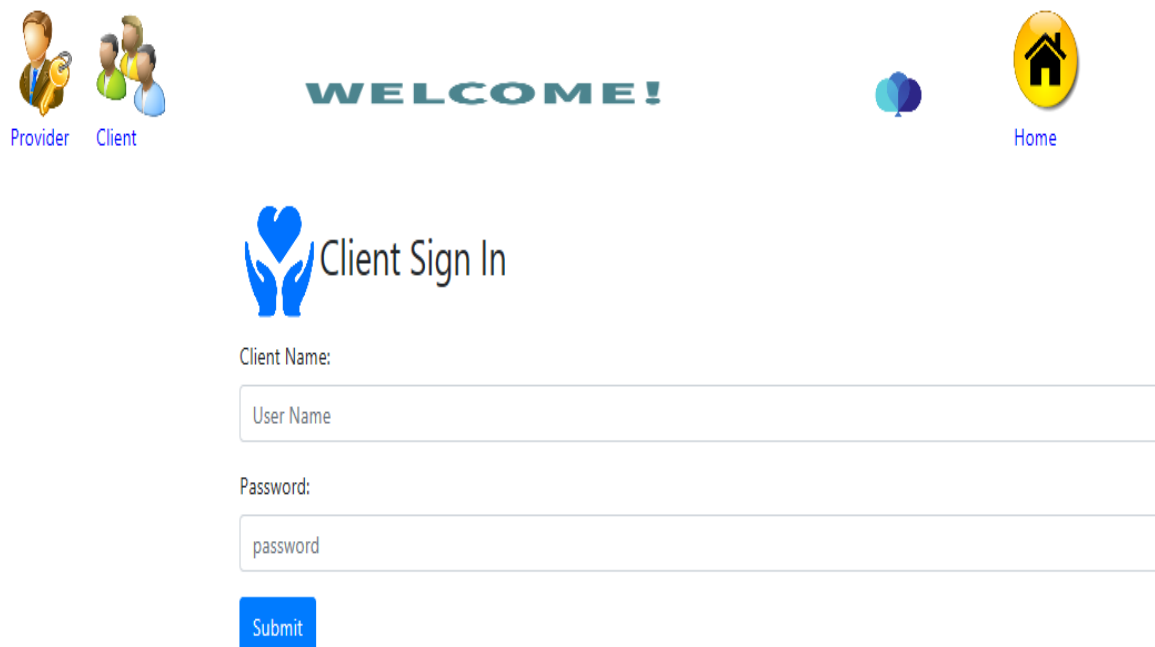
Last Name:

User Name:

Password:

[Already have an account?](#)

**Figure 27: Inscription client**



Provider Client

Home

WELCOME!

Client Sign In

Client Name:

Password:

**Figure 28: Connexion de client**

Un client doit choisir et confirmer le nom du domaine d'un service voulu, avant de confirmer la recherche sinon il aura une erreur ! , il peut se déconnecter à tout moment (figure 29, 30)

## Welcome to service page

Choose a service :

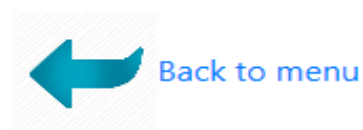
Confirm "The domain":

SEARCH

CLEAR



Figure 29: Chercher un service


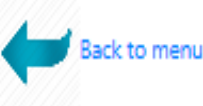


## Query Results

Please select a service domain and confirm it !

Figure 30: Erreur de recherche

En cas d'une recherche réussite, les services du domaine recherché sont affichés (figure 31).

### Query Results

Id	Domain	Name	Evolutivity	Relevance	Security	Cost	Response time	Fiability	Availability
<input type="checkbox"/>	Art	Dessin	0.5	0.5	0.4	0.4	0.2	0.2	0.2
<input type="checkbox"/>	Art	SSN	0.3	0.5	0.5	0.5	0.6	0.7	0.7
<input type="checkbox"/>	Art	FM	0.4	0.3	0.3	0.9	0.5	0.6	0.6

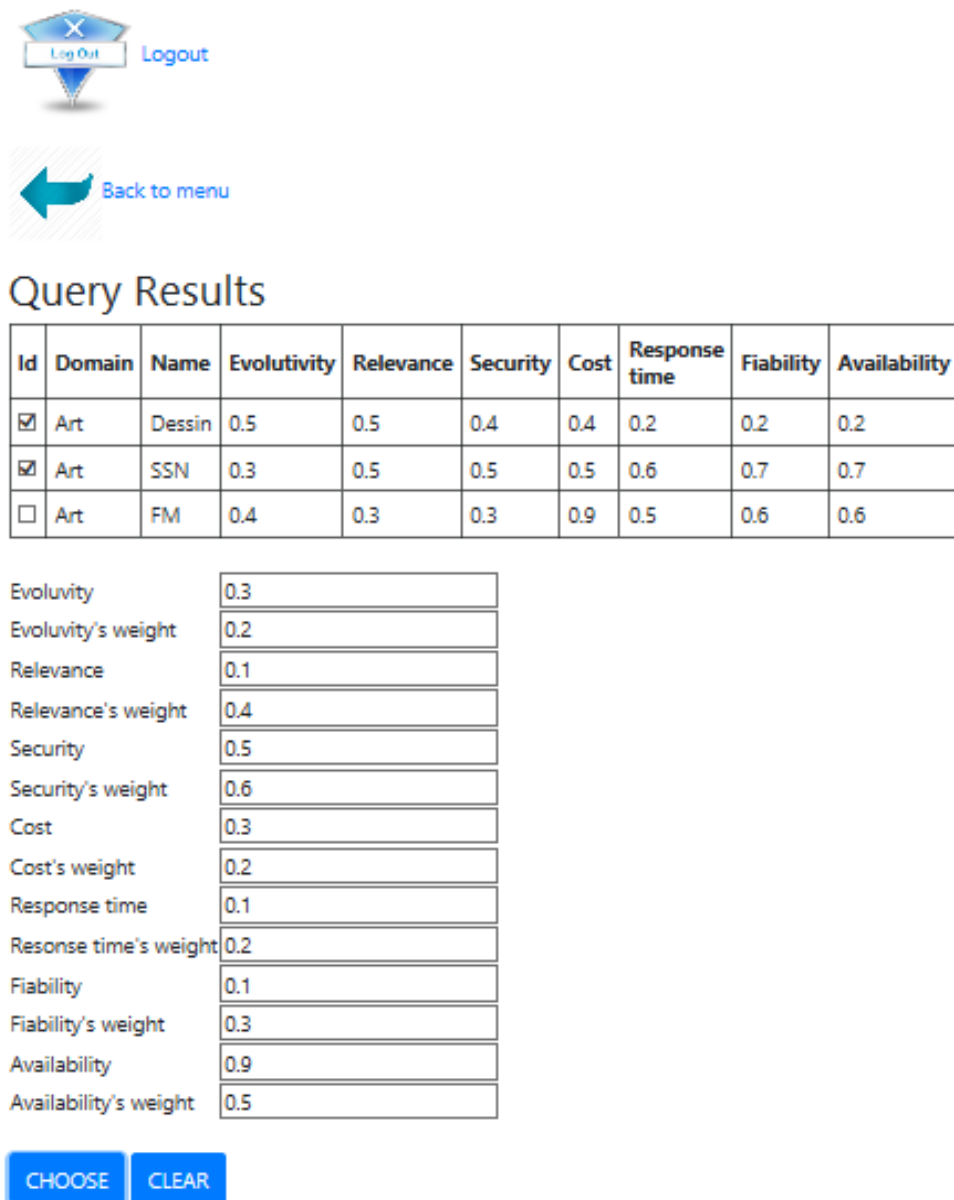
  

Evolutivity	<input type="text"/>
Evolutivity's weight	<input type="text"/>
Relevance	<input type="text"/>
Relevance's weight	<input type="text"/>
Security	<input type="text"/>
Security's weight	<input type="text"/>
Cost	<input type="text"/>
Cost's weight	<input type="text"/>
Response time	<input type="text"/>
Response time's weight	<input type="text"/>
Fiability	<input type="text"/>
Fiability's weight	<input type="text"/>
Availability	<input type="text"/>
Availability's weight	<input type="text"/>

