



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique

Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

**Département d'informatique**

N° d'ordre : IA /M2/2020

## Mémoire

Présenté pour obtenir le diplôme de master académique en

# Informatique

Parcours : Intelligence Artificielle

---

# La composition de services Cloud par les algorithmes évolutionnaires

---

Par :

**OUSSAMA DEBABECHE**

Soutenu le /09/2020, devant le jury composé de :

Nom	Grade	Président
MERIZIG Abdelhak	M.C. Classe B	Rapporteur
Nom	Grade	Examineur

Session 2020

## REMERCIEMENTS

*On remercie avant tout le Bon Dieu de nous avoir donné le savoir et la faculté de pouvoir pour suivre nos études.*

*Nous souhaitons adresser nos remerciements les plus sincères à notre promotrice Dr MERIZIG Abdelhak qui nous a permis de bénéficier de la qualité de son encadrement, les conseils qu'elle nous a prodigués, la patience et la confiance qu'elle nous a témoignée.*

*Nous tenons à remercier aussi tous les membres du laboratoire d'informatique de l'université de Biskra pour leur disponibilité et leur précieuse aide, ainsi que l'ensemble du personnel du département d'informatique pour la formation fournie durant les 5 années passées.*

## *Dédicace*

*Je dédie ce mémoire à.....*

*A mes très chers parents,*

*Affables, honorables, aimables la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi ; Votre prière et votre bénédiction m'ont été d'un grand secours pour mener à bien mes études. Je vous dédie ce travail en témoignage de votre profond amour;*

*A mes chers frères et à mes chères sœurs. Que Dieu vous assiste et vous accorde un avenir plein de joie, de bonheur, de réussite et de sérénité.*

*A tous les membres de ma famille, mes chers amis et A tous ceux qui m'ont soutenu et à toutes les personnes pour lesquelles j'ai une place dans leurs cœurs.*

***DEBABECHE Oussama***

## Abstract

Cloud Computing represents the ideal solution for the transition from a classic concept to a service-oriented concept. This paradigm brings together a growing set of technologies which makes it a hypercomplex system arousing significant interest from researchers and engineers. Since the emergence of Cloud Computing, several providers have been encouraged to use this technology by deploying services. The composition process in the cloud represents the kernel of each transaction due to the complexity of customer requirements. As well as the service limits that can respond to a single operation. To deal with this kind of problem and to meet all requirements including the non-functional aspects which are represented by the quality of service values, this project will use evolutionary algorithm as a solution. The problem addressed in this project aims to solve the problem of deploying services or applications in the Cloud. As long as there are thousands of services deployed in the Cloud it makes the composition process too complex which can be classified as an NP-complete problem. Also, cloud providers in the majority of time offer a package of services with a reduced cost which is included in the concept of combinatorial auctions as an offer or service package. The goal of this project is to use one of the optimization methods to find the best composite service among the thousands of services generated with optimal service qualities.

**Keywords :** *Cloud Computing, Service Composition, Evolutionary Algorithm, Quality of Service, QoS, NSGA-II, Service deployment .*

## Résumé

Le Cloud Computing représente la solution idéale pour le passage d'un concept classique à un concept orienté service. Ce paradigme regroupe un ensemble grandissant de technologies ce qui fait de lui un système hypercomplexe suscitant un intérêt important des chercheurs et des ingénieurs. Depuis l'arrivée du Cloud Computing, plusieurs fournisseurs sont encouragés d'utiliser cette technologie par le déploiement des services. Ces énormes services nécessitent des machines virtuelles pour être exécutés. Le processus de composition dans le cloud représente le noyau de chaque opération à cause de la complexité des exigences des clients. Ainsi que les limites de services qui peuvent répondre à une seule opération. Afin de traiter ce genre de problème et pour répondre à tous les besoins en incluent les aspects non-fonctionnel qui sont représentés par les valeurs de qualité de services, ce projet va utiliser les algorithmes évolutionnaires comme une solution. Le problème traité dans ce projet vise à résoudre le problème de déploiement des services ou des applications dans le cloud. Au tant qu'on des milliers des services déploies dans le cloud sa rendre l'opération de composition trop complexe qui peut être classé comme un problème NP-complet. Le but de ce projet est d'utiliser l'un des méthodes d'optimisation afin de trouver le meilleur service composite parmi les milliers services générés avec des qualités de services optimales.

**Mots clés :** *Cloud Computing, composition de services, les algorithmes évolutionnaires, Qualité de services, NSGA-II, Déploiement des services.*

## ملخص

يعتبر في وقتنا الحالي تركيب الخدمات واحدة من المشاكل الرئيسية في الخدمات السحابية نظرا للقيم الاستثنائية لنمو الخدمات والتي نشرها منتجو الخدمات. كما أن الخدمات البسيطة لم تتمكن مؤخرا من تلبية جميع متطلبات الزبائن، حيث تكوين الخدمة التقليدي يوفر للعملاء خدمة مركبة فقط دون إسناد عامل جودة الخدمة. من أجل الرد على الإعدادات الوظيفية وغير الوظيفية، نحن بحاجة إلى تركيب مجموعة من الخدمات. وباعتبار أن خدمات الويب غير قادرة على التواصل فيما بينها أو المشاركة ديناميكيا في تركيب الخدمة، أدت هذه المشكلة إلى استخدام كيان ديناميكي يمثله وكيل الذي يستند إلى بنية ديناميكية. يقترح هذا العمل تصميم يستند على الوكلاء عن طريق اضافة بروتوكول جديد للتعاون يمكن أن يوفر تكوين خدمة بطريقة أوتوماتيكية وقابل للتكيف من خلال توفير خدمة مركبة مع ضمان جودة الخدمة (QoS). ولأن مقدم الخدمة يستخدم السحابة لنشر خدماته، لضمان السير الحسن لهذه العملية ومن أجل تشغيل الخدمة في أفضل الظروف وفقا لقيم الجودة الخدمة (QoS). بغرض نشر الخدمات في السحابة، كل واحدة من هذه الخدمات تحتاج إلى نموذج موحد. عند الاشارة إلى مشكلة التجميع، يمكن اعتبار تركيب الخدمة حلا لهذه المشكلة. بالإضافة إلى ذلك، يبحث كل مزود خدمة عن أفضل نشر لها والذي يمنح الخدمات أفضل الظروف بفضل نوعية الخدمة التي يتميز بها. تحول هذه الأطروحة مشكلة نشر الخدمة إلى مشكلة التحسين، كما تقترح هذه المذكرة نموذجا لوصف الخدمة المساعدة لعملية البحث عن الخدمة.

### الكلمات المفتاحية:

تكوين الخدمة، الخدمات البسيطة، خدمة ويب، وكيل، نشر الخدمة، الحوسبة السحابية، الخدمة السحابية، نظام متعدد الوكلاء، جودة الخدمة، مشكلة التحسين

## Table de matières

Chapitre 1 .....	1
Introduction Générale.....	1
1.1 Contexte et Problématique .....	1
1.2 Objectif.....	2
2 Structure du mémoire .....	2
Chapitre 2 .....	3
Etat De L'art Sur Le Cloud Computing .....	3
2.1 Introduction .....	3
2.2 Historique .....	4
2.3 Définition du cloud computing.....	4
4.1 Caractéristiques du Cloud Computing .....	5
4.2 Les types de services cloud .....	7
2.5.1 Infrastructure-as-a-Service (IaaS) : ou le Cloud d'infrastructure.....	7
2.5.2 Software-as-a-Service (SaaS): logiciel à la demande.....	7
2.5.3 Platform-as-a-Service (PaaS): .....	8
2.6 Techniques de déploiement du cloud computing .....	8
2.6.1 Cloud publique .....	8
2.6.2 Cloud privé.....	9
2.6.3 Cloud communautaire .....	9
2.6.4 Cloud hybride.....	10
2.7 Les avantages et les inconvénients du cloud computing.....	11
2.7.1 Les avantages : .....	11
2.7.2 Les inconvénients .....	11
2.8 Conclusion.....	12
Chapitre 3 .....	13
<i>Travaux Connexes Et Composition De Services</i> .....	13
3.1 Introduction .....	13
3.2 Présentation du problème de composition de services .....	13
3.3 Définition de composition de services .....	14

3.4	Processus de composition de services .....	16
3.4.1	Définition abstraite .....	16
3.4.2	Découverte et sélection .....	17
3.4.3	Déploiement .....	17
3.4.4	Exécution.....	17
3.5	Les valeurs de qualités de services.....	17
3.5.1	Les valeurs de qualités de services utilisé .....	18
3.6	Travaux connexes.....	19
3.7	Synthèse bibliographique des travaux.....	20
3.8	Conclusion.....	20
	Chapitre 4.....	23
	Conception De Système .....	23
4.1	Introduction .....	23
4.2	Objectif.....	23
4.3	L'architecture globale de système .....	24
	.....	24
4.3.1	Description de l'architecture .....	24
4.4	Fonctionnement globale .....	25
4.5	Processus de composition.....	26
4.5.1	Normalisation .....	27
4.5.2	Recherche des services .....	28
4.5.3	Exécuter l'algorithme NSGA 2 .....	28
4.5.4	Choisir la meilleure solution .....	31
4.6	Conclusion.....	31
	Chapitre 5.....	33
	Implémentation Et Mise En Œuvre.....	33
5.1	Introduction .....	33
5.2	Outils et Plateformes Utilisées .....	33
5.3	Environnement de développement .....	33
5.3.1	Python.....	33
5.3.2	Plateforme PyCharm .....	34
5.3.3	Tkinter .....	34
5.4	Pseudo code de l'algorithme proposé.....	34
5.5	Présentation de l'interfaces du système.....	35
5.6	Résultat et discussion .....	36
5.7	Conclusion.....	37
6.1	Conclusion.....	39

