

**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**  
**MASTER EN INFORMATIQUE**  
**Option : Système Informatique & Réseaux**

***Thème***

**Solution à base de broker dans un environnement  
d'inter-Cloud de type SaaS**

***Présenté par : Mr CHERIET Aymene et Mr OUCHENE Omar***

***Le : 30/09/2019***

**Devant le jury**

- Dr : I. Cherfa  
- Dr : Fz. Zahra

**Encadré par :**

- Dr : Y.Mancer (USDB)

***Année universitaire 2018/2019***

## *Dédicace*

*À nos très chers parents,*

*Aucun mot, aucune dédicace ne peut exprimer notre respect, notre considération et l'amour éternel pour les sacrifices que vous avez consentis pour notre instruction*

*Et notre bien-être.*

*Votre générosité et votre bonneté ont toujours été un exemple pour nous tous.*

*Trouvez en ce travail le fruit de votre dévouement de votre patience et l'expression de notre gratitude et profond amour*

*Je t'aime papa je t'aime maman et sachez que je suis très reconnaissant*

*À toute notre chère famille,*

*À nos professeurs,*

*À nos chers amis,*

*À nos chers collègues,*

*À tous ceux qui m'aiment,*

*À tous ceux que j'aime,*

*À tous ceux qui nous ont aidé de près ou de loin,*

*Je dédie ce travail avec hommage*

## **Remerciements**

*Avant tout, nous remercions Dieu le très haut qui nous a donné le courage et la volonté de réaliser ce modeste travail.*

*« (CELUI QUI NE REMERCIE PAS LES GENS, NE REMERCIE PAS ALLAH.) »*

**[Authentique Hadith]**

*Nous remercions avant tout le Dieu tout puissant d'avoir veiller sincèrement sur notre état de santé pendant ces Cinq années de notre cursus universitaire en nous donnant ainsi la force et le courage de bien mener à terme ce modeste travail*

*Que Madame Y Mancer , professeur De l'Informatique à l'université «Saad Dahleb Blida», trouve ici le témoignage de notre profonde reconnaissance. Ses encouragements, et surtout ses critiques, Sa sensibilisation, ont largement contribué à l'accomplissement de notre travail.. Nous la remercions infiniment de nous avoir toujours poussés vers l'avant.*

*On tient également à remercier « TOUS » les Messieurs et dames, nos professeurs qui nous ont enseigné durant deux ans master en système D'informatique et réseaux, pour leurs précieux conseils et ses orientations,*

*Enfin, merci à nos familles (nos Chères mères et nos père) pour le soutien et l'encouragement qu'ils nous ont apporté tout au long de ce travail.*

## Résumé

Les leaders du marché de technologie de l'information, par exemple, Microsoft ; Google et Amazon, se dirigent de plus en plus vers des solutions à base de Cloud. Cependant, les solutions proposées sur le marché souffrent du problème de l'enfermement propriétaire. Cela, engendre une interopérabilité limitée causant ainsi l'adhésion d'un utilisateur à un seul fournisseur Cloud. De plus, la migration exigée par une application / donnée d'un fournisseur à un autre peut nécessiter un effort significatif pour adapter les standards du nouveau fournisseur et pour la mise en œuvre. D'un autre côté, les demandes des clients deviennent de plus en plus complexes, nécessitant la collaboration de plusieurs Clouds qui peuvent provenir de fournisseurs distincts. Pour faire face à ces problèmes, l'utilisation de médiateurs demeure indispensable.

Le présent mémoire introduit une solution de médiation dans un environnement de Cloud Computing pour la couche SaaS (Software as a Service) dans le but de garantir l'automatisation des interactions entre des fournisseurs de services Clouds et des clients. La solution proposée assure la description, la découverte, la sélection et la composition des services Clouds de type SaaS.

Afin d'illustrer l'applicabilité des propositions, un prototype est réalisé dans un environnement Java. La mise en œuvre de l'application reprend les concepts décrits dans la partie conceptuelle.

**Mots clés :** Cloud Computing, broker, Software as a Service (SaaS), inter-Cloud, description, sélection, composition.

# Abstract

Market leaders in information technology, for example, Microsoft; Google and Amazon, are moving more and more towards cloud-based solutions. However, the solutions offered on the market suffer from the problem of owner-locking. This creates limited interoperability, causing a user to join a single cloud provider. In addition, the migration required by one application / given from one provider to another may require a significant effort to adapt the standards of the new provider and for implementation. On the other hand, customer requests are becoming more complex, requiring the collaboration of multiple Clouds that can come from separate vendors. To deal with these problems, the use of mediators remains essential.

This work introduces a mediation solution in a Cloud Computing environment for the Software as a Service (SaaS) layer in order to ensure the automation of interactions between cloud service providers and clients. The proposed solution provides description, discovery, selection and composition of SaaS cloud services.

In order to illustrate the applicability of the proposals, a prototype is made in a Java environment. The implementation of the application uses the concepts described in the conceptual section.

**Keywords:** Cloud Computing, broker, Software as a Service (SaaS), inter-Cloud, description, selection, composition.

## ملخص:

إن رواد السوق في تكنولوجيا المعلومات، على سبيل المثال، مايكروسوفت؛ جوجل وأمازون تتحرك بشكل متزايد نحو الحلول المستندة إلى السحابة. ومع ذلك، فإن الحلول المعروضة في السوق تعاني من مشكلة الحبس في الملكية. يؤدي هذا إلى تقليص إمكانية التشغيل المتداخل بين السحب، مما تسبب في انضمام المستخدم إلى مزود سحابة واحد فقط. وبالإضافة إلى ذلك، فإن الترحيل المطلوب من أحد التطبيقات/المقدمة من بائع إلى آخر قد يتطلب بذل جهد كبير لتكثيف مع معايير المورد الجديد وللتنفيذ. ومن ناحية أخرى، أصبحت طلبات العملاء معقدة بشكل متزايد، مما يتطلب تعاون العديد من السحابات التي قد تأتي من موردين منفصلين. ولمعالجة هذه المشاكل، لا يزال استخدام الوسطاء أمراً أساسياً.

تقدم المذكرة الحالية حلاً للوساطة في بيئة الحوسبة السحابية لطبقة SaaS (البرمجيات كخدمة) لضمان أتمتة التفاعلات بين مزودي الخدمات السحابية والعملاء. يوفر الحل المقترح وصف خدمات الحوسبة السحابية لطبقة SaaS اكتشافها، اختيارها وتكوينها.

ولتوضيح إمكانية تطبيق المقترحات، يتم تقديم نموذج أولي في بيئة جافا (Java). ويتناول تنفيذ التطبيق المفاهيم الموصوفة في الجزء المفاهيمي.

**الكلمات الرئيسية :** الحوسبة السحابية، وسيط، البرمجيات كخدمة (SaaS)، بين السحابة، وصف الخدمة، اكتشاف الخدمة، وتركيب الخدمة.

# Sommaire

Introduction générale .....	1
<b>Chapitre I Généralités sur le Cloud Computing</b>	
I.1 Introduction.....	4
I.2 Définition du Cloud Computing .....	4
I.3 Eléments constitutifs du Cloud Computing .....	6
I.3.1 La virtualisation .....	6
I.3.2 Centre de Calcul (Datacenter).....	7
I.3.3 Plateforme collaborative.....	7
I.4 Caractéristiques du Cloud Computing .....	8
I.5 Types de services dans les environnements de Cloud Computing .....	10
I.5.1 SaaS (Software as a Service) .....	10
I.5.2 PaaS (Platform as a Service).....	11
I.5.3 IaaS (Infrastructure as a Service) .....	11
I.5.4 Autres types de services Clouds .....	11
I.6 Modèles de déploiement du Cloud Computing .....	12
I.6.1 Cloud public.....	13
I.6.2 Cloud privé .....	13
I.6.3 Cloud hybride.....	14
I.6.4 Cloud communautaire .....	15
I.7 Architecture de référence du Cloud Computing .....	15
I.7.1 Acteurs du Cloud Computing .....	17
I.7.1.1 Consommateur Cloud ‘Cloud Consumer’ .....	17
I.7.1.2 Fournisseur Cloud ‘Cloud Provider’ .....	18
I.7.1.3 Courtier en nuage ‘Cloud Broker’ .....	19
I.7.1.4 Relations entre les acteurs du Cloud Computing .....	20
I.8 Avantages et inconvénients du Cloud Computing .....	21
I.9 Evolution du Cloud Computing .....	23
I.9.1 Inter-Cloud.....	23
I.9.2 Multi-Cloud .....	24
I.10 Conclusion.....	24

## **Chapitre II Courtage en nuage**

II.1	Introduction.....	25
II.2	Cloud broker entre SaaS.....	26
II.2.1	Courtage de services en nuage.....	26
II.2.2	Rôle du courtier nuageux.....	27
II.2.2.1	Consultant et négociateur .....	27
II.2.2.2	Conseiller en conformité .....	28
II.2.2.3	Cloud orchestrateur hybride .....	28
II.2.2.4	Comptable et gestionnaire des dépenses .....	28
II.2.2.5	Fournisseur de services professionnels .....	29
II.3	Meilleurs courtiers en nuage de 2019 .....	29
II.4	Les composants essentiels d'un contrat de Cloud.....	30
II.5	Les langages de description de service.....	31
II.6	La découverte des services Cloud.....	35
II.6.1	Le processus de découverte de service.....	35
II.7	Sélection des services Cloud.....	36
II.7.1	Définition du but de la sélection du service Cloud .....	36
II.8	La qualité de service (QoS).....	37
II.9	Composition des services Cloud .....	40
II.9.1	Définition.....	40
II.9.2	Catégorisation des mécanismes de composition.....	40
II.9.3	Problème de la recherche de composition des services Cloud .....	41
II.9.3.1	Sélection de services composites .....	41
II.9.4	Structure d'exécution de services .....	42
II.9.5	Cycle de vie de composition de services .....	42
II.10	Synthèse et étude comparative des solutions à base de broker.....	43
II.10.1	Les critères d'évaluation de l'étude comparative .....	43
II.10.2	Comparaison entre les approches de description de services Cloud .....	45
II.10.3	Comparaison entre les approches de découverte de services Cloud .....	47
II.10.4	Comparaison entre les approches de sélection de service Cloud.....	50
II.10.4.1	Synthèse des travaux de sélection des services Cloud .....	52
II.10.5	Comparaison entre les approches de composition de service Cloud .....	54

II.11 Conclusion .....	59
------------------------	----

### ***Chapitre III Modélisation du Cloud broker***

III.1 Introduction.....	60
III.2 Architecture proposée du Cloud broker .....	60
III.3 Analyse des besoins .....	63
III.4 Description de service .....	65
III.5 Découverte des services .....	68
III.6 Sélection des services .....	69
III.7 Composition de services.....	74
III.8 Interface fournisseur.....	78
III.9 Interface client .....	78
III.10 Description des règles d'interaction entre les services Cloud .....	79
III.11 Description des mécanismes de communication .....	80
III.12 Conclusion .....	80

### ***Chapitre IV Expérimentation***

IV.1 Introduction.....	81
IV.2 Outils et environnement de développement .....	81
IV.3 Simulation.....	82
IV.4 Conclusion .....	100
Conclusion générale .....	101

Références Bibliographiques

## Liste des figures

Figure 1 : Exemple de virtualisation [5].....	9
Figure 2 : Datacenter [11].....	10
Figure 3 : Plateforme collaborative [6].....	11
Figure 4 : Types de services Clouds [12].....	13
Figure 5 : Cloud public [9].....	16
Figure 6 : Cloud Privé [9] .....	17
Figure 7 : Exemple de Cloud hybride [7].....	18
Figure 8 : Le modèle conceptuel de référence du Cloud Computing [17].....	19
Figure 9 : Exemple de services disponibles pour un consommateur de Cloud [17]..	21
Figure 10 : Relations entre les acteurs du Cloud Computing [18]. .....	23
Figure 11 : Obstacles à l'adoption du Cloud Computing [18]. .....	25
Figure 12 : Relations entre le consommateur, le courtier et le fournisseur [18] .....	26
Figure 13 : Exemple d'un fichier WSDL [23]. .....	32
Figure 14 : Exemple d'un fichier USDL [23]. .....	33
Figure 15 : Catégories de description de services Cloud. ....	46
Figure 16 : Niveaux de broker .....	61
Figure 17 : Architecture proposée du Cloud broker .....	62
Figure 18 : Processus de courtage du broker .....	63
Figure 19 : Diagramme de cas d'utilisation du Cloud broker .....	64
Figure 20 : Diagramme de cas d'utilisation d'un fournisseur Cloud .....	65
Figure 21 : Les parties essentielles de description de service .....	67
Figure 22 : Pseudo algorithme de découverte de service .....	68
Figure 23 : pseudo algorithme de la méthode agrégation linéaire pour la sélection ..	69
Figure 24 : Pseudo algorithme de la méthode MCDM pour la sélection de service..	74
Figure 25 : Matrice des chemins .....	76
Figure 26 : Interface login client .....	82
Figure 27 : Création d'un nouveau compte client.....	83
Figure 28 : Interface accueil client .....	83
Figure 29 : Interface de nous-contactez .....	84
Figure 30 : Capture de réception d'email .....	84

Figure 31 : Interface de création de compte fournisseur .....	85
Figure 32 : Interface de publication de service cloud .....	85
Figure 33 : Interface de la liste de services publiés .....	86
Figure 34 : interface de recherche de service .....	88
Figure 35 : interface de service consommer .....	89
Figure 36 : interface de contrat de niveau de service .....	89
Figure 37 : interface d'échange de donnée coté client .....	90
Figure 38 : interface d'échange de donnée coté broker .....	91
Figure 39 : interface d'échange de données partie fournisseur .....	92
Figure 40 : interface de recherche de service .....	93
Figure 41 : interface de composition de service .....	94
Figure 42 : Interface de résultat de chemin après la composition .....	95
Figure 43 : Résultat du meilleur chemin de composition .....	96
Figure 44 : Résultat de composition de service.....	97
Figure 45 : les inputs de service Sell Ticket .....	98
Figure 46 : les outputs de service Sell Ticket .....	98
Figure 47 : les inputs de service Touristic guide .....	98
Figure 48 : les outputs de service Touristic guide .....	98
Figure 49 : les inputs de service Insurance .....	99
Figure 50 : les outputs de service Insurance .....	99
Figure 51 : Recherche des services composés .....	99

## Lise des tableaux

Tableau 1 : Les grands acteurs mondiaux du Cloud Computing [12].....	15
Tableau 2 : Acteurs du Cloud Computing [17].....	20
Tableau 3 : Avantages et inconvénients des services Cloud [3] .....	24
Tableau 4 : Comparaison entre les approches de description de services Cloud .....	46
Tableau 5 : Comparaison des approches de services Cloud selon les aspects techniques, opérationnels, commerciaux et sémantiques. ....	47
Tableau 6 : Résumé de quelques approches de sélection des services Cloud.....	52
Tableau 7 : Comparaison de quelques approches de composition.....	56
Tableau 8 : Récapitulatif des comparaisons.....	58

## Liste des acronymes et abréviations

*AHP : Analytic hierarchy process*

*ANP : Analytic network process*

*BCO : Bee colony optimisation*

*CBMS : Cloud Brokerage Managed Services*

*DAML-S : Semantic Markup for Web Services*

*DBaaS : Database as a Service*

*GA : Genetic algorithm*

*IaaS : Infrastructure as a Service*

*IBM : International Business Machines Corporation*

*ISO : The International Organization for Standardization*

*MCDM : Multiple criteria decision making*

*NaaS : Network as a Service*

*NIST : National Institute of Standards and Technology*

*OWL-S : Semantic Markup for Web Services*

*PaaS : Platform as a Service*

*RDF : Resource Description Framework*

*SLA : Service Level Agreement*

*SaaS : Software as a Service*

*VM : machines virtuelles*

*XaaS : Anything as a Service*

Le Cloud Computing offre différentes options (infrastructure, plateforme, logiciel) en tant que services, il permet d'augmenter l'évolutivité des applications et de réduire le coût de l'infrastructure dans l'entreprise. Malgré ces potentiels bénéfiques, les processus de découverte, de sélection et de composition des services Cloud posent d'énormes problèmes aux consommateurs. Cela est dû au manque de standard pour la description des services Cloud.

Dans un environnement Cloud interopérable, les clients pourront comparer et choisir des offres parmi les différentes offres Cloud de différentes caractéristiques, en négociant avec des fournisseurs Cloud à chaque fois que cela s'avérerait nécessaire, sans mettre en danger les données et les applications et pourront ainsi composer de nouveaux services Cloud à partir de services existants dans le but de satisfaire des demandes plus complexes.

Dans un contexte de collaboration entre différents SaaS, chaque SaaS fournit un service spécifique. Les services vont être composés entre eux (c'est-à-dire le résultat d'un SaaS sera utilisé par un autre SaaS), dans le but de réaliser une application complexe et d'atteindre un objectif commun. La collaboration implique des fournisseurs de Cloud où chaque fournisseur peut fournir un ou plusieurs SaaS. De plus, plusieurs possibilités peuvent être envisagées pour l'ordonnancement et l'interaction des SaaS. De ce fait, choisir la meilleure solution ou la plus adéquate devient un enjeu majeur.

### **Problématique**

Certaines demandes complexes des clients ne peuvent pas être satisfaites par un seul Cloud, nécessitant ainsi la collaboration entre plusieurs Cloud, pour cela une stratégie d'interopérabilité efficace doit être adoptée.

Pour le problème d'interopérabilité, l'idéal c'est que les systèmes Clouds soient conçus dès le départ suivant des normes standards avec des interfaces standardisées

pour la communication et l'interaction. Cette solution peut s'avérer utopique actuellement. La rude concurrence entre les grands fournisseurs les pousse à utiliser des solutions propriétaires. Pour cela, l'utilisation d'un médiateur (broker) peut être une solution prometteuse car elle peut réduire la nécessité de devoir modifier le Cloud.

La problématique de ce sujet consiste à définir une solution pour le courtage entre Clouds de type SaaS.

### **Objectifs**

1. Etude comparative des solutions existantes pour le brokering entre SaaS.
2. Définir une solution à base de broker entre Clouds de type SaaS. Cette solution doit prendre en compte les modules suivants :
  - a. Description des SaaS.
  - b. Découverte des SaaS.
  - c. Le processus de sélection.
  - d. Le processus de composition.
  - e. Interface fournisseur.
  - f. Interface client.
  - g. Description des règles d'interaction entre les services Clouds.
  - h. Description des mécanismes de communication jusqu'au résultat final de la composition qui doit être transmis au client.
3. Validation de la solution proposée par une expérimentation à travers une application Java.

### **Plan de travail**

1. Etat de l'art sur les concepts de base du Cloud Computing (caractéristiques, types de déploiement, types de services, etc.) et l'évolution du Cloud Computing vers l'Inter-Cloud et le multi-Cloud.
2. Etat de l'art sur le brokering entre SaaS et les solutions existantes avec une étude comparative pour faire ressortir les problèmes traités et les problèmes non traités dans les solutions actuelles.

3. Modélisation de la solution proposée et ses différents concepts.
4. Expérimentation de la solution proposée.

### **Structure du mémoire**

Le présent mémoire est organisé en quatre chapitres. Il commence par une introduction générale décrivant le contexte, la problématique et les objectifs de notre travail.

- **Chapitre I** : Présente un aperçu des concepts de base du Cloud Computing, 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** : Introduit un état de l'art sur le courtage en nuage. Ce chapitre est organisé en deux parties. La première partie décrit les concepts de base relatifs aux brokers dans le Cloud. Elle expose aussi les problématiques de description, de découverte, de sélection et de composition des services Clouds. La deuxième partie présente les travaux connexes avec des synthèses et des études comparatives.
- **Chapitre III** : Dans ce chapitre, nous expliquons notre démarche conceptuelle pour l'élaboration de notre solution. Nous commençant d'abord par la description de l'architecture générale de notre broker. Ensuite, nous détaillons chaque étape du processus de courtage à travers une série d'algorithmes utilisés pour automatiser le travail du broker.
- **Chapitre IV** : Illustre la mise en œuvre 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.

## **I.1 Introduction**

Longtemps avant que l'expression "*Cloud Computing*" ne naisse, les architectes de réseaux (ceux qui conçoivent les réseaux intra et inter-entreprises) schématisaient internet par un nuage dans leurs croquis. En anglais, on parlait alors de "The cloud", ce qui signifiait à peu près l'internet que nous connaissons. Ce nuage évoquait alors une connexion vers une quantité indéfinie d'utilisateurs et non pas des services tel que nous l'entendons maintenant.

L'informatique dans le nuage (en anglais, *Cloud Computing*) est devenu un concept majeur faisant référence à l'utilisation de la mémoire et des capacités de calcul des ordinateurs et des serveurs répartis dans le monde entier et liés par un réseau, tel Internet.

Le but de ce chapitre est de se familiariser avec les notions de base du Cloud Computing (caractéristiques, types de déploiement, types de services, etc.) et de voir à la fin son évolution vers l'inter-Cloud.

## **I.2 Définition du Cloud Computing**

Techniquement, le concept de Cloud Computing est loin d'être nouveau, il est même présent depuis des décennies. On en trouve les premières traces dans les années 1960, quand John McCarthy<sup>1</sup> affirmait que cette puissance de traitement informatique serait accessible au public dans le futur. Le terme en lui-même est apparu plus couramment aux alentours de la fin du XXe siècle et il semblerait qu'Amazon.com soit l'un des premiers à avoir assemblé des Datacenter et fournit des accès à des clients.

Les entreprises comme IBM et Google ainsi que plusieurs universités ont seulement commencé à s'y intéresser sérieusement aux alentours de 2008, quand le Cloud Computing est devenu un concept à la mode. Réalisant ce qu'ils pourraient faire de toute cette puissance, de nombreuses compagnies ont ensuite commencé à montrer un

---

<sup>1</sup> *John McCarthy* (né le 4 septembre 1927, à *Boston, Massachusetts*) est le principal pionnier de l'intelligence artificielle. Il est également l'inventeur en 1958 du langage Lisp. A la fin des années 1950, il a créé avec *Fernando Cobarto* la technique du temps partagé, qui permet à plusieurs utilisateurs d'employer simultanément un même ordinateur.

certain intérêt, puis à échanger leurs anciennes infrastructures et applications internes contre ce que l'on appelle : les pay per-use service (services payés à l'utilisation) [1]. C'est la cinquième génération de l'informatique après les Mainframes, les PCs, les Clients/Serveurs et le Web [2].

Il existe dans la littérature plusieurs définitions du Cloud Computing. Le Cloud Computing traduit de l'anglais « informatique dans les nuages » est devenu un mot à la mode populaire, littéralement l'informatique dans les nuages est un concept qui consiste à déporter sur des serveurs distants des stockages et des traitements informatiques traditionnellement localisés sur des serveurs locaux ou sur le poste de l'utilisateur. Il consiste à proposer des services informatiques sous forme de service à la demande, accessibles de n'importe où, n'importe quand et par n'importe qui, grâce à un système d'identification, via un PC et une connexion à Internet. Cette définition est loin d'être simple à comprendre, toute fois l'idée principale à retenir est que le Cloud n'est pas un ensemble de technologies, mais un modèle de fourniture, de gestion et de consommation de services et de ressources informatiques [3].

Pour NIST <sup>2</sup> [4], le Cloud Computing est un modèle permettant un accès réseau pratique et à la demande à un pool partagé de ressources informatiques fiables (*par exemple : des réseaux, serveurs, stockage, applications, services*) pouvant être rapidement provisionnés et libérés avec un minimum d'effort de gestion du consommateur ou d'interaction avec le fournisseur de services.

Selon Cisco<sup>3</sup> : « Le Cloud Computing est une plateforme de mutualisation informatique fournissant aux entreprises des services à la demande avec l'illusion d'une infinité de ressources ».

Les différentes définitions affirment que le nuage est un ensemble de matériels, de raccordements réseau et de logiciels qui fournissent des services sophistiqués que les utilisateurs peuvent exploiter à volonté depuis n'importe où dans le monde.

---

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

<sup>3</sup> www.cisco.com

De ce fait, le Cloud Computing est un basculement de tendance : au lieu d'obtenir de la puissance de calcul par acquisition de matériel et de logiciel, le consommateur se sert de puissance mise à sa disposition par un fournisseur via internet. Ainsi, le Cloud Computing est une nouvelle façon de délivrer les ressources informatiques, et non une nouvelle technologie, il peut également être vu comme un réseau informatique distribué, dont les ressources peuvent être approvisionnées de manière dynamique et reposant sur un contrat de service entre un fournisseur et un client.

### I.3 Éléments constitutifs du Cloud Computing

Un environnement Cloud Computing ne peut être défini sans un ensemble d'éléments qui sont :

#### I.3.1 La virtualisation

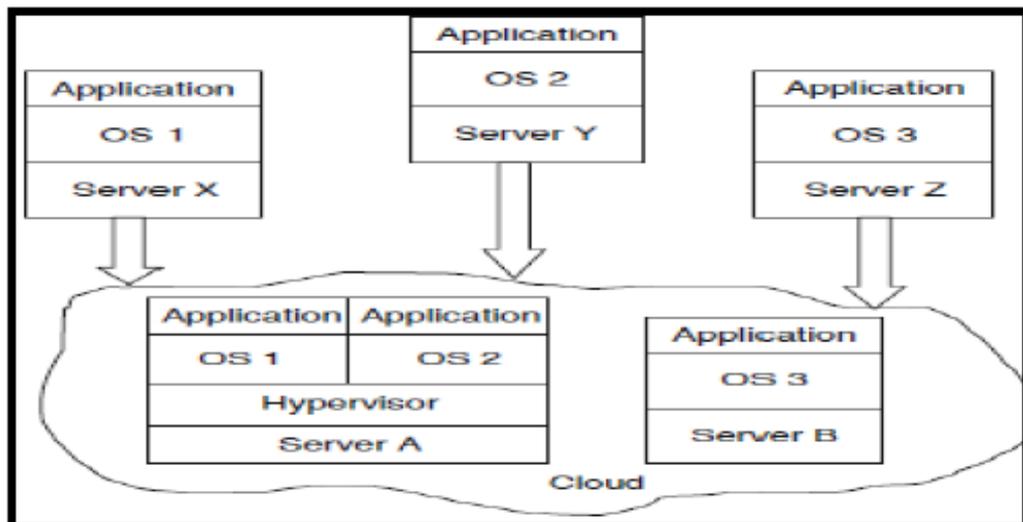


Figure 1: Exemple de virtualisation [5].

La virtualisation se définit comme l'ensemble des techniques matérielles et/ou logicielles qui permettent de faire fonctionner sur une seule machine, plusieurs systèmes d'exploitation, appelées machines virtuelles (VM), ou encore OS invitée [6].

La virtualisation des serveurs (Figure 1) permet d'avoir une plus grande modularité dans la répartition des charges et la reconfiguration des serveurs en cas d'évolution ou de défaillance momentanée.

### **I.3.2 Centre de Calcul (Datacenter)**



*Figure 2: Datacenter [11].*

Un centre de traitement de données (*data center en anglais*) est un site physique (Figure 2), sur lequel se trouvent regroupés des équipements constituant le système d'information de l'entreprise (*mainframes, serveurs, baies de stockage, équipements réseaux et de télécommunications, etc.*). Il peut être interne et/ou externe à l'entreprise, exploité ou non avec le soutien de prestataires, il comprend en général un contrôle sur l'environnement (*climatisation, système de prévention contre l'incendie, etc.*), une alimentation d'urgence et redondante, ainsi qu'une sécurité physique élevée. Cette infrastructure peut être propre à une entreprise et utilisée par elle seule ou à des fins commerciales. Ainsi, des particuliers ou des entreprises peuvent venir y stocker leurs données suivant des modalités bien définies [6].

### **I.3.3 Plateforme collaborative**

Une plate-forme de travail collaborative est un espace de travail virtuel. C'est un site qui centralise tous les outils liés à la conduite d'un projet et les met à disposition des acteurs (Figure 3). L'objectif du travail collaboratif est de faciliter et d'optimiser la communication entre les individus dans le cadre du travail ou d'une tâche.

Les plates-formes collaboratives intègrent généralement les éléments suivants : des outils informatiques ; des guides ou méthodes de travail en groupe pour améliorer la communication, la production et la coordination ; un service de messagerie ; un système de partage des ressources et des fichiers ; des outils de type forum ou pages de discussions ; un trombinoscope ou annuaire des profils des utilisateurs, des groupes, par projet ou par thématique et un calendrier [6].



*Figure 3: Plateforme collaborative [6].*

#### **I.4 Caractéristiques du Cloud Computing**

Généralement, un service, une solution ou un environnement d'exécution devrait satisfaire une liste de caractéristiques pour qu'il soit considéré comme étant du Cloud Computing. Parmi ces caractéristiques, il y a celles qui sont reconnues comme fondamentales. Par exemple, le NIST [4] définit cinq caractéristiques essentielles qui sont :

1. **Un service à la demande** : Un utilisateur peut allouer des ressources informatiques (serveurs, réseau, stockage, environnement d'exécution, application) au

besoin, de façon automatique et sans nécessité d'interaction humaine avec chaque fournisseur de service.

2. **Un accès aux ressources par le réseau** : Les fonctionnalités sont disponibles sur le réseau et accessibles via des mécanismes standards qui favorisent leurs utilisations à partir des appareils clients hétérogènes (*par exemple : téléphones mobiles, tablettes, ordinateurs portables et stations de travail*).
3. **Mise en commun des ressources** : Les ressources informatiques du fournisseur sont regroupées pour servir plusieurs consommateurs à l'aide d'un modèle multi-locataire. Différentes ressources physiques et virtuelles sont allouées et libérées dynamiquement selon la demande des consommateurs. Généralement, l'utilisateur n'a ni le contrôle ni la connaissance de l'emplacement exact des ressources allouées. Dans certains cas, il peut choisir l'emplacement géographique à un niveau haut (*par exemple : par pays, continent ou datacenter*). Les exemples de ressources incluent le stockage, le traitement, la mémoire et la bande passante du réseau.
4. **Flexibilité des ressources** : Les ressources sont allouées et libérées d'une façon élastique, dans certains cas automatiquement pour s'adapter rapidement à la demande qu'elle soit croissante ou décroissante. Pour le consommateur, les fonctionnalités disponibles pour l'allocation apparaissent comme illimitées et peuvent être adaptées à n'importe quelle quantité et à tout moment.
5. **Un service mesuré** : Les systèmes en nuage contrôlent et optimisent automatiquement l'utilisation des ressources afin de mesurer leurs consommations avec un niveau d'abstraction approprié selon le type du service (*par exemple : stockage, temps de calcul et bande passante*). L'utilisation des ressources peut être surveillée offrant ainsi une transparence à la fois au fournisseur et au consommateur du service utilisé.

### I.5 Types de services dans les environnements de Cloud Computing

La partie visible du Cloud Computing se décompose en 3 parties essentielles qui sont (Figure 4) :

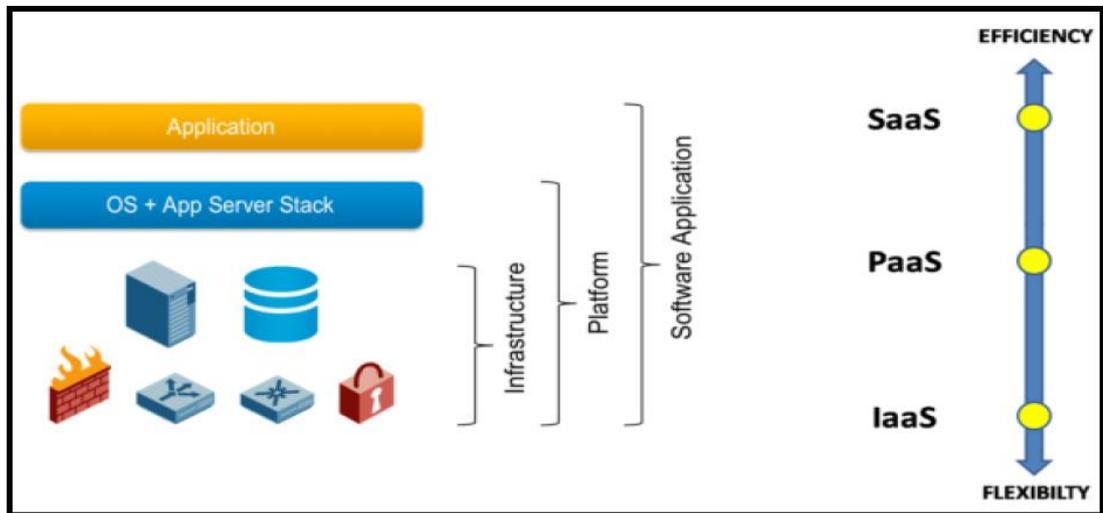


Figure 4: Types de services Clouds [12].