Figure 31: Résultat de la recherche

Parmi les services affichés, le client sélectionne ceux qui en trouve plus proche à satisfaire ses besoins, ensuite il remplit tous les champs par les valeurs de la QoS souhaitées (**figure 32**)



Logout

Back to menu

### Query Results

Id	Domain	Name	Evolutivity	Relevance	Security	Cost	Response time	Fiability	Availability
<input checked="" type="checkbox"/>	Art	Dessin	0.5	0.5	0.4	0.4	0.2	0.2	0.2
<input checked="" type="checkbox"/>	Art	SSN	0.3	0.5	0.5	0.5	0.6	0.7	0.7
<input type="checkbox"/>	Art	FM	0.4	0.3	0.3	0.9	0.5	0.6	0.6

Evolutivity:

Evolutivity's weight:

Relevance:

Relevance's weight:

Security:

Security's weight:

Cost:

Cost's weight:

Response time:

Response time's weight:

Fiability:

Fiability's weight:

Availability:

Availability's weight:

**Figure 32: La recherche du meilleur service**

Le but de la recherche est de trouver le meilleur service qui répond le plus aux exigences du client, le processus utilise la méthode MCDM pour en trouver, après le calcul le service approprié est retourné (**figure 33**)



Option

## Final Result

- Customer evolutivity: 0.3
- Customer evolutivity's weight: 0.2
- Customer relevance: 0.1
- Customer relevance's weight: 0.4
- Customer security: 0.5
- Customer security's weight: 0.6
- Customer cost: 0.3
- Customer cost's weight: 0.2
- Customer response time: 0.1
- Customer response time's weight: 0.2
- Customer fiability: 0.1
- Customer fiabilit's weight: 0.3
- Customer availability: 0.9
- Customer availability's weight: 0.5

Type	Evolutivity	Relevance	Security	Cost	Response time	Fiability	Availability	Score	
Art	Dessin	0.5	0.5	0.4	0.4	0.2	0.2	0.2	0.139999999999999987
Art	SSN	0.3	0.5	0.5	0.5	0.6	0.7	0.7	0.38
The BEST Score: 0,38									


Thank you for using our product 

Figure 33: Résultat de la méthode MCDM



✓ **Le processus de la substitution:**

Le processus de substitution est utilisé si il existe un dysfonctionnement au niveau des opérations des services ou bien son comportement s'arrête complètement (cas d'échec). Nous allons expliquer le processus de substitution utilisé dans notre application en utilisant l'exemple qui suit.

Soit les huit services : ABS, Cal, Dessin, FM, IAN, Mul, QTE, SSN

✓ **ABS :**

**Opération :** produce

**Paramètre1 :** submit

**Paramètre2 :** liste

**Type :** String

✓ **Cal :**

**Opération :** add

**Paramètre1 :** response

**Type :** int

✓ **Dessin :**

**Opération :** Create

**Paramètre1 :** submit

**Type :** String

✓ **FM:**

**Opération :** get

**Paramètre1 :** listetypes

**Type :** String

✓ **IAN:**

**Opération :** remove

**Paramètre1 :** add

**Paramètre2 :** soustra

**Type :** int

✓ **Mul :**

**Opération :** add

**Paramètre1 :** response

**Type :** int

✓ **QTE :**

**Opération :** produce

**Paramètre1 :** submit

**Type :** String

✓ **SSN :**

**Opération :** delete

**Paramètre1 :** listetypes

**Paramètre2 :** buy

**Paramètre3 :** filter

**Type :** String

Nous allons faire toutes les combinaisons possibles entre les paramètres des opérations des services sans répétition. que : **p.service** signifie le paramètre d'un service.

✓ **La combinaison de ABS :**

La combinaison de service ABS avec les autres services est détaillée dans la figure 34

P.service	Cal.P1					P.service	Dessin.P1					P.service	FM.P1				
ABS.P1						ABS.P1						ABS.P1					
ABS.P2						ABS.P2						ABS.P2					
P.service	IAN.P1	IAN.P2				P.service	Mul.P1					P.service	QTEI.P1				
ABS.P1						ABS.P1						ABS.P1					
ABS.P2						ABS.P2						ABS.P2					
P.service	SSN.P1	SSN.P2	SSN.P3														
ABS.P1																	
ABS.P2																	

**Figure 34: Combinaison de service ABS**

✓ **La combinaison de Cal :**

La combinaison de service Cal avec les autres service est détaillée dans la figure 35

P.service	Dessin.P1					P.service	FM.P1					P.service	IAN.P1	IAN.P2			
Cal.P1						Cal.P1						Cal.P1					
P.service	Mul.P1					P.service	QTE.P1					P.service	SSN.P1	SSN.P2	SSN.P3		
Cal.P1						Cal.P1						Cal.P1					

**Figure 35: Combinaison de service Cal**

✓ **La combinaison de Dessin :**

La combinaison de service Dessin avec les autres services est détaillée dans la figure 36

P.service	FM.P1	P.service	IAN.P1	IAN.P2	P.service	Mul.P1
Dessin.P1		Dessin.P1			Dessin.P1	

P.service	QTE.P1	P.service	SSN.P1	SSN.P2	SSN.P3
Dessin.P1		Dessin.P1			

Figure 36: Combinaison de Dessin

✓ **La combinaison de FM**

La combinaison de service FM avec les autres services est détaillée dans la figure 37

P.service	IAN.P1	IAN.P2	P.service	Mul.P1	P.service	QTE.P1
FM.P1			FM.P1		FM.P1	

P.service	SSN.P1	SSN.P2	SSN.P3
FM.P1			

Figure 37: Combinaison de FM

✓ **La combinaison de IAN :**

La combinaison de service IAN avec les autres services est détaillée dans la figure 38

P.service	Mul.P1	P.service	QTE.P1	P.service	SSN.P1	SSN.P2	SSN.P3
IAN.P1		IAN.P1		IAN.P1			
IAN.P2		IAN.P2		IAN.P2			

Figure 38: Combinaison de IAN

✓ **La combinaison de Mul**

La combinaison de service Mul avec les autres service est détaillée dans la figure 39

P.service	QTE.P1	P.service	SSN.P1	SSN.P2	SSN.P3
Mul.P1		Mul.P1			

**Figure 39: Combinaison de Mul**

✓ **La combinaison de QTE**

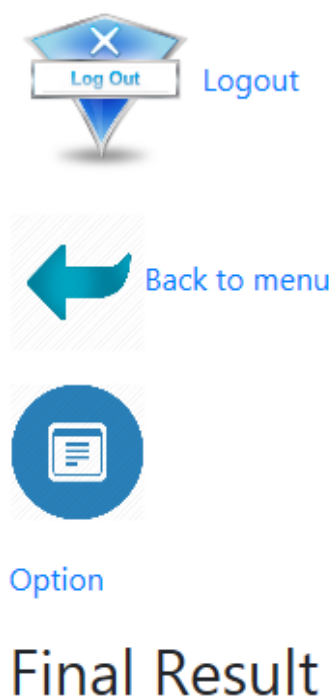
La combinaison de service QTE avec les autres service est détaillée dans la figure 40

P.service	SSN.P1	SSN.P2	SSN.P3
QTE.P1			

**Figure 40: Combinaison de QTE**

• **Substitution sémantique**

Cette substitution utilise la méthode de Wu-Palmer pour calculer la similarité sémantique entre les identifiants des opérations. Le client peut consulter les listes des services substituables en cliquant sur option après la phase de la sélection des services (**figure 41**)



**Figure 41: Option de substitution**

Ensuite il peut voir la liste des services remplaçant trouvés par Wu-Palmer en choisissant « Lexical substitution » (**figure 42,43**)



Lexical  
substitution



Global  
substitution



Precise  
substitution



Prediction



Back



**Figure 42: Menu de substitution**



Lexical  
substitution



Global  
substitution



Precise  
substitution



Prediction



Back

List of similar services:

Service	Replaced by
ABS	QTE, Dessin
Cal	Mul
Dessin	ABS, QTE
FM	/
IAN	SSN
Mul	Cal
QTE	ABS, Dessin
SSN	IAN

Figure 43: Résultat de la méthode Wu-Palmer

- **Substitution syntaxique :**

Cette analyse se base sur la mesure structurelle entre les différents caractères.