6.2 Perspectives.....	39
Références.....	40

# Table des figures

2.1 Vue générale de l’environnement Cloud Computing .....	6
2.2 Les trois modèles du cloud computing .....	8
2.3 Modèle de déploiement publique du cloud computing .....	10
2.4 Modèle de déploiement privé du cloud computing .....	11
2.5 Modèle de déploiement communautaire du cloud computing .....	12
2.6 représentation du cloud Hybrid .....	13
3.1 Principe de la composition de services. ....	17
3.2 Exemple de planification d’un voyage. ....	17
3.3 processus de la composition de services [Khanouche. 2016]. ....	18
4.1 Architecture générale du système .....	26
4.2 Diagramme de séquence du fonctionnement de système. ....	27
4.3 processus de composition. ....	28
4.4 Recherche des services. ....	29
4.5 Le processus de l’algorithme NSGA-II. ....	30

# Liste des tableaux

3.1 Les différentes travaux proposés pour résoudre le problème de composition.....	22
--	----

# **INTRODUCTION GENERALE**

# Chapitre 1

## Introduction Générale

### 1.1 Contexte et Problématique

Avec une évolution exponentielle dans le temps, les SI (Systèmes d'Informations) sont devenus plus complexes et hétérogènes en raison de la diversité des besoins, des exigences des clients, et de la grande masse d'information stockée et manipulée par les internautes. Les infrastructures publiques et privées du secteur informatique sont de plus en plus multiplateformes, multifournisseurs et distribuées à grande échelle. Dans une telle situation, nombreux sont les professionnels de l'informatique qui considèrent que l'interopérabilité est un aspect aussi important que la sécurité et la fiabilité pour la gestion de leurs SI et leurs environnements de fonctionnement.

Cependant, les services web possèdent aussi des inconvénients. Les Web services, tels qu'ils sont présentés, sont conceptuellement limités à des fonctionnalités relativement simples qui sont modélisés par une collection d'opérations. Toutefois, pour certains types d'applications, il est nécessaire de combiner un ensemble de services Cloud en services Cloud plus complexes (Web services agrégés ou composites) afin de répondre à des exigences plus complexes.

En effet, il se peut qu'un client ait des besoins spécifiques qui ne sont rendus par aucun des services web. A cause de l'élargissement d'internet, il est rare pour un consommateur de découvrir un service web qui réponde à ses besoins. Pour cette raison, plusieurs services peuvent interagir et échanger dynamiquement des informations. L'objectif de cette interaction est l'accomplissement d'un but complexe non concevable par un seul service Cloud. Cette agrégation des compétences pour réaliser un but commun est connue par « composition de services Cloud » cette composition tient compte des contraintes fonctionnelles et des contraintes non fonctionnelles, notamment en termes de qualité de service (*QoS*), requises par l'utilisateur.

Actuellement, la composition de services web représente un problème majeur qui est la sélection de services composites sensible à la QoS (Quality of Service) avec prise en charge des contraintes d'utilisateurs. En effet, la composition de services est une tâche délicate car elle ne consiste pas seulement à combiner un ensemble de services en tenant compte des aspects fonctionnels, mais aussi d'un assemblage non fonctionnel de services tout en assurant une fiabilité maximale.

## 1.2 Objectif

L'objectif de la composition de service est de créer de nouvelles fonctionnalités en combinant des fonctionnalités offertes par d'autres services existants, composés ou non en vue d'apporter une valeur ajoutée.

Notre objectif est de concevoir un prototype d'un système de composition de service. Ce prototype proposé assure la résolution du problème d'optimisation multi-objectifs (MOP) dans la composition. Il implique l'optimisation de deux objectifs ou plus soumis à certaines contraintes qui peuvent entrer en conflit les uns avec les autres, par exemple, maximiser la disponibilité et minimiser le coût ou maximiser la réputation et minimiser le temps de réponse. Ce phénomène est appelé optimalité de Pareto. Pour cela nous allons proposer L'Algorithme d'Optimisation Multi Objectif -NSGA2- pour résoudre ce problème.

## 2 Structure du mémoire

Le présent mémoire est organisé en quatre chapitres dont les thèmes sont donnés ci-dessous : Le premier chapitre présente le contexte et la problématique de ce projet. Ce chapitre identifie également l'objectif de ce travail.

Le deuxième chapitre donne couvre les notions de base sur la technologie du Cloud Computing. Dans le premier temps nous allons illustrer les différentes définitions du Cloud et ses différents caractéristiques. Ensuite, nous allons présenter les types de services cloud, les modèles de déploiement avec les avantages et les limites du cloud computing.

Le troisième chapitre donne une synthèse bibliographique sur les travaux réalisés pour résoudre le problème de composition de services. Il présente aussi le processus de composition de services dans le Cloud avec l'étude de cas utilisée dans ce projet.

Le quatrième chapitre commence notre contribution qui consiste à résoudre le problème de déploiement des services dans le Cloud. Après, nous allons présenter l'architecture globale du système en détaillant son fonctionnement à l'aide d'un diagramme de séquence UML pour simplifier la compréhension de notre travail. Ensuite, nous allons expliquer l'architecture détaillée de chaque composant et son fonctionnement, nous terminerons ce chapitre par une conclusion.

Le cinquième chapitre montre la mise en œuvre de notre approche, il commence par la présentation des outils et l'environnement de développement. Ensuite, nous allons présenter quelques interfaces qui montrent les résultats obtenus d'après l'implantation de notre modèle, nous terminons le par une discussion sur les résultats obtenus.

Finalement, le sixième chapitre conclut cette thèse. Il synthétise les contributions globales et met en évidence les perspectives de cette recherche.

# *Chapitre 2*

*Etat de l'art sur le cloud computing*

## **Chapitre 2**

# **Etat De L'art Sur Le Cloud Computing**

### **2.1 Introduction**

Les technologies de l'information et de la communication (TIC) se développent plus rapide et de manière progressive. Dans ces dernières années il y a une nouvelle destination, son but est d'améliorer les services dans le domaine TIC, il s'agit du "Cloud Computing". Ce dernier est un nouveau concept informatique qui consiste à proposer des services informatiques sous forme de services à la demande, accessibles de n'importe où, n'importe quand et par n'importe qui. Cette nouvelle technologie permet à des entreprises d'externaliser le stockage de leurs données et de leur fournir une puissance de calcul supplémentaire pour le traitement de grosse quantité d'information.

Dans ce chapitre nous donnons quelques généralités sur le Cloud Computing, à savoir sa définition, ses caractéristiques ses différents types, les services qu'il offre, ses techniques de déploiement .

Ce chapitre se termine par ses avantages et inconvénients, ainsi certaines notions pour la bonne compréhension de ce concept.

## **2.2 Historique**

La première énonciation du concept du Cloud Computing date de 1960, quand John McCarthy affirmait que la ressource informatique serait accessible et consommée par le public de la même façon que la distribution d'eau et d'énergie [1].

Le concept du Cloud Computing a été adopté pour la première fois en 2002 par Amazon, un leader du e-business, qui avait investi dans un parc de machines immense, dimensionné pour absorber la charge importante des commandes faites sur leur site au moment des fêtes de Noël, mais relativement inexploité le reste de l'année. Sous-dimensionner leur parc aurait causé des indisponibilités de leur site au moment des pics, mettant ainsi en péril leur business pendant les fêtes (soit une grosse partie de leur chiffre d'affaires). Leur idée a donc été d'ouvrir toutes ces ressources inutilisées aux entreprises, pour qu'elles les louent à la demande. Depuis, Amazon investit massivement dans ce domaine et continue d'agrandir son parc et ses services.

## **2.3 Définition du cloud computing**

L'ambiguïté qui entoure le concept du Cloud Computing rend à trouver plusieurs définitions pour ce concept. Aujourd'hui, le Cloud Computing est le plus important domaine dans l'industrie informatique. Avec ses nouveaux aspects et capacités qui ont été proclamés, le Cloud Computing est un modèle en évolution rapide. Le Cloud, pas à pas, est devenu la prochaine évolution de l'histoire informatique et modifie radicalement la manière dont une entreprise gère ses systèmes informatiques.

Beaucoup de chercheurs dans les domaines industriels et universitaires ont tenté de donner une définition au « Cloud Computing » et de préciser quelles sont les caractéristiques uniques qu'il présente :

Selon le "National Institute of Standards and Technology (NIST) " [2], "Le Cloud Computing est un modèle qui permet d'accéder rapidement à un pool de ressources informatiques mutualisées, à la demande (serveurs, stockage, applications, bande passante, etc.), sans forte interaction avec le fournisseur de service ”.