#### I.5.1 SaaS (Software as a Service)

Concept consistant à proposer un abonnement à un logiciel plutôt que l'achat d'une licence. On oublie donc le modèle client-serveur et aucune application n'est installée sur l'ordinateur, elles sont directement utilisables via le navigateur Web. L'utilisation reste transparente pour les utilisateurs, qui ne se soucient ni de la plateforme, ni du matériel, qui sont mutualisés avec d'autres entreprises. Le SaaS remplace l'ASP (Application Service Provider), aussi appelé fournisseur d'applications hébergées (FAH), qui est une entreprise qui fournit des logiciels ou des services informatiques à ses clients à travers un réseau.

Il s'agit de la mise à disposition d'applications sous forme de service (*CRM*<sup>4</sup>, *outils collaboratifs*, *messaging*, *ERP*<sup>5</sup>, ...). Ce concept consiste à proposer un abonnement à un logiciel plutôt que l'achat d'une licence.

<sup>4</sup> La gestion de la relation client (GRC), ou gestion des relations avec les clients, en anglais Customer Relationship Management (CRM), est l'ensemble des outils et techniques destinés à capter, traiter, analyser les informations relatives aux clients, dans le but de les fidéliser en leur offrant le meilleur service.

<sup>5</sup> Enterprise Resource Planning, signifiant littéralement en anglais, « planification des ressources de l'entreprise », et traduit en français par « progiciel de gestion intégré » (PGI).

De ce fait, pas d'installation, pas de mises à jour à gérer (elles sont gérées par le fournisseur) et le paiement est à l'usage.

### I.5.2 PaaS (Platform as a Service)

Il s'agit des plateformes du nuage, regroupant principalement les serveurs mutualisés et leurs systèmes d'exploitation. En plus de pouvoir délivrer des logiciels en mode SaaS, le PaaS dispose d'environnements spécialisés au développement comprenant les langages, les outils et les modules nécessaires.

### I.5.3 IaaS (Infrastructure as a Service)

Il s'agit de la mise à disposition, à la demande, de ressources d'infrastructures (*serveurs, moyens de stockage, réseau ...*) dont la plus grande partie est localisée à distance dans des Datacenter. L'IaaS permet l'accès aux serveurs et à leurs configurations pour les administrateurs de l'entreprise. Le client a la possibilité de louer des clusters, de la mémoire ou du stockage de données. Le coût est directement lié au taux d'occupation. Une analogie peut être faite avec le mode d'utilisation des industries des commodités (*électricité, eau, gaz*) ou des télécommunications [7].

### I.5.4 Autres types de services Clouds

Bien que les trois types de services de Cloud Computing (IaaS, PaaS et SaaS) soient la base de distinction du type de service, le style de nomenclature « as a Service » a été utilisé ailleurs pour caractériser des services et ressources Cloud Computing. Chacune des nouvelles abréviations peut-être vues comme un sous-ensemble d'un ou plusieurs des trois types de base. Ainsi, on parle aujourd'hui de XaaS (Anything as a Service). Parmi les abréviations existantes, nous citons [8]:

- **DBaaS (Database as a Service)** : Base de données en tant que service est un modèle de service d'informatique en nuage qui fournit aux utilisateurs un accès à une base de données, sans qu'il soit nécessaire de configurer du matériel physique, d'installer un logiciel ou de réaliser des performances. Le fournisseur de services s'occupe de toutes les tâches administratives et de la maintenance, de

sorte que tout ce que l'utilisateur ou le propriétaire de l'application doit faire est d'utiliser la base de données.

- **NaaS (Network as a Service) :** Un modèle de Cloud Computing dans lequel les locataires ont accès à des ressources informatiques supplémentaires co-localisées avec des commutateurs et des routeurs. C'est un modèle pour la fourniture virtuelle de services réseau, par le biais d'un modèle de service par abonnement ou «pay as you use». Grâce à NaaS, tout ce qui est requis du client est un ordinateur avec une connexion internet connectée au portail NaaS mis à disposition par l'intermédiaire d'un fournisseur NaaS / Cloud. NaaS simplifie l'architecture de réseau grâce à la virtualisation. Cisco, Aryaka et BigSwitch sont quelques-uns des fournisseurs bien connus de réseau en tant que service.

Le tableau 1 suivant donne des exemples d'offres de services Clouds existants sur le marché des leaders mondiaux.

IaaS	PaaS	SaaS
-Amazon offres EC2 et AWS	-Microsoft offre Azur	- Google offre Google Apps (messagerie et bureautique)
-Microsoft offre Azur	- Google offre Google App Engine	- Sales Force CRM (Customer Relationship Management)

Tableau 1 : Les grands acteurs mondiaux du Cloud Computing [12].

### I.6 Modèles de déploiement du Cloud Computing

Quel que soit le modèle de service utilisé (SaaS, PaaS ou IaaS), le NIST [4] a défini quatre modèles de déploiement pour les services de Cloud, avec des variantes qui répondent à des besoins spécifiques.

### I.6.1 Cloud public

Un Cloud public est basé sur un modèle standard de Cloud Computing, dans lequel un prestataire de service met les ressources, telles que les applications ou le stockage, à la disposition du grand public via internet (Figure 5).

Le Cloud public peut être gratuit ou fonctionner selon un paiement à la consommation. L'avantage de ce genre d'architecture est d'être facile à mettre en place, pour des coûts relativement raisonnables. La charge du matériel, des applicatifs, de la bande passante étant couverte par le fournisseur, ce modèle permet de proposer une souplesse et une évolutivité accrue afin de répondre rapidement au besoin. Il n'y a pas de gaspillage de ressources car le client ne paye que ce qu'il consomme [13].

Bien que le Cloud public offre un maximum de flexibilité, il demande de lourds investissements pour le fournisseur de services. De plus, ce type de Cloud n'est pas sécurisé.

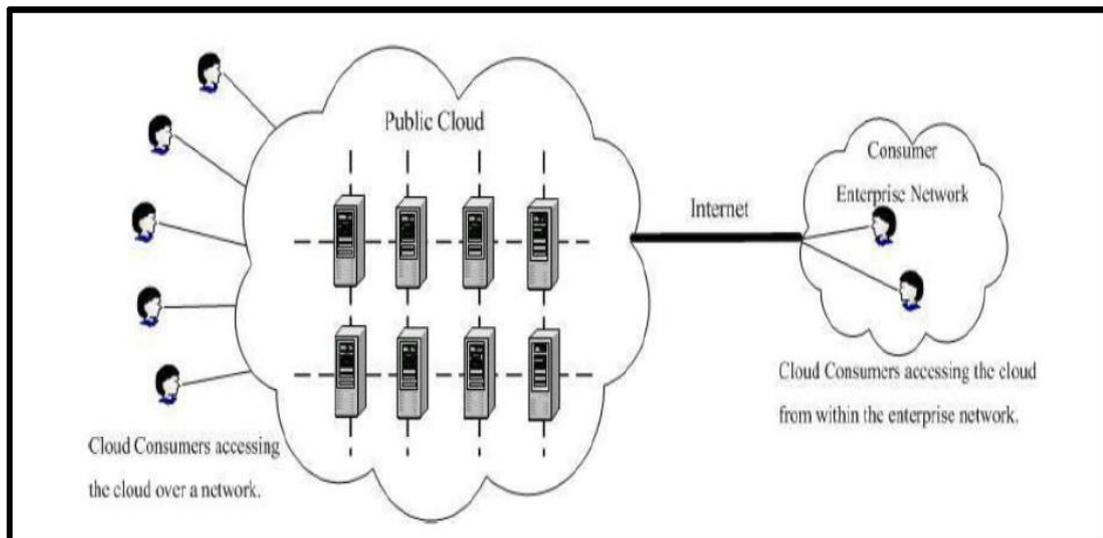


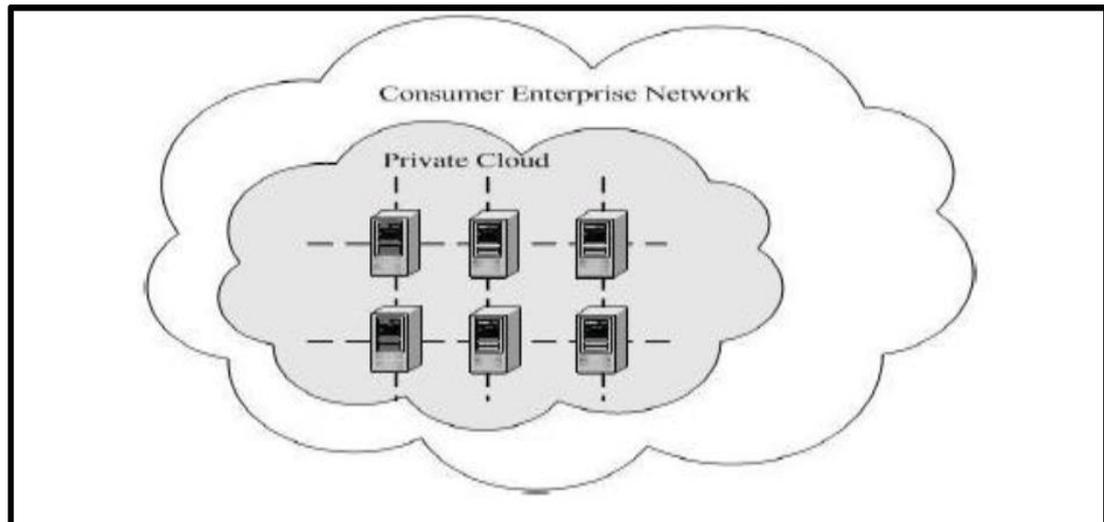
Figure 5 : Cloud public [9].

### I.6.2 Cloud privé

L'infrastructure en nuage est configurée pour une utilisation exclusive par une seule organisation comprenant plusieurs consommateurs (*par exemple, des unités*

*commerciales*) (Figure 6). Il peut être possédé, géré et exploité par l'organisation, un tiers, ou une combinaison de ceux-ci, et il peut exister sur site ou hors site [4].

L'avantage majeur du Cloud privée est qu'il est dédié et sécurisé. Cependant, il est moins flexible comparé au Cloud public et plus cher.



*Figure 6: Cloud Privé [9].*

### **I.6.3 Cloud hybride**

L'infrastructure de Cloud dans un Cloud hybride est une composition d'au moins de deux Clouds (*privé, communautaire ou public*) (Figure 7). Chacune d'entre elles reste une entité unique mais est reliée par une technologie standard ou exclusive. Cette technologie permet la portabilité des données et des applications. Les avantages des nuages hybrides sont l'évolutivité, la flexibilité et la rentabilité. Les inconvénients sont les problèmes de réseau et les règles de sécurité [7].

Avec un Cloud hybride, nous pouvons déporter les applications vers un Cloud public qui consommera des données stockées et exposées dans un Cloud privé, ou bien faire communiquer deux applications hébergées dans deux Clouds privés distincts, ou encore consommer plusieurs services hébergés dans des Clouds publics différents.

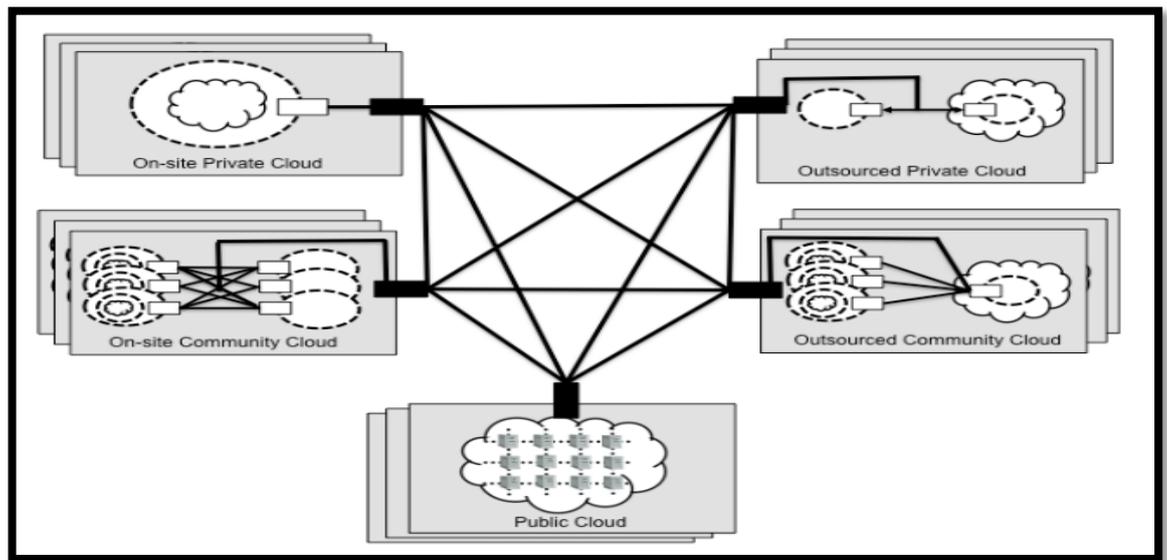


Figure 7: Exemple de Cloud hybride [7].

#### I.6.4 Cloud communautaire

L'infrastructure de Cloud Computing est configurée pour une utilisation exclusive par une communauté de consommateurs ou des organisations qui ont des préoccupations communes (par exemple : la mission et les exigences en matière de sécurité). Il peut être administré, géré et exploité par un ou plusieurs des organismes de la communauté, un tiers, ou une combinaison des deux, et il peut exister sur ou à l'extérieur des locaux [10].

#### I.7 Architecture de référence du Cloud Computing

La figure 8 présente un aperçu de l'architecture de référence du Cloud Computing de NIST [17], qui identifie les acteurs majeurs, leurs activités et leurs fonctions dans le Cloud Computing. Le diagramme représente une architecture générique de haut niveau et est destiné à faciliter la compréhension des exigences, des utilisations, des caractéristiques et des normes de l'informatique en nuage.

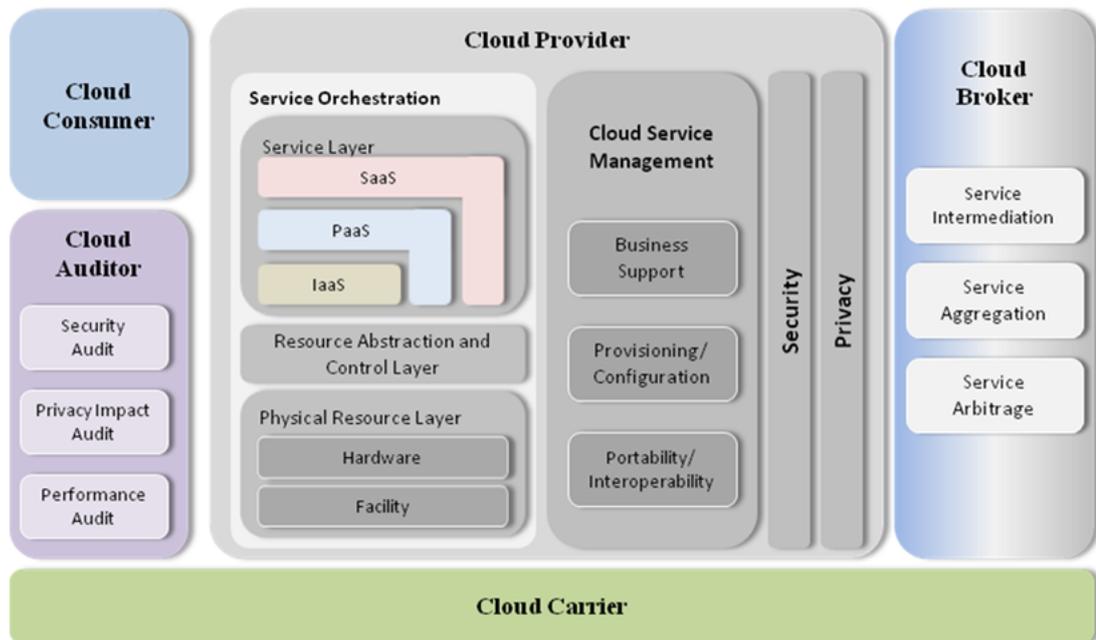


Figure 8 : Le modèle conceptuel de référence du Cloud Computing [17]

Comme le montre la figure 8, l'architecture de référence du Cloud Computing NIST définit cinq acteurs majeurs : le consommateur Cloud, le fournisseur de Cloud, le Cloud Carrier, le vérificateur Cloud et le Cloud Broker. Chaque acteur est une entité (une personne ou une organisation) qui participe à une transaction ou un processus et/ou effectue des tâches dans le Cloud Computing. Le tableau 2 énumère brièvement les acteurs définis dans l'architecture de référence du Cloud Computing NIST [17].

Acteurs	Role
<b>Cloud Consumer</b>	Une personne ou une organisation qui entretient une relation commerciale avec les fournisseurs de Cloud et utilise le service de ceux-ci.
<b>Cloud Provider</b>	Personne, organisation ou entité chargée de mettre un service à la disposition des parties intéressées.

<b>Cloud Auditor</b>	Une entité qui peut effectuer une évaluation indépendante des services Cloud, des opérations du système d'information, des performances et de la sécurité de la mise en œuvre du nuage.
<b>Cloud Broker</b>	Une entité qui gère l'utilisation, les performances et la fourniture de services de Cloud Computing, et négocie les relations entre les fournisseurs et les consommateurs de Cloud.
<b>Cloud Carrier</b>	Un intermédiaire qui assure la connectivité et le transport des services Clouds des fournisseurs aux consommateurs.

*Tableau 2 Acteurs du Cloud Computing [17].*

### **I.7.1 Acteurs du Cloud Computing**

Des acteurs importants jouent un rôle dans l'architecture de référence du Cloud Computing de NIST, notamment en tant que consommateur, courtier (broker) et fournisseur de Cloud. Chaque acteur est considéré comme une entité (une personne ou une organisation) pouvant participer à un processus ou à une transaction et / ou effectuer des tâches dans le Cloud Computing. Dans ce qui suit, nous allons donner plus de détails sur ces acteurs en reprenant les explications de [18].

#### **I.7.1.1 Consommateur Cloud 'Cloud Consumer'**

Le consommateur de services Cloud est le principal acteur dans le Cloud Computing. Un consommateur représente une personne ou une organisation qui entretient une relation commerciale avec un fournisseur de nuage en utilisant ses services. Il parcourt le catalogue de services à partir d'un fournisseur de nuage, demande le service approprié, établit des contrats de service avec le fournisseur de nuage et utilise le service. Le client en nuage peut être facturé pour le service fourni et doit assurer les paiements en conséquence.

Les consommateurs de services Cloud ont besoin de contrats de niveau de service pour spécifier les exigences de performances techniques remplies par un fournisseur de Cloud. Les accords de niveau de service peuvent couvrir des conditions relatives à la qualité de service, à la sécurité et aux solutions en cas de défaillance des performances. Un fournisseur de Cloud peut également inclure dans les contrats de

niveau de service un ensemble de promesses non explicitement faites aux consommateurs, c'est-à-dire des limitations et des obligations que les consommateurs de Cloud doivent accepter. La figure 9 donne un exemple de services disponibles pour un consommateur de Cloud.

Un consommateur de Cloud peut choisir librement un fournisseur de Cloud offrant une meilleure tarification et des conditions plus favorables. En règle générale, la politique de prix et les accords de niveau de service d'un fournisseur de Cloud ne sont pas négociables, sauf si le client désire une utilisation intensive, dans ce cas, il peut éventuellement être en mesure de négocier de meilleurs contrats [17].

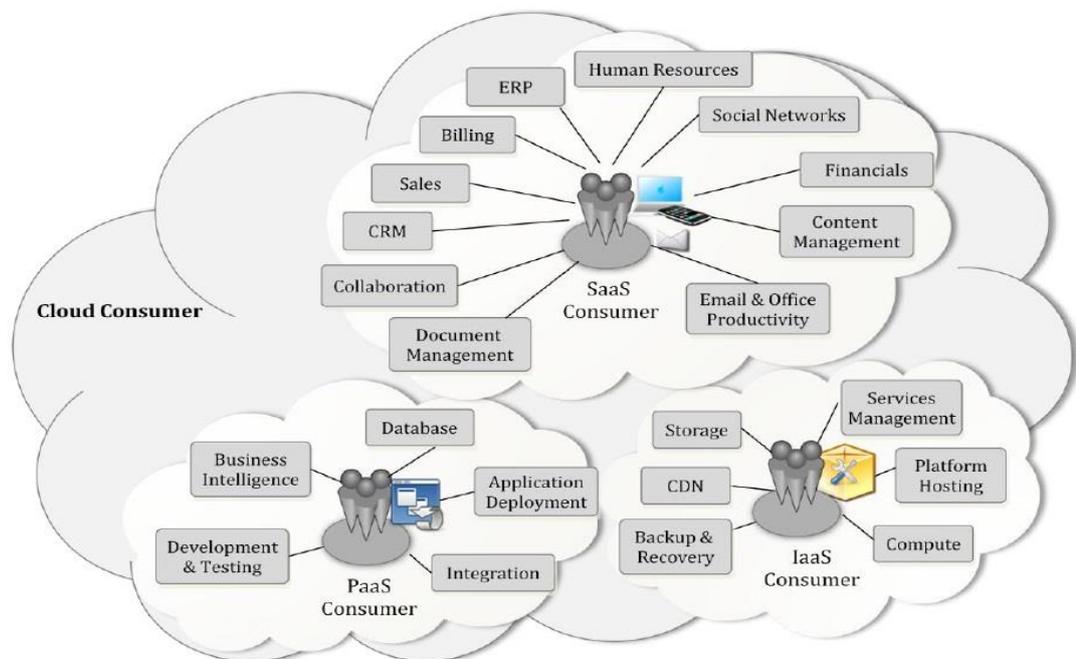


Figure 9 : Exemple de services disponibles pour un consommateur de Cloud [17]

### I.7.1.2 Fournisseur Cloud 'Cloud Provider'

Un fournisseur de Cloud est une personne ou une organisation, c'est l'entité chargée de mettre un service à la disposition des parties intéressées. Un fournisseur de Cloud acquiert et gère l'infrastructure informatique requise pour fournir les services, exécute le logiciel Cloud qui fournit les services et effectue des arrangements pour fournir les services Cloud aux consommateurs Cloud via l'accès réseau.

Pour le Software as a Service ‘SaaS’, le fournisseur de Cloud déploie, configure, maintient et met à jour le fonctionnement des applications logicielles sur une infrastructure Cloud afin que les services soient fournis aux niveaux de service attendus pour les consommateurs Cloud. Le fournisseur de SaaS assume la plupart des responsabilités dans la gestion et le contrôle des applications et de l’infrastructure, tandis que les consommateurs de Cloud ont un contrôle administratif limité des applications [17].

Il existe actuellement une multitude de fournisseurs Clouds qui délivrent des services de différents types (IaaS, PaaS et SaaS) et pour des domaines distincts. Parmi les fournisseurs les plus actifs sur le marché nous citons :

- Amazon ([www.amazon.com](http://www.amazon.com))
- Google
- Microsoft
- IBM
- Zoho ([www.zoho.com](http://www.zoho.com))
- Rackspace ([www.rackspace.com](http://www.rackspace.com))
- Citrix ([www.citrix.com](http://www.citrix.com))
- Intel
- Sales force ([www.salesforce.com](http://www.salesforce.com))

### **I.7.1.3 Courtier en nuage ‘Cloud Broker’**

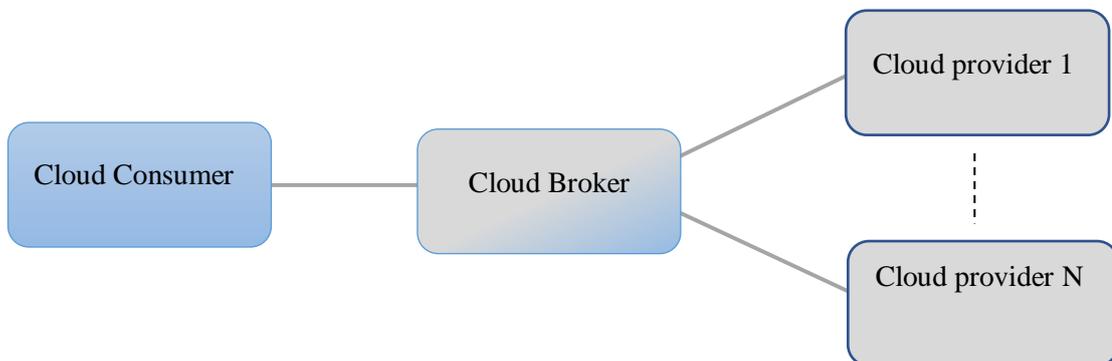
Un courtier en nuage est utilisé comme technique de médiation pour permettre une interaction entre différents systèmes en nuage qui ne sont pas équipés pour interagir les uns avec les autres. Cette technique peut réduire la nécessité de modifier les systèmes en nuage actuels [19].

L’institut national de normalisation et de technologie (NIST) définit un courtier en nuage comme « une entité qui gère l’utilisation, les performances et la fourniture de services en nuage et négocie les relations entre les fournisseurs de nuage et les consommateurs de nuage » [20].

L'Organisation Internationale de Normalisation (ISO) <sup>6</sup> a défini le courtier de services dans le nuage comme « un partenaire de services dans le nuage qui négocie les relations entre les clients des services et les fournisseurs de services dans le nuage ». Nous expliquerons l'entité Broker et son travail plus en détails dans le chapitre suivant.

#### **I.7.1.4 Relations entre les acteurs du Cloud Computing**

Les relations qui existent entre les acteurs du Cloud Computing sont illustrées dans la figure 10. Un consommateur de Cloud peut demander des services à un fournisseur de Cloud de deux manières. Soit une demande directe auprès d'un fournisseur de Cloud ou bien la négociation avec un courtier en Cloud. Un nouveau service pourrait être offert lorsqu'un courtier en Cloud est demandé. Un courtier peut offrir de nouveaux services en intégrant différents services ou en améliorant les services disponibles. Comme illustré à la figure 10, un consommateur de Cloud ne connaît pas le fournisseur de Cloud en question et doit contacter directement le courtier en Cloud [18].



*Figure 10 : Relations entre les acteurs du Cloud Computing [18].*

---

<sup>6</sup> ISO est une organisation indépendante et non gouvernementale dont les membres sont les organisations de normalisation des 164 pays membres. C'est le plus grand concepteur mondial de normes internationales. Il facilite le commerce mondial en fournissant des normes communes à tous les pays.

### **I.8 Avantages et inconvénients du Cloud Computing**

Le marché des services Clouds connaît une croissance fulgurante ces dernières années, prouvant ainsi qu'il ne s'agit pas juste d'un boom médiatique mais une vraie technologie dans le domaine des applications distribuées et aussi pour la délivrance de services. Cela est dû principalement à sa politique d'utilisation qui offre de nombreux avantages. Toutefois, l'adoption du Cloud Computing se heurte à plusieurs obstacles. Les avantages et les inconvénients du Cloud Computing diffèrent selon le type du service délivré. Le tableau 3 donne les principaux avantages et inconvénients selon le type du service Cloud.

	<b>Avantage</b>	<b>inconvénient</b>
<b>SaaS</b>	-pas d'installation -plus de licence -migration	-logiciel limité -sécurité -dépendance des prestataire
<b>Paas</b>	-pas d'infrastructure nécessaire -pas d'installation -environnement hétérogène	-limitation des langages -pas de personnalisation dans la configuration des machines virtuelles
<b>IaaS</b>	-administration -personnalisation -flexibilité d'utilisation	-sécurité -besoin d'un administrateur système

*Tableau 3 : Avantages et inconvénients des services Cloud [3]*

Les obstacles qui s'opposent à l'adoption du Cloud Computing (Figure 11) constituent actuellement de vraies pistes de recherche qui suscitent l'intérêt des chercheurs et des industriels. Parmi les problèmes majeurs du Cloud Computing, nous citons :

- **Interopérabilité** : L'un des principaux obstacles à l'adoption du Cloud Computing est l'interopérabilité. Il s'agit de la capacité des ressources d'un fournisseur de Cloud à communiquer avec des ressources d'un autre fournisseur de Cloud.
- **Latence** : Un réseau (ou internet si des Clouds publics doivent être fournis) rend les ressources Cloud accessibles. Cependant, il crée une latence (temps de retard) dans n'importe quel type de connexion entre les utilisateurs et l'environnement.
- **Sécurité** : La confidentialité des données inquiète les utilisateurs du Cloud. La plupart du temps, les responsables de l'organisation qui utilise les services ne sont pas parfaitement au courant des emplacements des fournisseurs de Cloud Computing qui sauvegardent leurs données [18].

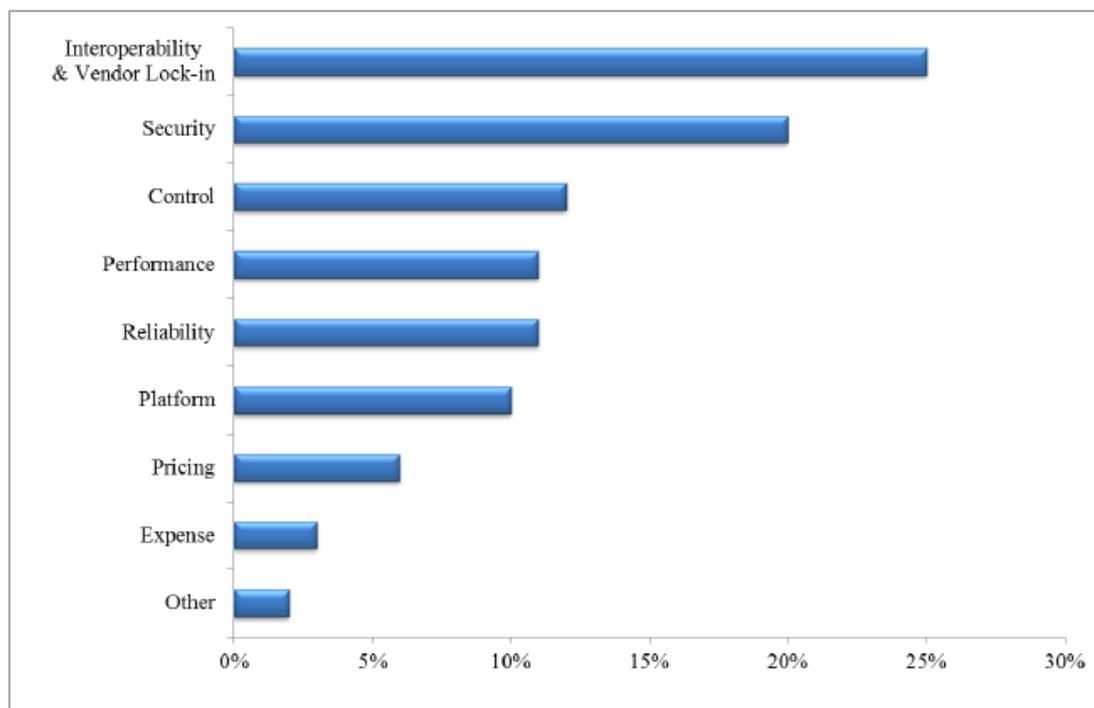


Figure 11 : Obstacles à l'adoption du Cloud Computing [18].

## **I.9 Evolution du Cloud Computing**

L'adoption récente et massive des services Cloud a augmenté le nombre des demandes des clients qui cherchent des services offrant toujours plus de fonctionnalités. Les demandes deviennent de plus en plus complexes nécessitant ainsi la collaboration de plusieurs Clouds, qui peuvent être même de différents fournisseurs, pour satisfaire une demande d'un client. Cette évolution a donné naissance à de nouveaux domaines tels que le multi-Cloud et l'inter-Cloud.

Pour bien comprendre les notions de multi-Cloud et d'inter-Cloud, il faut d'abord comprendre les notions d'interopérabilité et de portabilité dans le Cloud Computing.

- **L'interopérabilité :** Pour n'importe quel système informatique, l'interopérabilité est la capacité de deux ou plusieurs systèmes ou applications à échanger des informations et à utiliser mutuellement les informations échangées. Dans le Cloud Computing, l'interopérabilité est la capacité du système d'un client à interagir avec un service en nuage ou la capacité d'un service en nuage à interagir avec d'autres services en nuage en échangeant des informations selon une méthode prescrite pour obtenir des résultats prévisibles [14].
- **La portabilité :** Consiste à déplacer les données et / ou les applications d'un système à un autre et à les garder utilisables ou exécutables. La portabilité des données en nuage est la capacité de déplacer facilement des données d'un service en nuage à un autre sans avoir à ressaisir les données. La portabilité des applications en nuage correspond à la possibilité de migrer une application d'un service en nuage à un autre ou entre l'environnement du client et un service en nuage.