Notre but est de trouver la similarité entre les services, pour ce faire nous intéressons aux fonctionnalités qui sont décrites dans les opérations, plus précisément dans les sorties des opérations.

Du coup nous avons besoin du nom de service, nom de l'opération et nom de la sortie pour définir la fonctionnalité d'un service. On calcule la similarité entre les différents identifiants et entre leurs types aussi, ensuite on trouve leurs moyennes (paramètre de similarité).

Ce résultat est utilisé dans deux méthodes différentes :

- **Méthode 01(Substitution globale)**

On applique la méthode K-means sur les résultats des moyennes trouvées, pour extraire la liste des services substituables.

Le client peut voir la liste des services remplaçant trouvés par K-means en choisissant « Global substitution » (figure 44) dans le menu (figure 42)



Lexical substitution



Global substitution



Precise substitution



Prediction



Back

List of similar services:

Function(Outputs)	Clusters (Operation)	Services
submit liste response listetupes	produce add get delete	ABS QTE CaI Mul FM SSN
liste add soustra response submit listetypes buy filter	produce add remove create get delete	ABS CaI IAN Mul QTE Dessin FM SSN
liste submit buy filter listetypes response add soustra	produce add create get remove delete	ABS Dessin FM QTE SSN CaI IAN Mul

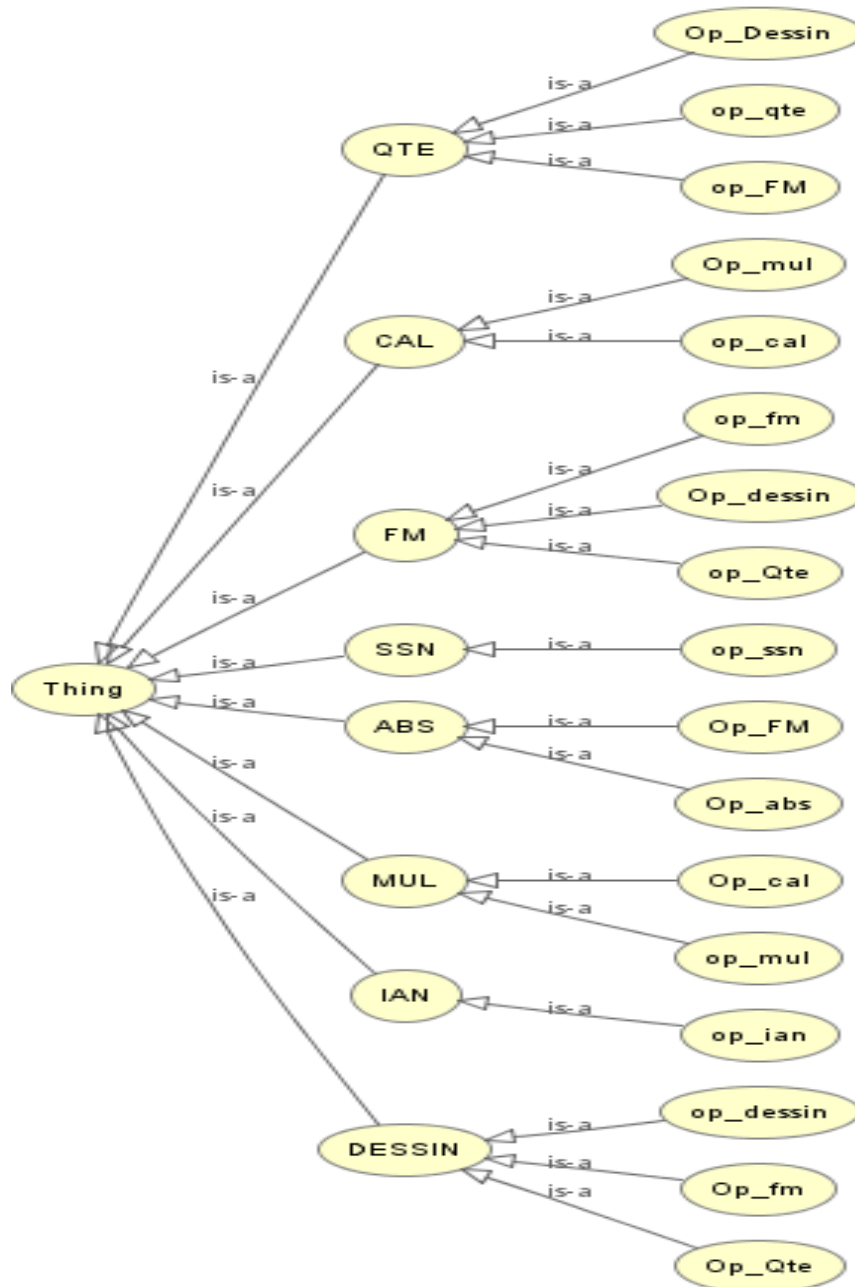
Figure 44: Résultat de la méthode K-means



**Méthode 02 (Precise substitution)**

On trouve les relations possibles (égalité, inclusion) entre les paramètres de similarité pour former un réseau similaire.

Le réseau de similarité des huit services est présenté dans la **figure 45**



**Figure 45: Réseau de similarité**

➤ **Explication**

Le réseau contient les différentes opérations des services existants, il les regroupe selon leurs appartenances, c'est-à-dire les opérations du même domaine ou qui fonctionnent de la même façon elles se rassemblent dans la même section.

Par exemple le service « Cal » comporte l'opération qu'il produise plus l'opération du service « Mul » qui actionne exactement comme la sienne, par contre le service IAN ne contient que son opération puisque il n'y avait aucune autre opération qui agit de la même manière.

Maintenant nous allons modéliser le graphe sous forme une matrice, pour extraire les services substituables tel que : 1 représente un nœud initial sinon 0 (**figure46**).

### Matrix Modeling the Graph:

Services	ABS	Cal	Dessin	FM	IAN	Mul	QTE	SNN
ABS	/	0	0	1	0	0	0	0
Cal	0	/	0	0	0	1	0	0
Dessin	0	0	/	1	0	0	1	0
FM	0	0	1	/	0	0	1	0
IAN	0	0	0	0	/	0	0	0
Mul	0	1	0	0	0	/	0	0
QTE	0	0	1	1	0	0	/	0
SNN	0	0	0	0	0	0	0	/

**Figure 46: La matrice représentative du réseau**

#### ➤ Interprétation de la matrice

La lecture de la matrice se fait à travers les colonnes tel que signifie une substitution possible, par exemple le service ABS de la première colonne ne peut pas être remplacé, puisque il n'y en a aucun 1 dans sa colonne, tandis que le service Cal peut être remplacé par le service Mul.