Selon IBM [3], "le Cloud Computing, souvent appelé simplement « nuage », est une fourniture de ressources informatiques à la demande, des applications aux centres de données, en passant par l'Internet et le paiement à l'utilisation ”.

Selon Gartner [4], “le Cloud Computing est un style d'informatique dans lequel des moyens informatiques disponibles, évolutives et élastiques sont fournies sous forme de service aux utilisateurs externes utilisant les technologies Internet. L'opération de fourniture est basée sur cinq attributs: la multi-location (ressources partagées), une extensibilité massive, l'élasticité, le paiement à l'utilisation et l'auto-provisionnement des ressources ”.

En analysant ces définitions on constate que la définition du Cloud tourne au tour de ses deux principaux groupes des caractéristiques: (1) Du point de vue fournisseur de services Cloud, on trouve l'élasticité, mis à jour automatique, évolutivité massive, l'auto-approvisionnement de ressources, et les ressources partagées. (2) et du point de vue utilisateur final du Cloud, on trouve l'instantanéité, la disponibilité, l'accès à la demande, le paiement à l'utilisation.

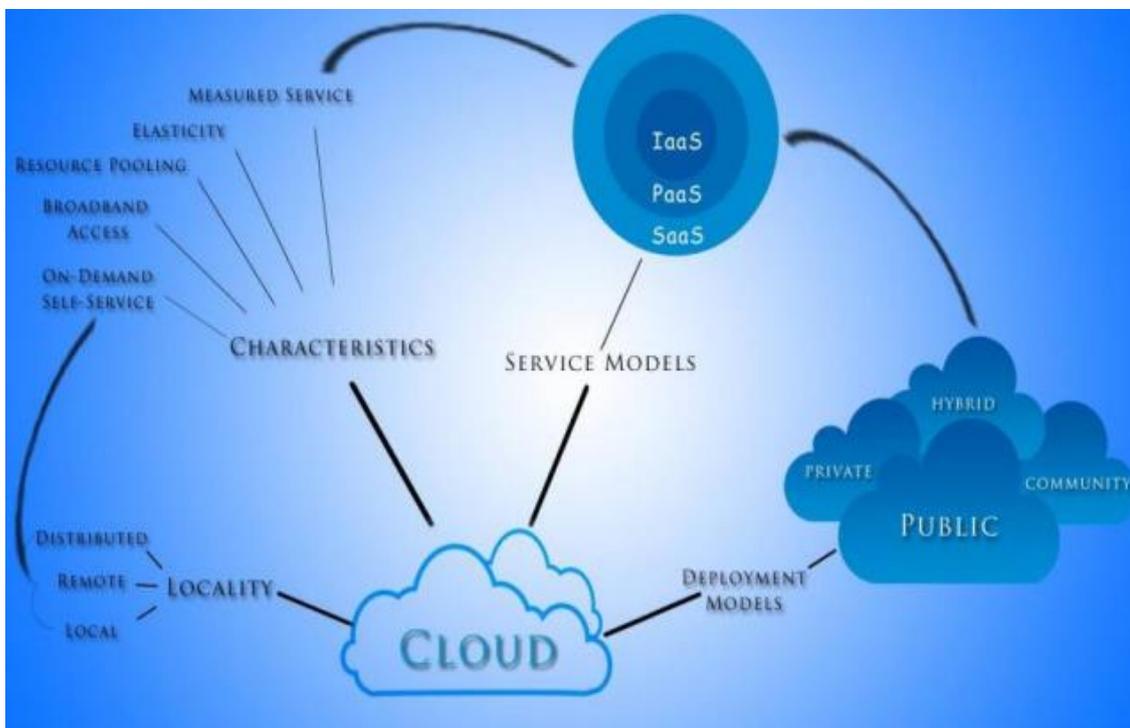


Figure 2.1 - Vue générale de l'environnement Cloud Computing.

#### 4.1 Caractéristiques du Cloud Computing

On peut caractériser un Cloud sur plusieurs aspects, dont voici les plus importants [5] :

- **Accès réseau universel via le réseau**

Un environnement de type Cloud s'appuie obligatoirement sur Internet et est accessible via ce réseau, quel que soit le périphérique utilisé (PC, Mac, TV, Tablette, et Smartphone).

- **Mise en commun de ressources (Pooling)**

Dans un environnement de type Cloud, on ne pense pas en nombre de serveurs, taille de disques ou nombre de processeurs, mais surtout en puissance de calcul, capacité de stockage et bande passante disponible.

- **Elasticité**

Des ressources supplémentaires peuvent être allouées au service pour assurer la continuité du service en cas de pic de charge, ou bien être ré-allouées à un autre service dans le cas inverse.

- **Libre-Service (self-service)**

Dans un Cloud, il est possible à un utilisateur de consommer les services ou ressources sans pour autant devoir faire une demande d'intervention auprès de son fournisseur (équipe IT). (ex: un développeur qui souhaite tester son application sur une machine virtuelle représentative d'un poste standardisé de son entreprise peut, seul et au travers d'un portail Web, provisionner et utiliser une machine sans devoir solliciter l'équipe IT).

- **Service mesurable et facturable (Pay As You Go)**

Dans un environnement Cloud, le fournisseur de la solution est capable de mesurer de façon précise la consommation des différentes ressources (CPU, stockage, bande passante, etc.). Cette mesure lui permet ensuite de facturer le client selon l'usage. Pay As You Go est une option de facturation typique pour un fournisseur de Cloud Computing, ce qui signifie que le client est facturé en fonction des ressources consommées. Cela peut être très utile dans le cas où ce dernier n'est pas certain des ressources dont il a besoin.

- **Tolérance aux pannes**

Les architectures Cloud doivent être tolérantes aux pannes, fournissant des services continus malgré la défaillance d'un ou plusieurs composants du système.

- **Lieu de travail dynamique (Dynamic Workplace)**

Le Cloud favorise la nouvelle vision de l'entreprise sur le lieu de travail de l'employé et sa présence physique sur les sites de l'entreprise. Alors, on n'est pas obligé de travailler dans les locaux de l'entreprise mais on peut travailler n'importe où. Le plus important est d'être rentable et efficace. Le Cloud est caractérisé aussi par :

- La flexibilité et l'accès aux ressources de grandes échelles ;
- L'évolutivité ;
- L'agilité et la rapidité de mise en service, développement des produits, ... ;
- Les données qui sont distribuées et répliquées sur de grandes distances géographiques.

## 4.2 Les types de services cloud

Selon NIST [2, 6] et comme illustré dans la Figure 2.2, il y a principalement trois types de services Cloud Computing qui sont : l'infrastructure en tant que service (IaaS), la plate-forme en tant que service (PaaS) et le logiciel en tant que service (SaaS) que nous allons détailler par la suite

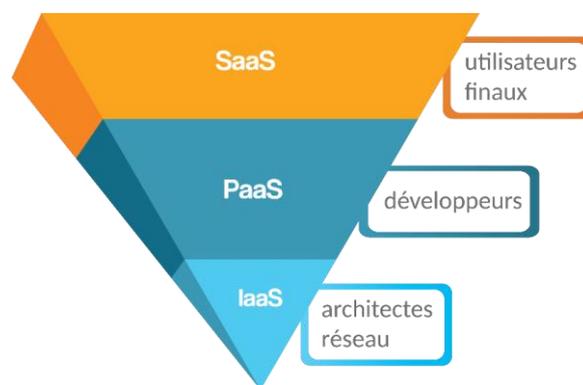


Figure 2.2 - les trois modèles du cloud computing.

### 2.5.1 Infrastructure-as-a-Service (IaaS) : ou le Cloud d'infrastructure

On donne aux utilisateurs des ressources accessibles et partagées, se présentant sous forme d'unités de calcul, de supports de stockage, de moyens de communication (réseau). . . etc. Avec ces moyens là, les consommateurs pourront dérouler leurs propres logiciels et outils à distance. Ceci veut dire que le client a le contrôle sur le système d'exploitation, l'aspect applicatif et le stockage. Mais il n'a pas la gestion des couches basses comme l'infrastructure et le réseau matériel. Ceci dit, il peut parfois avoir la gestion logicielle et partielle du réseau (si le fournisseur autorise cela) comme le choix d'un pare-feu [7].

### 2.5.2 Software-as-a-Service (SaaS): logiciel à la demande

Utiliser des applications web s'exécutant sur les serveurs du nuage. Les logiciels sont accessibles par des clients différents, répartis sur le globe. Ces derniers n'ont de contrôle que sur les paramètres laissés ouverts explicitement par le fournisseur du service. Mais n'ont aucun contrôle sur la gestion de l'infrastructure, le système d'exploitation, le réseau ou de l'application elle-même...etc [8,9].