### **I.9.1 Inter-Cloud**

L'inter-Cloud est le concept de réseaux de Clouds connectés, comprenant des Clouds publics, privés et hybrides. Il intègre un certain nombre d'activités technologiques regroupées pour améliorer l'interopérabilité et la portabilité entre les réseaux en nuage [15].

La notion d'inter-Cloud est encore ambiguë, les chercheurs divergent sur les définitions de base. Une stratégie inter-Cloud nécessite un accord préalable entre les différents fournisseurs Cloud contrairement à une stratégie multi-Cloud.

### **I.9.2 Multi-Cloud**

Le multi-Cloud consiste à utiliser plusieurs services de stockage et d'informatique en nuage dans une architecture réseau unique. Cela fait référence à la distribution des logiciels, applications et autres services dans plusieurs environnements Cloud. Avec une architecture multi-Cloud typique utilisant deux ou plusieurs Clouds publics ainsi que des Clouds privés, un environnement multi-Cloud vise à éliminer la dépendance vis-à-vis d'un fournisseur ou d'une instance de Cloud unique [16].

### **I.10 Conclusion**

Le Cloud Computing est considéré comme la nouvelle révolution de l'informatique à travers l'internet. Il offre des avantages pour les organisations et les individus. Dans ce chapitre, nous avons passé en revue les notions de base relatives à ce concept. Nous avons parlé aussi de l'évolution du Cloud Computing qui dépend de l'évolution du multi-Cloud et de l'inter-Cloud.

Dans le chapitre suivant, nous allons présenter et discuter le contexte de notre travail qui est le courtage en nuage avec ses concepts de base et les solutions existantes.

## **II.1 Introduction**

En raison de l'émergence rapide du marché de l'informatique en nuage au cours des dernières années, le nombre de fournisseurs de services en nuage a considérablement augmenté. Par ailleurs, les problèmes de « verrouillage du fournisseur » et l'absence de normes communes sur le Cloud entravent l'interopérabilité entre ces fournisseurs, ce qui constitue l'un des obstacles à la croissance du Cloud Computing. Ainsi, le client du Cloud se heurte aujourd'hui à un problème épineux de sélection des offres de Cloud adaptées à ses besoins. Par conséquent, des interfaces standardisées et des services intermédiaires sont nécessaires pour éviter les monopoles de fournisseurs de Cloud uniques.

L'un des cas d'utilisation prometteurs de la vision Inter-Cloud est la possibilité d'effectuer des transactions sur le marché via des courtiers. Dans un tel cas d'utilisation, une entité de courtier agit en tant que médiateur entre le consommateur de Cloud et plusieurs fournisseurs de services Cloud, afin d'aider le client à sélectionner le fournisseur qui répond le mieux à ses besoins. De plus, un courtier permet de faciliter le déploiement et la gestion du service pour l'utilisateur, quel que soit le fournisseur sélectionné.

Frank Kenney, ancien directeur de recherche chez Gartner, a expliqué le besoin de courtiers en Cloud, affirmant : "L'avenir du Cloud Computing sera imprégné de la notion de courtiers négociant des relations entre les fournisseurs de services de Cloud et les clients du service" [106].

Ce chapitre vise à faire une présentation des notions clés permettant de définir et comprendre le contexte général de notre travail à savoir le courtage en nuage. Le chapitre est structuré en deux parties. La première partie décrit les concepts de base relatifs aux brokers dans le Cloud. Elle expose aussi les problématiques de description, de découverte, de sélection et de composition des services Clouds. La deuxième partie présente les travaux connexes avec des synthèses et des études comparatives.

## II.2 Cloud broker entre SaaS

SaaS est le niveau sur lequel nous allons travailler, pour cela nous allons expliquer les concepts de courtage dans le Cloud en général et nous nous focaliserons sur le SaaS en particulier.

### II.2.1 Courtage de services en nuage

Le courtier en nuage est l'orchestrateur de l'interopérabilité, il assure les interactions entre les différents intervenants. Un logiciel de nuage en tant que consommateur de services peut demander un logiciel de nuage en tant que système de services SaaS à un courtier en nuage, au lieu de contacter directement un logiciel de nuage en tant que fournisseur de services (Figure 12).

Un courtier en nuage est une entité qui gère l'utilisation, les performances et la fourniture de services en nuage et négocie les relations entre un logiciel en nuage en tant que fournisseur de services et un logiciel en nuage en tant que consommateur de services [18].

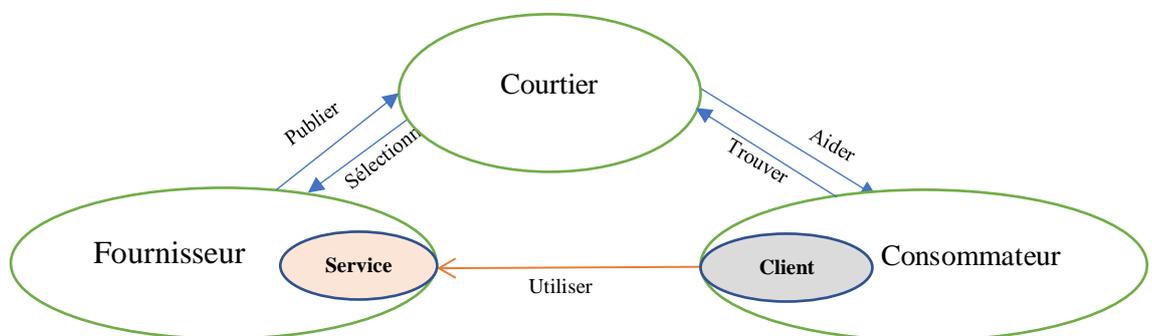


Figure 122 : Relations entre le consommateur, le courtier et le fournisseur [18].

En général, un courtier en nuage peut fournir des logiciels en tant que services selon des catégories. NIST a classé les services fournis par un courtier en Cloud dans trois catégories à savoir l'arbitrage, l'agrégation et l'intermédiation [20].

➤ **L'arbitrage** : améliore les fonctionnalités des services Cloud en offrant une flexibilité dans la sélection des services.

- **L'agrégation** : regroupe plusieurs services dans un seul service ou dans de nouveaux services afin d'améliorer les capacités du courtier.
- **Intermédiation** : Le broker joue le rôle d'intermédiaire entre les différents bénéficiaires du Cloud.

Gartner<sup>7</sup> a classé les services de courtage dans le Cloud en trois catégories : l'agrégation, l'intégration et la personnalisation [20].

- ❖ **L'agrégation** : consiste à fournir deux services ou plus à éventuellement de nombreux consommateurs, sans nécessairement fournir de nouvelles fonctionnalités, intégration ou personnalisation, mais en offrant généralement une gestion centralisée des SLA<sup>8</sup> et de la sécurité.
- ❖ **L'intégration** : faire fonctionner un ensemble de services indépendants comme une offre combinée. Cela pourrait être un interfonctionnement entre les couches de la pile de nuages verticale, ou pourrait impliquer l'intégration de données / processus au sein d'une seule couche. Des techniques telles que la transformation, la médiation et l'orchestration sont les solutions classiques [21].
- ❖ **La personnalisation** : consiste à modifier ou à ajouter des fonctionnalités afin d'améliorer la fonctionnalité de service et de fournir une analyse de service améliorée [21]. Elle personnalise les services de différents fournisseurs par composition ou décomposition [20].

## II.2.2 Rôle du courtier nuageux

Cette section résume le rôle du courtier en nuage dans les différentes classifications de catégories données par les chercheurs [22].

### II.2.2.1 Consultant et négociateur

- Le broker évaluera diverses options de Clouds publics disponibles et choisira le bon fournisseur pour le compte du consommateur de Cloud en fonction de la charge de travail du Cloud public, des exigences de performances et de la durée d'exécution.

---

<sup>7</sup> Gartner est une société mondiale de recherche fournissant des informations, des conseils et des outils aux entreprises dans les domaines de l'informatique.

<sup>8</sup> Service Level Agreement

- Le courtier en nuage devra agir en tant que représentant du consommateur de nuage en négociant les conditions du contrat et les accords de niveau de service avec le fournisseur de nuage.
- Le courtier en nuage sera chargé de choisir la combinaison de services du fournisseur de nuage.
- Le courtier en Cloud servira d'intermédiaire pour faire en sorte que le consommateur en nuage tire les avantages économiques et commerciaux du passage à un nuage public / privé.

### **II.2.2.2 Conseiller en conformité**

- Le courtier en nuage évaluera divers Clouds sur le marché et suggérera le Cloud adapté qui répond aux exigences de conformité du client.
- Le courtier en nuage traduira également les déclarations de conformité en un schéma de mise en œuvre et en des déclarations pouvant être facilement mises en œuvre par le consommateur de Cloud ou tout autre tiers agissant en son nom.

### **II.2.2.3 Cloud orchestrateur hybride**

- Le courtier en nuage fournira des services d'orchestration via une plate-forme de gestion de Cloud auto-développée ou reconnue par l'industrie.
- En tant qu'orchestrateur, le courtier en nuage fournira un volet via lequel un consommateur en nuage peut commander des services auprès de plusieurs fournisseurs de nuage SaaS.
- Le courtier en nuage fournira également une plate-forme utilisant les API publiées des fournisseurs de nuage pour déplacer de manière transparente les charges de travail entre différentes configurations de nuages publics et de nuages privés.

### **II.2.2.4 Comptable et gestionnaire des dépenses**

- Les fournisseurs d'informatique en nuage fournissent divers types de services aux clients, à l'aide d'un modèle basé sur un catalogue. Bien que cela offre une grande souplesse aux consommateurs de Cloud pour choisir uniquement les

services qu'ils veulent, il introduit également des défis supplémentaires pour concilier ce qui a été consommé et ce qui était nécessaire.

- Les courtiers en nuage agissent en tant que comptables pour suivre et gérer les dépenses des consommateurs sur les services en nuage.
- Les courtiers en nuage, dans le cadre de leurs services, auront la possibilité d'examiner et de vérifier les factures mensuelles des fournisseurs de nuage publics.
- Ils seront responsables d'assurer un modèle de consommation optimal et économiquement meilleur pour le consommateur de Cloud. Cela peut impliquer la sélection des meilleurs plans, la négociation périodique des taux unitaires avec les fournisseurs et le rééquilibrage des charges.

### II.2.2.5 Fournisseur de services professionnels

- Dans certains cas, les courtiers en nuage devront également fournir des services professionnels et aider les consommateurs à configurer leurs environnements nuage, en migrant leurs charges de travail sur des nuages de fournisseurs.
- Le courtier en nuage aura la charge de migrer le client d'une configuration sur site vers une configuration en nuage avec le partenaire choisi.
- Cela créera un flux de travail distinct dans l'organisation du courtier en nuage. Un flux qui comprendra des professionnels hautement qualifiés en implémentation du Cloud, indépendants du fabricant du Cloud et des équipementiers.

## II.3 Meilleurs courtiers en nuage de 2019

Selon [116], les meilleurs courtiers en nuage de 2019 sont :

- **AWS Service Broker**<sup>9</sup>: fait partie d'Amazon Web Services. La solution est open source, ce qui permet aux services AWS d'être exposés directement via des applications tierces telles que Red Hat OpenShift<sup>10</sup>. Les avantages de cette solution sont une interface unifiée dans laquelle les utilisateurs peuvent gérer

---

<sup>9</sup> <https://aws.amazon.com/fr/partners/servicebroker/>

<sup>10</sup> <https://www.openshift.com/>

toutes leurs applications et leurs ressources, la possibilité de faire un essai gratuit et le fait qu'elle soit open source. Cependant, elle nécessite un abonnement AWS.

- **IBM Cloud Brokerage Managed Services(CBMS)<sup>11</sup>** : IBM propose sa propre plate-forme Cloud et propose également CBMS pour permettre au client de regrouper sous un même toit les services Cloud de plusieurs fournisseurs. Les avantages de cette solution sont une meilleure gestion informatique, conservation de la conformité et efficacité. Son inconvénient majeur est les frais de démarrage.
- **Cloudmore<sup>12</sup>** : Cloudmore est une société suédoise ayant des bureaux en Estonie, au Royaume-Uni et aux États-Unis. Elle a été fondée en 2007. La solution permet de regrouper tous les services informatiques en nuage sur une plate-forme unifiée. Elle offre une intégration d'API facile avec tarification par répartition et une interface intuitive.
- **Jamcracker Cloud Services Brokerage<sup>13</sup>** : offre aux entreprises la possibilité de gérer divers services Cloud à partir d'une plate-forme unique. Cela apporte la possibilité d'unifier les opérations de Cloud public, privé et hybride et de les livrer à la fois aux différents départements et aux clients. Elle prend en charge plusieurs langues avec une bonne variété d'options.
- **Dell Boomi<sup>14</sup>** : La prise en charge de Boomi pour divers modèles de Cloud et son intégration facile font de lui un outil idéal pour plusieurs entreprises. Cet outil prend en charge plusieurs architectures de Cloud.

### II.4 Les composants essentiels d'un contrat de Cloud

Un contrat dans le Cloud, SLA (Service Level Agreement), est un document entre le fournisseur et le client. Afin de bien comprendre ce concept, dans cette section, nous allons présenter les composants essentiels de ce dernier. Le contenu typique d'un SLA, selon [23], est présenté comme suit :

---

<sup>11</sup> <https://www.ibm.com/us-en/marketplace/cloud-brokerage-solutions>

<sup>12</sup> <https://web.cloudmore.com/>

<sup>13</sup> <https://www.jamcracker.com/>

<sup>14</sup> <https://boomi.com/>

- **Définition de service :** La première chose très importante dans un contrat est de décrire les besoins de l'utilisateur pour fournir un service qui répond à ses besoins. La définition des informations d'un service est une tâche sensible. Pour cela, le contrat doit prendre en charge des spécifications détaillées du service à délivrer.
- **La gestion du rendement :** Un point très important dans un contrat SLA est d'assurer un moyen efficace pour mesurer le rendement. Chaque service doit être mesuré et les résultats doivent être analysés et rapportés. Afin de garantir le niveau de performance du service, le contrat doit être régulièrement examiné par les deux parties (fournisseur et le client).
- **La gestion des problèmes :** Le but de ce point est de minimiser l'impact négatif des incidents et des problèmes. Cela consiste à gérer les problèmes imprévus et les propositions ou le plan à suivre pour les résoudre.
- **Les responsabilités et les obligations du client :** Le client doit connaître qu'il peut prendre des responsabilités sur l'opération de livraison du service, dans lequel il doit faciliter les opérations d'accès, d'installation et de fournir des ressources nécessaires pour les employés du fournisseur.
- **Garanties :** ce point comporte trois sujets : indemnités de qualité du service, les réclamations et les solutions en cas d'infractions.
- **Sécurité :** La sécurité dans n'importe quel domaine représente une caractéristique critique. Chaque client doit fournir un accès physique et logique contrôlé à ses informations. De même, le fournisseur doit respecter les politiques de sécurité du client.
- **Résiliation :** comporte les points suivants :
  - Résiliation à la fin de la période initiale.
  - Résiliation pour plus de commodités.
  - Résiliation pour cause.

### II.5 Les langages de description de service

Dans la littérature, il existe plusieurs travaux proposés pour décrire les services Cloud. Les modèles sont utilisés pour présenter les services de façon compréhensible. Les modèles de description de service sont repartis en deux catégories

qui sont : description à base syntaxique et sémantique [23]. Dans cette section nous allons présenter quelques modèles utilisés pour décrire les services Cloud.

### ❖ WSDL<sup>15</sup>

WSDL (Web Service Description Language) est un fichier XML pour décrire les services et leur lien avec les adresses réseaux spécifiques (Figure 13). Les fichiers WSDL contiennent les définitions et les opérations. Généralement, les définitions exprimées en XML incluent à la fois les définitions de type de données et les définitions de messages qui utilisent les définitions de type de données.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:s0="http://www.myapp.org"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" targetNamespace="http://www.myapp.org"
  xmlns:head="http://www.intersystems.com/SOAPheaders">
  <types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://www.myapp.org">
    <s:element name="GetCustomerInfo">
      <s:complexType>
        <s:sequence>
          <s:element name="ID" type="s:decimal" minOccurs="0" />
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:schema>
  </types>
</definitions>
```

Figure 13 : Exemple d'un fichier WSDL [23].

### ❖ USDL<sup>16</sup>

Le langage de description de service unifié USDL (Unified Service Description Language) est utilisé pour décrire les paramètres commerciaux, opérationnels et techniques des services. Les descriptions de service incluent des informations telles que le prix, le fournisseur de services, les méthodes d'interaction et les accords de niveau de service. La figure 14 donne un exemple d'un fichier USDL.

<sup>15</sup> <https://www.w3.org/TR/2001/NOTE-wsdl-20010315>

<sup>16</sup> <https://www.w3.org/2005/Incubator/usdl/charter>