Le client peut consulter la liste des services remplaçants trouvées, par la deuxième méthode de substitution en cliquant sur « Precise substitution » dans le menu (**figure42**), ensuite la page appropriée est affichée (**figure47**)



Lexical  
substitution



Global  
substitution



Precise  
substitution



Prediction



Back

List of similar services:

Service	Replaced by
ABS	FM
Cal	Mul
Dessin	FM, QTE
FM	Dessin, QTE
IAN	/
Mul	Cal
QTE	Dessin, FM
SSN	/

Figure 47: Résultat du réseau similaire

Un service peut tomber en panne, comme il peut aussi y avoir des violations au niveau des critères fonctionnelles exigée dans le contrat SLA entre le client et le fournisseur.

Afin d'éviter le dysfonctionnement imprévisible des services on doit les prédire à l'avance l'échec des services.

✓ **Le processus de prédiction :**

La prédiction utilisée passe par deux étapes :

➤ **Etape 01(Phase d'apprentissage)**

L'administrateur doit être en surveillance pour collecter des informations utiles présentées comme un arbre de décision, ce dernier est analysé par la suite sous forme « fichier.arff » qui représente les règles d'associations pour le service qui convient.

Soit « QTE.arff » un fichier qui décrit les données utilisées pour le service QTE sous forme règles d'associations (**figure 48**).

```
@relation Cal
@attribute provider{p1,p2,p3,p4,p5,p6}
@attribute x NUMERIC
@attribute y NUMERIC
@attribute sommx+y>0 {yes,no}
@attribute violation{viol,noviol}

@data
p1,4,17,yes,viol
p1,11,14,yes,noviol
p1,19,4,yes,viol
p1,50,10,yes,viol
p1,17,15,yes,viol
p1,19,5,yes,viol
p1,10,5,yes,noviol
p2,3,16,yes,noviol
p2,11,7,yes,noviol
p2,12,9,yes,noviol
p2,14,10,yes,viol
p2,15,10,yes,viol
p2,6,10,yes,noviol
p3,3,4,yes,noviol
p3,14,17,yes,viol
p4,4,4,yes,noviol
p3,9,4,yes,noviol
p4,5,10,yes,noviol
p5,7,5,yes,noviol
p6,19,15,yes,viol
```

**Figure 48: Règles d'associations pour le service Cal**

➤ **Etape 02 (La prédiction)**

Les accords du service sont mentionnés dans le SLA comme suit dans la figure 49 :

▣ **5. Service Agreement**

The following are the responsibility of the Service Provider in the ongoing support of this Agreement.

**Service Scope**

The following Services are covered by this Agreement;

1. X(1 to 12)
2. Y(0 to 15)
3. Sumx+y>0 "yes"

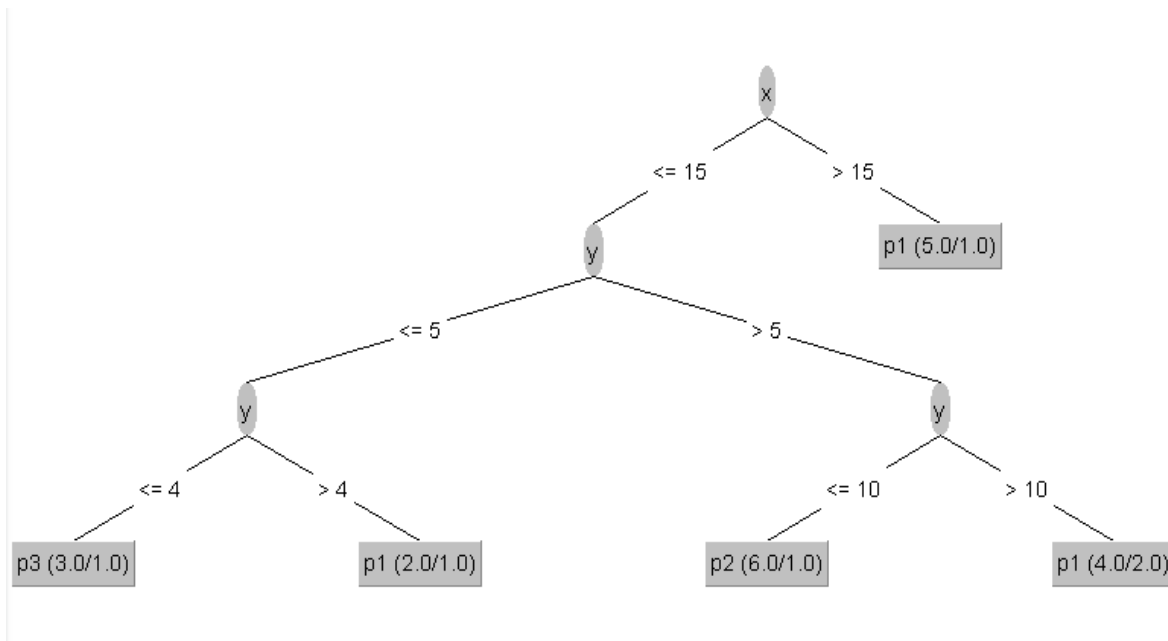
**b. Customer Requirements**

Customer responsibilities and/or requirements in support of this Agreement include:

- Payment for all support costs at the agreed interval.

**Figure 49: Accord SLA pour le service QTE**

On remarque que les conditions du fonctionnement « de la sortie » de ce service sont : X (1 à 12), Y(0 à 15) , SumX+Y>0 est toujours vrai  
La prédiction de résultat est montrée dans la figure 50 :



**Figure 50: Prédiction pour le service Cal**

On constate qu'il y'a eu une violation dans les cas suivants :

- ✓  $X > 15$  par le fournisseur p1
- ✓  $X \leq 15$  par les fournisseurs p1, p2, p3

Alors le client doit éviter ces fournisseurs, sinon il doit substituer son service puisque il sera en panne dans les cas trouvés.

Un client peut éviter les services échoués en cliquant sur « Prediction » dans le menu (**figure48**) dans le but de consulter les prévisions des services existants (**figure51**).



Lexical substitution



Global substitution



Precise substitution



Prediction



Back

## Prediction of violation

Service	Operation	Parameters	SLA requirement	Violation reason	The Responsible provider
ABS	Produce	List Submit	Format of list: classic verification of submit: Yes	Verification: No	Provider4
Cal	Add	Response	x(1 to 12) y(0 to 15) sumx+y>0: yes	1. x>15 2. x<=15, Y>5 3. x<=15, Y<=5	1.Provider1 2. Provider1 or Provider2 3. Provider1 or Provider3
Dessin	Create	Submit List	verification of submit: no Format of list: standard	/	Provider2
FM	get	Listtypes	list:static verification: yes	1. Verification=no, list=static 2. Verification=no, list=dynamic	1.Provier14 2. Provider3
IAN	Remove	Add Soustra	x(0 to 50) y(0 to 20) soustrax-y>0: yes sumx+y>0: yes	/	/
QTE	Produce	Submit	Format: modern verification: no	/	/
Mul	Add	Response	x(4 to 10) y(5 to 20) sumx+y>0:yes	1. y>5 2. y<=5	1.Provider6 2. Provider1 or Provider3
SSN	Delete	Listtyp Buy Filter	Format:dynamic of listtypes Buying of buy Filter of filter	nobuying	Provier4

Figure 51:Prédiction de violation

## 4 Comparaison entre les substitutions proposées

Chacune des substitutions utilise une méthode différente, nous allons montrer une évaluation des trois substitutions mentionnées précédemment dans la figure 52:

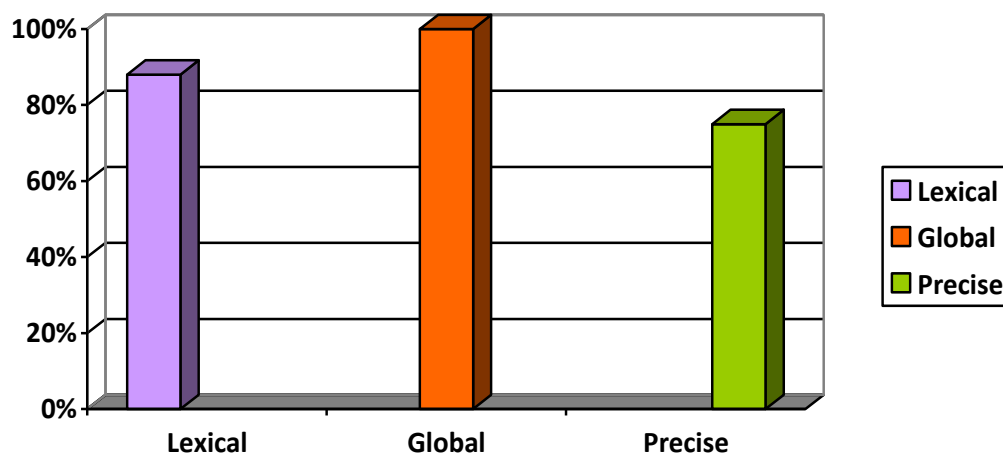


Figure 52: Comparaison des substitutions

On remarque qu'avec la substitution globale tous les services trouvent des services remplaçants, donc elle est classée la première avec un pourcentage de 100% , ensuite la substitution lexicale avec 88%, et enfin la substitution précise avec 75%.

### - Substitution globale :

Cette substitution prend en entrée un ensemble de paramètres de similarité, calculés par la méthode de jaro-winkler, et applique la méthode K-means pour donner en sortie plusieurs clusters, sachant que chacun regroupe des services similaires.

Cette méthode n'est pas très efficace, puisque les services du même cluster ne peuvent pas être toujours substituables. Dans le cas où un service produit une même fonctionnalité et plus qu'un autre service, alors le premier service peut remplacer le deuxième, mais en réalité le deuxième service est incapable de faire la même chose, bien qu'ils sont dans la même communauté.

### - Substitution lexicale :

Cette substitution prend en entrée l'ensemble d'identifiants des opérations pour plusieurs services. Elle est valable seulement si les identifiants existent dans la base de données WordNet, ensuite elle les compare par rapport au sens via la méthode de Wu-palmer, et en résulte une liste des services sémantiquement similaires.

Cette méthode dépend de ce qui existe dans la base de données lexicale, c'est-à-dire si les deux identifiants n'étaient pas mentionnés la substitution ne sera pas possible.



- **Substitution précise :**

Cette substitution prend en entrée un ensemble de paramètres de similarité, calculés par la méthode de jaro-winkler, et construit un réseau de similarité, pour donner en sortie une des services similaires.

Cette méthode est la plus efficace, car elle prend en considération les relations entre les services d'une façon précise, c'est-à-dire, si on a un service avec une fonctionnalité, et un autre service avec plusieurs fonctionnalités, une parmi eux est similaire à celle du premier service, donc le deuxième service est considéré similaire au premier puisque il contient sa fonctionnalité, par contre le premier service ne peut pas remplacer le deuxième puisque il a une seule fonctionnalité similaire parmi toutes les autres.

## 5 Evaluation de la solution proposée

Nous allons évaluer notre proposition dans le tableau 27 selon les critères de comparaison discuté dans le chapitre 02 :

Critères	Utilisation
Extraction	WSDL✓
	USDL×
Composition	×
Sélection	✓
Comparaison	Syntaxique✓
	Sémantique✓
Classification	Communauté✓
	Réseau✓
SLA	✓
Type de service	Web×
	Cloud✓

Tableau 27: Evaluation

➤ **Discussion**

Notre proposition utilise la description du fichier WSDL, elle permet une sélection selon la méthode MCDM, et compare les services cloud selon l'aspect syntaxique ainsi que l'aspect sémantique, elle fournit aussi une classification de communauté et de réseau tout en prédisant les violations des SLA.

## 6 Conclusion

Dans ce chapitre nous avons présenté notre solution de substitution des services cloud, le prototype réalisé fournit la possibilité aux clients de sélectionner les différents services publiés selon leurs besoins et retourne le service le plus approprié, il permet aussi une substitution des services échoués et donne une prédiction qui aide le client à éviter l'échec des services.

Nous avons mis en pratique les concepts étudiés dans les chapitres précédents en les modélisant par des exemples pour expliquer tous les cas de figures possibles.

## Conclusion Générale

Dernièrement, les applications connaissent une croissance de complexité due à une évolution des besoins, il est devenu nécessaire de communiquer et de partager de nombreuses ressources ou applications hétérogènes. Plusieurs approches ont amélioré la productivité par exemple l'architecture orientée services (SOA) est l'un des modèles qui permet de résoudre cette problématique.

Actuellement, la gestion des services dans le Cloud Computing représente un défi surtout dans les cas d'échecs, où un client pense trouver le service qui satisfait son besoin, malheureusement le service choisi n'est plus valable à ce moment, il doit le remplacer par un autre répondant à ses exigences. C'est le problème de substitution.

Nous avons proposé une solution qui permet de remplacer les services échoués en rassemblant les services qui fonctionnent de la même façon ; notre travail a le but aussi de faciliter la sélection des services en retournant le meilleur.

Dans l'intention de rendre notre client plus satisfait nous avons pensé à une méthode de prédiction de violation qui sert à trouver l'erreur pour alerter le client d'avance.

Les travaux discutés dans ce mémoire, ouvre la voie à plusieurs pistes de recherche qui peuvent faire l'objet de perspectives, en modifiant les méthodes de calcul de la similarité (Jaro Winkler pour la similarité syntaxique , ou Wu Palmer pour la similarité sémantique), la technique de la sélection (MCDM) ainsi que l'algorithme de la prédiction (J48).

## Références

- [1] P. Mell and T. Grance, \The NIST -National Institute of Standards and Technology- Definition of Cloud Computing," NIST Special Publication 800-145, p. 7, 2011.
- [2] S. Srinivasan, \Cloud Computing Evolution," no. December, pp. 1-16, 2014.
- [3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, \Cloud computing and emerging it platforms : Vision, hype, and reality for delivering computing as the 5th utility," Future Generation computer systems, vol. 25, no. 6, pp. 599-616, 2009.
- [4] Y. Jadeja and K. Modi, \Cloud computing-concepts, architecture and challenges," in 2012 International Conference on Computing, Electro-nics and Electrical Technologies (ICCEET). IEEE, 2012, pp. 877-880.
- [5] D. Rountree and I. Castrillo, The basics of cloud computing : Understanding the fundamentals of cloud computing in theory and practice. Newnes, 2013.
- [6] Q. Zhang, Q. Zhu, M. F. Zhani, R. Boutaba, and J. L. Hellerstein, \Dynamic service placement in geographically distributed clouds," IEEE Journal on Selected Areas in Communications, vol. 31, no. 12, pp. 762-772, 2013.
- [7] I. Foster, Y. Zhao, I. Raicu, and S. Lu, \Cloud computing and grid computing 360-degree compared," in 2008 grid computing environments workshop. Ieee, 2008, pp. 1-10.
- [8] W. Kim, \Cloud computing : Today and tomorrow." Journal of object technology, vol. 8, no. 1, pp. 65-72, 2009.
- [9] M. G. Xavier, M. V. Neves, F. D. Rossi, T. C. Ferreto, T. Lange, and C. A. De Rose, \Performance evaluation of container-based virtualization for high performance computing environments," in 2013 21st Euro-micro International Conference on Parallel, Distributed, and Network-Based Processing. IEEE, 2013, pp. 233-240.
- [10] R. Jain and S. Paul, \Network virtualization and software defined networking for cloud computing : a survey," IEEE Communications Magazine, vol. 51, no. 11, pp. 24-31, 2013.
- [11] I. Tafa, E. Beqiri, H. Paci, E. Kajo, and A. Xhuvani, \The evaluation of transfer time, cpu consumption and memory utilization in xen-pv, xen-hvm, openvz, kvm-fv and kvm-pv hypervisors using ftp and http approaches," in 2011 Third International Conference on Intelligent Networking and Collaborative Systems. IEEE, 2011, pp. 502-50