### 2.5.3 Platform-as-a-Service (PaaS): ou le Cloud applicatif

Un niveau intermédiaire entre les 2 niveaux précédemment cités est le PaaS. Le client peut dérouler des applications de sa création sur les outils offerts ou proposés par le fournisseur. Ces derniers peuvent regrouper les bases de données et les langages de programmation. Le consommateur ne gère pas les couches basses comme les serveurs et les systèmes d'exploitation. Mais il a tout de même le contrôle sur les applications déployées ainsi qu'une possibilité de la configuration de l'environnement les hébergeant [10].

## 2.6 Techniques de déploiement du cloud computing

Les modèles de Cloud Computing sont distingués en fonction de l'utilisation des ressources physiques par les intervenants. Les ressources peuvent être localisées chez l'utilisateur ou chez un fournisseur, peuvent être partagées ou non, et peuvent être pour des entreprises ou pour des autres types d'utilisateurs. Pour cela, le Cloud offre quatre modèles ou typologies de déploiement [11,12].

### 2.6.1 Cloud publique

Bien qu'il soit géré par une entreprise particulière ( Azure par Microsoft ) proposant des services de Cloud Computing, le modèle publique du Cloud Computing reste un modèle accessible par tout internaute connecté. Par conséquent, il est le moins sécurisé, mais pas pour autant le moins utilisé [13].

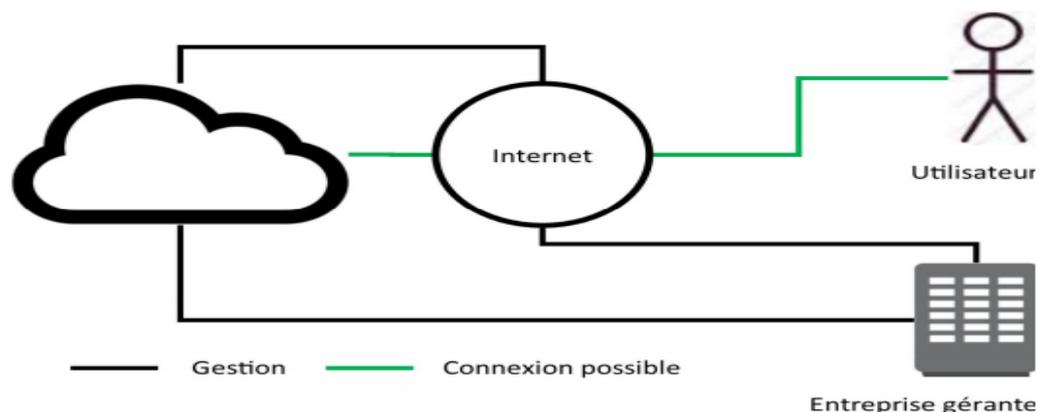


Figure 2.3 - Modèle de déploiement public du cloud computing.

La Figure 2.3 illustre un cas d'exemple pour le modèle public du cloud computing. Un service public en nuage reste toujours accessible au grand public. La seule condition est qu'il doit être connecté à Internet. Comme tout matériel, il doit être géré par un organisme, une entreprise..., etc. Mais ceci pourrait être fait d'une manière locale ou via Internet.

### 2.6.2 Cloud privé

L'infrastructure du nuage ou d'une partie du nuage est dédiée à une entreprise spécifique. Un internaute non concerné par l'entreprise ne pourra pas accéder aux installations privées. Ceci dit, l'infrastructure pourrait être gérée par l'entreprise en question (cliente) ou une tierce (gérantes).

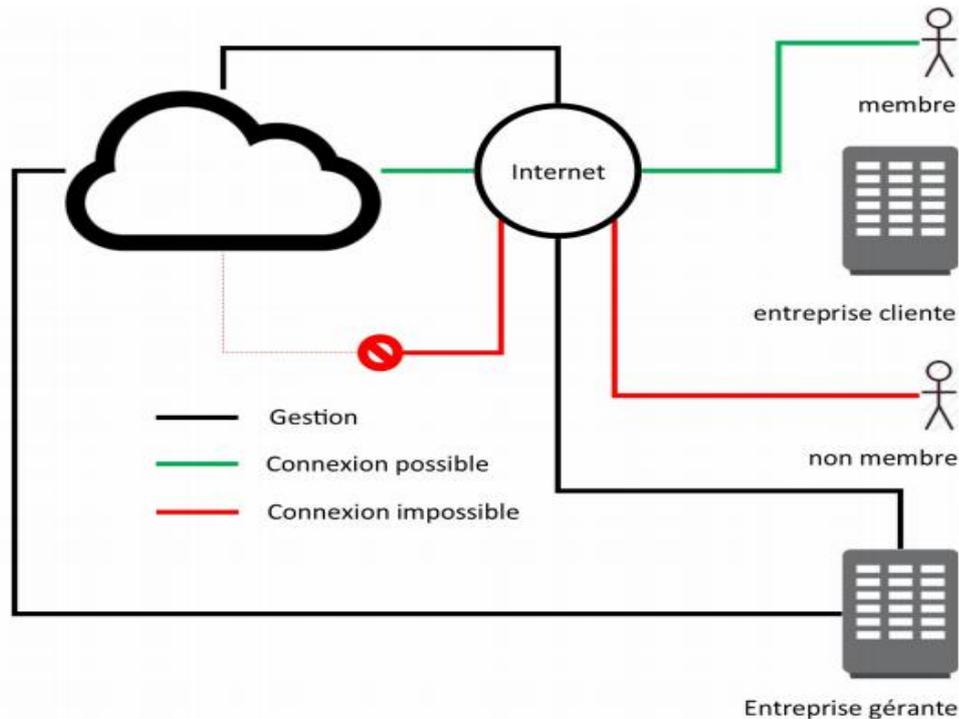


Figure 2.4 - Modèle de déploiement privé du cloud computing.

La Figure 2.4, illustre un exemple d'un modèle de déploiement privé du cloud computing. Un membre de l'entreprise cliente peut accéder aux installations qui lui sont dédiées alors qu'un autre ne pourra pas. D'un autre côté, le matériel peut être géré par un organisme tiers (entreprise gérante) ou l'entreprise elle-même (cliente) [14].

### 2.6.3 Cloud communautaire

L'infrastructure du nuage ou une partie du nuage est dédiée à une communauté d'entreprises partageant les mêmes requis et préférences spécifiées dans leurs contrats de type Service Level Agreement. L'infrastructure en question peut être gérée par la communauté en totalité, une partie de cette dernière, une entreprise membre ou un organisme tiers [13].

La Figure 2.5, illustre à titre d'exemple un modèle de déploiement communautaire. La gestion de l'infrastructure peut être exercée par l'entreprise en dehors de la communauté (entreprise gérante). Mais, peut être aussi faite par la communauté elle-même ou par une partie de la communauté. Ceci dit, un groupe d'entreprises doit avoir au moins un minimum de requis

et de principes en commun afin de bénéficier et d'en tirer au maximum des bénéfices d'un tel modèle [15].

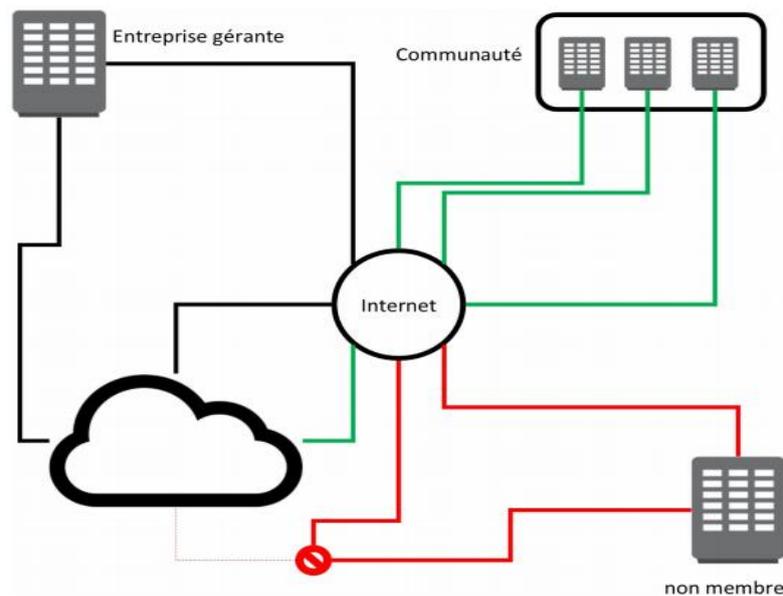


Figure 2.5 - Modèle de déploiement communautaire du cloud computing.

#### 2.6.4 Cloud hybride

Un Cloud hybride est une combinaison d'un Cloud privé combiné à l'utilisation de services de Cloud public. Certains services sont accessibles via un réseau privé et d'autres via Internet. Le futur devrait confirmer l'émergence du Cloud hybride qui consiste à la cohabitation et la communication entre un Cloud privé et un Cloud publique dans une organisation partageant des données et des applications [14].



Figure 2.6 - représentation du cloud Hybrid.