```
<http://linked-usdl.org/ns/usdl> a owl:Ontology;  
  dc:title "Linked-USDL Core";  
  dc:description ""<p>This vocabulary provides ...most of the original  
USDL specification with some useful simplifications. """;  
  dc:modified "2012-09-20"^^xsd:date;  
  vann:preferredNamespaceUri "http://www.linked-usdl.org/ns/usdl#";  
  vann:preferredNamespacePrefix "usdl";  
  foaf:page <http://linked-usdl.org/ns/usdl.html>;  
  dc:creator  
    <http://linked-usdl.org/ns/usdl#cpedrinaci>,  
    <http://linked-usdl.org/ns/usdl#jcardoso>,  
    <http://linked-usdl.org/ns/usdl#tleidig> .
```

Figure 14 : Exemple d'un fichier USDL [23].

### ❖ *RDF*<sup>17</sup>

RDF (Resource Description Framework) est un cadre de description des ressources proposé comme une méthode générale pour la description conceptuelle ou la modélisation de services à travers des graphes basés sur les modèles de données. Il est représenté comme une méthode générale pour décomposer n'importe quel type de connaissance en petit morceaux. Le graphe d'un *RDF* contient trois composants principaux qui sont : l'objet, l'attribut et la valeur.

### ❖ *DAML-S*<sup>18</sup>

DAML-S (Semantic Markup for Web Services) est un langage de description de service utilisant l'aspect sémantique à travers des ontologies. DAML-S offre des moyens afin de créer une description d'un service qui va être interprétée par un programme. Il facilite la description sémantique de services, leurs interfaces et leurs comportements.

---

<sup>17</sup> <https://www.w3.org/RDF/>

<sup>18</sup> <http://www.daml.org/services/daml-s/0.7/daml-s.html>

❖ *OWL-S*<sup>19</sup>

OWL-S (Semantic Markup for Web Services) fournit aux fournisseurs de services un ensemble essentiel de constructions de langage de balisage pour décrire les propriétés et les capacités de leurs services sous une forme non-ambiguë et intelligente. Le balisage OWL-S des services facilite l'automatisation des tâches du service, la détection, l'exécution, l'interopérabilité, la composition et la surveillance de l'exécution du service.

❖ *WSMO*<sup>20</sup>

WSMO (Web Service Modeling Ontology) définit les principaux aspects liés aux services sémantiques tels que : les ontologies, les services Web, les buts et les médiateurs. Les ontologies WSMO sont exprimées à l'aide du langage de modélisation des services Web (WSML), qui se base sur les différents formalismes logiques : logique de description, logique de premier ordre et la programmation logique.

❖ *WSDL-S*<sup>21</sup>

WSDL-S (Web Service Semantics) est une extension du WSDL traditionnel en ajoutant des aspects sémantiques. L'aspect sémantique relié à ce type de fichier (WSDL-S) est contrôlé de l'extérieur du fichier WSDL.

❖ *SAWSDL*<sup>22</sup>

SAWSDL (Semantic Annotations for WSDL and XML Schema) est défini comme un standard pour spécifier comment les liaisons de données du service Web peuvent être mises en correspondance avec les modèles formels. De plus, SAWSDL offre un moyen pour connecter RDF ou OWL aux services Web sémantiques avec les liaisons de données, facilitant ainsi la recherche.

---

<sup>19</sup> <https://www.w3.org/Submission/OWL-S/>

<sup>20</sup> <https://www.w3.org/Submission/WSMO/>

<sup>21</sup> <https://www.w3.org/Submission/WSDL-S/>

<sup>22</sup> <https://www.w3.org/TR/sawSDL/>

## **II.6 La découverte des services Cloud**

La découverte de service est nécessaire pour prendre en charge toute infrastructure de service telle que le Cloud Computing. Une infrastructure de service multi-domaines à grande échelle nécessite un système de découverte de services ouvert, évolutif, robuste et efficace plus qu'un système à domaine unique. Dans un environnement multi-domaines tel qu'un Cloud hébergeant des milliers d'instances de service, l'exactitude et l'intégralité de la découverte sont plus importantes que dans un petit environnement restreint. L'intégralité de la découverte de service nécessite que le système récupère des résultats pertinents dans les différents domaines constituant le système [24].

La découverte de services dans un environnement en nuage est rendue difficile par le nombre potentiellement important de services hétérogènes disponibles et par un grand nombre de fournisseurs de services. La découverte de service est en outre rendue complexe par le niveau variable de qualité de service (QoS) offert par les différents fournisseurs. Les problèmes de sécurité représentent également un défi de taille pour la recherche d'un service Cloud approprié [24].

### **II.6.1 Le processus de découverte de service**

L'informatique en nuage a été perçue comme une agrégation complexe de ressources informatiques de différents domaines avec des règles administratives différentes, mais présentant d'immenses avantages susceptibles d'améliorer le mode de calcul. L'analyse du problème posé par la découverte d'un service Cloud fiable révèle que le processus implique une série d'événements liés. Le processus de découverte de service peut être décomposé comme suit :

- i. Fonction de publication** : un fournisseur de services utilise cette fonction pour publier des informations sur les services fournis. Cette fonction annonce la disponibilité des services pour une éventuelle découverte.
- ii. Fonction de médiation (Broker)** : elle fournit un intermédiaire pour que les utilisateurs et le fournisseur interagissent.

**iii. Fonction de recherche (requête) :** cette fonction détermine l'emplacement actuel de la disponibilité du service demandé en interagissant avec la liste des services pour une éventuelle occurrence du service requis.

**iv. Fonction de comparaison :** cette fonction vérifie le degré de correspondance entre le service demandé et les services trouvés.

**v. Fonction de considération de réputation :** cette fonction détermine le niveau de confiance du fournisseur de service pour le service exact trouvé pour le service demandé, afin de garantir aux utilisateurs un service fiable.

**vi. Fonction d'octroi d'accès :** cette fonction permet à l'utilisateur d'accéder au service de confiance pour une utilisation réelle [24].

## **II.7 Sélection des services Cloud**

La sélection de services en nuage est un processus permettant de découvrir et de sélectionner des fournisseurs de services en nuages appropriés pouvant satisfaire les exigences en fonction de diverses contraintes, telles que les exigences fonctionnelles, la qualité de service et les accords de niveau de service.

L'absence d'un cadre normalisé pour la représentation de ces exigences rend difficile l'évaluation de la qualité des services (de différents fournisseurs) de manière à répondre à toutes les exigences des utilisateurs qui respectent les SLA des fournisseurs de services. Il est donc important de concevoir des procédures de sélection de services appropriées qui évaluent les services Cloud et sélectionnent les services les plus appropriés [44].

### **II.7.1 Définition du but de la sélection du service Cloud**

Dans la littérature, les objectifs de la sélection de services en nuage peuvent généralement être classés en deux types : sélection du meilleur service et sélection d'une composition optimale de services.

- **Sélection du meilleur service :** le résultat de ce type de sélection de services en nuage est généralement le meilleur service alternatif ou un classement de service selon les exigences du consommateur en nuage demandant la sélection de service. Le processus de sélection est généralement basé sur la performance

des aspects multi-facettes des services alternatifs, et est donc modélisé comme un problème de prise de décision multicritère (MCDM). Le consommateur peut déterminer le niveau d'importance pour chaque aspect de la performance. Le résultat de la sélection peut donc refléter les exigences personnalisées du consommateur. Les approches courantes du problème MCDM incluent AHP, ANP, la prise de décision pondérée et floue, etc.

- **Sélection d'une composition optimale de services** : dans ce type de sélection de services dans le Cloud, le consommateur doit généralement sélectionner un groupe de services composés afin de répondre à ses exigences en matière de fonctionnalités complexes [45].

## **II.8 La qualité de service (QoS)**

Selon [23], la qualité de service QoS (Quality of Service) désigne la capacité d'un service à répondre par ses caractéristiques aux différentes exigences de ses utilisateurs en termes par exemple de disponibilité (taux de rejet), fiabilité (temps moyen entre deux pannes), performance (temps de réponse) et coût (économique, énergétique).

Afin de quantifier un service ou un service composite, il existe des mesures. Ces dernières sont présentées par la qualité de service. De plus, les QoS représentent l'aspect non-fonctionnel d'un service. Ces valeurs permettent la sélection des services pertinents aux demandes des utilisateurs. Dans ce qui suit, nous allons présenter quelques critères de QoS.

- **Le coût** : Il se réfère aux frais d'accès et d'utilisation d'un service que le demandeur de service doit payer. En outre, cela dépend du nombre de tâches qu'un utilisateur de service doit exécuter.
- **Temps de réponse** : Le plus important dans la délivrance de services est de fournir un service au consommateur dans un délai raisonnable. Le temps de réponse est mesuré en fonction de certains sous-facteurs tels que le temps de réponse moyen et le temps de réponse maximal promis par le fournisseur de services. Le temps de réponse est calculé selon la formule (1) suivante :

$$\text{Temp de réponse} = \sum_i^N Ti/n \quad (1)$$

$Ti$  représente le temps où l'utilisateur  $i$  a demandé un service disponible et  $n$  est le nombre total de demandes de service.

- **Fiabilité** : se réfère à la capacité du service à fonctionner correctement et de manière cohérente selon les conditions définies dans l'accord de niveau de service (SLA). Il est mesuré en termes de défaillances de transaction par année ou par mois. La fiabilité est calculée selon la formule (2) :

$$\text{Fiabilité} = \left(1 - \frac{\text{Nombre d'échec}}{n}\right) \times Pmttf \quad (2)$$

*Nombre d'échec* est le nombre d'utilisateurs qui ont subi un échec dans un intervalle de temps inférieur à celui prévu par le fournisseur du Cloud,  $n$  est le nombre d'utilisateur. *Pmttf* représente le temps d'échec estimé qui est défini par le fournisseur de services.

- **Réputation** : est le niveau de confiance à partir duquel le service est accepté par les utilisateurs. Il mesure la confiance gagnée par le service en fonction d'expériences antérieures. Il est calculé selon la formule (3) suivante :

$$\text{Réputation} = \frac{\sum_{r=1}^N Fr}{N} \quad (3)$$

$Fr$  est le rang donné par le  $r$ ième utilisateur pour un service,  $N$  est le nombre d'utilisateurs qui ont classé le service.

- **Disponibilité** : représente la valeur qui mesure si le service est accessible aux utilisateurs. Elle est définie par le pourcentage de temps que le client consomme pour accéder au service. En outre, cela représente la mesure dans laquelle le service est opérationnel. La disponibilité est calculée par la formule (4) suivante :

$$\text{Disponibilité} = \frac{MTBF}{MTBF + MTTR} \quad (4)$$

- **Pertinence** : signifie le degré d'intersection entre les exigences du client et le fournisseur du service. Il existe deux cas pour le calcul de la pertinence, le cas où nous avons plus d'un fournisseur de services à la fin du processus de découverte qui satisfont toutes les exigences essentielles et non essentielles du client. Elle est calculée selon la formule (5) suivante.

$$\text{Pertinence} = \frac{\text{Nombre de caractéristiques non_essentielles fournies par le service}}{\text{Nombre de caractéristiques non_essentielles requises par le client}} \quad (5)$$

Si le filtrage aboutit à une liste de fournisseurs de services vide, les opérateurs qui satisfont les caractéristiques essentielles sont choisis (6).

$$\text{Pertinence} = \begin{cases} 1, & \text{Si toutes les contraintes sont satisfaites} \\ 0, & \text{dans le cas inverse} \end{cases} \quad (6)$$

Un critère de qualité de service peut être soit positif, soit négatif.

- **Les critères positifs** : sont les attributs qu'il faut maximiser (disponibilité, fiabilité, sécurité, etc.).
- **Les critères négatifs** : sont les attributs qu'il faut minimiser (temps de réponse, le coût, etc.).

## **II.9 Composition des services Cloud**

Dans cette section nous présentons les principaux concepts liés à la composition des services Cloud.

### **II.9.1 Définition**

La notion de composition a toujours fait polémique, essentiellement à cause de l'étape de découverte, puisque dans la littérature, une partie de chercheurs considèrent que cette étape est une étape primordiale et faisant partie de la composition, alors que d'autres pensent le contraire.

Selon [46], la composition est un processus qui permet de générer des services complexes à partir de services atomiques. Ces derniers ont la faculté de négocier et interagir de façon intelligente afin de découvrir automatiquement d'autres services qui servent à l'enrichissement du service composite.

Selon [47], la composition est un mécanisme qui permet de regrouper un ensemble de services, qui doivent satisfaire une requête complexe, en utilisant des structures de contrôle et d'échange de messages.

En général, la composition est une opération qui permet de créer des chaînes de connectivité bien ordonnées à partir de services atomiques, afin de satisfaire des requêtes complexes qui ne peuvent pas être satisfaites à partir de services atomiques isolés.

### **II.9.2 Catégorisation des mécanismes de composition**

Dans la littérature, nous distinguons deux grands types de catégorisation :

- Selon Chakraborty et Joshi [48], les mécanismes de composition peuvent être classés en 4 catégories :
  - 1) **Composition proactive** : les services sont composés « hors-ligne ».
  - 2) **Composition réactive** : les services sont composés « enligne ».
  - 3) **Composition obligatoire** : tous les services atomiques composants doivent obligatoirement participer à l'exécution du service composite.

**4) Composition optionnelle :** la participation d'un certain nombre de services atomiques composants est suffisante pour la réussite de l'exécution du service composite.

➤ Selon Medjahed [49], les mécanismes de composition peuvent également être classés en 4 catégories :

**1) Composition statique :** les services sont composés, compilés, et déployés lors de l'élaboration du mécanisme de composition avant même la réception de la requête.

**2) Composition dynamique :** l'opération de composition de services est lancée lors de la réception de la requête.

**3) Composition manuelle :** les services composites sont créés à la main à l'aide d'un éditeur XML par exemple, sans l'aide d'un moteur de génération automatique de services composites.

**4) Composition automatique :** l'outil de génération de services composites génère ces derniers sans l'intervention de l'utilisateur qui n'a comme seule tâche, la spécification de ces besoins.

### **II.9.3 Problème de la recherche de composition des services Cloud**

La composition des services traditionnels est examinée du point de vue des utilisateurs [50, 51]. Comme lors du processus de composition, il existe de nombreux services candidats avec la même fonctionnalité, mais des attributs de qualité de service (QOS) différents, l'objectif est de choisir les bons services à composer afin d'obtenir une solution optimale pour répondre aux exigences QOS demandées par un utilisateur de service (comme la disponibilité, le débit, le temps de réponse, la sécurité, etc.).

#### **II.9.3.1 Sélection de services composites**

La sélection de service consiste à choisir un service parmi un ensemble de services sur la base de propriétés fonctionnelles ou des paramètres de qualité de services [52].

En d'autres termes, la sélection de services consiste à sélectionner le meilleur service répondant aux besoins de l'utilisateur. Nous distinguons deux plans de sélection :

- **Sélection locale** : consiste à trouver pour chaque service abstrait d'un plan de composition, indépendamment des autres, le service le plus approprié. Si la recherche locale permet de prendre en compte des exigences de QoS d'un service de la composition, elle ne permet pas de prendre en compte les contraintes globales de QoS au niveau de toute la composition.

- **Sélection globale** : vise à résoudre les limitations de la sélection locale. A l'instar de la sélection locale, une prise de décision multicritères est appliquée sur le résultat composite pour décider des services permettant d'atteindre une QoS optimale au niveau du service composite.

#### **II.9.4 Structure d'exécution de services**

Selon [53], Il existe quatre structures d'exécution de service, parallèle, séquentielle, boucle et conditionnelle.

- **Parallèle (ET)** : plusieurs services sont exécutés en même temps.
- **Séquentielle** : les services sont exécutés l'un après l'autre.
- **Boucle (Loop)** : un ou plusieurs services sont exécutés de manière répétitive n fois.
- **Conditionnelle** : sélectionne un service parmi plusieurs services.

#### **II.9.5 Cycle de vie de composition de services**

Nous allons utiliser le processus défini dans [23] afin d'illustrer les étapes nécessaires de composition de services. La composition de services est basée sur un ensemble d'étapes présentées comme suit :

- ❖ **Phase de définition** : Dans cette étape, l'utilisateur introduit la demande qui comporte les informations sur le service voulu. Ensuite, les demandes seront décomposées (requête comporte plusieurs informations) de façon automatique en utilisant des ontologies ou d'autres modèles en tenant en compte les qualités de services suggérées par l'utilisateur.
- ❖ **Phase de sélection** : Dans cette phase, à l'intérieur de chaque activité dans le service composite, on cherche dans un registre de services ceux qui concernent

le type de service convenables à chaque activité à partir de la description du service déployé. Dans cette étape, on peut trouver des milliers de services qui satisfont les demandes de l'utilisateur. Pour choisir le service adéquat à une certaine activité, on doit utiliser un algorithme de matching. A la fin de cette étape, on peut composer un ensemble de services qui répondent aux besoins de l'utilisateur.

- ❖ **Phase de déploiement** : Dans cette phase, le service composite va être déployé afin d'être utilisé à partir des opérations d'instanciation et d'invocation d'après l'utilisateur final. A la fin de cette phase, nous obtenons un service composite exécutable.
- ❖ **Phase d'exécution** : Dans cette étape, une instance d'un service composite sera créée et exécutée par le moteur d'exécution qui est responsable de l'invocation des composants du service individuel. Au cours de l'exécution, plusieurs opérations seront réalisées telles que : les tâches de surveillance, l'enregistrement, suivi de l'exécution, mesure du rendement et gestion des exceptions. On peut trouver des méthodes de composition automatique qui fusionnent les deux premières étapes dans une seule étape. De telle façon, le service composite produit est directement généré selon les conditions de composition sans créer le service composite abstrait.

## **II.10 Synthèse et étude comparative des solutions à base de broker**

Dans cette section, nous allons présenter une synthèse et des études comparatives de quelques approches de description, sélection et composition de service pour un broker dans le Cloud. Nous commençons d'abord, par donner les critères d'évaluation utilisés pour effectuer les comparaisons. A la fin, nous dressons un tableau récapitulatif des comparaisons effectuées selon les critères choisis.

### **II.10.1 Les critères d'évaluation de l'étude comparative**

Afin de mieux comparer les différents travaux, nous utilisons les critères d'évaluation suivants :

- **Interopérabilité** : Une caractéristique essentielle d'une approche de composition de service. C'est la capacité de deux ou plusieurs systèmes ou éléments d'échanger des informations et d'utiliser les informations échangées.

- **Description de service** : C'est un critère qui permet de capturer les caractéristiques fonctionnelles et non fonctionnelles d'un service dans un format pouvant être traité et préparé pour la publication du service.
- **Découverte de service** : Cette caractéristique permet aux utilisateurs de rechercher et récupérer la liste des services qui satisfont leurs besoins.
- **Sélection de service** : En phase de sélection des services, les utilisateurs sélectionnent le service le plus approprié pour leurs besoins fonctionnels et non fonctionnels.
- **Composition de service** : permet de regrouper deux ou plusieurs services dans un seul service ou dans de nouveaux services afin d'améliorer les capacités du courtier et répondre au besoin de l'utilisateur.
- **SLA automatique** : Les utilisateurs négocient le contrat de niveau de service (SLA) avec le fournisseur.
- **Module business**: indique le montant d'argent à payer pour accéder à un service Cloud convenu.
- **Interface web** : C'est un critère de côté fournisseur qui permet de publier des descriptions de services dans l'annuaire de broker. Ce composant est situé chez les deux types d'utilisateurs : le demandeur de service et le fournisseur de service. Le but principal de cette unité est d'introduire les requêtes du client et d'afficher les résultats. De plus, dans le cas du fournisseur et à travers cette interface, il peut réaliser deux opérations : (i) soit le déploiement du service en présentant la description du service afin de fournir une définition du service (ii) ou bien envoyer des mises à jour sur les services déployés. Ce composant présente un avantage en facilitant l'interaction entre les utilisateurs et le système via l'interface graphique.

En plus de ces critères, nous allons comparer les approches selon les outils, langages et concepts utilisés.

### II.10.2 Comparaison entre les approches de description de services Cloud

Nous comparons, dans le tableau 4, les approches de service en nuage selon quatre critères de comparaison : (i) Service déploiement ; (II) la technique utilisée ; (III) la représentation proposée du service Cloud ; et IV) le domaine couvert.

Reference	Technique utilisée	La représentation proposée du CS	Service déploiement	Domaine couvert
[27]	Ontology (OWL)	Cloud Ontology	SaaS, PaaS, IaaS	Description, découverte
[54]	Ontology (OWL)	Cloud ontology	SaaS, PaaS	Description
[36]	Ontology (OWL)	Cloud service Ontology	SaaS, PaaS, IaaS	Description, Storage
[35]	Ontology	Cloud service ontology	SaaS, PaaS, IaaS	Description, découverte
[55]	Ontology (OWL)	Business Ontology	SaaS, PaaS	Description, découverte
[33]	Ontology (OWL)	Cloud service ontology	SaaS, PaaS, IaaS	Description, découverte
[56]	Model	Reference model of Cloud application	SaaS	Description, découverte
[57]	Model (UML, model)	Extended UML profile	SaaS, PaaS, IaaS	Description, composition
[58]	Model (Feature model)	Cloud service feature	SaaS, PaaS, IaaS	Description
[30]	Langage USDL	CSDM model	SaaS, PaaS, IaaS	Description
[59]	Langage USDL	Cloud model	SaaS, PaaS, IaaS	Description
[60]	Langage (CSDL, CSCL et CSM)	CSDL, CSCL et CSM	SaaS, PaaS, IaaS	Description
[61]	XML, OWL	Qos model, RoS model	SaaS, PaaS, IaaS	Description, Storage
[62]	Blueprint Template	Blueprint Template	SaaS, PaaS, IaaS	Description,

[63]	Tree structure	Taxonomy based on a tree structure	SaaS, PaaS, IaaS	Description
[25]	Langage USDL	Cloud model	SaaS	Description
[26]	Langage USDL	Cloud model	SaaS	Description
[28]	Langage wsdl-s	WSDL model	SaaS	Description
[29]	Langage wsdl	WSDL model	SaaS	Description
[31]	Langage USDL	Cloud model	SaaS, PaaS	Description
[32]	Langage USDL +WSMO ontology	Cloud ontology	SaaS, PaaS	Description

Tableau 4 : Comparaison entre les approches de description de services Cloud

### Discussion

Comme le montre le tableau 4, nous notons que chaque approche utilise une technique spécifique pour décrire les services Cloud : ontologies [27,33,35,54,55] ; langages [25,26,28,29,30,31,32,59,60,61] ; Modèles [56,57,58] ; Template [62] ou structure arborescente [63].

Cela signifie qu'il n'y a pas de description de service Cloud standardisée. Le choix d'une technique pour décrire les services Cloud se fait en fonction de l'objectif du travail (description, découverte ou composition, etc.). La figure 15 donne les pourcentages des catégories les plus utilisées dans la description des services Cloud.

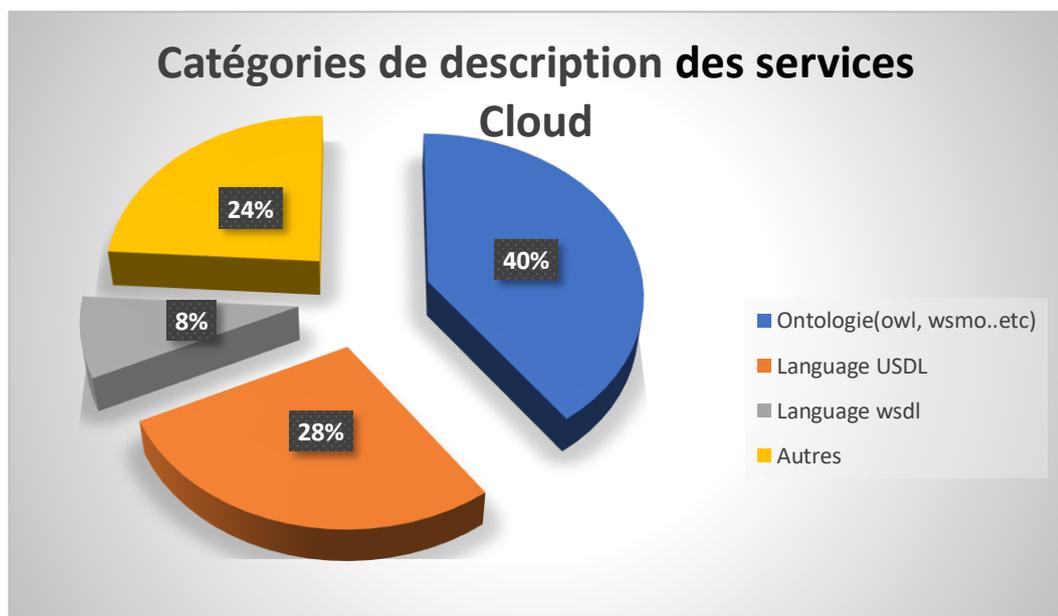


Figure 15 : Catégories de description de services Cloud

Les services Cloud peuvent être décrits selon quatre aspects : techniques, opérationnels, commerciaux et sémantiques.

Les aspects techniques et opérationnels décrivent l'interface et les fonctionnalités des services. L'aspect commercial décrit les acteurs qui interagissent avec un service, la tarification, les SLA, les obligations légales, etc. L'aspect sémantique se distingue par l'utilisation d'ontologies (OWL, WSMO, etc.).

Comme indiqué dans le tableau 5, la plupart des travaux ont décrit les services en considérant certains aspects. Ainsi, nous trouvons des travaux qui traitent des aspects opérationnels et techniques [56, 60, 62], des travaux qui prennent en charge les aspects commerciaux et sémantiques [36,55], ceux qui couvrent les aspects techniques, opérationnels et commerciaux [58, 59, 61, 63, 26], des travaux décrivant les aspects techniques, opérationnels et sémantiques [35, 54, 57]. D'autres travaux couvrent les aspects techniques, opérationnels, commerciaux et sémantiques [27,30].

Travaux de recherche	Aspect technique	Aspect opérationnel	Aspect business	Aspect sémantique
[35,54,57]	√	√	X	√
[56,60,62]	√	√	X	X
[36,55]	X	X	√	√
[58,59,61,63,26]	√	√	√	X
[27], [30]	√	√	√	√

Tableau 5 : Comparaison des approches de services Cloud selon les aspects techniques, opérationnels, commerciaux et sémantiques.

En conclusion, les approches qui ne prennent pas en charge tous les aspects ne sont pas utiles pour décrire les services Cloud, en raison de l'absence d'expressivité.

### II.10.3 Comparaison entre les approches de découverte de services Cloud

**Jing Zhou et al.** dans [28] proposent une étude préliminaire sur la découverte de services en nuage en adoptant un paradigme de P2P non structuré. Ils développent un mécanisme efficace pour le routage des demandes de service en couplant un certain nombre de composants : réplication en un saut, routage sémantique des messages et

réorganisation de la topologie. Un certain nombre d'expériences ont été réalisées qui ont démontré les performances attendues du système de recherche P2P proposé.

**Kadda Beghdad Bey et al.** dans [34] présentent une nouvelle approche pour la découverte de services SaaS basée sur des systèmes Multi-Agents dans les environnements de Cloud Computing. Plus précisément, le but de cette approche est de satisfaire les besoins de l'utilisateur en termes de taux de précision des résultats et de temps de traitement de la demande. Pour établir l'intérêt de la solution proposée, des expériences sont effectuées sur un jeu de données simulé. Le schéma de modélisation de système de découverte basé sur l'agent est divisé en plusieurs éléments principaux, y compris la description des services et la publication, l'approche de clustering, la génération de demandes, le processus de correspondance et la méthode de découverte. Pour améliorer la qualité de la découverte de services SaaS, ils ont proposé un nouvel algorithme pour regrouper les services SaaS dans le Cloud en fonction de leur description sémantique.

**Abdullah Alfazi et al.** dans [35] proposent un moteur de recherche de services Cloud pour identifier et catégoriser les services Cloud sur Internet. Ils développent une ontologie complète pour découvrir et catégoriser les services Cloud dans des environnements réels. Une technique basée sur l'ontologie est également proposée pour catégoriser les services Cloud afin d'améliorer la découverte du service Cloud en catégorisant les services Cloud. Ces études expérimentales approfondies démontrent l'applicabilité de l'approche proposée, ainsi que sa capacité à identifier efficacement les services Cloud sur Internet.

**Amirreza Tahamtan et al.** dans [36] fournissent une ontologie intégrée pour les fonctions métier et les fournisseurs de Cloud qui aide les utilisateurs à trouver des services Cloud en fonction de leurs exigences fonctionnelles et non fonctionnelles. La structure fournie peut non seulement servir de référentiel de services, mais également répondre à des problèmes tels que la description des services Cloud, leur flexibilité et leur possibilité d'échange. À ce stade, les données fournies sur les fournisseurs de Cloud reposent sur une étude de marché.

### II.10.3.1 Synthèse des travaux de découverte des services Cloud

D'après les recherches qui existent, les approches qui s'attaquent au problème de découverte de services en nuage peuvent être classées en quatre catégories :