- [12] A. Kovari and P. Dukan, \Kvm & openvz virtualization based iaas open source cloud virtualization platforms : Opennode, proxmox ve," in 2012 IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics. IEEE, 2012, pp. 335-339.
- [13] M. Terrell and N. Meghanathan, \Setting up of a cloud cyber infra-structure using xen hypervisor," in 2013 10th International Conference on Information Technology : New Generations. IEEE, 2013, pp. 648-652.
- [14] L. Malhotra, D. Agarwal, and A. Jaiswal, \Virtualization in Cloud Computing," Journal of Information Technology & Software Engineering, vol. 04, no. 02, pp. 2-4, 2014.
- [15] M. U. Bokhari and Q. Makki, \A Survey on Cloud Computing A Cloud architecture A Security," pp. 149-164, 2018.
- [16] D. Rani and R. K. Ranjan, \A Comparative Study of SaaS , PaaS and IaaS in Cloud Computing," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 6, pp. 458-461, 2014.
- [17] S. Lohr, \Google and ibm join in ‘cloud computing’research," New York Times, vol. 8, 2007.
- [18] W. Wang, Y. Xu, and M. Khanna, \A survey on the communication architectures in smart grid," Computer networks, vol. 55, no. 15, pp. 3604-3629, 2011.
- [19] W. A. Jansen, T. Grance et al., \Guidelines on security and privacy in public cloud computing," 2011.
- [20] S. B. Shaw, \A Survey on Cloud Computing Subhadra Bose Shaw," pp. 1-6, 2014.
- [21] S. Patidar, D. Rane, and P. Jain, \A survey paper on cloud computing," in 2012 Second International Conference on Advanced Computing & Communication Technologies. IEEE, 2012, pp. 394-398.
- [22] \A survey paper on cloud computing," Proceedings - 2012 2nd International Conference on Advanced Computing and Communication Technologies, ACCT 2012, pp. 394-398, 2012.
- [23] T. Eymann, \Cloud computing|enzyklopaedie der wirtschaftsinformatik," 2008.
- [24] C. Storage, \Contribution a la securit du Cloud Computing : Application des algorithmes de chi rement pour securiser les donnees dans le Cloud Storage," pp. 1-127, 2016.

- [25] S. B. Shaw and A. Singh, "A survey on cloud computing," in 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE). IEEE, 2014, pp. 1-6.
- [26] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, "NIST Special Publication 500-292 : Cloud Computing Reference Architecture," p. 28, 2011.
- [27] "Nist cloud computing reference architecture," NIST special publication, vol. 500, no. 2011, p. 292, 2011.
- [28] R. B. Bohn, J. Messina, F. Liu, J. Tong, and J. Mao, "Nist cloud computing reference architecture," in 2011 IEEE World Congress on Services. IEEE, 2011, pp. 594-596.
- [29] C. Service, M. Perspective, and C. Lim, "06588059," 2011.
- [30] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," IEEE Security & Privacy, vol. 8, no. 6, pp. 24-31, 2010.
- [31] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in 2010 3rd International symposium on parallel architectures, algorithms and programming. IEEE, 2010, pp. 89-96.
- [32] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. IEEE, 2010, pp. 826-831.
- [33] Y. Demchenko, C. Ngo, C. De Laat, J. A. Garcia-Espin, S. Figue-rola, J. Rodriguez, L. M. Contreras, G. Landi, and N. Ciulli, "Inter-cloud architecture framework for heterogeneous cloud based infrastructure services provisioning on-demand," Proceedings - 27th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2013, pp. 777-784, 2013.
- [34] C. Cachin and R. Haas, "Dependable Storage in the Intercloud," Ibm, pp. 1-6, 2010.
- [35] Y. Zhu and H. Hu, "for Integrity Verification in Multicloud Storage," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 12, pp. 2231-2244, 2012.
- [36] M. Singhal, S. Chandrasekhar, T. Ge, R. Sandhu, R. Krishnan, G.-j. Ahn, and E. Bertino, "Rese a Rch Fe atuRe," pp. 76-84, 2013.
- [37] X. Wang, J. Cao, and Y. Xiang, "Dynamic cloud service selection using an adaptive learning mechanism in multi-cloud computing," Journal of Systems and Software, vol. 100, pp. 195-210, 2015. [Online]. Available : <http://dx.doi.org/10.1016/j.jss.2014.10.047> « dernière visite le 16 décembre 2019 »
- [38] L. U. Claude and B. Lyon, "Un canevas pour l' adaptation et la substitution de services Web Yehia Taher To cite this version : HAL Id : tel-00996101 D'elivr'ee par," 2014

- [39] M. F. Abouzaid, “Analyse formelle d’orchestrations de services web,” Ph.D. dissertation, Ecole Polytechnique de Montréal, 2010.
- [40] Z.-y. Di, “A Service-cluster Based approach to service substitution of web service composition - Op1 Op2 Op3 / ; eSi . ef . Input , L eSj . ef . Input \ ; finput E I ; , : : 3f ( input ) E I j,” pp. 564–568, 2012.
- [41] J. Wang, Y. Han, J. Wang, and G. Li, “An approach to dynamically reconfiguring service-oriented applications from a business perspective,” in *Advanced workshop on content computing*. Springer, 2004, pp. 357–368.
- [42] F. Hematian and C. Cheriki, “Indirect Substitution Method in Combinable Services by Eliminating Incompatible Services,” vol. 7, no. 7, pp. 358–366, 2016.
- [43] H. Al-Helal and R. Gamble, “Introducing replaceability into web service composition,” *IEEE Transactions on services computing*, vol. 7, no. 2, pp. 198–209, 2013.
- [44] R. Iordache and F. Moldoveanu, “Qos-aware web service semantic selection based on preferences,” *Procedia Engineering*, vol. 69, pp. 1152–1161, 2014.
- [45] K. Saeedi, L. Zhao, and P. R. F. Sampaio, “Extending bpmn for supporting customer-facing service quality requirements,” in *2010 IEEE International Conference on Web Services*. IEEE, 2010, pp. 616–623.
- [46] D. Z. G. Garcia and M. B. F. de Toledo, “Achieving autonomic web service integration : A quality of service policybased approach,” *International Transactions on Systems Science and Applications*, vol. 3, pp. 45–63, 2007.
- [47] S. Bosse, M. Splieth, and K. Turowski, “Multi-objective optimization of it service availability and costs,” *Reliability Engineering & System Safety*, vol. 147, pp. 142–155, 2016.
- [48] D. D. Informatique, “Composition et interop´eration des services web s´emantiques,” 2014.
- [49] F. Xinyang, S. Jianjing, and F. Ying, “REST : An alternative to RPC for web services architecture,” *2009 1st International Conference on Future Information Networks, ICFIN 2009*, pp. 7–10, 2009.
- [50] G. Goth, “Critics Say Web Services Need a REST,” *IEEE Distributed Systems Online*, vol. 5, no. 12, pp. 1–1, 2004.
- [51] R. Sara, A. Bakhta, and L. Lakhdar, “A similarity network for web services operations substitution,” *Journal of King Saud University - Computer and Information Sciences*, 2018. [Online].
- [52] M. Bichier and K.-J. Lin, “Service-oriented computing,” *Computer*, vol. 39, no. 3, pp. 99–101, 2006.
- [53] M. P. Papazoglou, “Service-oriented computing : Concepts, characteristics and directions,” in *Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003. WISE 2003*. IEEE, 2003, pp. 3–12.