## 2.7 Les avantages et les inconvénients du cloud computing

### 2.7.1 Les avantages :

Le Cloud Computing offre de nombreux avantages que plusieurs travaux ont listés[13,12,16,17]. Parmi ces avantages on trouve :

- Réduction des coûts d'infrastructure.
- Réduction des coûts de développement.
- Réduction des coûts des logiciels.
- Des ressources et services plus rapide à allouer et plus simple à utiliser.
- Accès aux ressources plus flexible.
- Meilleure utilisation, plus efficace, des ressources.
- Augmentation de la puissance de calcul.
- Grande capacité de stockage (quasi illimitée).
- Moins de problèmes d'entretien.
- Gestion des mises à jour plus simple et rapide.
- Pas de perte de données.
- Tout est considéré comme un service défini par un SLA.
- Infrastructure allouée et disponible juste à temps.
- Réduction du temps de mise sur le marché

### 2.7.2 Les inconvénients

Le Cloud Computing n'a pas que des avantages, il possède quelques obstacles et désavantages qui sont abordés dans[13,12, 18-20] . Parmi ces obstacles, il y a :

- Confidentialité des données.
- Chiffrement des données.
- Nécessité d'un accès réseau constant.
- Mauvais fonctionnement avec les connexions à basse vitesse.
- Faible niveau de la qualité de service dans le réseau.
- Risque d'engorgements lors des transferts de données.
- Problème d'interopérabilité.
- Problème de portabilité.
- Faible contrôlabilité.
- Manque de fonctionnalités d'audit

## **2.8 Conclusion**

Au cours de ce chapitre, nous avons essayé de faire un survol sur les principaux concepts et points clés du Cloud Computing. La naissance du Cloud Computing a donné un nouveau mode de consommation de l'informatique et a changé la manière d'investissement des entreprises dans les infrastructures informatiques. Un coût faible, stockage évolutif illimité et une grande puissance de calcul sont les promesses du Cloud Computing. Tandis que la sécurité est l'une des plus grandes préoccupations des responsables informatiques lorsqu'il est question de Cloud Computing .

# *Chapitre 3*

## *Travaux connexes et composition de services*

## Chapitre 3

### *Travaux Connexes Et Composition De Services*

#### 3.1 Introduction

La composition de services est considérée comme l'une des motivations les plus importantes du paradigme SOA (Service Oriented Architecture), en effet si une application ou un client requièrent des fonctionnalités, et qu'aucun service n'est seul apte à les fournir, il devrait être possible de combiner ou de composer des services existants afin de répondre aux besoins de cette application ou de ce client. C'est ce que l'on appelle la composition de services [21].

La composition de services vise à faire inter-opérer, interagir et coordonner plusieurs services pour la réalisation d'un but. Malgré les efforts de recherche et de développement autour de la problématique de la composition des services, elle reste une tâche hautement complexe et pose un certain nombre de défis. Sa complexité provient généralement des sources suivantes:

- L'augmentation dramatique du nombre des services web sur le web rend très difficile la recherche et la sélection des services web pouvant répondre à un besoin donné.
- Les services sont créés et mis à jour de façon hautement dynamique.
- Les services web sont d'habitude développés par différentes organisations qui utilisent différents modèles conceptuels pour décrire les caractéristiques des services web.

Dans ce chapitre nous présenterons le problème de composition des services Web et le processus de composition de services . Ensuite nous mettrons en relief les valeurs de qualités des services. Ce chapitre se termine par les Travaux connexes et synthèse bibliographiques.

#### 3.2 Présentation du problème de composition de services

Un service est dit composite lorsque son exécution nécessite d'invoquer plusieurs autres services en faisant appel à leurs fonctionnalités, sinon il est dit atomique.

Le problème de composition de service peut être présenté comme la transformation des besoins des utilisateurs en un plan d'exécution de service composite qui les satisfait.

Les utilisateurs des plates-formes clouds utilisent des services pour satisfaire leurs demandes. Mais ces demandes deviennent très complexes, et ils ont besoin plus d'un seul service pour accomplir une demande. Le processus de collecte d'un ensemble des services pour satisfaire une demande de l'utilisateur est appelé la composition de services [22].

### 3.3 Définition de composition de services

Plusieurs définitions ont été proposées pour le concept de composition, nous citons dans ce qui suit les plus communes :

La composition de services est aujourd'hui un sujet de grand intérêt autant pour le monde de la recherche que pour le monde industriel. De nombreuses recherches visent à développer des modèles de composition de services et à fournir les outils nécessaires pour la composition de services [24].

La composition de services web est le processus de construction de nouveaux services web à valeur ajoutée, à partir de deux ou plusieurs services web déjà présents et publiés sur le web. Un service Web est dit composé ou composite lorsque son exécution implique des interactions avec d'autres services web, des échanges de messages entre eux afin de faire appel à leurs fonctionnalités. La composition de services web spécifie quels services ont besoin d'être invoqués, dans quel ordre et comment gérer les conditions d'interaction [23].

Une autre définition plus raffinée est celle proposée par Georges Gardarin [24], qui définit la composition des services web comme étant : une technique permettant d'assembler des services Web afin d'atteindre un objectif particulier, par l'intermédiaire de primitives de contrôle (boucle, test, traitement d'exception, etc.) et d'échange (envoi et réception de messages). Les services composants existent au préalable et peuvent ne pas être fournis par la même organisation.

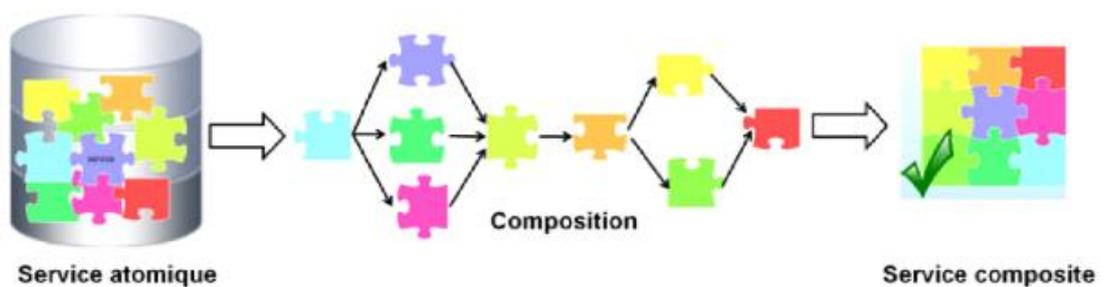


Figure 3.1 - Principe de la composition de services.

Nous allons illustrer la sélection et la composition de service par un scénario appelé

« Partir en voyage ».

Pour partir en voyage l'utilisateur (client) doit trouver un service web pour chaque réservation qu'il désire effectuer (avion, voiture, train, hôtel), parmi une grande masse de service qui ont des fonctionnalités similaires. Pour résoudre ce problème, des services web peuvent être sélectionnés en se basant sur des propriétés non-fonctionnelles, puis regroupés et interconnectés pour former un seul service d'une manière transparente pour l'utilisateur.