- **Recherche sémantique centrée sur le Cloud :** La recherche sémantique centrée sur le Cloud pour les approches de service Cloud est présentée dans [64, 65, 66]. Une ontologie de service métier et une ontologie Cloud avec des capacités d'interrogation de service sont proposées dans [66]. L'ontologie unifiée capture les services métier requis dans une organisation et fournit le mappage entre les fonctions métier et les services offerts dans le Cloud.
- **Cloud service annotations :** La découverte du nuage à l'aide de l'annotation des services est présentée dans [67, 68, 69]. Dans [68], les annotations de service en nuage sont utilisées pour la découverte sémantique de services pertinents en nuage.
- **Approches QoS basées sur la sémantique :** Les approches qui prennent en charge la découverte de service sensible à la QoS basée sur la sémantique sont présentées dans [70,71]. La découverte d'unités virtuelles en nuage basée sur l'ontologie est présentée dans [70]. Cependant, l'architecture ne s'applique qu'aux services IaaS. Une autre approche est présentée dans [71]. Il est déployé en tant qu'application Cloud pour fournir des services de découverte de services sensibles au comportement et à la qualité de service. Son efficacité sera évaluée lors de son déploiement dans un Cloud commercial.
- **Autres approches :** D'autres efforts incluent un système de recommandation de concept de service dans [72] et une architecture de découverte de service centralisée dans [73]. Un framework pour un système de recommandation de concept de service est proposé dans [72], pour la récupération de service dans l'écosystème de service. La structure est intégrée dans un dispositif de mise en correspondance de services sémantiques afin d'améliorer la fiabilité du dispositif de mise en correspondance de services sémantiques dans l'écosystème de services.

En conclusion, la découverte automatique des services est souvent difficile, car les descriptions de service peuvent impliquer des contraintes complexes et nécessitant

une correspondance sémantique flexible. De plus, les fournisseurs utilisent souvent des formats non standards ce qui pose des problèmes d'interopérabilité sémantique.

#### **II.10.4 Comparaison entre les approches de sélection de service Cloud**

**Hela Malouche et al.** dans [38] proposent une architecture de courtage en nuage qui permet de sélectionner les services Cloud en fonction des exigences fonctionnelles et non fonctionnelles identifiées par l'utilisateur. La sélection du service Cloud par le courtier se fait à l'aide d'une version développée de l'algorithme CM-Factory qui prend en compte les préoccupations transversales de l'organisation.

**Falak Nawaz et al.** dans [39] développent une architecture de Cloud Broker pour la sélection des services Cloud en trouvant un modèle des priorités changeantes des préférences utilisateur (UPs). Ils ont proposé un framework qui trouve un modèle de changement UPs en utilisant une chaîne de Markov indépendamment des préférences de l'utilisateur initial. Le modèle est ensuite lié aux critères de QoS de tous les services disponibles pour trouver des pondérations à l'aide de la méthode BWM (Best Worst Method). Les pondérations des critères de tous les services sont ensuite utilisées pour déterminer le classement général des options pour la sélection des services en nuage, ainsi que le modèle d'UP.

**Shuai Ding et al.** dans [40] proposent une méthode de prédiction de classement innovante pour la sélection de services Cloud personnalisée, qui prend en compte l'attitude du client et ses attentes en matière de qualité de service. Pour améliorer la précision de la prévision du classement de service, la méthode proposée a défini une mesure KRCC (Kendall Rank Correlation Coefficient) améliorée permettant de réduire l'influence des clients négatifs dans le calcul de la similarité de classement et a présenté une fonction de satisfaction du client afin de déterminer les valeurs de préférence concernant les paires de services.

**Qiang He et al.** dans [41] proposent MSSOptimiser, une approche QoSdriven qui prend en charge la sélection de services pour les applications logicielles SaaS. En utilisant les techniques d'optimisation, en particulier la programmation d'entiers (Integer Programming), il aide les développeurs SaaS à déterminer les services optimaux pour un SaaS mutualisé qui répondent aux exigences de QoS de différentes parties

prenantes, y compris l'objectif d'optimisation du fournisseur SaaS et les différents niveaux de contraintes de QoS de différents utilisateurs finaux.

**Dan Lin et al.** dans [42] ont présenté une architecture basée sur le courtage pour les systèmes de Cloud Computing, ainsi qu'un algorithme de sélection de service Cloud efficace qui fournit une liste des fournisseurs de services Cloud recommandés aux consommateurs Cloud en fonction de leurs besoins. En particulier, ils ont conçu une nouvelle structure d'indexation, à savoir le Bcloud-Tree, pour faciliter l'arrangement et la récupération des informations sur les prestataires de services. En plus du Bcloud-Tree, ils ont développé un algorithme de requête de sélection de service efficace qui récupère rapidement les fournisseurs de services souhaités en fonction des demandes de service des utilisateurs.

**Godse et Mulik** dans [74], proposent une approche basée sur AHP de sélection de nuages SaaS. La sélection de produits SaaS est modélisée selon une hiérarchie composée de cinq facteurs principaux (niveau 1), notamment la fonctionnalité, l'architecture, l'utilisabilité, la réputation du fournisseur et le coût, ainsi que seize attributs (niveau 2).

**Gonçalves et al** dans [75], proposent une sélection de déploiement en nuage basée sur AHP, fondée sur des attributs de performance non fonctionnels, notamment efficacité, coût et évolutivité. L'approche proposée est utilisée dans un cas réel (WordPress déployé dans le Cloud Amazon).

**Habib et al** dans [76], proposent un système de gestion de la confiance dans les environnements de Cloud, dont le but est d'aider les consommateurs de Cloud à trouver des fournisseurs de Cloud de confiance à partir de descriptions de services vagues. Le système peut prendre en compte à la fois des évaluations subjectives issues des réactions des utilisateurs et des évaluations objectives issues de l'audit de services. Pour chaque attribut de performance d'un service en nuage, la confiance correspondante est évaluée en fonction de la note moyenne du service, de la certitude associée à la note moyenne et de l'attente initiale du service. Ensuite, toute la confiance d'attribut est agrégée via un algorithme basé sur la logique.

**Ghosh et al.** dans [77], proposent un cadre appelé «SelCSP» pour calculer le risque d'interaction des fournisseurs de Cloud pour les consommateurs. Le calcul du risque repose sur l'agrégation de l'évaluation de la fiabilité des fournisseurs par le biais

des retours d'informations ou des expériences directes des consommateurs et de l'évaluation de la compétence des fournisseurs par le biais de la transparence des garanties SLA. Enfin, les niveaux de risque d'un fournisseur de Cloud dans différents contextes peuvent être estimés pour un consommateur particulier.

#### II.10.4.1 Synthèse des travaux de sélection des services Cloud

Dans cette section, nous résumons des approches de sélection de services Cloud selon le contexte de sélection de service et les méthodes utilisées (tableau 6).

Référence	Méthode de sélection	Contexte
[78]	MCDM-AHP	General
[74]	MCDM-AHP	SaaS
[33]	MCDM-AHP	SaaS
[79]	MCDM-ANP	General
[81]	MCDM-MAUT	General
[80]	MCDM-MAUT	SaaS
[82]	MCDM-ELECTRE	General
[83]	MCDM-SAW	General
[84]	MCDM-SAW	General
[85]	MCDM	General
[86]	MCDM- weighting average aggregation	General
[87]	Optimization-dynamic programming & greedy	General
[88]	Optimization-dynamic programming	SaaS
[89]	Optimization-greedy & tree search	General
[90]	Optimization-greedy	SaaS
[91]	Optimization-greedy & integer programming & skyline	SaaS
[92]	Optimization	General
[93]	Optimization-SAW & MDP	SaaS
[94]	Logic-satisfiability module theory & constraint programming	General
[95]	Matrix factorization	General
[96]	Selection strategies	SaaS

Tableau 6 : Résumé de quelques approches de sélection des services Cloud.

Sur la base de la dimension des techniques de sélection de service, nous classons les approches existantes en quatre groupes : *basées sur MCDM, basées sur l'optimisation, basées sur la confiance et autres techniques.*

❖ **Approches MCDM pour la sélection de services dans le Cloud**

[78,74,33,79, 81,80,82,83,84,85,86] ont utilisé MCDM (prise de décision multicritères) qui est une technique courante appliquée à la sélection de services dans le Cloud. Dans ce scénario, les performances globales d'un service Cloud sont exprimées via l'agrégation d'un ensemble d'attributs finis, chacun représentant un aspect de performance. Les principaux problèmes de la sélection basée sur le MCDM sont les suivants : comparaison de la qualité de la performance de chaque attribut parmi tous les services alternatifs et agrégation des attributs de performance en fonction des besoins des consommateurs [43].

❖ **Approches basées sur l'optimisation pour la sélection de services en nuage**

Dans le domaine de la sélection des services, l'optimisation est définie comme suit : « Trouver les services les plus appropriés pour les clients ou les fournisseurs, ce qui maximise ou minimise un ou plusieurs critères tout en respectant les contraintes ». La littérature [87,88,89,90,91,92,93] montre que de nombreuses méthodes d'optimisation ont été appliquées à la sélection de services dans le Cloud, telles que la programmation dynamique, la programmation en nombre entier, l'algorithme glouton, etc. Dans les environnements généraux orientés services, la sélection basée sur l'optimisation est largement étudiée dans le domaine de la composition des services [43].

❖ **Sélection basée sur la confiance**

Le concept de « confiance » a été introduit dans de nombreux contextes différents. Dans des environnements orientés services, Jøsang et al. [70] définissent la confiance comme «la probabilité subjective par laquelle un individu s'attend à ce qu'un autre effectue une action donnée dont dépend son bien-être». La sélection basée sur la confiance dans les environnements en nuage fait référence à la sélection de services basée sur l'évaluation de la fiabilité des fournisseurs ou services en nuage. Dans les approches de sélection basées sur la confiance, les modèles de confiance sont généralement proposés sur la base soit d'évaluations subjectives, soit d'évaluations objectives, soit de leur combinaison. Les évaluations subjectives sont généralement extraites

de jugements de consommateurs ordinaires ou d'avis d'experts ; et les évaluations objectives sont généralement obtenues à partir d'une évaluation de performance quantitative ou de descriptions de QoS [45].

### **II.10.5 Comparaison entre les approches de composition de service Cloud**

**Thar Baker et al.** dans [97] ont proposé une approche de composition de service qui détermine le nombre de services composites le moins possible en fonction d'une combinaison efficace de fournisseurs de services Cloud qui satisfont la demande de l'utilisateur. Cette approche aborde le besoin croissant d'optimiser la consommation d'énergie associée à l'augmentation des services et de demandes d'utilisateurs complexes basés sur le Cloud, qui se caractérisent par un grand nombre de fournisseurs et de services Cloud.

**Seghir et al.** [107] ont proposé une approche hybride utilisant des algorithmes d'optimisation génétiques pour la composition de services en nuage tenant compte de la qualité de service. Où l'algorithme génétique est utilisé pour la recherche globale et l'optimisation est utilisée pour la recherche locale.

**Karimi et al.** [108] ont proposé une approche de composition réduisant l'espace de recherche lors de la sélection de services locaux en utilisant des règles d'association, un regroupement de services et un algorithme génétique pour obtenir la solution optimale globale.

**Jrad et al.** [109] ont étudié l'appariement SLA dans plusieurs nuages. Ils ont proposé un modèle ontologique pour la description sémantique des services composites multi-Cloud. En outre, un algorithme génétique basé sur un utilitaire a été proposé pour sélectionner les ressources Cloud optimales en fonction de la correspondance entre les exigences de l'utilisateur et les propriétés Clouds.

**Kurdi et al.** [99] proposent un algorithme d'optimisation combinatoire pour la composition de services en nuage (COM2) qui vise à composer efficacement des services avec un petit nombre de services examinés et de nuages combinés. L'algorithme proposé sélectionne le Cloud avec le nombre maximal de services afin d'accroître la possibilité de répondre aux demandes de service avec des frais généraux minimales.

Le tableau 7 suivant résume la comparaison entre quelques approches de composition.

Référence	Solution	Complexité	Avantage	Inconvénient
[98]	global optimization with local selection for Efficient QoS-aware Service Composition	$O(l)$ , où $l$ est le nombre de services candidats dans chaque groupe	<ul style="list-style-type: none"> <li>* Efficace en termes de temps de calcul et complexité.</li> <li>* Solution quasi optimale.</li> </ul>	<ul style="list-style-type: none"> <li>* Elle ne convient pas à la composition de service basée sur plusieurs critères de qualité de service.</li> </ul>
[99]	A combinatorial optimization algorithm		<ul style="list-style-type: none"> <li>* Nombre de nuages réduit.</li> <li>* Temps de composition court.</li> </ul>	<ul style="list-style-type: none"> <li>* Solution optimale dans quelques cas.</li> <li>* Manque d'équilibrage de la charge entre les serveurs Cloud.</li> </ul>
[100]	Ant Colony Optimization (ACO) algorithm	$O(N^m)$ , où $m$ services abstraits, et chaque service abstrait concerne $n$ services concrets	<ul style="list-style-type: none"> <li>* Solution quasi optimale dans un court laps de temps.</li> <li>* Présente des avantages en termes de qualité de solution et d'efficacité.</li> </ul>	<ul style="list-style-type: none"> <li>* faible évolutivité en raison de sa complexité de calcul exponentielle et est donc impraticable pour les applications en temps réel et dynamiques.</li> </ul>
[101]	bee colony algorithm (BCO)			
[102]	genetic algorithm	$O(T) = N \log l$ , où $N$ est le nombre de générations et $l$ la taille initiale de la population.	<ul style="list-style-type: none"> <li>* Solutions réalisables, et évolutives.</li> </ul>	<ul style="list-style-type: none"> <li>* Temps d'exécution élevé.</li> <li>* Convergence lente.</li> <li>* Haute complexité temporelle.</li> </ul>
[103]	Genetic Algorithm			
[97]	An energy-efficient multiple cloud-computing	exponentiel	<ul style="list-style-type: none"> <li>• Temps d'exécution court.</li> </ul>	<ul style="list-style-type: none"> <li>• La QoS n'est pas prise en compte.</li> </ul>

	service composition algorithm		<ul style="list-style-type: none"> <li>• Faible nombre de services examinés.</li> </ul>	<ul style="list-style-type: none"> <li>• Solution optimale dans quelques cas.</li> </ul>
[104]	an agent-based approach to compose services	Toutes les fonctions impliquées dans les comportements d'agent prennent $O(n)$ temps pour s'exécuter	<ul style="list-style-type: none"> <li>* Les informations incomplètes sur les participants au Cloud distribué sont prises en compte.</li> <li>* Les besoins changeants des consommateurs sont traités</li> </ul>	<ul style="list-style-type: none"> <li>* Coûts de communication et de traitement élevés.</li> <li>* Efficace uniquement pour les exigences fonctionnelles et financières.</li> </ul>
[105]	approach based on Formal Concept Analysis (FCA).		<ul style="list-style-type: none"> <li>* Réduction du nombre de fournisseurs.</li> <li>* Nombre minimal de nuages.</li> <li>* Qualité de composition de service élevée.</li> </ul>	<ul style="list-style-type: none"> <li>* consomme du temps.</li> </ul>

Tableau 7 : Comparaison de quelques approches de composition

## Discussion

La composition de service est le processus permettant de rechercher les composants de service existants et appropriés pour composer un nouveau service doté d'une fonction d'agrégation. Il existe actuellement trois méthodes de composition :

**1) Méthode de sélection locale :** Différentes valeurs d'indices de qualité de service sont mappées en une seule valeur par le biais de la fonction utilitaire et le processus de prise de décision multicritères (MCDM) est appliqué pour sélectionner les services concrets appropriés dans chaque groupe de services abstraits pour la composition [98]. Cette méthode présente la meilleure efficacité, alors qu'elle ne peut garantir que les contraintes de QoS locales. Par exemple, la demande de la durée d'exploitation minimale globale du service composite ne peut être satisfaite [101].

**2) Méthode d'optimisation globale :** Le problème de la composition de service globale est transformé en un problème de programmation linéaire à nombres entiers mixtes (MILP) et résolu par un solveur linéaire [105]. Il peut rapidement obtenir une solution de haute qualité, mais il faut que la fonction objective et les contraintes soient des fonctions linéaires, ce qui limite l'algorithme dans une certaine mesure [101].

**3) Méthode d'optimisation intelligente :** C'est aussi une méthode d'optimisation globale. La différence réside dans le fait que le problème est non linéaire et qu'il peut être résolu avec des algorithmes d'optimisation intelligents, tels que les algorithmes génétiques (GA) [102,103] ; les algorithmes d'optimisation de particules (PSO) [9, 10] ; l'optimisation de la colonie des fourmis et d'abeilles (ACO, BCO) [100,101]. Des méthodes d'optimisation intelligentes peuvent résoudre le problème complexe de composition de services dans un court laps de temps avec une qualité élevée. C'est pourquoi cette préoccupation a été généralisée ces dernières années [101].

Le processus de composition de services constitue un point de divergence entre les chercheurs. Il existe des chercheurs qui considèrent la sélection de service comme une étape à part et il y a d'autres qui considèrent la sélection de service comme une étape qui fait partie de la composition. De même, il existe des articles avec des titres de composition mais en réalité, ils traitent juste l'ordonnancement par exemple.

Nous concluons que les articles vus ne traitent pas tout le processus de composition à la fois.

Le tableau 8 récapitule l'ensemble des comparaisons expliquées dans les sections précédentes.

Références	Interopérabilité	Description	Découverte	Sélection	Composition	SLA automatique	Module business	Interface client	Interface fournisseur	Classement	Clustering	Publication
[25]	×	✓	✓	×	×	×	×	×	×	×	×	×
[26]	×	✓	×	×	×	×	×	×	×	×	×	×
[27]	×	✓	✓	×	×	×	×	×	×	×	×	×
[29]	✓	✓	✓	✓	✓	×	×	×	×	✓	×	×
[30]		✓	×	×	×	×	×	×	×	×	×	×
[28]	×	✓	✓	×	×	×	×	×	×	×	×	×
[34]	×	×	✓	×	×	×	×	✓	×	×	✓	✓
[35]	×	×	✓	×	×	×	×	×	×	×	✓	×
[36]	×	×	✓	×	×	×	×	×	×	×	×	×
[38]	×	×	×	✓	×	×	×	×	×	×	×	×
[39]	×	×	×	✓	×	×	×	×	×	×	×	×
[40]	×	×	×	✓	×	×	×	×	×	×	×	×
[33]	×	×	×	✓	×	×	×	×	×	×	×	×
[42]	×	×	×	✓	×	×	×	×	×	×	×	×
[115]	✓	×	×	×	✓	×	✓	×	×	×	×	×
[97]	✓	×	×	×	✓	×	×	×	×	×	×	×
[114]	✓	×	×	×	✓	×	×	×	×	×	×	×
[19]	✓	×	×	×	×	×	×	×	×	×	×	×
[111]	✓	×	×	×	×	×	×	×	×	×	×	×
[110]	✓	×	×	×	×	×	×	×	×	×	×	×
[[112]	✓	×	×	×	×	×	×	×	×	×	×	×
[113]	✓	×	×	×	×	×	×	×	×	×	×	×

Tableau 8 : Récapitulatif des comparaisons

## **II.11 Conclusion**

Malgré le nombre croissant des services Cloud offerts sur le marché et le besoin grandissant de collaboration entre plusieurs fournisseurs de Cloud, les courtiers proposés sur le marché sont conçus pour des solutions propriétaires.

Les langages de description n'ont pas été conçus au départ pour le Cloud Computing mais pour les services web donc ne prennent pas en charge tous les aspects des services Cloud.

Dans ce chapitre, nous avons présenté un état de l'art sur le courtage dans le Cloud Computing. Nous avons structuré ce chapitre en deux parties. La première décrivant les concepts de base et la deuxième consacrée aux travaux connexes avec des comparaisons et des synthèses.

Dans le chapitre qui suit, nous allons présenter l'aspect conceptuel de notre solution de broker. Ce chapitre expliquera en détail les étapes de description, sélection et le processus de composition

### **III.1 Introduction**

Après avoir étudié les concepts de base du courtage en nuage et effectué des études comparatives des solutions existantes dans le chapitre précédent, nous allons dans ce chapitre, présenter notre solution pour le courtage en nuage entre SaaS.

La solution proposée consiste à offrir un broker permettant de traiter les problèmes de description, de découverte, de sélection et le processus de composition à travers une série d'algorithmes utilisés dans le but de satisfaire des requêtes complexe nécessitant l'interopérabilité entre plusieurs SaaS.

#### **Démarche de travail**

La démarche de travail que nous avons suivi afin de réaliser notre projet correspond à l'XP-eXtreme Programming<sup>23</sup>. L'eXtreme Programming (XP) est une méthode agile de gestion de projet informatique adaptée aux équipes réduites avec des besoins changeants.

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

Les principes de l'eXtreme Programming sont basés sur : la revue du code en permanence, les tests systématiques avant chaque mise en œuvre, l'intégration des modifications, la communication et le courage.

### **III.2 Architecture proposée du Cloud broker**

L'étude et l'analyse des solutions existantes nous ont permis d'identifier deux niveaux de broker (Figure 16) :

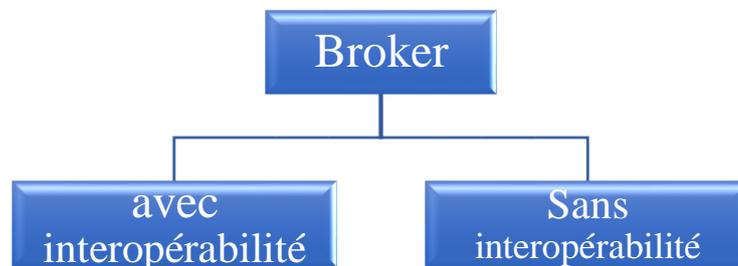
- **Niveau 1 sans interopérabilité** : Lorsque le client communique avec un seul fournisseur de service. Dans ce cas, le client n'a pas le choix de choisir les meilleurs services SaaS qui existent. Donc il est verrouillé avec un seul fournisseur, et s'il veut migrer

---

<sup>23</sup> <http://www.extremeprogramming.org/>

vers un autre fournisseur SaaS il est obligé de prendre toutes ses données stockées chez le premier fournisseur et de les transférées vers un autre fournisseur SaaS. Cela va couter cher et même va prendre du temps.

- **Niveau 2 avec interopérabilité** : Dans ce niveau, les ressources peuvent être échangées et utilisées facilement entre les différents fournisseurs SaaS sans aucun problème. Il permet au consommateur de choisir les meilleurs services. En plus, il donne la capacité de composer des services pour obtenir un nouveau service s'il n'existe pas dans l'entrepôt du broker.



*Figure 16 : Niveaux de broker*

La solution que nous proposons pour le Cloud broker entre SaaS prend en charge les deux niveaux. Elle est décrite dans la figure 17 suivante.

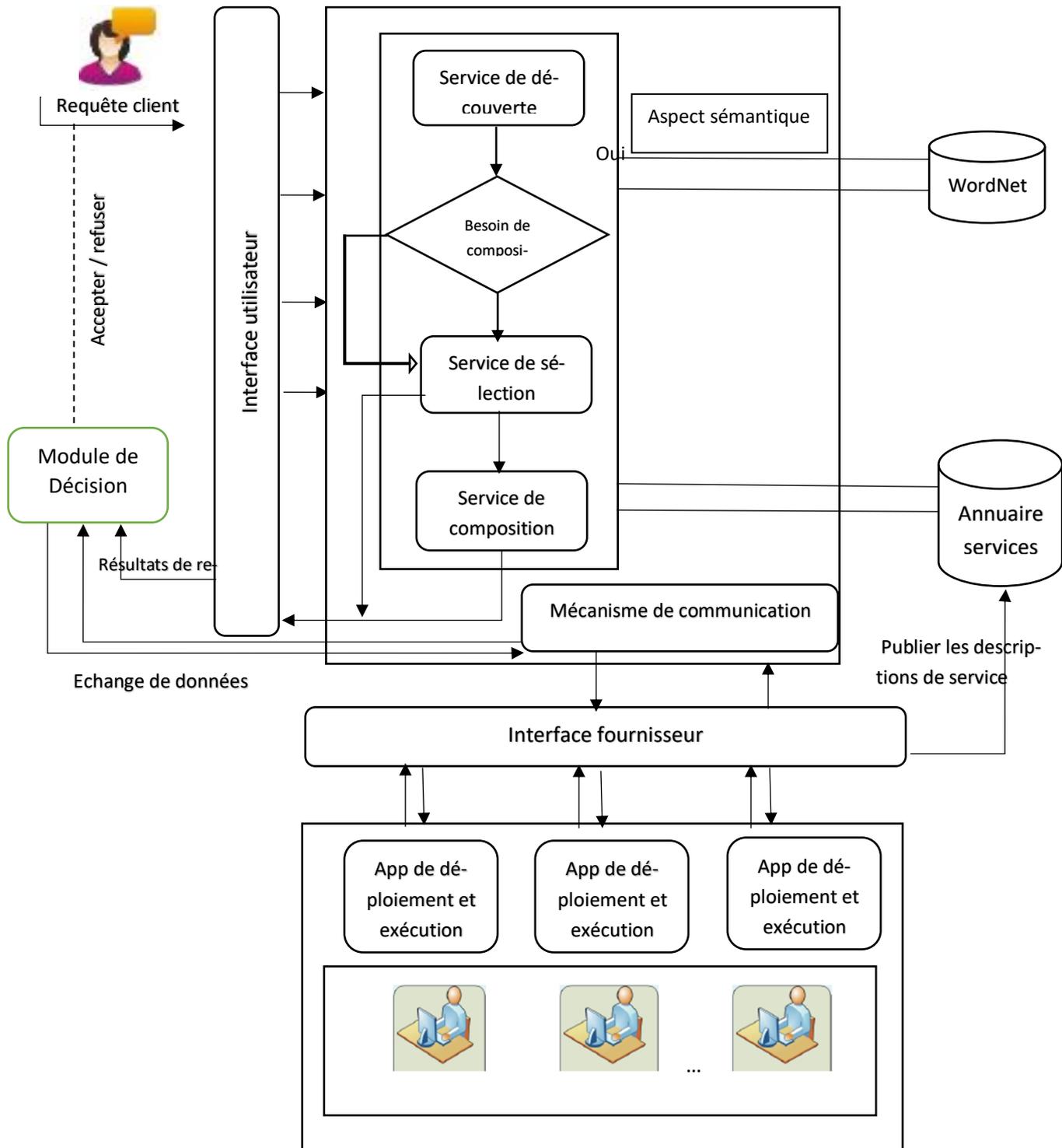
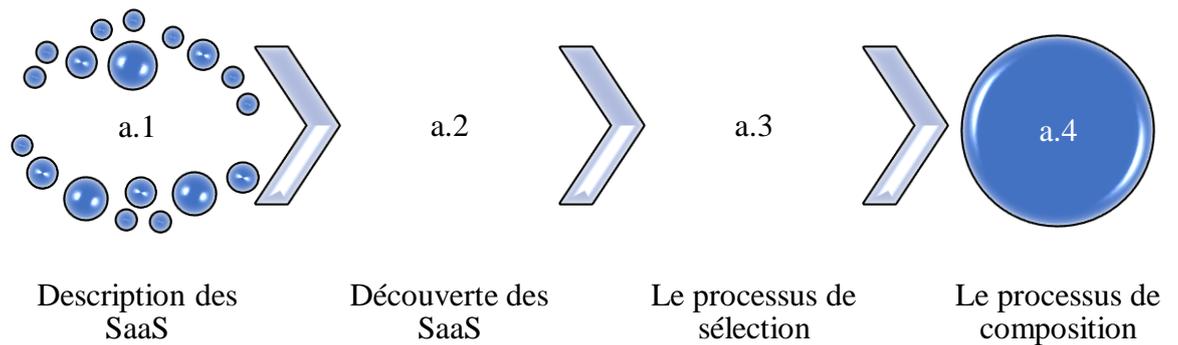


Figure 17 : Architecture proposée du Cloud broker

Le travail du broker suit le processus décrit dans la figure 18 :



*Figure 18 : Processus de courtage du broker*

Dans ce qui suit, nous allons détailler le processus de courtage en expliquant les étapes une à une. Dans un premier temps, nous identifions nos acteurs et les différentes opérations qu'ils peuvent effectuer. Pour les illustrés, nous utilisons le diagramme de cas d'utilisation.

### **III.3 Analyse des besoins**

Nous identifions dans notre système 3 acteurs principaux, qui sont :

**Utilisateur** : la personne désirant profiter du service.

**Fournisseur** : la personne qui fournit les services.

**Admin broker** : l'équipe technique chargée d'apporter son soutien aux clients.

Les fonctionnalités des acteurs sont illustrées par les diagrammes de cas d'utilisation suivants :

- **Diagramme de cas d'utilisation du Cloud broker:** Ce diagramme détermine les principales fonctions de chaque acteur du broker.

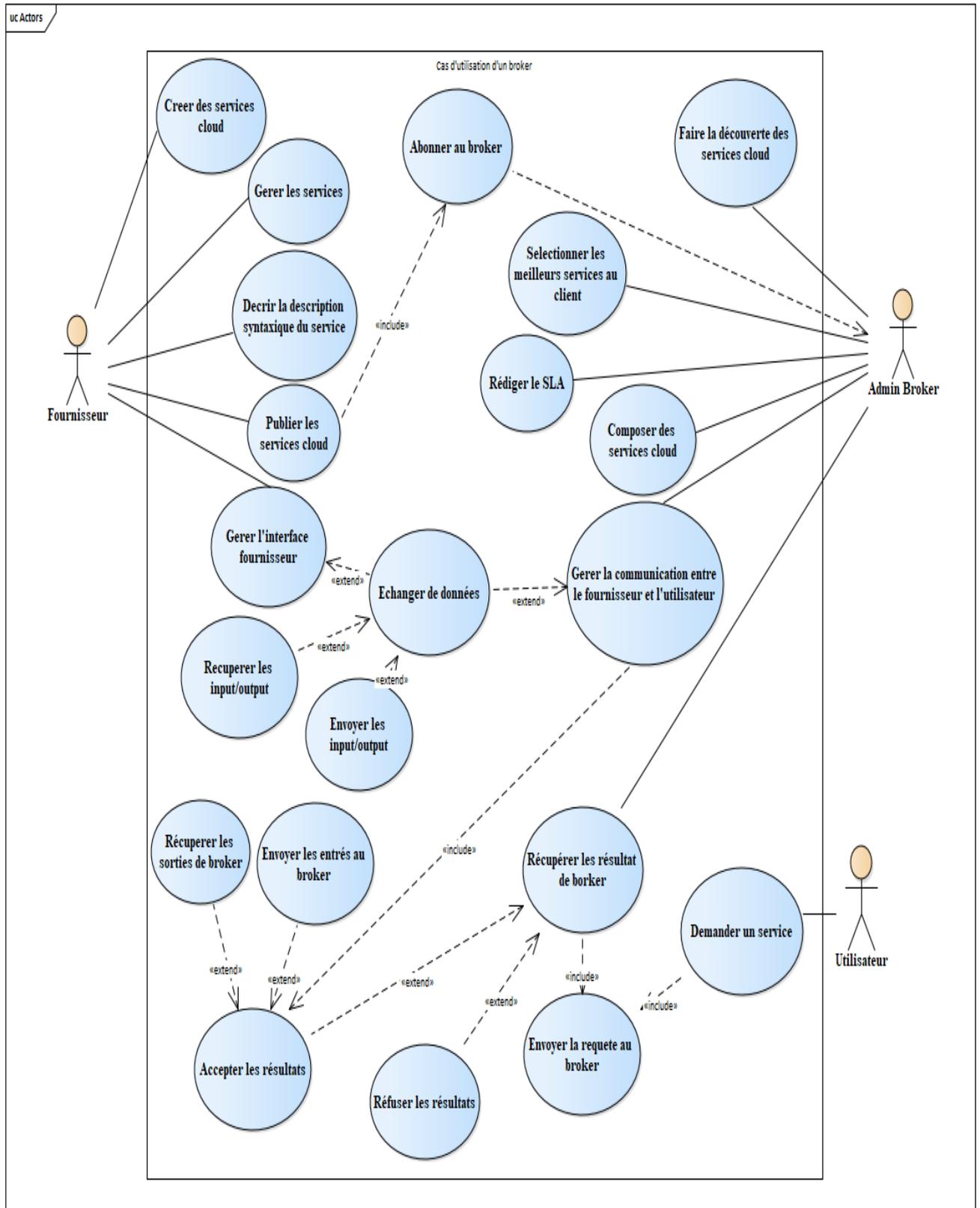


Figure 19 : Diagramme de cas d'utilisation du Cloud broker

- ✓ **Diagramme de cas d'utilisation du fournisseur :** Dans ce diagramme nous allons détailler les différentes fonctions du fournisseur.

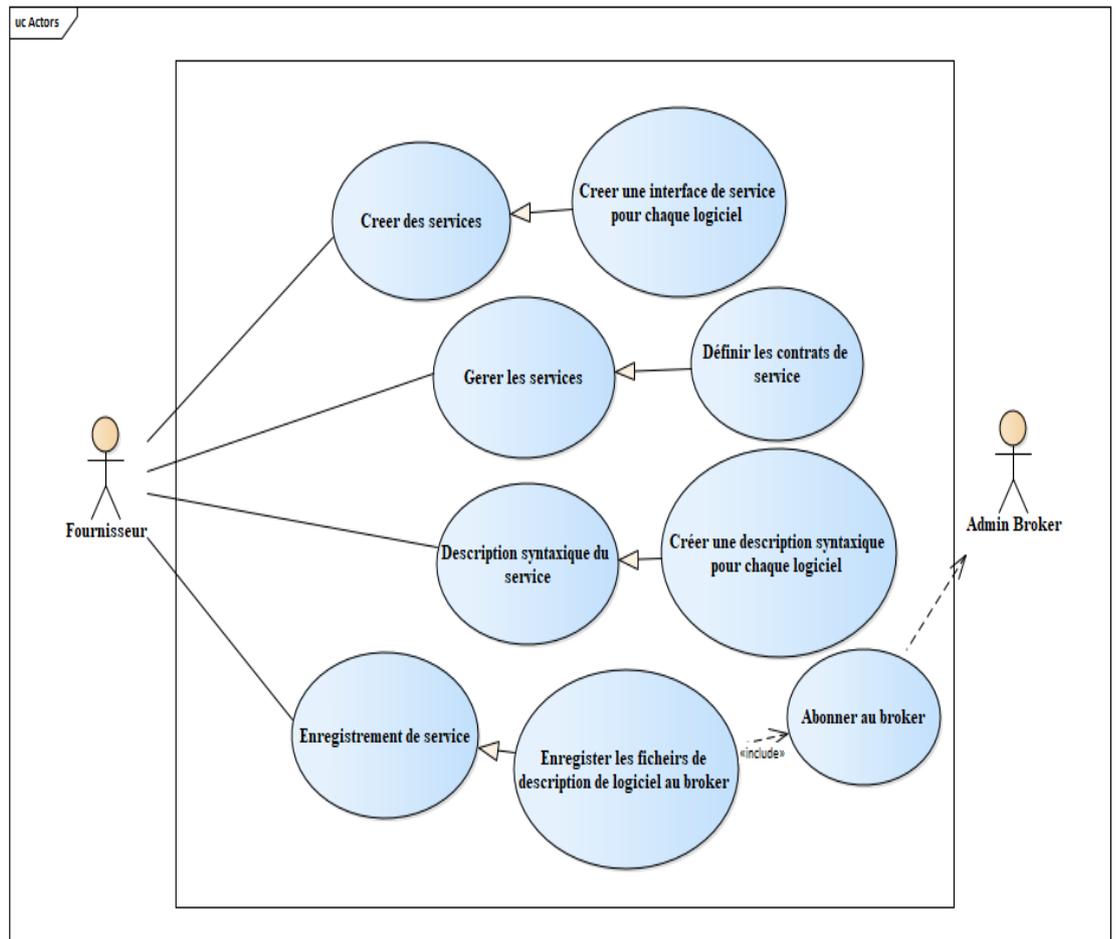


Figure 20 : Diagramme de cas d'utilisation d'un fournisseur Cloud

### III.4 Description de service

Le Cloud Computing est une sorte de pool de services à partir duquel des services sont extraits et utilisés. Nous proposons une description pour les services Cloud de type SaaS basée sur le langage WSDL.

**Pourquoi WSDL ?**

- WSDL (Web Service Description Language) est un fichier XML pour décrire les services et leur lien avec les adresses réseaux spécifiques.
- WSDL est standardisé.
- La possibilité d'ajouter ou modifier des éléments (car XML est extensible).
- Bien structuré.

Et pour que ça soit une description adéquate à tout SaaS, il faut au moins quatre parties (Figure 21) :

- ✓ **Partie catégorisation** : Cette partie décrit le domaine et l'emplacement du service. Les paramètres utilisés dans cette partie sont le nom du service et le domaine.
- ✓ **Partie fonctionnelle** : Cette partie décrit la fonction générale du service et les fonctions techniques de chaque opération de service. Les paramètres utilisés dans cette partie sont les suivants : matricule, nom du type de port, noms des opérations et noms des entrées / sorties.
- ✓ **Partie non fonctionnelle** : Cette partie décrit le niveau de qualité de service. De nombreux paramètres sont utilisés dans cette partie (par exemple, le temps de réponse, la disponibilité, la fiabilité et la transparence).
- ✓ **Partie business** : Cette partie décrit les caractéristiques industrielles et économiques déterminées par le prestataire de services. Les paramètres utilisés dans cette partie sont les SLA et la tarification.

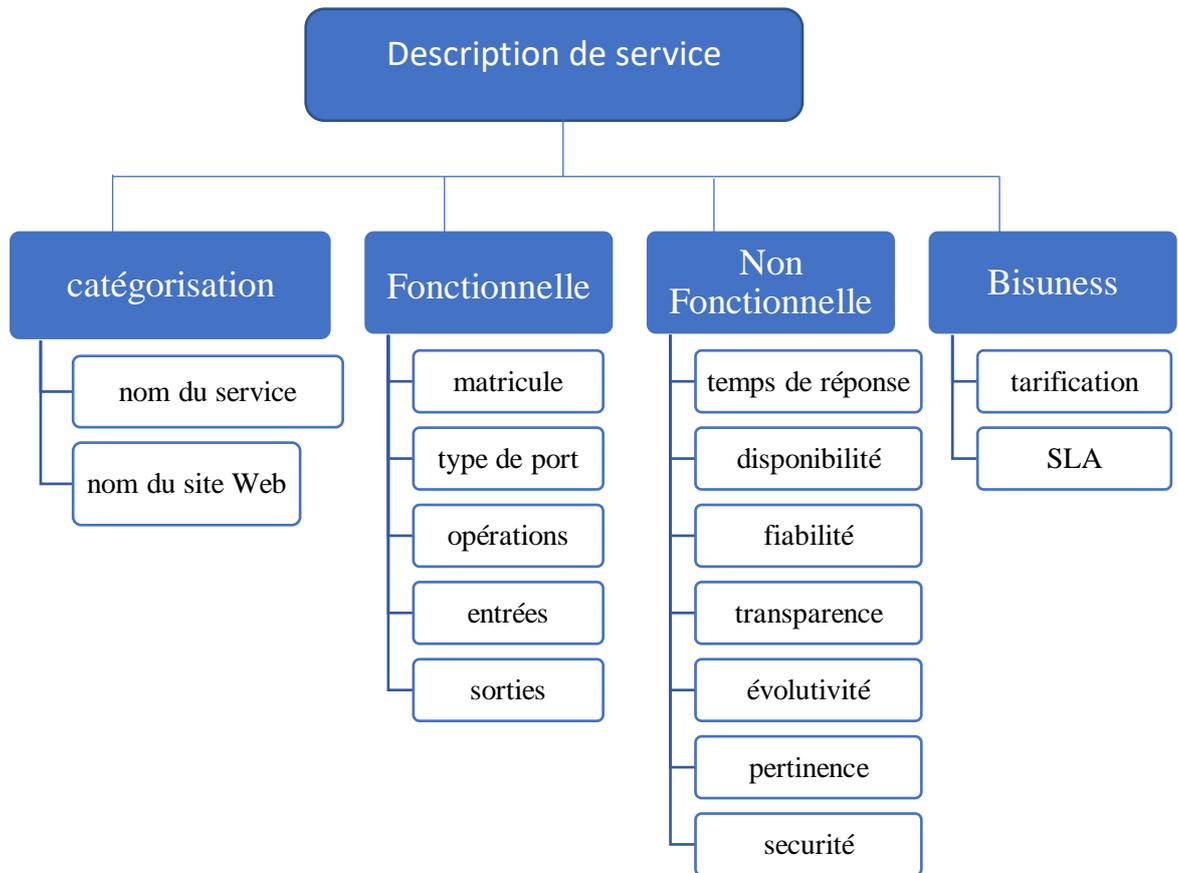


Figure 21 : Les parties essentielles de description de service

- ❖ Cette description nous permet de couvrir le nombre maximum des aspects (Technique, opérationnel, bisness et sémantique) :
  - ✓ Les aspects techniques et opérationnels représentés par l'interface et les fonctionnalités des services.
  - ✓ L'aspect commercial (bisness) à partir de décrire les acteurs qui interagissent avec un service, la tarification, les SLA, les obligations légales, etc.
  - ✓ Et pour assurer l'aspect sémantique et améliorer l'interopérabilité dans notre framework broker nous avons utilisé la bibliothèque Wordnet.

- Les avantages de notre description service :
  - Une description unifier pour tous les services SaaS.
  - La possibilité de modifier ou d'ajouter des nouvelles fonctionnalités.
  - Exploitable par tous fournisseur SaaS.
  - Couvrir un maximum des aspects.

### **III.5 Découverte des services**

La découverte de service permet aux utilisateurs recherchent et récupèrent la liste des services qui satisfont leurs besoins.

Dans notre cas la découverte est faite à partir de l'entrepôts des services SaaS qui ont déjà publier par les fournisseurs sur le broker. Lorsque le client cherche un service SaaS, il va sélectionner le domaine de service et exprimer son besoin dans une requête précise qui représente leur demande. Puis le broker va utiliser les entrés de client comme illustre le pseudo algorithme suivant :