- [54] M. N. Huhns and M. P. Singh, "Service-oriented computing : Key concepts and principles," IEEE Internet computing, vol. 9, no. 1, pp. 75–81, 2005.
- [55] Z. Wu, K. Gomadam, A. Ranabahu, A. P. Sheth, and J. A. Miller, "Automatic composition of semantic Web services using process mediation," ICEIS 2007 - 9th International Conference on Enterprise Information Systems, Proceedings, vol. SAIC, no. ii, pp. 453–461, 2007.
- [56] L. Chen and H. Qin, "Graph planning approach for adaptive substitution of conversational services," 2012 9th International Conference on Service Systems and Service Management - Proceedings of ICSSSM'12, pp. 598–603, 2012.
- [57] E. Aljournah, F. Al-Mousawi, I. Ahmad, M. Al-Shammri, and Z. AlJady, "SLA in cloud computing architectures : A comprehensive study," International Journal of Grid and Distributed Computing, vol. 8, no. 5, pp. 7–32, 2015.
- [58] D. D. Informatique, "Composition et interop eration des services web s emantiques," 2014.
- [59] Limam N, Boutaba R. Assessing software service quality and trust worthiness at selection time. IEEE Trans Softw Eng 2010;36(4):559–74.
- [60] Silas S, Rajsingh EB, Ezra K. Efficient service selection middleware using ELECTRE methodology for Cloud environments. Inf Technol J 2012;11(7):868–75.
- [61] Saripalli P, Pingali G. MADMAC: multiple attribute decision methodology for adoption of Clouds. In: Proceedings of the IEEE international conference on Cloud computing (CLOUD). Washington, DC; 2011.
- [62] Zhao L, Ren Y, Li M, Sakurai K. Flexible service selection with user-specific QoS support in service-oriented architecture. J Netw Comput Appl 2012;35 (3):962–73.
- [63] Wittern E, Kuhlenkamp J, Menzel M. Cloud service selection based on variability modeling. In: Proceedings of the 10th international conference on service-oriented computing (ICSOC), Shanghai, China; 2012.
- [64] Dustdar S, Schreiner W. 'A survey on web services composition', Int. J. Web and Grid Services, 2005, Vol. 1, No. 1, p 2.
- [65] R. Sara, E. Des, O. A. Partir, D. E. S. Fichiers, S. Des, O. Entre, and ` S. Web, "THESE Pr esent ee par : Sp ecialit e : Informatique," 2019.
- [66] J. Zhou, N. A. Abdullah, and Z. Shi, "A hybrid P2P approach to service discovery in the cloud," Int. J. Inf. Technol. Comput. Sci., vol. 3, no. 1, pp. 1–9, 2011.
- [67] V. S. K. Nagireddi and S. Mishra, "An ontology based cloud service generic search engine," in 2013 8th International Conference on Computer Science & Education, 2013, pp. 335–340.
- [68] J. Cardoso, A. Barros, N. May, and U. Kyla u, "Towards a unified service description language for the internet of services: Requirements and first developments," in 2010 IEEE International Conference on Services Computing, 2010, pp. 602–609.



- [69] N. Ghosh, Soumya K. Ghosh, Sajal K. Das. SelCSP: A Framework to Facilitate Selection of Cloud Service Providers, 2168-7161 2013
- [70] S. K. Garg, S. Versteeg, and R. Buyya, “A framework for ranking of cloud computing services,” *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012–1023, 2013.
- [71] H. Malouche, Y. Ben Halima, and H. Ben Ghezala, “A brokerage architecture: Cloud service Pelection,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10380 LNCS, pp. 45–55, 2017.
- [72] M. Godse and S. Mulik. An approach for selecting Software-as-a-Service (SaaS) product. In *IEEE International Conference on Cloud Computing, CLOUD*, pages 155–158, 2009.
- [73] Abderrahmane Maaradji, Hakim Hacid, Johann Daigremont and Noel Crespi, « Composition des services Web Basée sur les Réseau Sociaux », *Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances*, Tunisie, 2010.
- [74] Rao J and Su X. ‘A survey of automated Web service composition methods’, *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition*, California, USA: Springer, 2004.
- [75] M. Fluegge, I. J. G. Santos, N. P. Tizzo, and E. R. M. Madeira, “Challenges and techniques on the road to dynamically compose Web services,” *Proceedings of the 6th international conference on Web engineering*, (New York, USA), ACM, 2006.
- [76] R. Sara, A. Bakhta, and L. Lakhdar, “A similarity network for web services operations substitution,” *Journal of King Saud University - Computer and Information Sciences*, 2018.
- [77] A. Meskini, Y. Taher, A. El Gammal, B. Finance, and Y. Slimani, “Proactive learning from SLA violation in cloud service based application,” *CLOSER 2016 - Proceedings of the 6th International Conference on Cloud Computing and Services Science*, vol. 1, no. March, pp. 186–193, 2016
- [78] J. L. Mathieu, M. Gonz, F. Oldewurtel, T. Borsche, M. Bucher, P. Fortenbacher, T. Haring, M. Olivier, and E. Vrettos, “Personal use of this material is permitted . Permission from IEEE must be obtained for all other users , including reprinting / republishing this material for advertising or promotional purposes , creating new collective works for resale or redistribution,” 2013.
- [79] Chantal Cherifi, Vincent Labatut, Jean-François Santucci “Topological Properties of Web Services Similarity Networks”, *Journal of Strategic Advantage of Computing Information Systems in Enterprise Management*, ATINER, pp. 105-117, 2010.
- [80] Boutahar Jaouad, Rachad Taoufik, El Houssaini Souhail «A New Efficient Matching Method For Web Services Substitution », *Journal Of Computer Science Issues*, Vol.11, No 2, September 2014.
- [81] Boudjemaa Boudaa « Vers Une Substitution Des Services Web Sans Inconsistance Sémantique », *Journal CoRR* volume abs/1305.1666, 2013.
- [82] Dong, X., Halevy, A., Madhavan, J., Nemes, E. And Zhang, J. 2004 « Similaraty Recherche For Web Services ». In *Proceedings Of The 30<sup>th</sup> Vldb Conference*, Toronto, Canada, August

2004,372-383.

[83] Rachad.T ,Boutahar. T,El Ghazi. S , « A New Efficient Method For Calculating Similarity Between Web Services »,Journal Of Advanced Computer Science And Applications,Vol.5, No 08, P. 60-67, 2014.

[84] VishkaeiMahsa Jamal, Baraani-Dastjerdi Ahmad And Jamshidi Kamal « An Architecture For Web Service Similarity Evaluation Based On Their Functional And Qos Aspects », International Journal On Web Service Computing (Ijwsc), Vol.2, No.2, June 2011.

[85] Huang, .A «Similarity Measures For Text Document Clustering ». In Holland, J., Nicholas, A., AndBrignoli, D., Editors, New Zealand Computer Science Research Student Conference, Pages 49-56, 2008.

[86] OkbaTibermacine, ChoukiTibermacine, FoudilCherif, « A Practical Approach To The Measurement Of Similarity Between Wsdl-Based Web Services» ,In Proceedings Of The Frensh-Speaking Conference On Software Architecture (Cal'2014),France,2014.

[87] YehiaTaher « Un Canevas Pour L'adaptation Et La Substitution De Services Web », L'université Claude Bernard Lyon 1, 2009.

[88] Y. Taher, D. Benslimane, M.-C.Fauvet, and Z.Maamar,” Towards an Approach for Web Services Substitution”, in 10 th International Database Engineering and Applications Symposium,2006,pp.166-173.