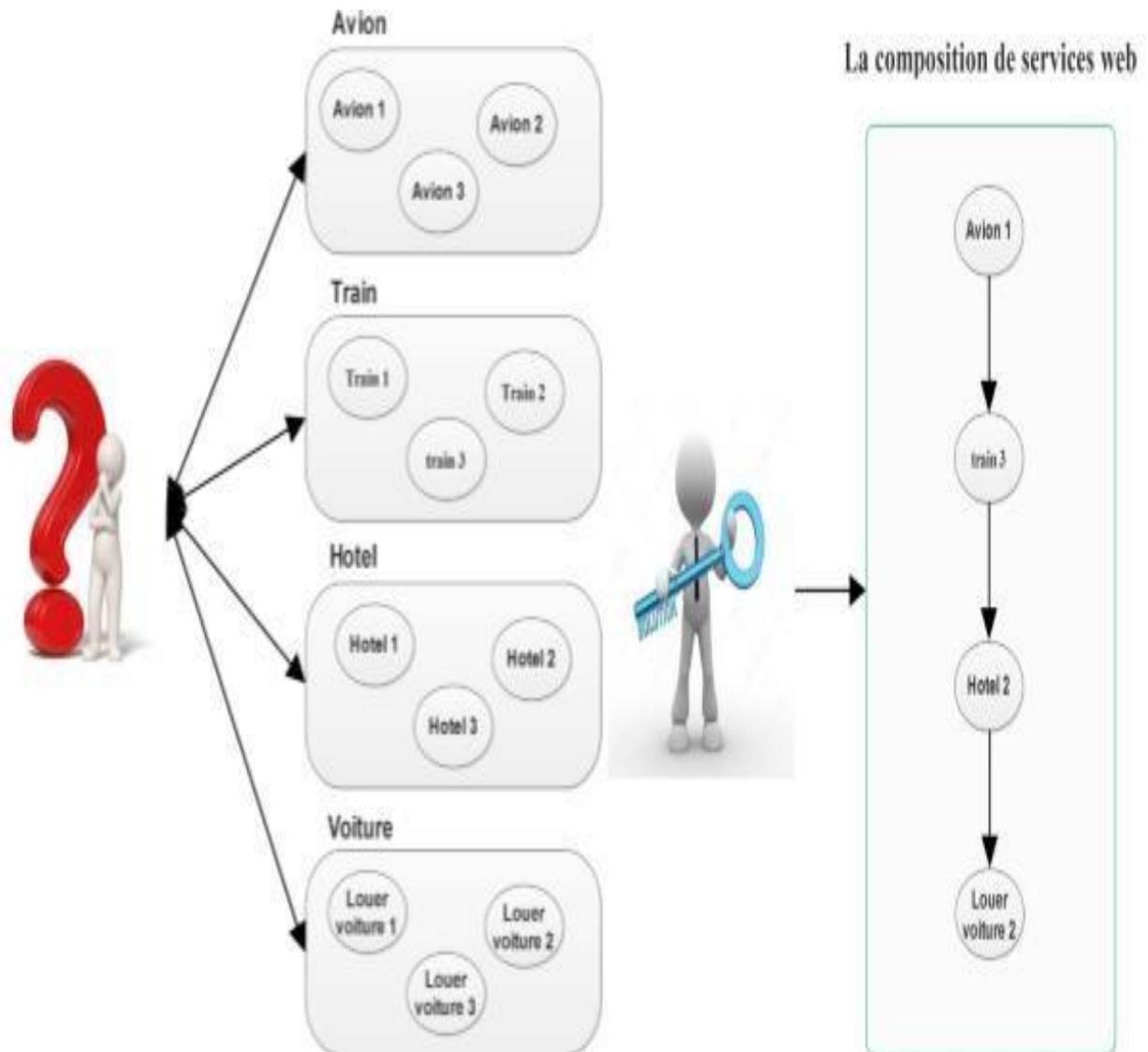


Figure 3.2 - Exemple de planification d'un voyage.

### 3.4 Processus de composition de services

La réalisation d'une application par composition de services comporte plusieurs étapes qui permettent le passage incrémental d'une spécification abstraite vers une composition concrète de services, c'est à dire une composition prête à être exécutée [25].-

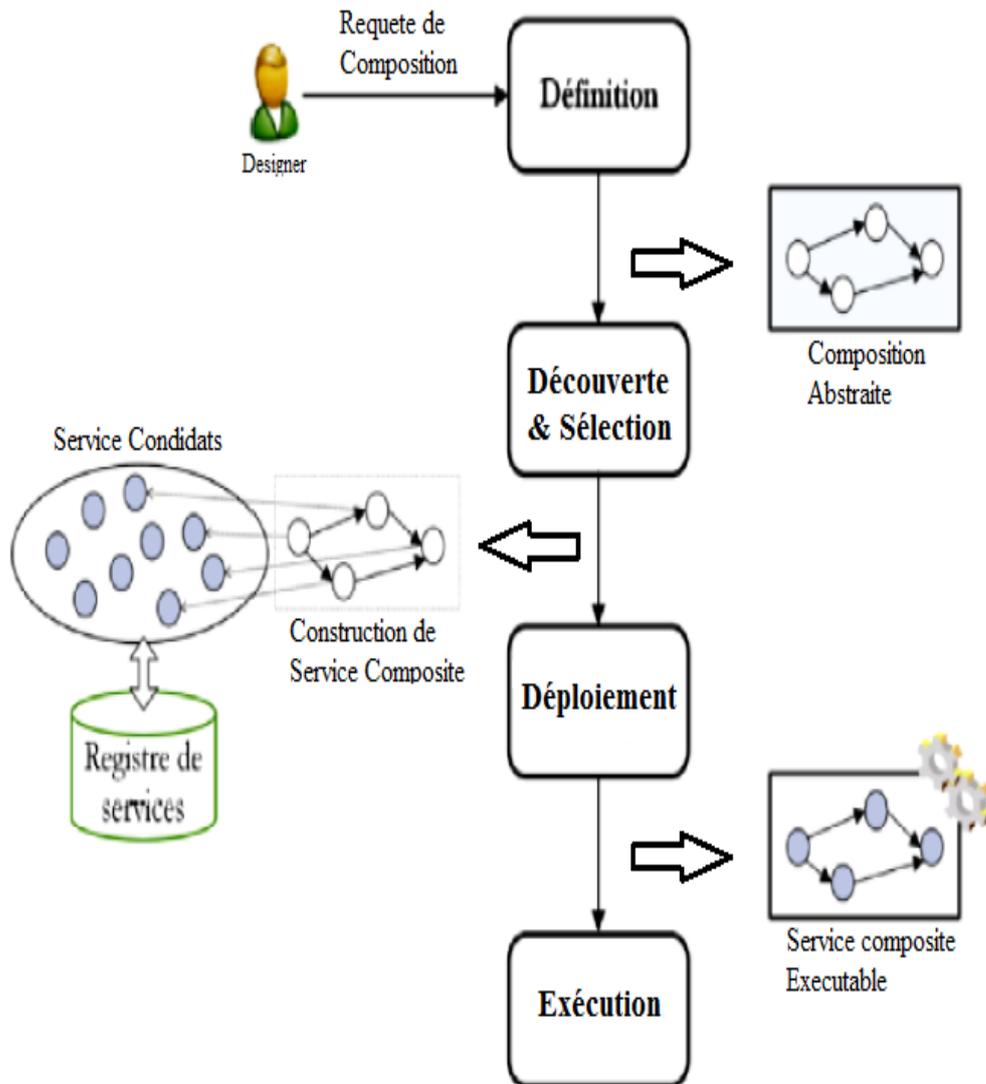


Figure 3.3 - Processus du composition de services.

#### 3.4.1 Définition abstraite

La définition de la composition de services a comme point de départ l'identification des fonctionnalités qui doivent être remplies par l'application ainsi créée. Elle demande l'identification des fonctionnalités abstraites que les différents participants doivent fournir ainsi que celle des interactions qui auront lieu entre ces participants.

### 3.4.2 Découverte et sélection

Cette phase consiste à identifier les services nécessaires à la composition afin de répondre aux besoins fonctionnels identifiés au préalable. Une recherche dans le registre de services permet la découverte des services répondant aux fonctionnalités de chaque service abstrait de la composition. Cette recherche se base généralement sur la description (syntaxique ou sémantique) des services disponibles. Le résultat de la phase de découverte consiste en général en plusieurs services candidats possédant des fonctionnalités similaires, mais des propriétés non-fonctionnelles (*QoS*) différents. La phase de sélection permet, quant à elle, selon les propriétés non fonctionnelles, de déterminer, dans l'ensemble des services candidats précédemment identifiés, les services les plus appropriés pour la composition

### 3.4.3 Déploiement

Lors de cette phase, les services candidats sélectionnés précédemment sont déployés sur des plateformes permettant leur instanciation et invocation par les utilisateurs finaux.

### 3.4.4 Exécution

Le service composite instancié est exécuté par le moteur d'exécution qui est aussi responsable de l'invocation des services composants. L'exécution d'un service composite peut être vue comme une séquence d'échange de message entre les services composants. Cet échange consiste en transfert des données de sortie d'un service composant aux services composants qui le suivent immédiatement dans la description de service composite.

## 3.5 Les valeurs de qualités de services

Les services web sont des entités logicielles qui répondent généralement à des besoins fonctionnels et des besoins non fonctionnels, les besoins non fonctionnels des services web s'expriment généralement à travers la qualité de service (*QoS*), vue la prolifération des services offrant des fonctionnalités similaires, l'établissement des critères de choix entre ces services est basé donc sur les paramètres des *QoS* de ces derniers. En général, la recommandation ITU-X.90212 [26] définit la *QoS* comme « un ensemble d'exigences dans le comportement collectif d'un ou plusieurs objets ». Donc on peut dire que la qualité de service web *QoS* [27] désigne l'aptitude Intrinsèque d'un service à répondre de manière adéquate à des exigences, qui visent à satisfaire ses usagers. Ces exigences peuvent être liées à plusieurs aspects d'un service tels que : la disponibilité, le temps de réponse, le coût et la fiabilité... etc. Dans cette partie, nous examinons les valeurs de qualités de services en matière de service Web

### 3.5.1 Les valeurs de qualités de services utilisé

Un attribut de qualité de service  $Q_i$  est une valeur représentant la propriété non fonctionnelle d'un service Web, telle que le temps de réponse, le débit, le coût, la réputation, ... etc. Les attributs QoS peuvent être classés en deux classes. L'un est négatif, plus la valeur est élevée, plus la qualité est faible, comme le temps de réponse et le coût. L'autre est positif, plus la valeur est élevée, plus la qualité est élevée, comme le débit et la réputation.

Nous citons dans ce qui suit, une liste de certains des valeurs QoS applicables aux services web, définis par le Consortium W3C :

- **Le coût** (ou le prix) : désigne le Coût de chaque invocation du service web.
- **Temps de réponse** : désigner le temps pris par un service pour répondre à une requête.
- **Le débit** : désigne la quantité de demandes que le service est en mesure de traiter dans un intervalle de temps donné.
- **La fiabilité** (aussi connue comme le taux d'exécution avec succès) : ce paramètre désigne la capacité du service à accomplir sa fonction correctement pendant une période de temps spécifiée. Il peut être mesuré par le temps moyen entre pannes.
- **La disponibilité** : désigne la probabilité que le service soit actif, qui s'exprime par le rapport entre le temps où le service est opérationnel en répondant aux demandes des clients d'une part et de la durée dont laquelle le client souhaite que le service soit actif de l'autre part.
- **La sécurité** : désigne les conditions assurant que les échanges de messages entre le client et le service soient sécurisés à savoir : la confidentialité, le cryptage des messages et le contrôle d'accès.
- **La réputation** : c'est une mesure de la crédibilité du service. Elle dépend principalement des expériences d'utilisateurs finaux.