```
Input : requête_client RC, domaine D, entrepôt E, existe
Output: liste_services LS,
Begin
  SelectDomaine (D) ; existe =false ;
  EnterRequest (RC);
  String Pre-processing_request (RC) { //stopword, lemmatization, token.
    ...
  }
  Return RC' ; } // return les mots clés de la requête
  Select LS from E where (domaine == D);
  If (existe = true) then {
    For (int i=0 ; i < LS.size(); i++){ //prétraitement pour toutes les descriptions de LS.
      String Pre_processing_descption(desc){...} // return les mots clés de la description
    }
    Double Similarity (RC', desc) { //calcul de similarité sémantique entre la requête
      // et service avec WordNet.
      ...
      If similarity > Val // la valeur minimale de similarité
      Return LS' // liste final de la découverte
    }
  }
Else
  Println ("service non existe");
End
```

*Figure 22 : Pseudo algorithme de découverte de service*

- Il va extraire tous les services de différent fournisseur qui appartient au domaine sélectionné par le client (les services qui existe dans l'annuaire de broker).
- A partir de ces services le broker va chercher les services les plus proches à la requête entré d'une manière automatique, pour faire ça le broker passe par ces étapes :
  - D'abord, un pré-traitement à la requête de client en enlevant tous les mots de vide, tokenisation, stemming/lemmatization pour extraire l'ensemble de mots clés de cette requête. Idem pour la partie fonctionnelle de chaque description de service de ce domaine.
  - Puis, Calculer la similarité sémantique (correspondance sémantique) entre les mots clés de requête et les mots clés de chaque partie fonctionnelle de service à travers API WordNet (est une base de données lexicale pour la langue anglaise).
  - Enfin, A partir des résultats obtenus, le broker choisi les services dont lequel son résultat de similarité sémantique est grande comme une réponse de requête.
- S'il n'y a aucun résultat, le broker donné le choix au client de reformuler sa requête ou de composer des services pour créer un nouveau service composé répondre à son besoin.
- Les résultats de la découverte seront utilisés dans l'étape de sélection afin de sélectionner le meilleur service.

### **III.6 Sélection des services**

Comme nous l'avons mentionné précédemment dans le chapitre précédant sur les travaux connexes, un nombre limité de travaux ont examiné la question de la sélection des services dans le Cloud.

Dans cette section, nous décrivons quelques algorithmes pour la sélection de fournisseurs SaaS pouvant répondre à la demande du consommateur de

services. Le client choisit le meilleur service qui satisfait leur requête à partir des résultats qui ont été obtenus précédemment dans la phase de découverte. Il fait un choix entre les méthodes de sélection qui vont calculer le meilleur service qui existe.

Nous avons proposé deux méthodes les plus efficaces et les plus fréquemment utilisées dans le domaine de sélection de services, ils sont comme suit :

- **Méthode 01** : prend en compte uniquement les problèmes non fonctionnels (principalement la qualité de service) dans le processus de sélection.

Un consommateur de services peut avoir ses propres préférences en termes de qualité de service, pour le service qu'il souhaite obtenir. De même, les fournisseurs SaaS peuvent avoir des offres QoS différentes pour chaque service fourni. Nous supposons que les indicateurs de qualité de service sont sous forme normalisée avec des valeurs comprises entre 0 et 1. Une valeur de 1 correspond à la qualité la plus élevée et 0 à la qualité la plus basse.

Soit  $P = \{X_1, X_2, X_3, X_4, \dots, X_n\}$  être le vecteur de paramètres de QoS pris en compte dans le système.

Soit  $S_i = \{SP_1, SP_2, SP_3, SP_4, \dots, SP_k\}$  l'ensemble des fournisseurs SaaS pouvant fournir le service demandé par le consommateur de services. Soit  $Q_i = \{q_{1i}, q_{2i}, q_{3i}, q_{4i}, \dots, q_{ni}\}$  avec  $0 < q_{ij} < 1$ , l'offre QoS du fournisseur SaaS  $SP_i$ .

Le gestionnaire de sélection du CSB calcule la fonction score qui peut être définie comme suit :

$$U = w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n$$

tel que  $w_i$  est le poids d'un critère de QoS attribué par le consommateur, en calculant la fonction score pour chaque fournisseur SaaS, nous obtenons le classement des services de l'offre la plus élevée au plus bas.

Figure 23 : pseudo algorithme de la méthode agrégation linéaire pour la sélection

À la réception d'une demande de contrat de niveau de service, Cloud Service Broker mappe les attentes du consommateur de service en valeurs de QoS normalisées.

○ **Méthode 02 :**

Le MCDM (Multiple-criteria decision-making) est devenu un nouveau domaine de recherche important qui a une immense portée pratique dans de nombreux problèmes scientifiques, logistiques, techniques et industriels. Il peut être défini comme un ensemble de méthodologies pour la comparaison, le classement et la sélection de plusieurs alternatives ayant plusieurs attributs.

➤ Etant donné les vecteurs suivants :

*1- Services :* Soit  $S = \{S_1, S_2, S_3, \dots, S_l\}$  ensemble des services qui existe dans l'annuaire de broker, tel que  $l \geq 2$

2- *Critère de performance* : Soit  $C = \{c_1, c_2, c_3, \dots, c_m\}$  ensemble des critères de service tel que  $m \geq 2$ .

3- *Fonctions de mesurent de performance* : Soit  $F = \{f_1, f_2, f_3, \dots, f_m\}$   $f_i$  représente la fonction de chaque critère  $C_i$ , tel que  $m \geq 2$ .

4- *Vecteur de description de service de fournisseur* : Soit  $D = \{d_1, d_2, d_3, \dots, d_m\}$  un ensemble qui décrit un service  $S_i$ , tel que chaque  $d_i$  représente la performance d'un service  $S_i$  et  $d_i = f_i(S_i)$ .

5- *Matrice de décision* : les vecteurs de description de service peuvent combinés dans une matrice de décision  $A$  de taille  $l * n$  comme suit :

$$\begin{array}{cccc} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ & & & \vdots \\ & a_{l,2} & & a_{l,n} \end{array}$$

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- *Vecteur de description de service de consommateur* : Soit  $R = \{r_1, r_2, r_3, \dots, r_m\}$  est un vecteur de QoS requière donné par le consommateur tel que  $r_i$  est la valeur minimale de critère  $c_i$  accepté.

La sélection de service se fait par la comparaison entre le vecteur de QoS de fournisseur  $D$  et le vecteur de consommateur  $R$  afin de choisir les meilleurs services.

7- *Vecteur de poids* : chaque consommateur peut avoir sa propre préférence de QoS, et chaque critère a une importance relative. L'utilisateur exprime la priorité de chaque critère par un poids Tel que  $W = \{w_1, w_2, w_3, \dots, w_m\}$ .

- D'abord, Nous soustrayons chaque vecteur de la matrice de décision 'A' la forme vectorielle des besoins de l'utilisateur.
- Ensuite, Nous calculons le produit entre le vecteur de poids et résultat de la soustraction de la matrice 'A'.

- Ainsi, en multipliant la matrice par le scalaire -1.
- Puis, Nous utilisons une fonction exponentielle pour limiter l'effet de la suppression mutuelle entre les critères supérieurs et inférieurs aux besoins de l'utilisateur.
- Après, on calcule la somme vectorielle de la matrice pour chaque ligne et la mettre dans un vecteur list\_result.
- Enfin, on fait le tri descendant de list\_result pour obtenir la liste des meilleurs services.

Le pseudo algorithme de la méthode MCDM :

```
Input: S, C, F, D, R, W;
Output: list_result;
Begin
    Create matrix A ; create float list_result; float sum
    S = {S1, S2, S3, ..., Sl} // liste des services
    C = {c1, c2, c3, ..., cm} // les valeur de critère de fournisseur
    R = {r1, r2, r3, ..., rm} // les valeurs de critère requièrent
    W = {w1, w2, w3, ..., wm} // les poids de critère

    For (int i=0; i<m; i++) { // remplir la matrices
        { For(int k=0; k<l; k++){
            A[i][k] = ci - ri; }}
        Sum =0, // la somme d'une ligne de matrice
        j=0; // indice de la list_result
```