[89] “Vous avez dit ”Cloud” ? Tous savoir sur l'informatique dans les nuages.” [Online]. Available : <https://mbamci.com/cloud/> « dernière visite le 30 septembre 2020 »

[90]“hybrid cloud - Globb Security FR.” [Online]. Available : <http://globbsecurity.fr/?attachment{id=45612{#}prettyPhoto>« dernière visite le 30 septembre 2020 »

[91] “Le Cloud Computing, une meilleure solution ? – Communication et ´edition num´erique.” [Online]. Available : <https://lewebpedagogique.com/presencesenligne/2014/10/28/cloud-computing/>« dernière visite le 30 septembre 2020 »

[92] “Cisco Global Intercloud : Open and Interoperable - Cisco Blogs.” [Online]. Available : <https://blogs.cisco.com/datacenter/cisco-global-intercloud-open-and-interoperable>« dernière visite le 30 septembre 2020 »

[93] “Multi-Cloud Platforms offer up key new data capabilities and differentiation - Adaderana Biz English — Sri Lanka Business News.” [Online]. Available : <http://bizenglish.adaderana.lk/multi-cloud-platforms-offer-up-key-new-data-capabilities-and-differentiation/>« dernière visite le 30 septembre 2020 »

[94] “SOAP As Melted ! ! ! Drained By REST, But Not Dried — by Venkatesh Vellore — Medium.” [Online]. Available : <https://medium.com/@venkatesh.dev.vijayarathy/soap-as-melted-drained-by-rest-but-not-dried-6d5c5a904bc4> « dernière visite le 30 septembre 2020 »

[95] P. Plebani and B. Pernici, “Urbe : Web service retrieval based on similarity evaluation,” IEEE Transactions on Knowledge and data engineering, vol. 21, no. 11, pp. 1629–1642, 2009.

- [96] G. M. Winch and R. Courtney, “The organization of innovation brokers : An international review,” *Technology Analysis and Strategic Management*, vol. 19, no. 6, pp. 747–763, 2007.
- [97] Tibermacine, Okba, Tibermacine, Chouki, Cherif, Foudil, 2013. WSSim: a Tool for the Measurement of Web Service Interface Similarity. In: proceedings of the frenchspeaking conference on software Architecture (CAL’2013).
- [98] J. Boutahar, T. Rachad et al., “A new efficient matching method for web services substitution,” arXiv preprint arXiv :1501.05983, 2015.
- [99] Taher, Y., Benslimane, D., Fauvet, M.-C., Maamar, Z., 2006. Towards an Approach for Web Services Substitution. In: 10 th International Database Engineering and Applications Symposium. pp. 166–173
- [100] Dong, X., Halvey, A., Madhavan, J., Nemes, E., Zhang, J., 2004. Similarity search for web services. In: Proceedings of the 30th VLDB Conference, Toronto, Canada, August 2004, pp. 372–383.
- [101] A. Konduri, “Clustering of web services based on semantic similarity,” Ph.D. dissertation, 2008
- [102] Eclipse (logiciel) : d’efinition et explications.” [Online]. Available : <https://www.techno-science.net/definition/517.html> « dernière visite 21/10/2020 »
- [103] “What is Java Development Kit (JDK) ? - Definition from WhatIs.com.” [Online]. Available : <https://www.theserverside.com/definition/Java-Development-Kit-JDK> « dernière visite 21/10/2020 »
- [104] “D’efinition — MySQL — Futura Tech.” [Online]. Available : <https://www.futura-sciences.com/tech/definitions/internet-mysql-4640/> « dernière visite 21/10/2020 »
- [105] Tutoriel Protégé 5.2 - Création d’une petite ontologie.” [Online]. Available : <http://www.iro.umontreal.ca/~lapalme/ift6282/OWL/EtapesCreationOntologie.html> « dernière visite 21/10/2020 »
- [106] Y. Lechevallier Dauphine and Y. Lechevallier, “Approche Data Mining par WEKA WEKA, un logiciel libre d’apprentissage et de data mining,” Tech. Rep. [Online]. Available : <http://www.cs.waikato.ac.nz/ml/weka/>
- [107] A. Fella, M. Malki, and A. Zahaf, “Alignement des ontologies : utilisation de WordNet et une nouvelle mesure structurelle,” Tech. Rep.
- [108] “Extreme Programming : A Gentle Introduction.” [Online]. Available : <http://www.extremeprogramming.org/> « dernière visite 21/10/2020 »
- [109] “Java EE Guide de développement d’applications web en Java - Qu’est-ce que Tomcat ?” [Online]. Available : <https://www.editions-eni.fr/open/mediabook.aspx?idR=98db25384380c51b79e46e170641bf45> « dernière visite 21/10/2020 »
- [110] S. . Khaled ; Ranka and V. Singh, “An efficient k-means clustering algorithm,” Tech. Rep., 1997. [Online]. Available : <https://surface.syr.edu/eecshttps://surface.syr.edu/eecs/4>
- [111] “K-means : Definition & Avantages / Inconvénients - Le DataScientist.” [Online]. Available : <https://le-datascientist.fr/k-means> « dernière visite 21/10/2020 »

## Annexe A : Algorithme K-Means

K-means est une méthode de clustering non-hiérarchique (ne sont pas ordonnés en fonction de leurs ressemblance) et non supervisée (les groupes n'existent pas avant), regroupe les objets dans des communautés différentes, son fonctionnement est décrit dans l'algorithme de la figure 53

---

```
function Direct-k-means()
  Initialize  $k$  prototypes  $(w_1, \dots, w_k)$  such that  $w_j = i_l$ ,  $j \in \{1, \dots, k\}$ ,  $l \in \{1, \dots, n\}$ 
  Each cluster  $C_j$  is associated with prototype  $w_j$ 
  Repeat
    for each input vector  $i_l$ , where  $l \in \{1, \dots, n\}$ ,
      do
        Assign  $i_l$  to the cluster  $C_{j^*}$  with nearest prototype  $w_{j^*}$ 
        (i.e.,  $|i_l - w_{j^*}| \leq |i_l - w_j|$ ,  $j \in \{1, \dots, k\}$ )
    for each cluster  $C_j$ , where  $j \in \{1, \dots, k\}$ , do
      Update the prototype  $w_j$  to be the centroid of all samples currently in  $C_j$ , so that  $w_j = \sum_{i_l \in C_j} i_l / |C_j|$ 
  Compute the error function:


$$E = \sum_{j=1}^k \sum_{i_l \in C_j} |i_l - w_j|^2$$


  Until  $E$  does not change significantly or cluster membership no longer changes
```

---

Figure 53: Algorithme K-Means[110]

### **Complexité:**

K-means est considérée comme la technique la plus rapide de classification tel que sa complexité est de «  $O(K - \text{means}) = n \times m$  » sachant que :

- ✓ n: Représente le nombre total d'opérations.
- ✓ m: Le nombre de clusters non vides

### **Avantages :**

La méthode K-means est considérée connue grâce à ces avantages, nous présentons quelques uns selon [110] :

- ✓ Rapide, facile à implémenter et flexible (s'adapte aux divers changements de vos données) avec une efficacité.
- ✓ Produit des clusters plus proches et serrées, qui donne une interprétation simple ainsi que précise .

### **Inconvénients:**

Les K-means est une technique de classification utilisée pour simplifier des ensembles de données volumineux en ensembles de données simples, malgré ces avantages elle contient aussi des point de faiblesse présentés comme suit [110] :

- ✓ K-means fournit un ensemble non optimal de clusters, par conséquent la construction des ensembles affecte les résultats des données.
- ✓ Limitation des calculs et de types c'est-à-dire elle ne comprend pas un grand ensemble de données, ainsi qu'elle traite les données numériques seulement.
- ✓ Pour une efficace classification par K-moyennes (Means) , une spécification du nombre des cluster(K) doit être faite au début puisque il est difficile de prévoir leurs valeurs d'avance.