Les attributs QoS peuvent être classés en deux (02) grandes parties :

- ❖ **Attributs non mesurables** : La sécurité, la réputation, ... etc.
- ❖ **Attributs mesurables** : Comme la disponibilité, la fiabilité, le Coût et le temps de réponse.

Les attributs QoS mesurables peuvent aussi être classés en deux catégories :

- ❖ **Les attributs à maximiser** : Comme la disponibilité et la fiabilité lesquels le plan résultant doit contenir les valeurs maximales.
- ❖ **Les attributs à minimiser** : Comme le temps de réponse et le coût lesquels le plan résultant doit contenir les valeurs minimales.

### 3.6 Travaux connexes

Dans cette section nous allons discuter sur quelques travaux réalisés reliaient à notre travail qui est la composition des services et noter quelques points.

Dans le travail présenté en [28], les auteurs ont travaillé sur le problème de déploiement des services dans le cloud. Les auteurs utilisent la méthode de composition de service qui se base sur l'Algorithme évolutionnaire Niched Pareto Genetic Algorithm -NPGA- comme solution à ce problème. En outre dans ce travail les auteurs prennent en considération l'aspect non-fonctionnel. Les auteurs se basent sur le cloud computing comme un domaine d'application de leur travail. Mais nous voyons que l'impasse de ce travail est posée lorsque la sélection des niches Pareto.

Les auteurs de [29], ont présenté une approche pour résoudre le problème de maintenir un équilibre convenable entre exploration exploitation posée par la composition de service. De plus l'idée proposée dans ce travail est basée sur la stratégie Eagle avec l'algorithme d'optimisation des baleines ESWOA. L'aspect non fonctionnel est tenu en compte dans ce travail. Mais ils ont ignoré quelques valeurs de qualité comme la disponibilité.

Munghai Yuan et Zhuo [30], ont proposé une approche pour résoudre le problème de composition de service dans le cloud manufacturing. Cette méthode est basée sur des exigences des tâches et la combinaison des objectifs de la qualité de service. Mais ils ont éliminé les objectifs réputation et la disponibilité.

Dans le travail présenté en [31], les auteurs ont proposé une méthode permettant d'assister la résolution du problème de composition de service. La méthode se base sur une nouvelle composition de service cloud prenant en compte la métrique de fitness optimale en utilisant l'algorithme -IWO modifié - Modified Invasive Weed Optimization. Ce travail prend en considération l'aspect non fonctionnel dans le problème de composition. Mais je pense que ce travail à manquer un objectif important de qualité de service qui est la réputation.

### 3.7 Synthèse bibliographique des travaux

Dans le tableau.1, nous représentons une comparaison entre les différents travaux réalisés lier a notre travail citer au-dessus :

Tableau 3.1- les différentes travaux proposés pour résoudre le problème de composition.

Critère de Travail	Méthode utilisée	Valeurs de QoS (mentionnée ou non)	Domaine d'application	Limites
<b>Merizig et al., 2019 [7].</b>	Algorithme évolutionnaire NPGA	Oui	Déploiement des services	La sélection selon Niched Pareto
<b>Gavvala et al., 2019 [8].</b>	La stratégie Eagle avec l'algorithme ESWOA	Oui	Composition des services dans le cloud	Absence de quelques objectifs de Qos
<b>Jatoth et al., 2019 [9].</b>	Optimal fitness aware avec l'algorithme IWO	Oui	Composition des services dans le cloud	Absence de l'attribut : réputation de Qos
<b>Yuan et al., 2020 [10].</b>	- Modèle mathématique - Méthode d'évaluation	Oui	Composition des services dans le cloud Manufacturing	Absence de quelques objectifs de Qos

### 3.8 Conclusion

La composition de service est un point crucial qui a un grand impact sur plusieurs domaines de recherches. Elle pose un challenge pour les chercheurs à cause de la diversité de ces derniers. De nombreux efforts ont été fournis afin de permettre une composition optimale utilisable et acceptable de service répondre aux exigences des clients et satisfaire ses besoins.

Ce chapitre, comme nous avons vu, dresse une présentation des problèmes de composition de service et son processus. De plus il discute les valeurs de qualité de service. En outre dans ce chapitre nous mettrons en relief une synthèse bibliographique concernant les travaux réalisés pour résoudre le problème de composition de service et nous avons discuté la variété des solutions proposées. Parmi les travaux réalisés il existe quelques points qui ne sont pas tenus en

compte lors de la résolution du problème de composition tel que l'absence de quelques objectifs de qualité de service en particulier comme nous avons vu au-dessus.

# *Chapitre 4*

*conception de système*

# Chapitre 4

## Conception De Système

### 4.1 Introduction

Il est important d'avoir un lot de services atomiques simples fonctionnant ensemble ; par conséquent, il y a un fort besoin d'intégrer un système de composition de service dans les environnements Cloud Computing [30]. Malheureusement, les services individuels optimaux qui sont fournis par les fournisseurs de services sélectionnés ont une qualité variée en termes d'attributs système. Cela signifie que les rassembler pour former le service compliqué nécessaire doit également être optimal [31].

L'idée de composer des processus commerciaux en déterminant et en exécutant les services les plus appropriés plutôt que de créer de nouvelles applications pour satisfaire un besoin commercial a conduit au lancement de la composition des services [32].

La composition des services a pour objectif de déterminer une combinaison de services en fonction d'une requête d'un utilisateur (client). Du côté du client cette composition semblera un unique service. La composition sera transparente au client même si cette composition sera la combinaison de plusieurs services. Alors dans ce chapitre nous allons décrire les différentes étapes de la conception de notre système de composition de services dans le cloud .

Tout d'abord nous examinons l'architecture globale en termes de description de l'architecture et le fonctionnement du système. Ensuite, nous présentons le processus de déploiement. Finalement ce chapitre se termine par l'architecture détaillée.

### 4.2 Objectif

Notre objectif est de concevoir un prototype d'un système de composition de service. Ce prototype proposé assure la résolution du problème d'optimisation multi-objectifs (MOP) dans la composition. Il implique l'optimisation de deux objectifs ou plus soumis à certaines contraintes qui peuvent entrer en conflit les uns avec les autres, par exemple, maximiser la disponibilité et minimiser le coût ou maximiser la réputation et minimiser le temps de réponse. Ce phénomène est appelé optimalité de Pareto. Pour cela nous allons proposer L'Algorithme d'Optimisation Multi Objectif -NSGA2- pour résoudre ce problème.

### 4.3 L'architecture globale de système

Dans cette section nous allons présenter une vue globale sur l'architecture générale de notre système qui se base sur la composition de services comme une approche pour résoudre le problème de déploiement. De plus nous allons expliquer le rôle de chaque composant. L'architecture de notre système est représentée sur la figure 4.1.

Notre système comporte quatre composants de service (illustrés à la figure 4.1) et agit comme un Broker indépendant, ce qui peut aider les utilisateurs à composer et à gérer le déploiement de services. Premièrement, l'interface système qui va interagir avec l'utilisateur et connaître ses besoins. Ensuite, la recherche & sélection qui va chercher et identifie les services disponibles sur le cloud en fonction des exigences spécifiques. Ensuite, méthode d'optimisation celle qui va faire exécuter l'algorithme d'optimisation NSGA 2 pour chercher les meilleures solutions. Enfin, la phase finale qui va sélectionner et choisir la meilleure solution potentielle afin de l'exposer à l'utilisateur.