```
For (int i=0; i<m; i++) {
  {For (int k=0; k<l; k++) {
    sum= Math.expntiel ((A[i][k] * w1) * (- 1)) + sum;
  }
  list_result[j] = sum;
  j++;
}
Collections.sort(list_result); // trie la liste list_result descendant
```

*Figure 24 : Pseudo algorithme de la méthode MCDM pour la sélection de service*

### **III.7 Composition de services**

La composition est une opération qui permet de créer des chaînes de connectivité bien ordonnées à partir de services atomiques, afin de satisfaire des requêtes complexes qui ne peuvent pas être satisfaites à partir de services atomiques isolés.

Dans cette section, Nous allons proposer une méthode qui permet de résoudre le problème de composition de service comme suit :

- Lorsque le consommateur n'a pas trouvé le service qu'il cherche, il passe au deuxième scénario qui est la composition de service, cet fois il exprime son besoin avec plusieurs requête simple  $R = \{r_1, r_2, r_3 \dots r_n\}$  ces requêtes représentent la découverte de différents services nécessaires qui doivent être inclus lors de la composition de service cloud afin de satisfaire leur besoin.
- Le broker reçoit les requêtes et faire les étapes suivantes :
  - Au premier lieu, une découverte est faite pour chaque requête 'R<sub>i</sub>' afin d'extraire les services candidats 'SC' (générer la population initial) qui ont une relation avec la requête de manière automatique, et les mettre

dans une liste 'L\_candidat' (exemple :  $L\_candidat = \{c1, c2, c3 \dots cm\}$ ), 'm' est le nombre de candidat d'une requête  $R_i$  (chaque requête  $R_i$  a une liste de candidat unique).

- Ensuite, il passe à la deuxième étape qui est la sélection, il applique l'un des méthodes de composition qui est la sélection locale, tel que sur chaque liste de candidat il calcule le meilleur service candidat qui existe par l'un des fonctions de sélection (Score /MCDM) qui sont déjà traité au-dessus, et le mettre dans une nouvelle liste 'Sol\_candidat'.
- Lorsque l'étape précédente terminée, on obtient une liste de solution candidat prête à utiliser.
- Pour qu'on puisse composer les services un algorithme d'ordonnement est obligatoire, dans notre cas on va utiliser l'algorithme de graphe plan et l'algorithme FIFO pour ordonner les services. Cet ordonnancement se fait en 3 étapes :

 *Etape 1* : sélectionner les services initiaux

Dans cette étape on va sélectionner les services initiaux sur les quel l'ordonnancement va l'utiliser pour commencer son travail ; pour faire ça il faut :

- Faire une correspondance entre les inputs d'un service et tous les outputs de tous les autres services qui reste, s'il n'y a pas une correspondance entre eux donc le service est désigné comme un service initial, et le mettre dans une liste Service\_initiaux. Et passe à un autre service et faire le même processus.

 *Etape 2* : après avoir la liste des services initiaux, on va les placer dans matrice  $M\_chemin$  horizontalement comme illustre l'exemple suivant :

$Sol\_candidat = \{S1, S2, S3, S4, S5, S6, S7, S8\}$  // le meilleur service de chaque requête  $R_i$

Service\_initiaux= {S1, S3} // résultat de service initiaux

Service\_resté= {S2, S4, S5, S6, S7} // liste de service qui reste

$\begin{pmatrix} S1 \\ S3 \end{pmatrix}$  // matrice M\_chemin

A partir de la matrice M\_chemin on va faire toutes les correspondances sémantiques entre les outputs des services qui existe dans la matrice et la liste Service\_resté pour construire les chemins et avoir l'enchaînement des services comme illustre l'exemple suivant.

$$\begin{pmatrix} S1 & S7 & S8 \\ S1 & S5 & S6 & S8 \\ S3 & S2 & S4 & S8 \end{pmatrix}$$

Figure 25 : Matrice des chemins

- ❖ Les chemins obtenus :
  - Chemin 1 : S1 => S7 => S8
  - Chemin 2 : S1 => S5 => S6 => S8
  - Chemin 3 : S3 => S2 => S4 => S8

✚ Etape 3 : Deux méthode au choix

Méthode 1 : Algorithme de Dijkstra

Dans cette étape on va choisir le meilleur chemin parmi les chemins obtenus qui satisfait le besoin de consommateur de manière automatique, en utilisant l'algorithme de Dijkstra pour l'optimisation comme suit :

Chaque chemin on le calculé sa valeur de fitness à base de fonction score de chaque service qui construit le chemin, et la plus grande valeur de fitness représente le meilleur chemin.

Le meilleur chemin représente les services qui forme une meilleure composition de service qui répond au besoin de client.

Méthode 2: A\* star

L'algorithme A star combine RMC & RA :

- RMC : Priorité par cout du chemin parcouru  $g(n)$  (qui est le score dans notre cas).
- RA : Priorité par l'approche du but  $h(n)$
- Recherche A\* : Priorité par la somme  $g(n) + h(n)$
- Idée : Eviter de développer les chemins qui sont déjà coûteux
- La Fonction d'évaluation  $F(n) = g(n) + h(n)$
- $g(n) =$  Cout pour arriver à  $n$ .
- $h(n) =$  Le Cout estime de  $n$  au But
- $F(n) =$  Le cout estime total du chemin de  $n$  jusqu'au but.

Une heuristique, est une fonction, qui prend un état du monde et elle retourne un nombre.

- D'abord, nous calculons la fonction  $h(n)$  en utilisant la distance Euclidienne entre le dernier nœud (le service final) et le nœud courant pour tous les nœuds de chemin.
- Ensuite, nous calculons la fonction  $g(n)$  qui représente dans notre cas le score de chaque service.
- Enfin, nous calculons la fonction  $F(n) = h(n) + g(n)$  pour chaque chemin.

Le chemin qui a une grande valeur de  $F(n)$  est le meilleur chemin qui satisfait les besoins du client.

- Le résultat obtenu dans les deux méthodes représente un nouveau service composé prêt à utiliser et sera stocker dans l'annuaire de service broker pour réutiliser la prochaine fois. Concernant sa description de service nous avons utilisé la plateforme Bpel qui permet de générer sa description en langage WSDL.

### **III.8 Interface fournisseur**

Elle se compose de 5 interfaces qui sont comme suit :

- Interface « Register » : elle permet au n'importe quel fournisseur cloud de créer un compte au niveau de broker afin de pouvoir publier ses services.
- Interface « login » : elle permet au fournisseur cloud d'accéder à tout moment à son compte et lui permettre de publier ou consulter ses services.
- Interface « Home\_Provider » : cette interface est permise de voir tous les services publier par le fournisseur lui-même et également la possibilité de chercher des services.
- Interface « Publier\_service » permet au fournisseur de publier la description de service, le SLA, et donner des information (nom service, domaine, date de publication...etc) à propos du service à déployer.
- Interface « echange\_donnée » elle permet au fournisseur cloud de recevoir les inputs et envoyer les outputs au broker.

### **III.9 Interface client**

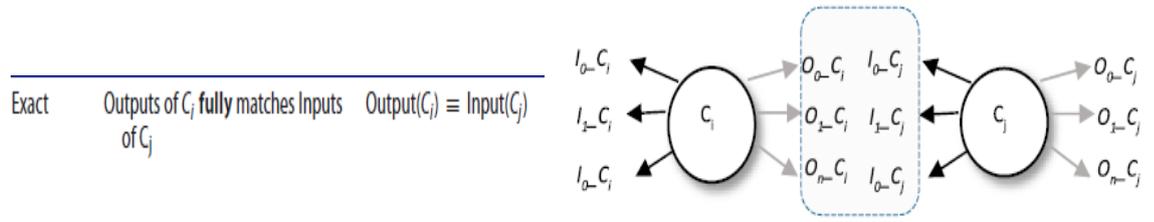
Elle se compose en plusieurs interfaces comme suit :

- Interface « Register » : elle permet au n'importe quel consommateur cloud de créer un compte au niveau de broker afin de pouvoir demander des services.
- Interface « login » : elle permet au consommateur cloud d'accéder à tout moment à son compte et lui permettre de faire plusieurs fonctionnalités.
- Interface « recherche\_service » elle permet au consommateur de chercher et demander les services qui répondent à ses besoins.
- Interface « compose\_service » elle permet au consommateur de composer des services lorsque y'a pas un service qui répond à son besoin.
- Interface « echange\_donnée » elle permet au consommateur d'envoyer les inputs et récupérer les outputs de broker.

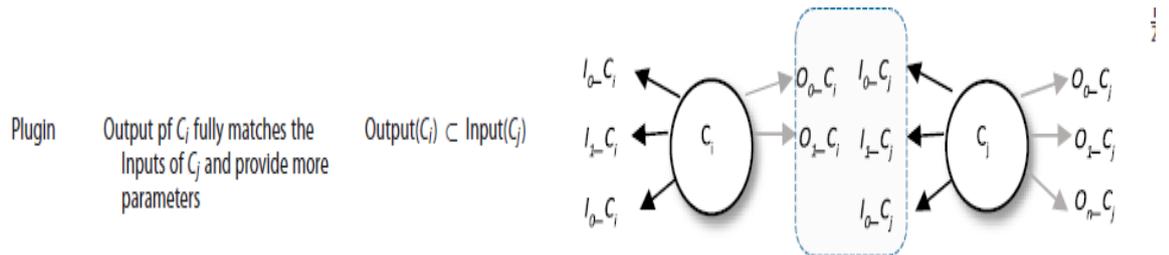
### III.10 Description des règles d'interaction entre les services Cloud

Les relations entre les services qui forment le chemin de composition peuvent être classées en trois catégories qui sont :

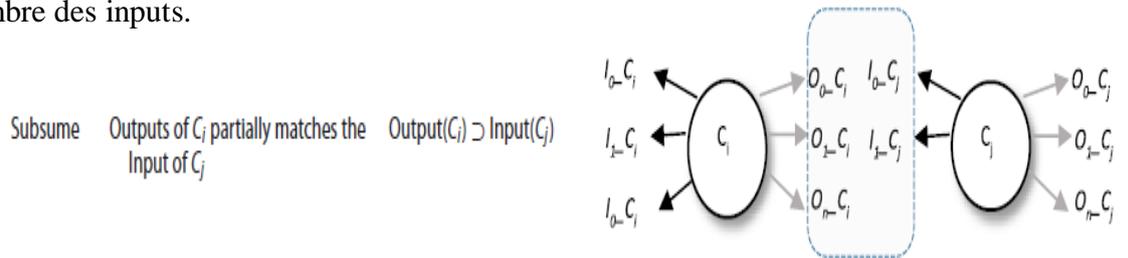
- ✓ Catégorie 01 : « Exact » lorsque le nombre des outputs égale le nombre des inputs.



- ✓ Catégorie 02 : « Plugin » lorsque le nombre des outputs est inférieur au nombre des inputs.



- ✓ Catégorie 03 : « Subsume » lorsque le nombre des outputs est supérieur au nombre des inputs.



- Donc pour faire les correspondances entre les inputs et les outputs des services de composition, il faut respecter les trois catégories ci-dessus.
- Chacun de service candidat doit respecter son tour lors de l'échange de donnée par exemple un cloud SaaS (SaaS1) qui a des input (a,b) et des outputs (c,d) et un autre cloud SaaS (SaaS2) a comme input (c,d) : dans

ce cas le SaaS2 ne peut pas commencer son travail avant que le SaaS1 termine son travail.

### **III.11 Description des mécanismes de communication**

Cette partie est très importante dans le Cloud broker, elle décrit comment le client et le fournisseur communiquent avec le broker. Dans notre conception nous avons utilisé l'architecture client/serveur à travers les sockets tel que le client peut envoyer ses inputs quel que soit sa nature (fichier XML, image, variable, etc.) au broker et reçoit les outputs. La même chose pour le fournisseur il peut envoyer et recevoir les données de la part de broker.

La connexion entre le client et le broker se fait automatiquement c'est-à-dire le client n'a pas besoin de configurer la session de communication.

### **III.12 Conclusion**

Dans ce chapitre, nous avons présenté la conception de notre solution « Cloud broker » en tant que courtier entre les fournisseurs de type SaaS et les consommateurs.

Le broker proposé peut faire la découverte des SaaS, la sélection des meilleurs services et même la composition des services Cloud pour créer de nouveaux services afin de répondre au besoin du consommateur.

Nous avons utilisé différentes méthodes et algorithmes à savoir : MCDM, Agrégation linéaire, A\* et Dijkstra, afin de garantir toutes les fonctionnalités.

L'implémentation de la solution proposée sera décrite dans le chapitre suivant.

### IV.1 Introduction

Arrivé à ce stade nous pouvons nous estimer heureux, il ne reste qu'à commencer à écrire notre code en se basant sur l'analyse et la conception qu'on a obtenu dans les chapitres précédents et faire des tests sur les résultats de notre application.

Dans ce chapitre, nous allons décrire notre prototype de broker à travers des scénarios d'utilisation.

### IV.2 Outils et environnement de développement

Pour la réalisation de notre application, nous avons utilisé des outils différents qui sont :

- **Netbeans**

NetBeans est un environnement de développement intégré pour Java. NetBeans permet aux applications d'être développées à partir d'un ensemble de composants logiciels modulaires appelés modules. NetBeans fonctionne sous Windows, macOS, Linux et Solaris.

- **Java Development Kit:**

Le kit de développement Java est une implémentation de l'une des plates formes Java Edition, Standard Edition, Java Platform, Enterprise Edition ou Java Platform, Micro Edition publiées par Oracle Corporation sous la forme d'un produit binaire destiné aux développeurs Java sous Solaris, Linux, macOS ou Windows.

- **API WordNet :**

WordNet est une grande base de données lexicale en anglais. Les noms, verbes, adjectifs et adverbes sont regroupés en ensembles de synonymes cognitifs (synsets), chacun exprimant un concept distinct. Les synsets sont liés entre eux au moyen de relations conceptuelles-sémantiques et lexicales.

- **Apache Derby**

Apache Derby est un système de gestion de base de données relationnelle développé par Apache Software Foundation qui peut être intégré à des programmes Java et utilisé pour le traitement de transactions en ligne. Il a une empreinte d'espace disque de 3,5 Mo. Apache Derby est développé en tant que projet open source sous

la licence Apache 2.0.

- **Business Process Execution Language (Bpel)**

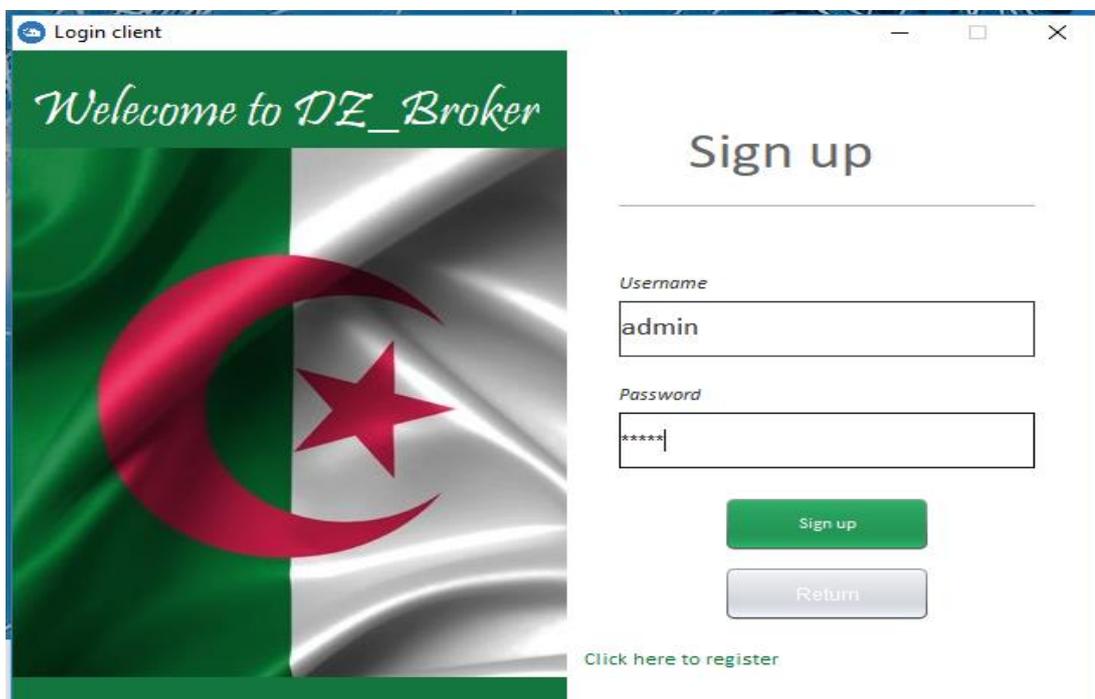
Le langage d'exécution de processus métier de services Web, plus communément appelé BPEL, est un langage exécutable standard OASIS permettant de spécifier des actions au sein de processus métier avec des services Web. Les processus dans BPEL exportent et importent des informations en utilisant exclusivement des interfaces de service Web.

### IV.3 Simulation

Dans cette partie nous allons voir les principales figures de notre application et nous allons essayer de les expliquer avec des exemples.

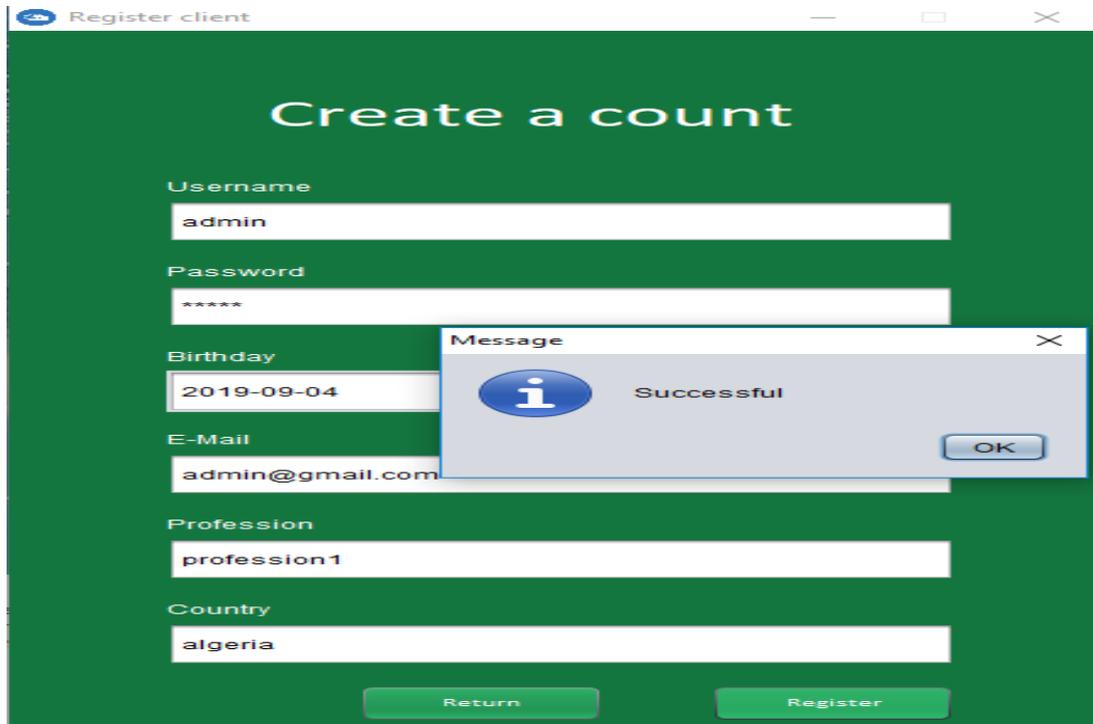
Tous d'abord, nous avons décrit les interfaces de base comme suit :

- Interface Login\_client permet au client de connecter et accéder à son compte comme représenté ci-dessous :



*Figure 26 : Interface login client*

- Cette interface permet de créer un compte client dans le cas où le client n'a pas encore inscrit, il suffit de remplir les champs présenter dans la capture ci-dessous :



The screenshot shows a web browser window titled "Register client". The main heading is "Create a count". The form contains the following fields: Username (admin), Password (masked with asterisks), Birthday (2019-09-04), E-Mail (admin@gmail.com), Profession (profession1), and Country (algeria). At the bottom are "Return" and "Register" buttons. A "Message" dialog box is overlaid on the form, displaying an information icon, the text "Successful", and an "OK" button.

Figure 27 : Création d'un nouveau compte client

- Cette interface représente la page d'accueil de client qui lui permet de chercher un service, consulter les services consommés et contacter l'administrateur broker.

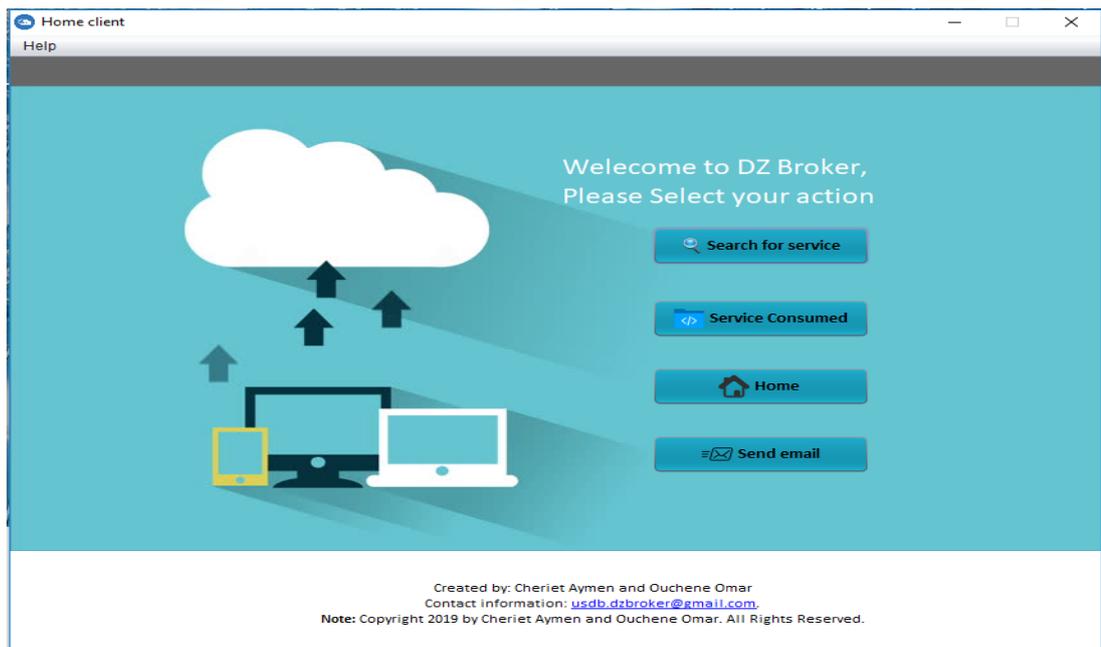


Figure 28 : Interface accueil client

- Cette interface permet de contacter l'admin broker au cas de besoin d'aide ou d'information, il suffit de remplir les champs avec informations valide et existe.
  - Nous avons utilisé ici un API de Java Mail qui permet d'envoyer des emails via une connexion à internet.

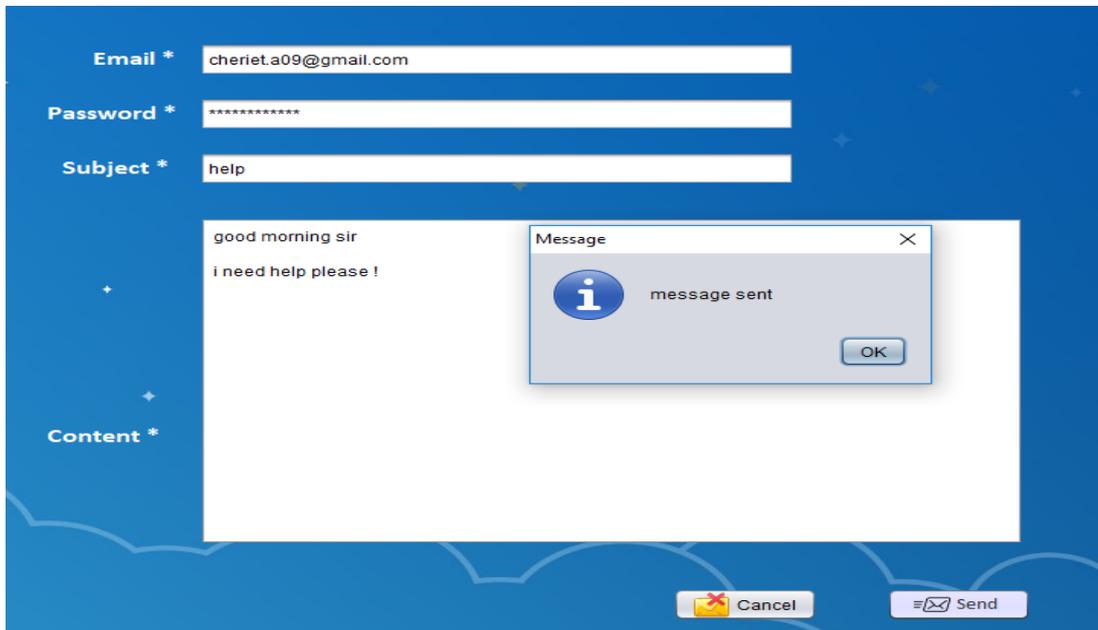


Figure 29 : Interface de nous-contactez

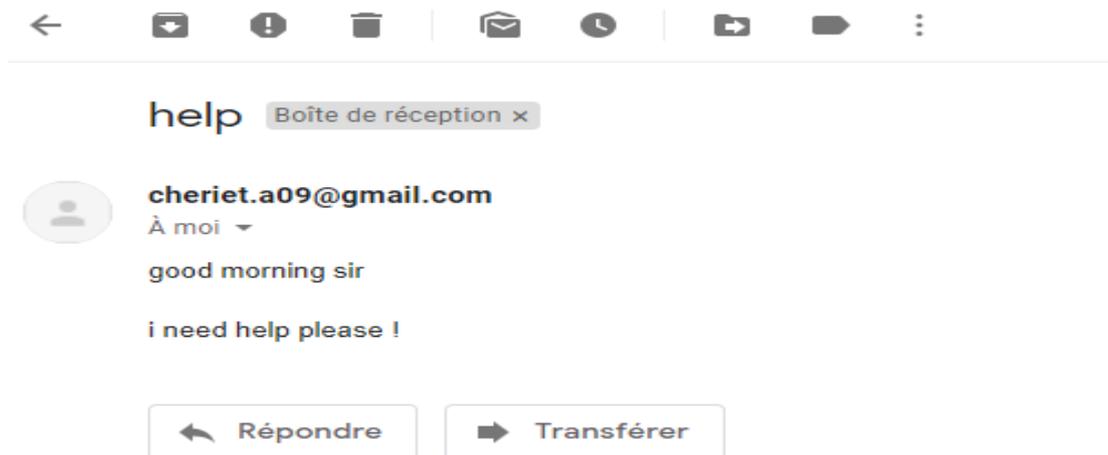


Figure 30 : Capture de réception d'email

- Dans le coté fournisseur, il peut créer un compte, publier un service et consulter la liste des services publier :

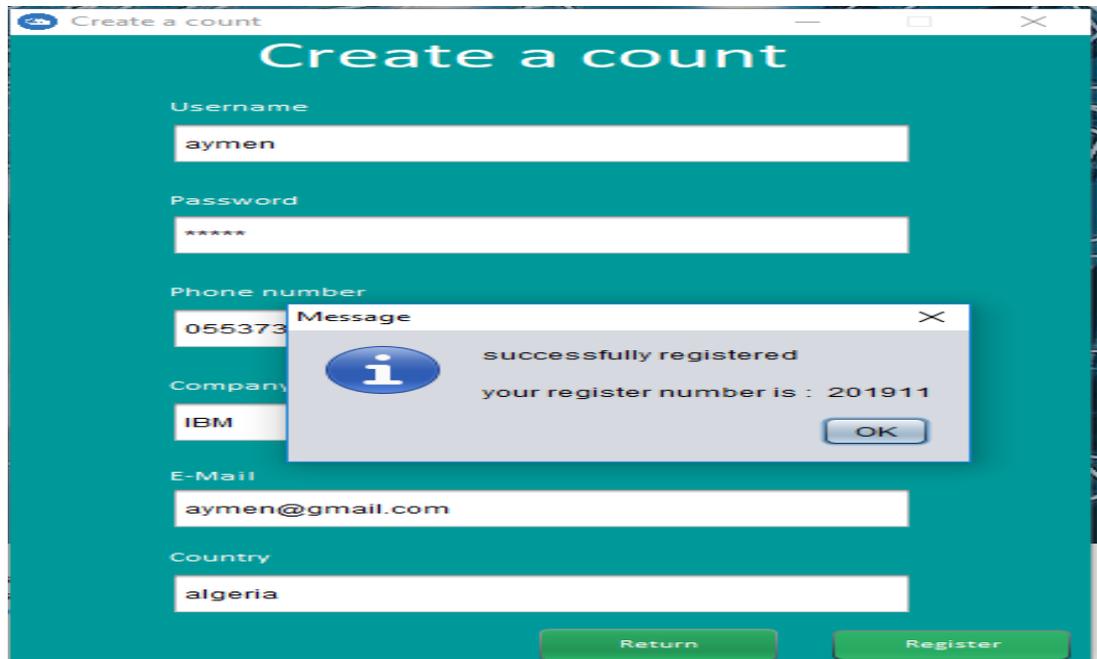


Figure 31 : Interface de création de compte fournisseur

- Cette interface permet au fournisseur cloud de publier leurs services, tous les champs marqués avec étoile sont obligatoires de les remplir ; aussi, il doit attacher la description de service et le contrat de niveau de service.

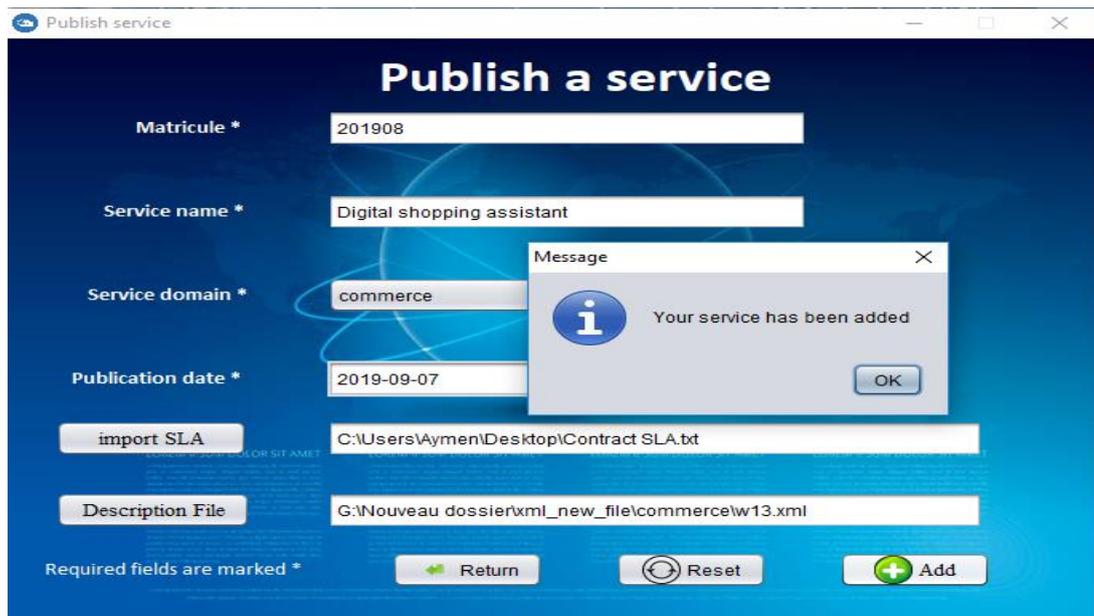


Figure 32 : Interface de publication de service cloud

- Cette interface affiche tous les services de fournisseur qui sont déjà publiés.

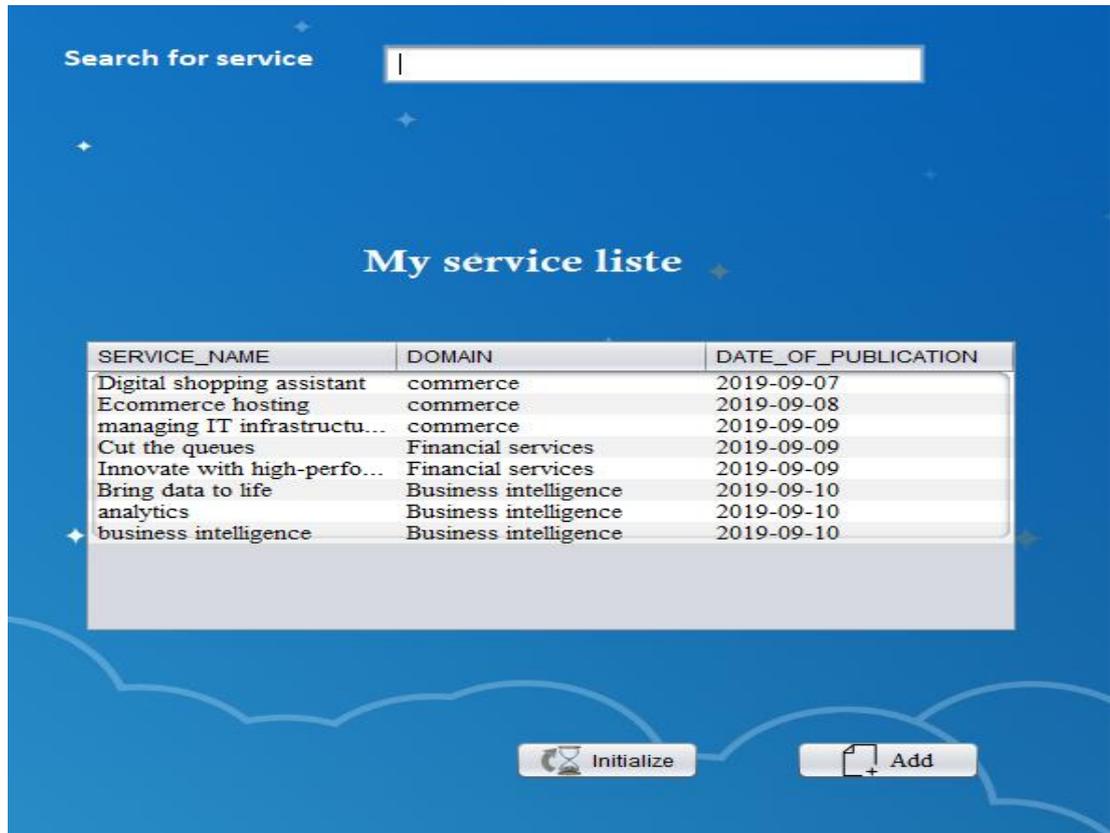


Figure 33 : Interface de la liste de services publiés

❖ Comme présenté dans le chapitre de conception nous allons voir deux scénarios dans la partie simulation qui sont :

### ✚ *Scénario 01 : requête simple (sans interopérabilité)*

Nous allons illustrer ce scénario avec un exemple complet jusqu'à l'obtention de résultat :

➤ Supposons qu'un client cherche un service SaaS qui fait l'hébergement des plateformes de commerce électronique et qui garantit le bon fonctionnement de sa plateforme au période d'hébergement ; le client va suivre le processus suivant afin de trouver son besoin :

- Au premier lieu, le client connecter avec son compte (comme présenté au-dessus) ou créer un compte si n'a pas encore inscrit.
  - Au deuxième lieu, le client passe à l'interface "Search for service" qui est ci-dessous et sélectionne le domaine de son besoin ; dans cet exemple le client sélectionne le domaine "Commerce" et soumet sa requête « search for flexible and scalable e-commerce hosting capabilities ».
  - Ensuite, il clique sur le bouton "Search" ; toute un processus est fait dans le background de l'application avant retourner les résultats.
- 
- D'abord, la requête soumise par le client va être traité de manière automatique pour éliminer les mots vides et extraire les mots clés (voir le chapitre de conception)
  - Après, des correspondances est faites entre les mots clés et les descriptions du service afin d'extraire les services qui ont une relation avec la requête de client.
  - Enfin, les services qui ont une grande valeur sémantique retourné par le Wordnet sera renvoyer à l'interface pour d'afficher.

- Une liste des services similaires apparue au client ; dans ce moment, le client a besoin de sélectionner le meilleur service qui existe entre eux donc il remplir le poids de chaque critère de qualité de service et les valeurs requises selon son besoin et choisir la méthode de sélection en cliquant sur le radio-bouton qui est devant la méthode de sélection.

Comme nous avons vu dans la conception dans la partie sélection nous avons deux méthodes de sélection MCDM et Agrégation linéaire. Le meilleur service est apparu dans le TextField, dans notre exemple “Ecommerce hosting” est celui le meilleur dans les deux méthodes de sélection.

Select a domain	
commerce	

Enter your request

search for Flexible and scalable hosting capabilities

Service list	
SERVICE_NAME	DOMAINE
Digital shopping assistant	commerce
Ecommerce hosting	commerce
managing IT infrastructure	commerce

performance criteria

	Wights	Required
availability	0.8	0.8
Reliability	0.7	0.7
Response time	0.2	0.2
Evolutivity	0.1	0.1
Cost	0.8	0.8
Security	0.9	0.9
Transparence	0.8	0.8
Relevance	0.5	0.8

Select a method

Lineare method  MCDM method

Best service with lineare methode: Ecommerce hosting

Best service with MCDM methode: Ecommerce hosting

[Click here to compose service](#)

Return Select

Figure 34 : interface de recherche de service

- Après avoir le meilleur service, le client a le choix d'accepter ou refuser le service ; dans le cas où il accepte, le service sélectionné sera ajouté à sa liste de service consommé comme représenté dans la capture suivante de notre exemple.
- Lorsque le client veut utiliser le service choisi, il clique sur son nom ensuite, clique sur le bouton " Use service "

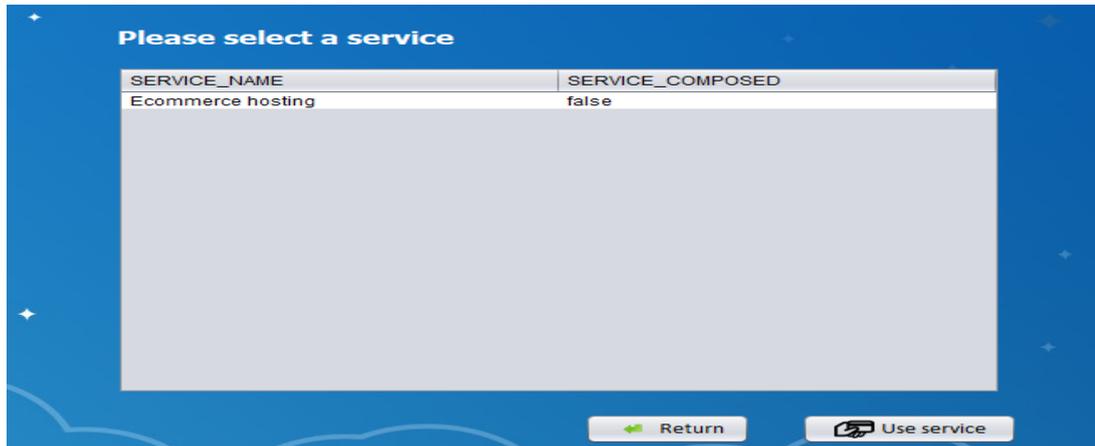


Figure 35 : interface de service consommé

- Une fois cliquer sur " Use service ", le contrat de ce service apparaît pour continuer le processus de la demande de service, une confirmation est obligatoire pour qu'il puisse profiter des services de ce service SaaS. Le contrat de ce service s'appare comme ci-dessous :

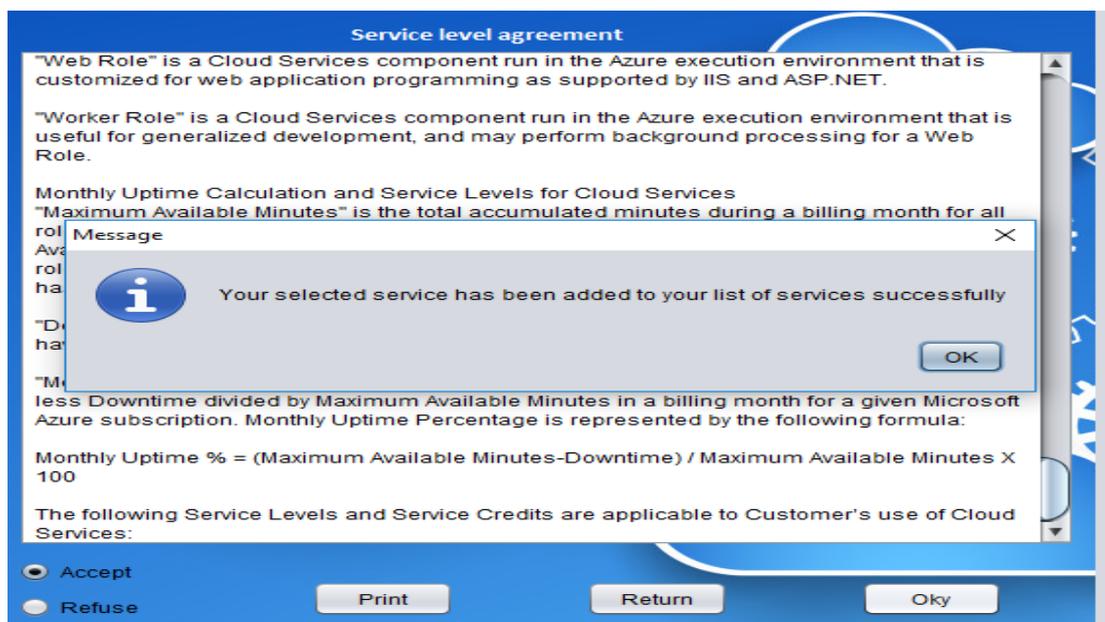


Figure 36 : interface de contrat de niveau de service

- Comme une dernière étape, il reste au client que l'échange de données, cet fois le client sélectionne les inputs à envoyer quel que soit sa nature (fichier, image, xml, etc.) nous avons utilisé l'architecture client-serveur à travers les sockets pour envoyer et recevoir les données comme suit :
  - Interface client : le client clique sur le bouton "Connect" pour ouvrir une session de communication avec le broker afin d'envoyer ses inputs, et clique sur le bouton "Start" pour joue le rôle d'un serveur afin de recevoir les outputs. Et ça se fait sans configuration demander au client (par exemple : numéro de port ou, adresse IP) car sont déjà configuré statiquement. Lorsque la connexion a établi, des informations sur la connexion s'affiche sur le rectangle noir.
  - Pour commencer a envoyé les inputs, le client doit sélectionner le type de fichier et donner son nom et clique sur le bouton "Send" pour les envoyés.

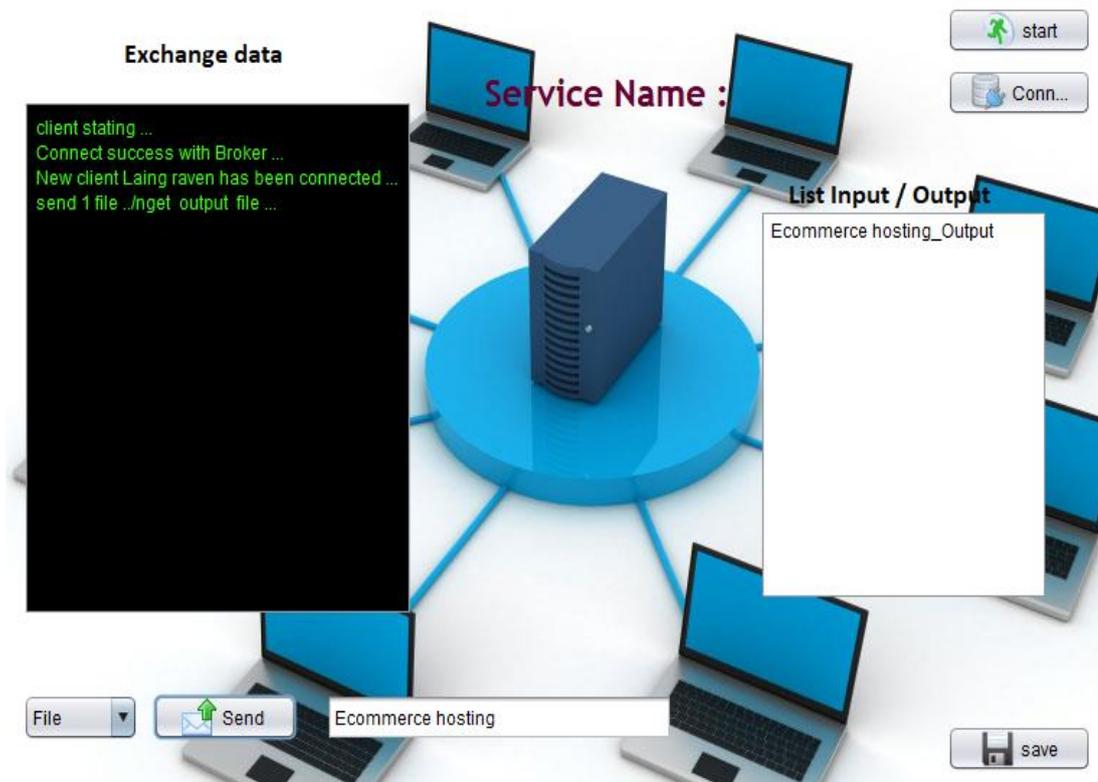


Figure 37 : interface d'échange de donnée coté client

- Interface broker : lorsque on clique sur le bouton “Start server” le broker se mit en attente d’une connexion arriver afin de répondre au besoin de client, après la réception des inputs, le broker consulte le fournisseur cloud propriétaire de service consommer et lui envoyé les inputs de client, et attendre la réponse ; le fournisseur à son tour, il va exécuter la demande de broker et renvoyer les résultats.

Dans notre Framework broker le fournisseur est désigné comme une boîte noire on le donne des inputs et on récupère les outputs.

Lorsque le broker reçoit les outputs, il va les retransmettre au client.

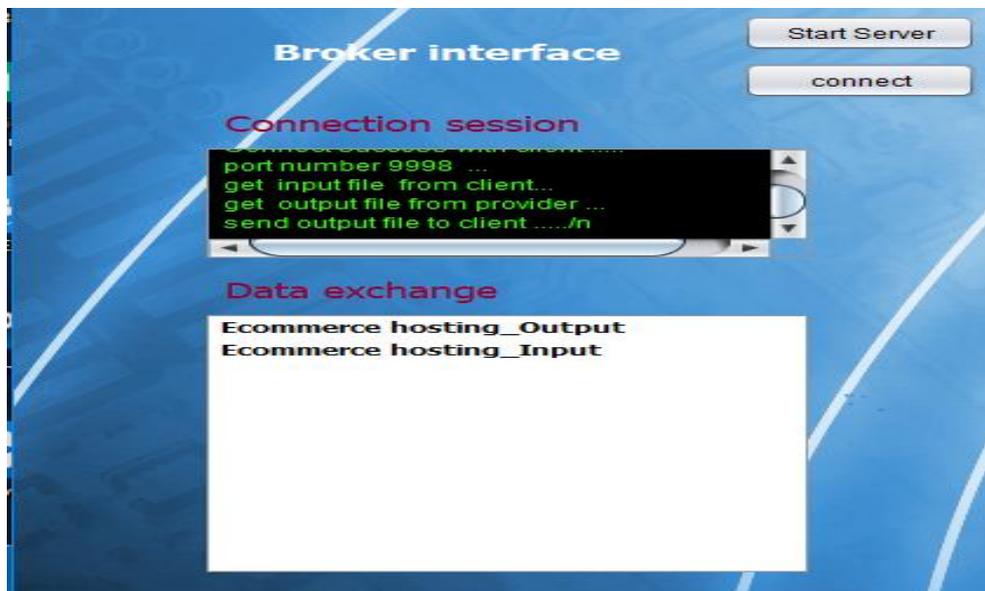


Figure 38 : interface d’échange de donnée coté broker

- Interface fournisseur : pour la communication et l’échange de données entre le broker et le fournisseur, nous avons utilisé l’architecture client-serveur à travers les sockets qui est le même concept que le client et le broker ; le fournisseur lance le serveur afin de recevoir les inputs en cliquant sur ‘Start server’ de la part de broker, et lorsque termine l’exécution de la demande il renvoi les outputs au broker après de l’établissements de session avec le broker grâce au bouton ‘Connect’ ; qui à son tour renvoie les outputs au client. Comme le montre la capture ci-dessous :

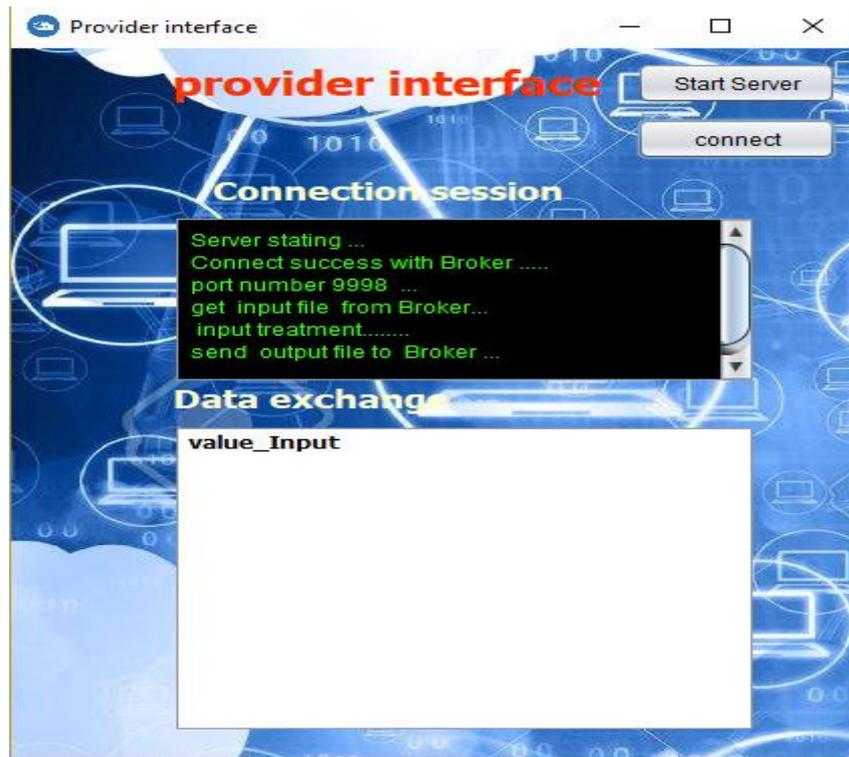


Figure 39 : interface d'échange de données partie fournisseur

- A la fin de l'opération, le client reçoit les résultats de la part de broker, qui s'affichent sur le rectangle blanc comme une notification afin de sauvegarder les résultats. Et dans ce cas le client a pu réussir dans son opération.

✚ Scénario 02 : requête complexe (avec interopérabilité)

Nous allons illustrer ce scénario avec un exemple complet jusqu'à l'obtention de résultat :

- Supposons qu'un client cherche un service SaaS d'agence de voyage qui offre plusieurs fonctionnalités (réservation d'hôtel, guide touristique, assurance de voyage, ..., etc.) ; il a essayé d'appliquer le processus qui est d'écrit au scenario 01, mais n'a pas réussi à obtenir un résultat ; et la capture ci-dessous est apparu, propose une composition de service comme suit :

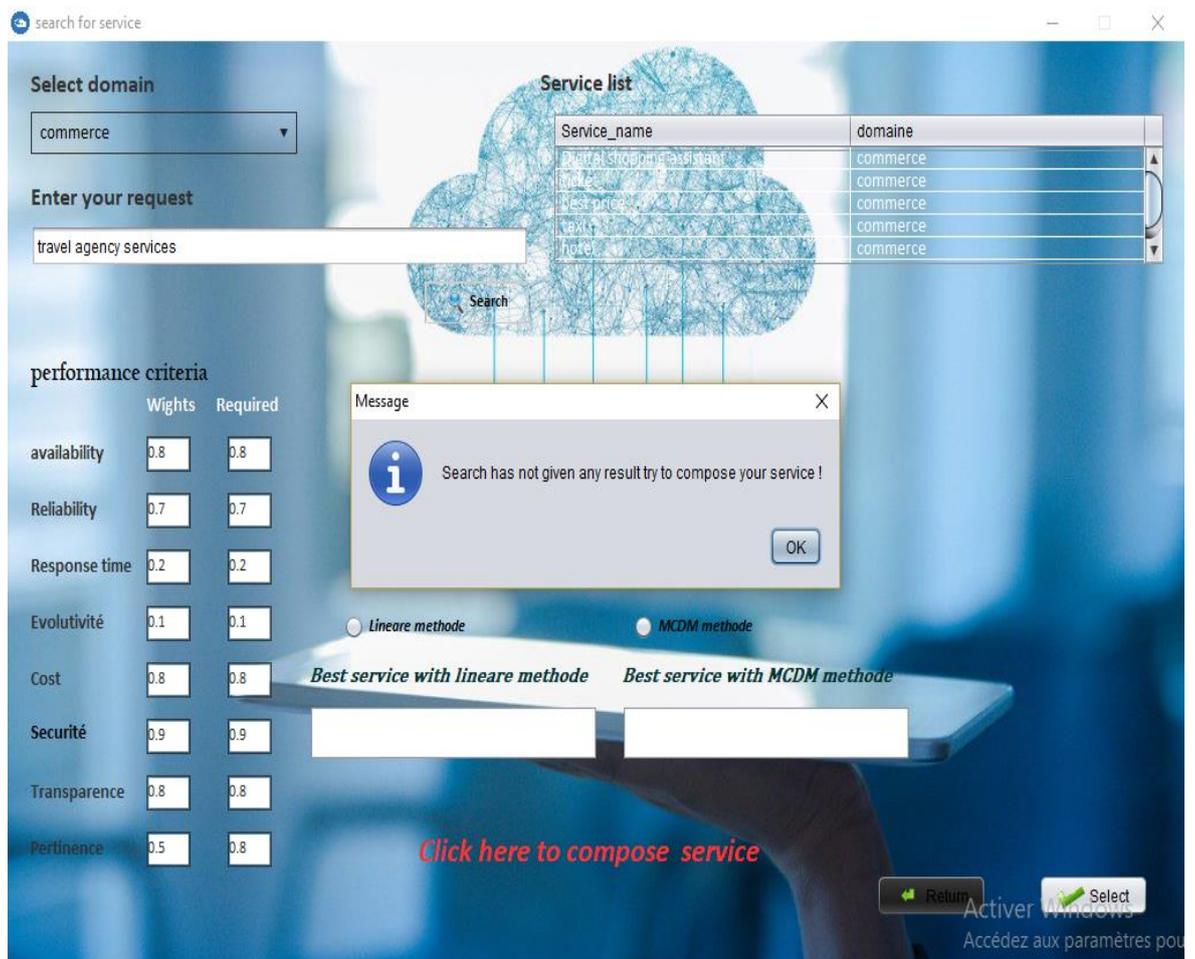


Figure 40 : interface de recherche de service

- Le client n'a pas obtenu de résultat en raison de :

Lorsque le processus de découverte n'a pas trouvé aucun service simple dans l'annuaire de service broker qui répond au besoin de client, il lui propose une composition de service qui regroupe plusieurs services candidats selon les requêtes soumises par le client afin de créer un nouveau service composé qui répond au besoin de client et sera intégré dans l'annuaire de service pour une nouvelle utilisation. La composition de service se fait au niveau de l'interface suivante :

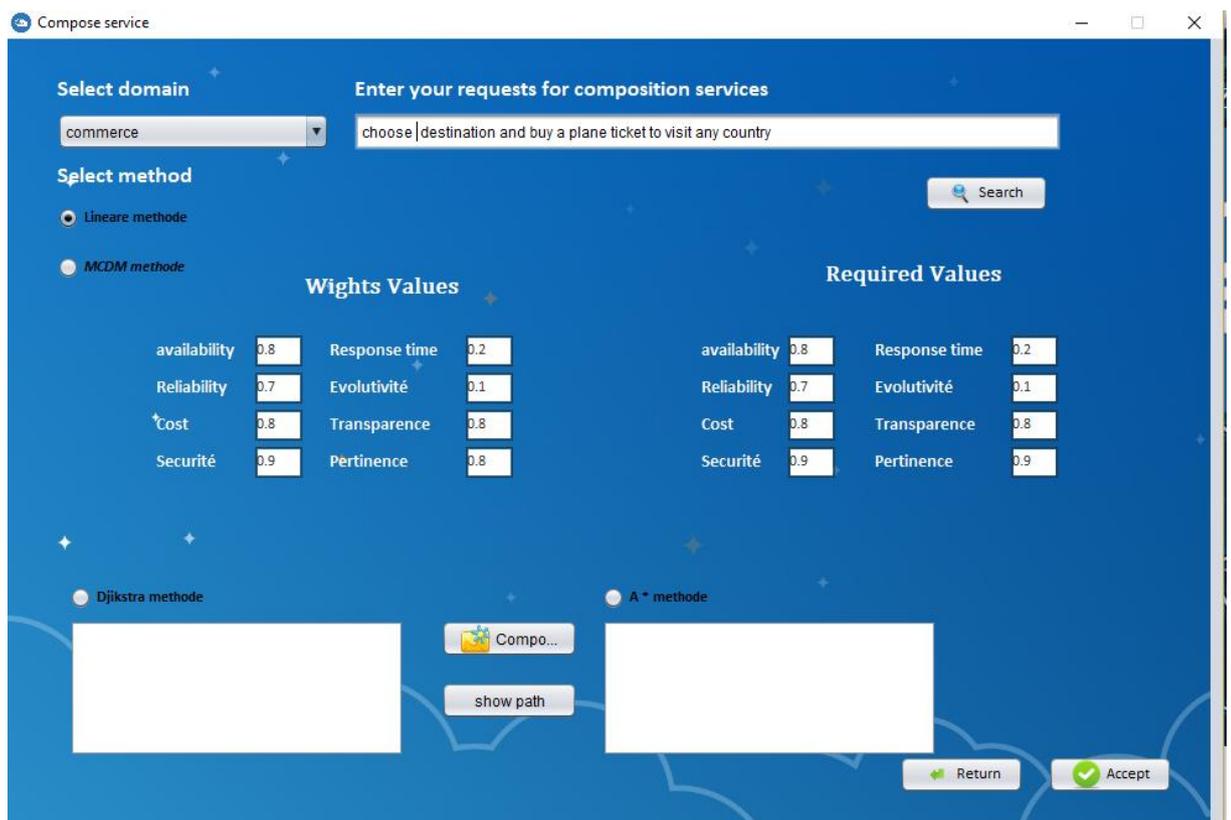


Figure 41 : interface de composition de service

- Dans ce scénario le client sélectionne le domaine de son besoin, et comme un exemple le client sélectionne le domaine "Commerce" et soumet leurs requêtes :
  - Requête 01: « choose destination and buy a plane ticket to visit any country »
  - Requête 02: « rental a car to move and find all the models »
  - Requête 03: « book a hotel with a good price, find five stars hotel in a good place »
  - Requête 04: « tourist guide to visit a beautiful place »
  - Requête 05: « the best insurance with a good price »

- A chaque fois le client saisi sa requête il doit cliquer sur le bouton search après choisi la méthode de sélection de service :
  - Lorsqu'il clique sur le bouton "Search" le broker fait une découverte de service selon la requête entrée et retourne un ensemble de service ; et après fait la sélection de meilleur service selon la méthode de sélection (MCDM ou Agrégation linéaire) utilisé par le client (le processus de sélection se fait de même manière de scenario 01 précédant) ;
- Lorsque le client termine la saisi des requêtes, il clique sur le bouton "Composition" pour générer les chemins, dans ce cas toute un processus est fait lors de la composition, comme nous avons expliqué dans le chapitre conception nous avons utilisé l'algorithme de « Graphe plan » pour faire l'ordonnancement de chemin à l'aide des fonctions de correspondances entre les inputs et les outputs de chaque service. La capture suivante montre les résultats de chemin qui sont construit :
- Une fois les chemins de composition sont construits grâce à l'algorithme « Graphe plan », il passe à l'étape suivante qui est l'optimisation de composition de service, dans cette étape nous avons utilisé deux méthodes d'optimisation A\* (star) et Dijkstra.
- Le client choisit l'un des deux méthodes ; afin de sélectionner le meilleur chemin qui existe et ça se fait grâce à la méthode choisie de composition de service.

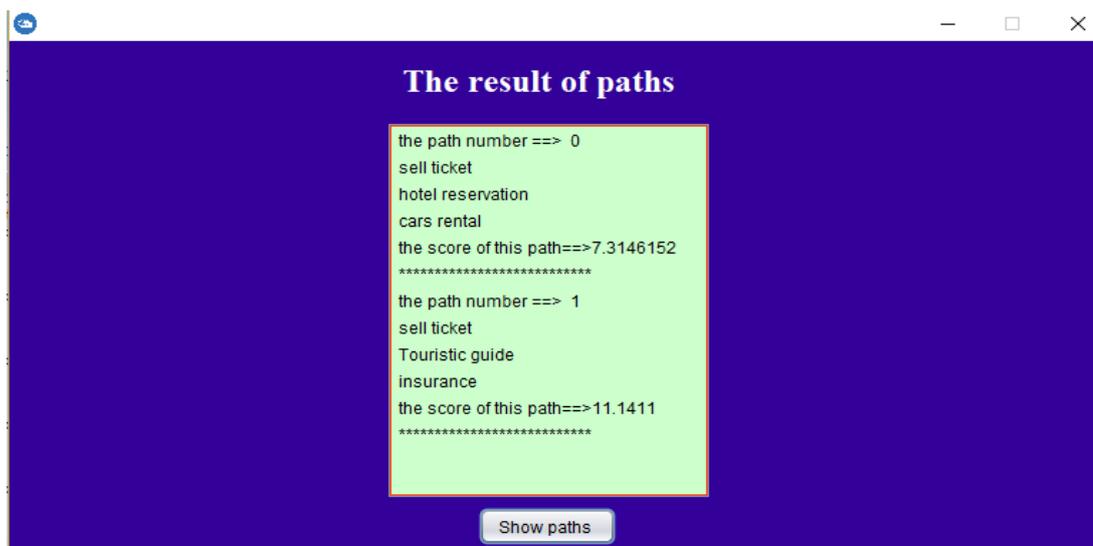


Figure 42 : Interface de résultat de chemin après la composition

- Nous avons utilisé dans notre exemple la méthode d'optimisation A\* qui a donnée comme résultat de meilleur chemin les noms des service suivants :  
Sell ticket → Touristic guide → Insurance ; comme indiqué dans la capture ci-dessous :

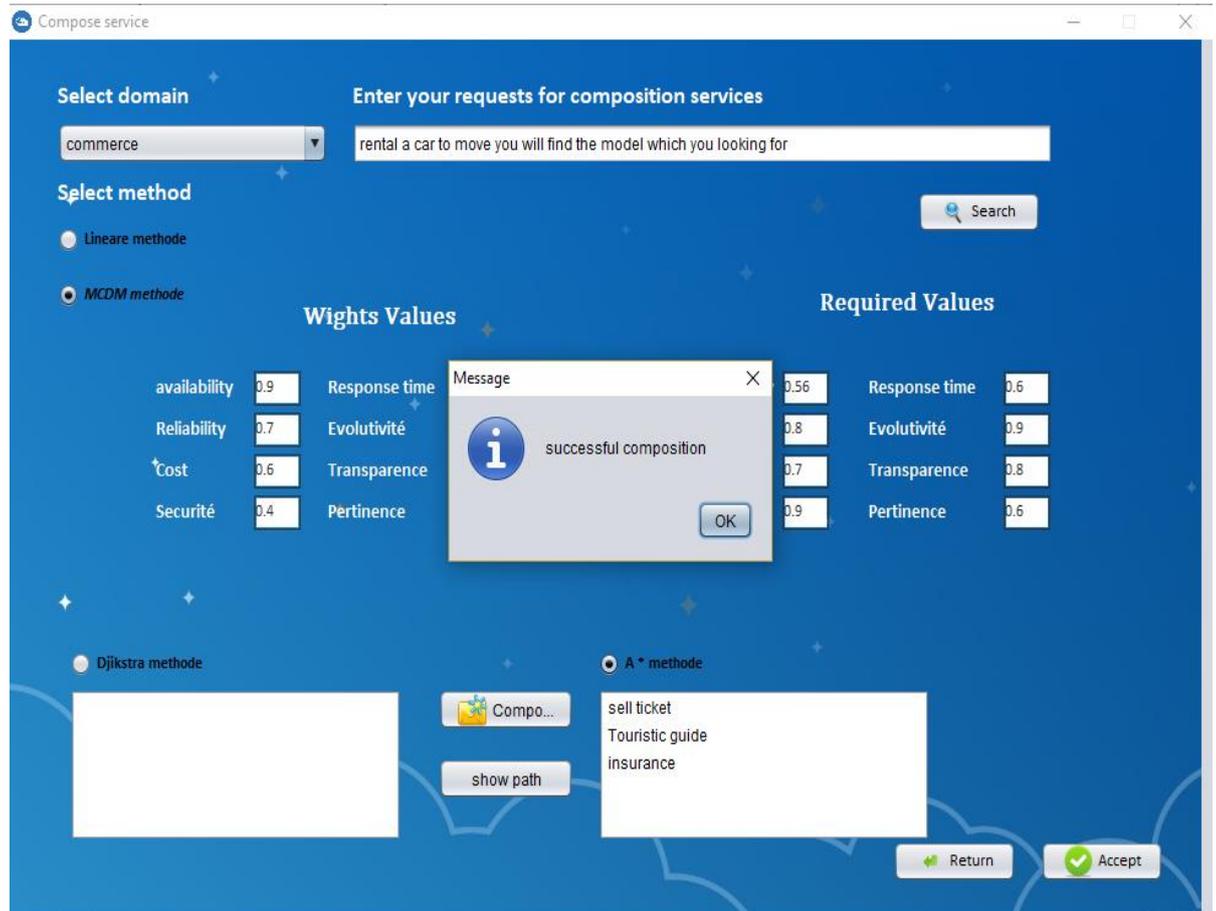


Figure 43 : Résultat du meilleur chemin de composition

- Comme une dernière étape de composition de service, nous allons utiliser la plateforme Bpel qui va générer la description de service composé, pour faire ça nous a besoin le chemin de service composé et tous les inputs et les outputs de chaque service afin de générer le diagramme Bpel et obtenu la description WSDL de service composé.
- Le résultat de notre exemple est présenté dans les captures ci-dessous elle regroupe les trois services candidat qui forme le meilleur chemin afin de répondre au besoin de client.

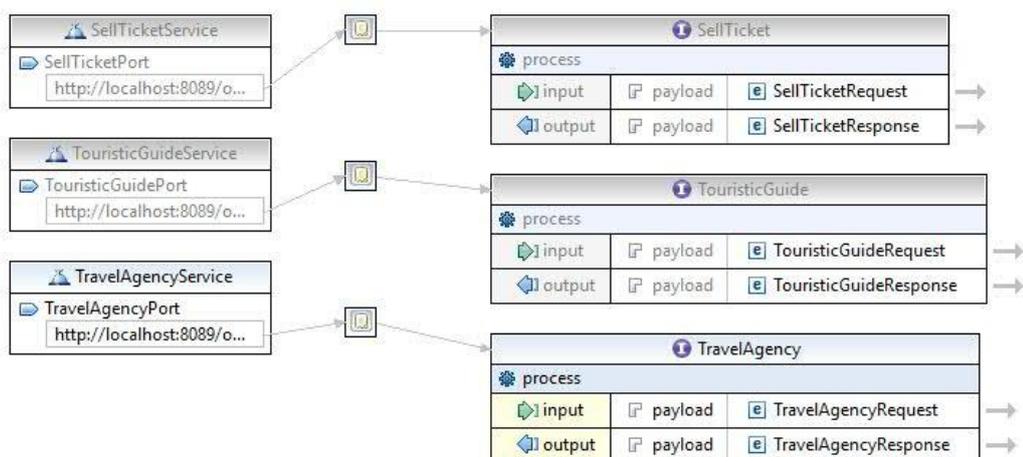
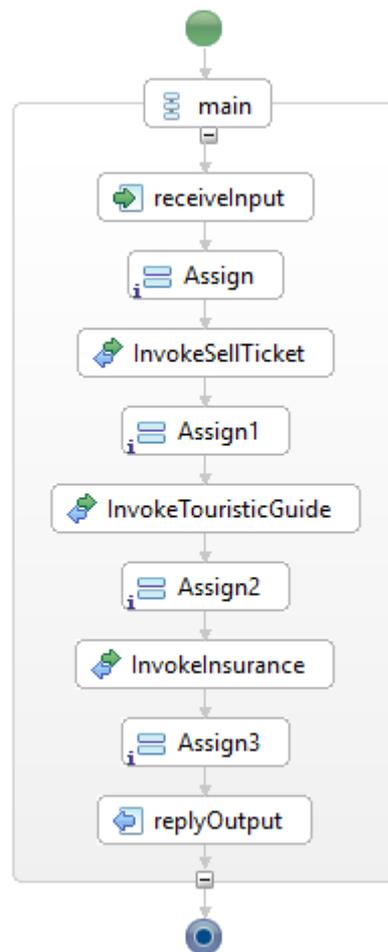


Figure 44 : Résultat de composition de service

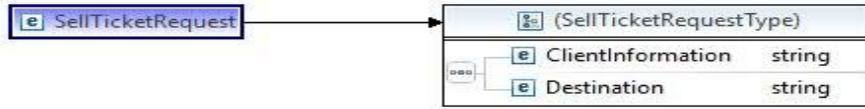


Figure 45 : les inputs de service Sell Ticket

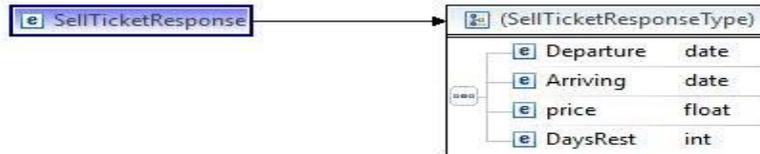


Figure 46 : les outputs de service Sell Ticket

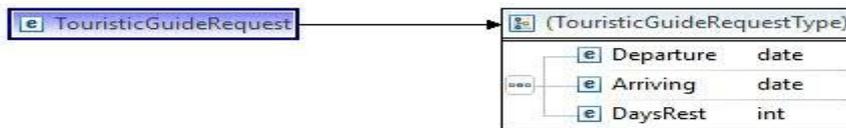


Figure 47 : les inputs de service Touristic guide



Figure 48 : les outputs de service Touristic guide

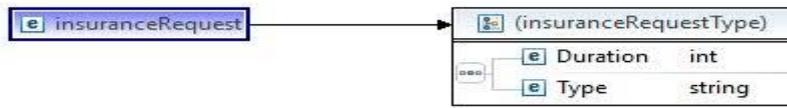


Figure 49 : les inputs de service Insurance



Figure 50 : les outputs de service Insurance

➤ Le point le plus important dans notre Framework est arrivé à répondre au besoin de l'utilisateur quel que soit la complexité de sa requête, donc la composition de services joue un rôle très sensible sur ça ; et pour bénéficier de cette composition, il faut que réutiliser les services composés afin de réduire le temps de réponse et minimiser le coût de tarification ; pour cela nous avons ajouter dans notre Framework la possibilité d'enregistrer et rechercher des services composés comme le montre la capture suivante :

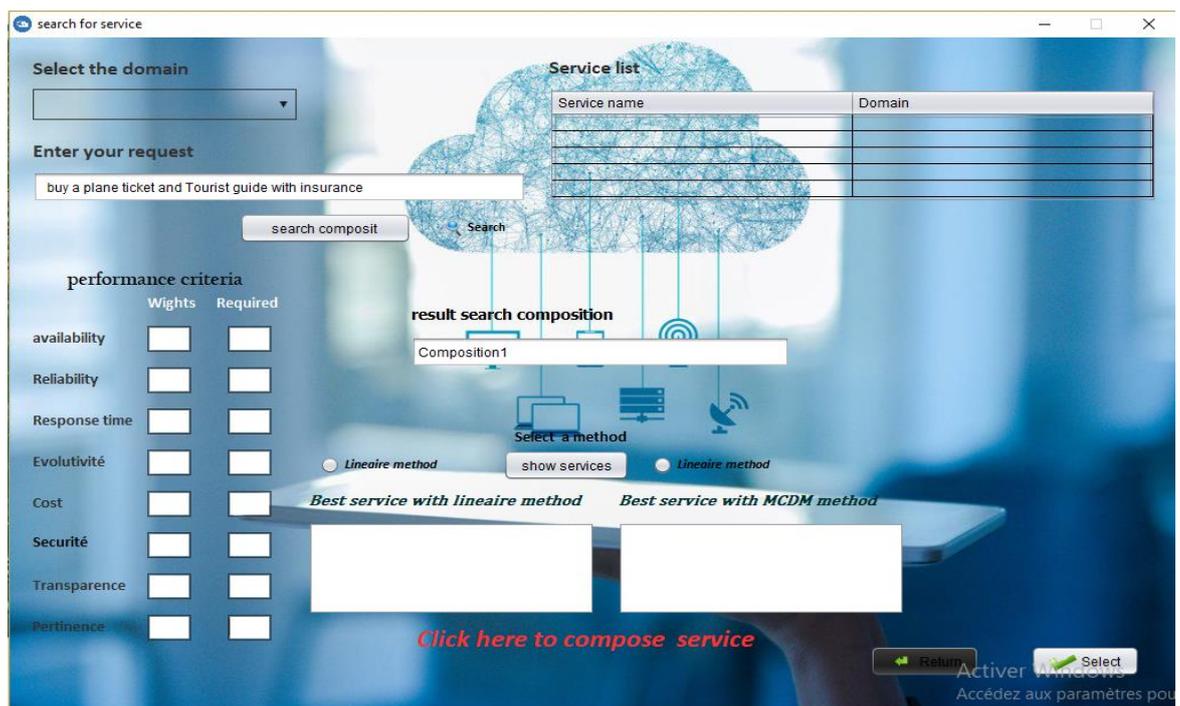


Figure 51 : Recherche des services composés

Le résultat de recherche de la requête « buy a plane ticket and tourist guide with insurance » donne le service composé ‘‘composition1’’ qui est affiché dans le rectangle blanc.

#### **IV.4 Conclusion**

Au fil de ce dernier chapitre, nous avons présenté la mise en œuvre de notre solution à base de broker dans un environnement d’inter-Cloud de type SaaS. L’application réalisée est un médiateur entre le consommateur de Cloud et plusieurs fournisseurs de Cloud, dans le but d’aider le client à sélectionner le fournisseur qui répond le mieux à ses besoins.

Nous avons montré l’applicabilité des concepts, décrits dans la partie conceptuelle, à travers des scénarios d’utilisation prouvant l’interopérabilité entre le broker et le client d’un côté et le broker avec les différents fournisseurs d’un autre côté.

Le Cloud Computing présente une technologie prometteuse qui facilite l'exécution des applications scientifiques et commerciales. Il fournit des services flexibles et évolutifs, à la demande des utilisateurs, via un modèle de paiement à l'usage.

Dans un environnement Cloud interopérable, les clients pourront comparer et choisir des offres parmi les différentes offres Cloud de différentes caractéristiques, en négociant avec des fournisseurs Cloud à chaque fois que cela s'avérerait nécessaire, sans mettre en danger les données et les applications et pourront ainsi composer de nouveaux services cloud à partir de services existants dans le but de satisfaire des demandes plus complexes.

Le but de notre travail est de proposer une solution pour le courtage dans le Cloud entre des SaaS. Pour atteindre cet objectif nous avons commencé, dans un premier temps, par étudier les notions de base du Cloud Computing, après nous avons exploré le problème de courtage dans le Cloud en général et les problématiques de description, sélection et composition en particulier. Ensuite, nous avons fait une recherche bibliographique des travaux de recherche qui sont en relation avec notre problématique. Après cette étape, nous avons réalisé des études comparatives entre les solutions existantes.

L'idée de notre solution consiste à utiliser une série d'algorithmes partant d'une description WSDL et la base WordNet et aboutissant à un schéma d'exécution du Cloud composite.

Pour l'expérimentation de la solution proposée, nous avons réalisé une application Java concrétisant l'ensemble des concepts proposés.

Plusieurs perspectives s'ouvrent à ce travail. En effet, des perspectives d'élargissement du schéma du broker peuvent être envisagées, en ajoutant d'autres méthodes de sélection de service et de composition de services. Le problème de substitution peut aussi faire l'objet d'une extension du broker proposé.

- [1] K. Rubenstein, *Cloud Computing in Life Sciences R&D*. Insight Pharma Reports. Cambridge Healthtech Institute, 2010.
- [2] S. Rajan and A. Jairath, “Cloud computing: The fifth generation of computing,” in *2011 International Conference on Communication Systems and Network Technologies*, 2011, pp. 665–667.
- [3] É. Michon, J. Gossa, S. Genaud, L. Unbekandt, and V. Kherbache, “Schlounder: A broker for IaaS clouds,” *Futur. Gener. Comput. Syst.*, vol. 69, pp. 11–23, 2017.
- [4] E. Simmon, “Evaluation of Cloud Computing Services Based on NIST SP 800-145,” *NIST Spec. Publ.*, vol. 500, p. 322, 2018.
- [5] B. Furht and A. Escalante, *Handbook of cloud computing*, vol. 3. Springer, 2010.
- [6] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering cloud computing: foundations and applications programming*. Newnes, 2013.
- [7] R. Bhojar and N. Chopde, “Cloud computing: Service models, types, database and issues.,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 3, 2013.
- [8] D. C. Marinescu, *Cloud computing: theory and practice*. Morgan Kaufmann, 2017.
- [9] E. Simmon, “Draft-Evaluation of Cloud Computing Services Based on NIST 800-145,” pp. 1–24, 2017.
- [10] B. Sosinsky, *Cloud computing bible*, vol. 762. John Wiley & Sons, 2010.
- [11] «Quadtec solutions INC,» [En ligne]. Disponible : [www.quadtec.com](http://www.quadtec.com). [Visité le 10/12/2018].
- [12] « Le cloud computing » [En ligne]. Disponible : [www-igm.univ-mlv.fr/~dr/XPOSE2009/cloudcomputing/types.html](http://www-igm.univ-mlv.fr/~dr/XPOSE2009/cloudcomputing/types.html). [Visité le 12/12/2018].
- [13] « cloud computing » [En ligne] disponible ; [www.azure.microsoft.com](http://www.azure.microsoft.com) [visité le 1/12/2018]

- [14] « cloud-interoperability-and-portability » [En ligne] Disponible : <http://insightsaas.com/cloud-interoperability-and-portability-necessary-or-nice-to-have/> [Visité le 1/12/2018].
- [15] « what-is-intercloud » [En ligne] Disponible: <https://www.sdxcentral.com/sdn/network-virtualization/definitions/what-is-intercloud/> [Visité le 1/12/2018].
- [16] « what-is-multicloud » [En ligne] Disponible <https://avinetworks.com/glossary/multi-cloud/> [Visité le 1/12/2018].
- [17] F. Liu *et al.*, “NIST cloud computing reference architecture,” *NIST Spec. Publ.*, vol. 500, no. 2011, pp. 1–28, 2011.
- [18] R. Rezaei, T. K. Chiew, S. P. Lee, and Z. S. Aliee, “A semantic interoperability framework for software as a service systems in cloud computing environments,” Elsevier, 2014.
- [19] H. Ali, R. Moawad, and A. A. F. Hosni, “A Cloud Interoperability Broker (CIB) for data migration in SaaS,” *Futur. Comput. Informatics J.*, vol. 1, no. 1–2, pp. 27–34, 2016.
- [20] S. S. Chauhan, E. S. Pilli, R. C. Joshi, G. Singh, and M. C. Govil, “Brokering in interconnected cloud computing environments: A survey,” *J. Parallel Distrib. Comput.*, 2018.
- [21] F. Fowley, C. Pahl, P. Jamshidi, D. Fang, and X. Liu, “A classification and comparison framework for cloud service brokerage architectures,” *IEEE Trans. Cloud Comput.*, vol. 6, no. 2, pp. 358–371, 2016.
- [22] G. Girish and N. J. Nischita, “Cloud broker and their role in a hybrid multi cloud environment,” in *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, 2017, pp. 1532–1535.
- [23] A. Merizig, “Approche de composition de services web dans le Cloud Computing basée sur la coopération des agents.” Université Mohamed Khider-Biskra, 2018.
- [24] M. Rajeswari, G. Sambasivam, N. Balaji, M. S. S. Basha, T. Vengattaraman, and P. Dhavachelvan, “Appraisal and analysis on various web service composition

- approaches based on QoS factors,” *J. King Saud Univ. Inf. Sci.*, vol. 26, no. 1, pp. 143–152, 2014.
- [25] L. Sun, H. Dong, and J. Ashraf, “Survey of service description languages and their issues in cloud computing,” in *2012 Eighth International Conference on Semantics, Knowledge & Grids*, 2012, pp. 128–135.
- [26] S. Ghazouani and Y. Slimani, “A survey on cloud service description,” *J. Netw. Comput. Appl.*, vol. 91, pp. 61–74, 2017.
- [27] 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.
- [28] 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.
- [29] H. Saouli, O. Kazar, and A. cha N. Benharkat, “SaaS-DCS: software-as-a-service discovery and composition system-based existence degree,” *Int. J. Commun. Networks Distrib. Syst.*, vol. 14, no. 4, pp. 339–378, 2015.
- [30] L. Sun, J. Ma, H. Wang, Y. Zhang, and J. Yong, “Cloud service description model: an extension of USDL for cloud services,” *IEEE Trans. Serv. Comput.*, vol. 11, no. 2, pp. 354–368, 2018.
- [31] J. Cardoso, A. Barros, N. May, and U. Kylau, “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.
- [32] S. Ghazouani and Y. Slimani, “Towards a standardized cloud service description based on USDL,” *J. Syst. Softw.*, vol. 132, pp. 1–20, 2017.
- [33] R. Karim, C. Ding, A. Miri, and X. Liu, “End-to-end QoS mapping and aggregation for selecting cloud services,” in *2014 International Conference on Collaboration Technologies and Systems (CTS)*, 2014, pp. 515–522.
- [34] K. B. Bey, H. Nacer, M. E. Y. Boudaren, and F. Benhammadi, “A Novel Clustering-based Approach for SaaS Services Discovery in Cloud Environment.,” in *ICEIS (1)*,

2017, pp. 546–553.

- [35] A. Alfazi, Q. Z. Sheng, Y. Qin, and T. H. Noor, “Ontology-based automatic cloud service categorization for enhancing cloud service discovery,” in *2015 IEEE 19th International Enterprise Distributed Object Computing Conference*, 2015, pp. 151–158.
- [36] A. Tahamtan, S. A. Beheshti, A. Anjomshoaa, and A. M. Tjoa, “A cloud repository and discovery framework based on a unified business and cloud service ontology,” in *2012 IEEE Eighth World Congress on Services*, 2012, pp. 203–210.
- [37] Y. M. Afify, I. F. Moawad, N. L. Badr, and M. F. Tolba, “Cloud services discovery and selection: survey and new semantic-based system,” in *Bio-inspiring Cyber Security and Cloud Services: Trends and Innovations*, Springer, 2014, pp. 449–477.
- [38] 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.
- [39] F. Nawaz, M. R. Asadabadi, N. K. Janjua, O. K. Hussain, E. Chang, and M. Saberi, “An MCDM method for cloud service selection using a Markov chain and the best-worst method,” *Knowledge-Based Syst.*, vol. 159, pp. 120–131, 2018.
- [40] S. Ding, Z. Wang, D. Wu, and D. L. Olson, “Utilizing customer satisfaction in ranking prediction for personalized cloud service selection,” *Decis. Support Syst.*, vol. 93, pp. 1–10, 2017.
- [41] Q. He, J. Han, Y. Yang, J. Grundy, and H. Jin, “QoS-driven service selection for multi-tenant SaaS,” in *2012 IEEE Fifth International Conference on Cloud Computing*, 2012, pp. 566–573.
- [42] D. Lin, A. C. Squicciarini, V. N. Dondapati, and S. Sundareswaran, “A cloud brokerage architecture for efficient cloud service selection,” *IEEE Trans. Serv. Comput.*, 2016.
- [43] L. Sun, H. Dong, F. K. Hussain, O. K. Hussain, and E. Chang, “Cloud service selection: State-of-the-art and future research directions,” *J. Netw. Comput. Appl.*, vol. 45, pp. 134–150, 2014.

- [44] M. Eisa, M. Younas, K. Basu, and H. Zhu, “Trends and directions in cloud service selection,” in *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, 2016, pp. 423–432.
- [45] L. Qu, “Credible service selection in cloud environments,” 2016.
- [46] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, “Semantic matching of web services capabilities,” in *International semantic web conference, 2002*, pp. 333–347.
- [47] G. Gardarin, G. Gardarin, F. Informaticien, G. Gardarin, and G. Gardarin, “XML: Des bases de données aux services Web,” Dunod, 2002.
- [48] D. Chakraborty and A. Joshi, “Dynamic service composition: State-of-the-art and research directions,” 2001.
- [49] B. Medjahed, “Semantic web enabled composition of web services.” Virginia Tech, 2004.
- [50] G. Canfora, M. Di Penta, R. Esposito, and M. L. Villani, “An approach for QoS-aware service composition based on genetic algorithms,” in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 2005, pp. 1069–1075.
- [51] Z. Ye, X. Zhou, and A. Bouguettaya, “Genetic algorithm based QoS-aware service compositions in cloud computing,” in *International Conference on Database Systems for Advanced Applications*, 2011, pp. 321–334.
- [52] Y. Charif, “Chorégraphie dynamique de services basée sur la coordination d’agents introspectifs.” Paris 6, 2007.
- [53] S. K. Gavvala, C. Jatoth, G. R. Gangadharan, and R. Buyya, “QoS-aware cloud service composition using eagle strategy,” *Futur. Gener. Comput. Syst.*, vol. 90, pp. 273–290, 2019.
- [54] B. Di Martino, G. Cretella, and A. Esposito, “Towards a unified OWL ontology of cloud vendors’ appliances and services at paas and saas level,” in *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems*, 2014, pp. 570–575.
- [55] G. Di Modica, G. Petralia, and O. Tomarchio, “A business ontology to enable semantic

- matchmaking in open cloud markets,” in *2012 Eighth International Conference on Semantics, Knowledge and Grids*, 2012, pp. 96–103.
- [56] M. Hamdaqa, T. Livogiannis, and L. Tahvildari, “A Reference Model for Developing Cloud Applications,” in *CLOSER*, 2011, pp. 98–103.
- [57] F. Amato and F. Moscato, “A Modeling Profile for Availability Analysis of Composite Cloud Services,” in *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2014, pp. 406–413.
- [58] S. Gudenkauf, M. Josefiok, A. Göring, and O. Norkus, “A reference architecture for cloud service offers,” in *2013 17th IEEE International Enterprise Distributed Object Computing Conference*, 2013, pp. 227–236.
- [59] P. Hoberg, J. Wollersheim, and H. Krcmar, “Service Descriptions for Cloud Services- The Customer’s Perspective,” in *Proceedings of ConLife Academic Conference*, 2012.
- [60] W. A. Goddard III, D. Brenner, S. E. Lyshevski, and G. J. Iafrate, *Handbook of nanoscience, engineering, and technology*. CRC press, 2012.
- [61] L. Sun, J. Ma, H. Wang, Y. Zhang, and J. Yong, “Cloud service description model: an extension of USDL for cloud services,” *IEEE Trans. Serv. Comput.*, vol. 11, no. 2, pp. 354–368, 2015.
- [62] D. K. Nguyen, F. Lelli, M. P. Papazoglou, and W.-J. Van Den Heuvel, “Blueprinting approach in support of cloud computing,” *Futur. Internet*, vol. 4, no. 1, pp. 322–346, 2012.
- [63] M. Zhang, R. Ranjan, A. Haller, D. Georgakopoulos, M. Menzel, and S. Nepal, “An ontology-based system for Cloud infrastructure services’ discovery,” in *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2012, pp. 524–530.
- [64] Sashi Kanth. Cloud service discovery system using cloud ontology. In National conference on parallel computing technologies (PAR-COMPTECH), February 2013.
- [65] Kwang Mong Sim. Agent-based cloud computing. *IEEE Transactions on Services Computing*, 5(4):564–577, 2012.

- [66] A. Tahamtan, S.A. Beheshti, A. Anjomshoaa, and A.M. Tjoa. A cloud repository and discovery framework based on a uni\_ed business and cloud service ontology. In IEEE Eighth World Congress on Services (SERVICES2012), pages 203{210, 2012.
- [67] Miguelngel Rodrguez-Garca, Rafael Valencia-Garca, Francisco Garca-Snchez, JosJavier Samper-Zapater, and Isidoro Gil-Leiva. Semantic annotation and retrieval of services in the cloud. In Sigeru Omatu, Jos Neves, Juan M. Corchado Rodriguez, Juan F Paz San-tana, and Sara Rodriguez Gonzalez, editors, Distributed Computing and Arti\_cial Intelligence, volume 217 of Advances in Intelligent Sys-tems and Computing, pages 69{77. Springer International Publishing, 2013.
- [68] Chen, Fei, Bai, Xiaoli, Liu, Bingbing: Efficient service discovery for cloud computing environments.In: Shen, Gang, Huang, Xiong (eds.) Advanced Research on Computer Science and Information Engineering. Communications in Computer and Information Science, vol. 153,pp. 443–448. Springer, Berlin Heidelberg (2011)
- [69] Wang, J., Zhang, J.,Hung, P.C.K., Li, Z., Liu, J., He,K.: Leveraging fragmental semantic data to enhance services discovery. In: 2011 IEEE 13th International Conference on High Performance Computing and Communications (HPCC), pp. 687–694. (2011)
- [70] Dastjerdi, A.V., Tabatabaei, S.G.H., Buyya, R.: An effective architecture for automated appliance management system applying ontology-based cloud discovery. In: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), pp. 104–112. (2010)
- [71] Chen, H.P., Li, S.C.: Src: A service registry on cloud providing behavior-aware and qos-aware service discovery. In: 2010 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp 1–4. (2010)
- [72] Dong, H., Hussain, F.K., Chang, E.: A service concept recommendation system for enhancing the dependability of semantic service matchmakers in the service ecosystem environment. J. Netw. Comput. Appl. **34**(2), 619–631 (2011) (Efficient and Robust Security and Services of Wireless Mesh Networks)

- [73] Wang, J., Zhang, J., Hung, P.C.K., Li, Z., Liu, J., He, K.: Leveraging fragmental semantic data to enhance services discovery. In: 2011 IEEE 13th International Conference on High Performance Computing and Communications (HPCC), pp. 687–694. (2011)
- [74] 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.
- [75] R. G. Junior, T. Rolim, A. Sampaio, and N. C. Mendonça. A multi-criteria approach for assessing cloud deployment options based on non-functional requirements. In Proceedings of the 30th Annual ACM Symposium on Applied Computing, pages 1383–1389, 2015.
- [76] S. Habib, S. Ries, and M. Muhlhauser. Towards a trust management system for cloud computing. In IEEE Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pages 933–939, 2011.
- [77] N. Ghosh, Soumya K. Ghosh, Sajal K. Das. SelCSP: A Framework to Facilitate Selection of Cloud Service Providers, 2168-7161 2013
- [78] 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.
- [79] Menzel M, Schönherr M, Tai S. (MC2)2: criteria, requirement sand a software prototype-for Cloud infrastructure decisions. *Softw Pract Exp* Nov. 2013; 43 (11):1283–97.
- [80] Limam N, Boutaba R. Assessing software service quality and trust worthiness at selection time. *IEEE Trans Softw Eng* 2010; 36(4):559–74.
- [81] Zeng W, Zhao Y, Zeng J. Cloud service and service selection algorithm research. In: Proceedings of the first ACM/SIGEVO summit on genetic and evolutionary computation (ACM SIGEVO). Shanghai, China; 2009.
- [82] Silas S, Rajsingh EB, Ezra K. Efficient service selection middleware using ELECTRE methodology for Cloud environments. *Inf Technol J* 2012; 11(7):868–75.
- [83] Saripalli P, Pingali G. MADMAC: multiple attribute decision methodology for adop-

- tion of Clouds. In: Proceedings of the IEEE international conference on Cloud computing (CLOUD). Washington, DC; 2011.
- [84] 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.
- [85] 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.
- [86] Martens B, Teuteberg F, Gräuler M. Design and implementation of a community platform for the evaluation and selection of Cloud computing services: a market analysis. In: ECIS 2011 proceedings, 2011.
- [87] Chang CW, Liu P, Wu JJ. Probability-based Cloud storage providers selection algorithms with maximum availability. In: Proceedings of the international conference on parallel processing (ICPP). Pittsburgh, PA; 2012.
- [88] Ye Z, Bouguettaya A, Zhou X. QoS-aware Cloud service composition based on economic models. In: Proceedings of the international conference on service-oriented computing (ICSOC), Shanghai, China; 2012.
- [89] Sundareswaran S, Squicciarini A, Lin D. A brokerage-based approach for Cloud service selection. In: Proceedings of the IEEE 5th international conference on Cloud computing (CLOUD). Honolulu, HI; 2012.
- [90] Zheng Z, Wu X, Zhang Y, Lyu M, Wang J. QoS ranking prediction for Cloud services. *IEEE Trans Parall Distrib* 2013; 24(99):1213–22.
- [91] He Q, Han J, Yang Y, Grundy J, Jin H. QoS-driven service selection for multi-tenant SaaS. In: Proceedings of the international conference on Cloud computing (CLOUD). Honolulu, HI; 2012.
- [92] Martens B, Teuteberg F. Decision-making Cloud computing environments : a cost and risk based approach. *Inform Syst Front* 2012; 14(4):871–93.
- [93] Yang J, Lin W, Dou W. An adaptive service selection method for cross-Cloud service composition. *Concurr Comput Pract Exp* 2013; 25(18):2435–54.

- [94] Chen C, Yan S, Zhao G, Lee B S, Singh A S. A systematic framework enabling automatic conflict detection and explanation in Cloud service selection for enterprises. In: Proceedings of the IEEE 5th international conference on Cloud computing (CLOUD). Honolulu, HI; 2012.
- [95] Yu Q., CloudRec: a framework for personalized service recommendation in the Cloud, Knowl Inf Syst, <http://dx.doi.org/10.1007/s10115-013-0723-x>, in press.
- [96] Ghezzi C., Manna V.P.L., Motta A. and Tamburrelli G., Performance-driven dynamic service selection, *Concurr Comput Pract Exp*, <http://dx.doi.org/10.1002/cpe.3259>, in press.
- [97] T. Baker, B. Aldawsari, M. Asim, H. Tawfik, Z. Maamar, and R. Buyya, “Cloud-SEnergy: A bin-packing based multi-cloud service broker for energy efficient composition and execution of data-intensive applications,” *Sustain. Comput. Informatics Syst.*, vol. 19, pp. 242–252, 2018.
- [98] M. Alrifai and T. Risse, “Combining global optimization with local selection for efficient QoS-aware service composition,” in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 881–890.
- [99] H. Kurdi, A. Al-Anazi, C. Campbell, and A. Al Faries, “A combinatorial optimization algorithm for multiple cloud service composition,” *Comput. Electr. Eng.*, vol. 42, pp. 107–113, 2015.
- [100] Q. Wu and Q. Zhu, “Transactional and QoS-aware dynamic service composition based on ant colony optimization,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 5, pp. 1112–1119, 2013.
- [101] Y. Huo, Y. Zhuang, J. Gu, S. Ni, and Y. Xue, “Discrete gbest-guided artificial bee colony algorithm for cloud service composition,” *Appl. Intell.*, vol. 42, no. 4, pp. 661–678, 2015.
- [102] H. Gao, J. Yan, and Y. Mu, “Trust-oriented QoS-aware composite service selection based on genetic algorithms,” *Concurr. Comput. Pract. Exp.*, vol. 26, no. 2, pp. 500–515, 2014.
- [103] S. Y. Toh and M. Tang, “Software-as-a-Service Composition in Cloud Computing

Using Genetic Algorithm,” 2018, pp. 542–551.

- [104] J. O. Gutierrez-Garcia and K. M. Sim, “Agent-based cloud service composition,” *Appl. Intell.*, vol. 38, no. 3, pp. 436–464, 2013.
- [105] D. Ardagna and B. Pernici, “Adaptive service composition in flexible processes,” *IEEE Trans. Softw. Eng.*, vol. 33, no. 6, pp. 369–384, 2007.
- [106] H. Malouche, Y. Ben Halima, and H. Ben Ghezala, “A brokerage architecture: Cloud service Pelection,” 2017.
- [107] F. Seghir, A. Khababa, A hybrid approach using genetic and fruit fly optimization algorithms for QoS-aware cloud service composition, *J. Intell. Manuf.* (2016) 1–20.
- [108] M.B. Karimi, A. Isazadeh, A.M. Rahmani, QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm, *J. Supercomput.* 73 (4) (2017) 1387–1415.
- [109] F. Jrad, J. Tao, and A. Streit, “SLA based Service Brokering in Intercloud Environments.,” *CLOSER*, vol. 2012, pp. 76–81, 2012.
- [110] F. Garcia-Sanchez *et al.*, “Adding semantics to software-as-a-service and cloud computing,” *WSEAS Trans. Comput.*, vol. 9, no. 2, pp. 154–163, 2010.
- [111] P. Mohagheghi, “REMICS-REuse and migration of legacy applications to interoperable cloud services,” 2011.
- [112] E. A. Marks and B. Lozano, *Executive’s guide to cloud computing*. John Wiley and Sons, 2010.
- [113] M. H. Dodani, “The Silver Lining of Cloud Computing.,” *J. object Technol.*, vol. 8, no. 2, pp. 29–38, 2009.
- [114] A. Bastia, M. Parhi, B. K. Pattanayak, and M. R. Patra, “Service composition using efficient multi-agents in cloud computing environment,” in *Intelligent Computing, Communication and Devices*, Springer, 2015, pp. 357–370.
- [115] F. Junker, J. Vogel, and K. Stanoevska, “Aggregating price models for composite

## *Références bibliographiques*

---

services in cloud service marketplaces,” in *2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications*, 2012, pp. 479–486.