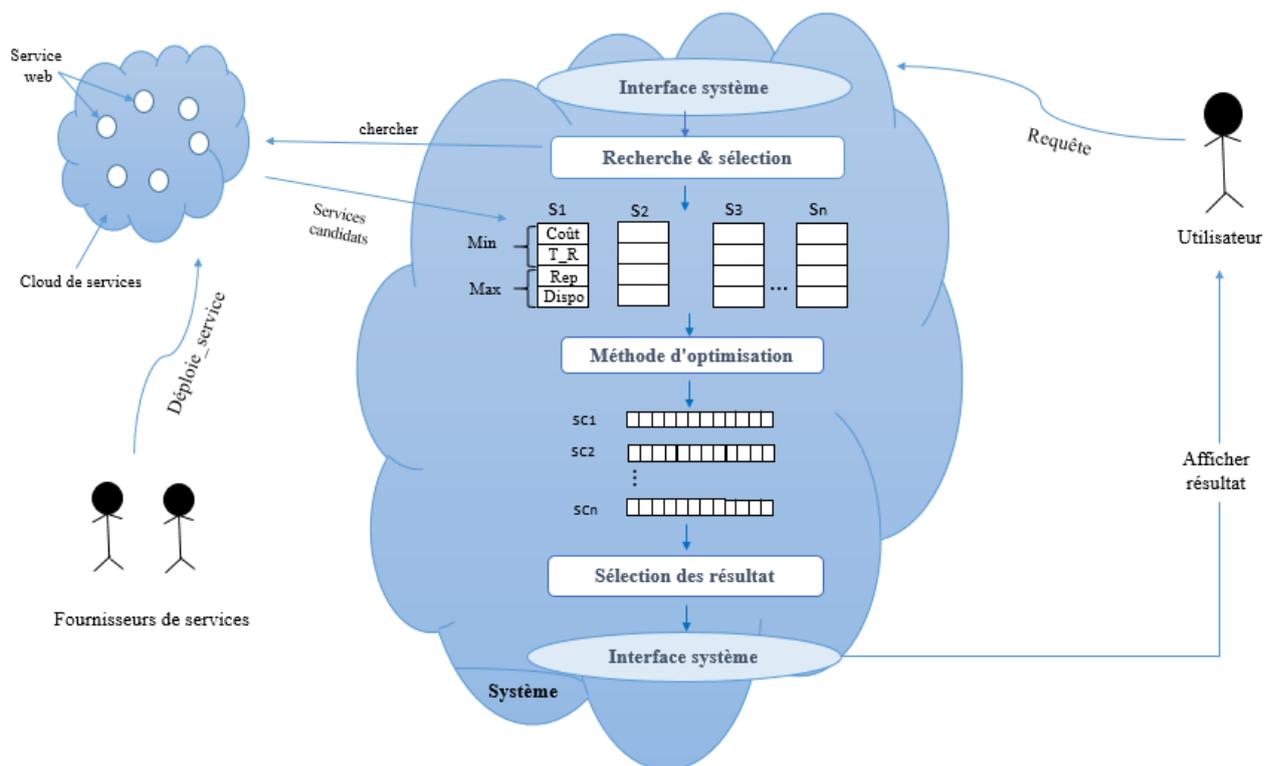


Figure 4.4- Architecture générale du système.

#### 4.3.1 Description de l'architecture

Comme la figure 4.1 représente, l'architecture proposée est composée des composants suivants :

**1. Interface système :**

Dans notre architecture proposée, ce composant représente l'intermédiaire ou bien le lien entre l'utilisateur et le système à travers une interface graphique. En outre, ce composant est responsable principalement de collecter les exigences de service voilait par l'utilisateur et l'affichage des résultats obtenus. Cette unité facilite l'interaction entre le système et le demandeur de service.

**2. Recherche & sélection :**

La tâche principale de ce composant est la recherche des services candidats déployées sur les centres de données (cloud) convenable et satisfiable aux exigences de l'utilisateur, puis il fait la sélection parmi les services concurrents déployés dans l'espace de recherche, et entreprendre la préparation de l'algorithme.

**3. Méthode d'optimisation :**

Ce composant représente la tâche principale de notre système. De plus, ce composant est responsable sur l'exécution de l'algorithme d'optimisation multi objectives NSGA 2 qui est le noyau de notre système.

**4. Sélection des résultats :**

Après l'exécution de l'algorithme d'optimisation proposé, nous allons dans ce composant par la suite, choisir la meilleur solution (solution optimal) parmi un large espace de solutions concurrents, afin de répondre au mieux aux exigences du client. Par la suite, ce composant envois la solution (résultats) obtenue à l'interface système pour l'afficher à l'utilisateur.

**4.4 Fonctionnement globale**

Après avoir eu une vue globale sur l'architecture proposé de notre système dans la section précédente, nous allons présenter, dans cette partie, le fonctionnement général et les interactions fonctionnelle entre les différents composants qui se forme notre système, sous forme d'un diagramme de séquence. La figure ci-dessous représente le diagramme de séquence de toute les taches de notre système.

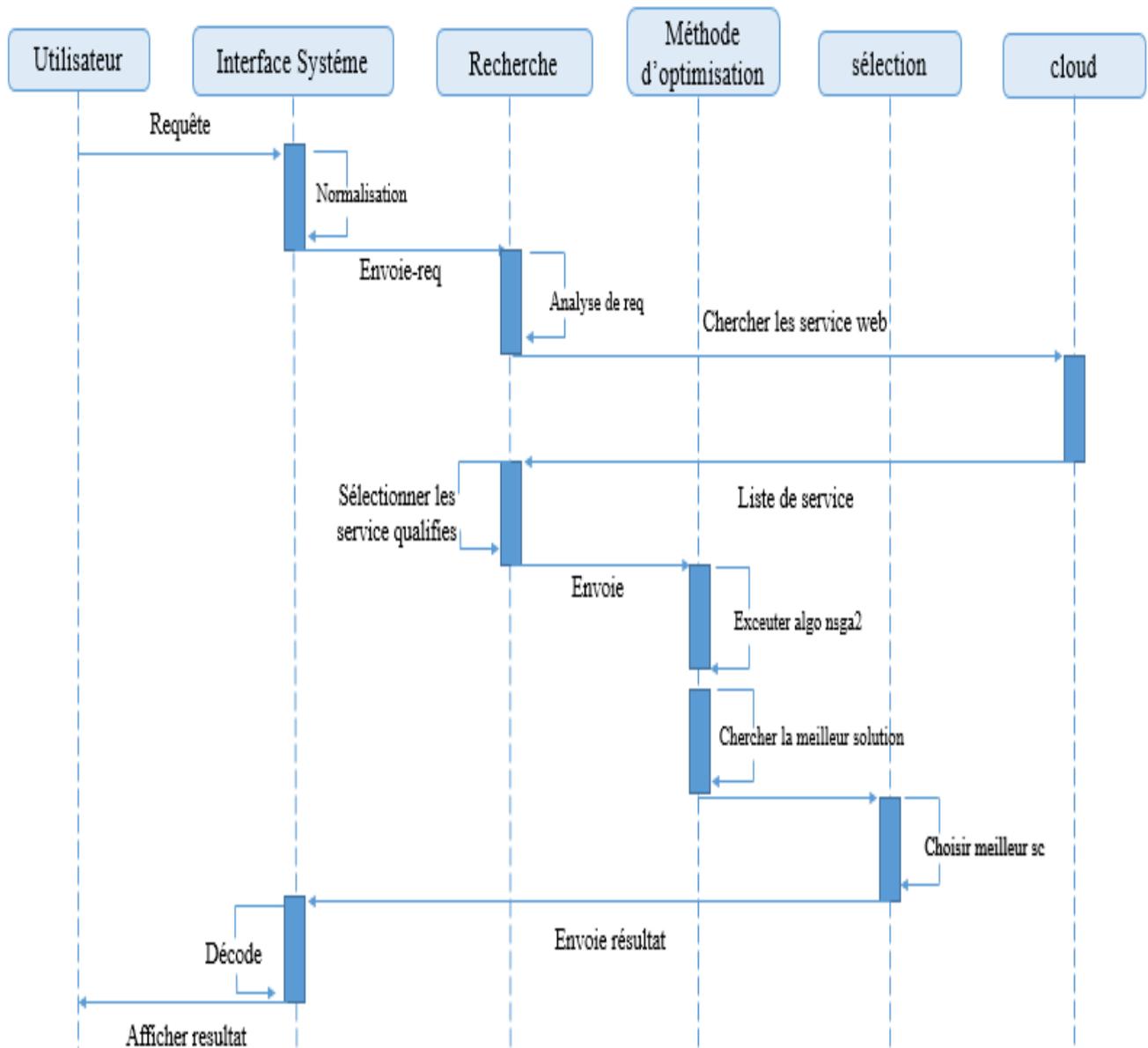


Figure 4.2 - Diagramme de séquence du fonctionnement de système.

### 4.5 Processus de composition

Dans cette partie, nous allons présenter le processus de composition de services et mettre en relief le déroulement de ce dernier. La figure ci-dessous illustre le processus de composition proposé :

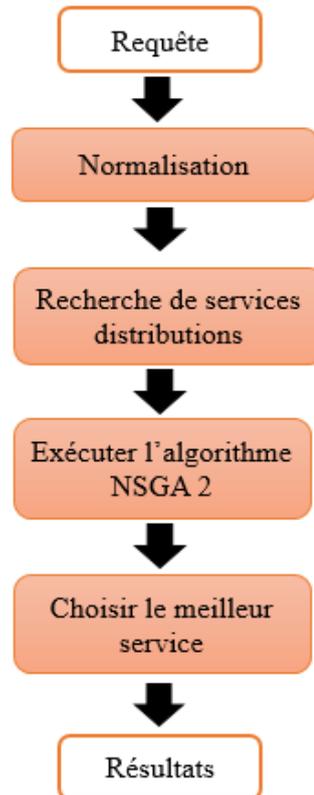


Figure 4.3 - processus de composition.

#### 4.5.1 Normalisation

Lorsque les variables de données générées à partir des distributions peuvent être différentes (et non normales), une transformation (standardisation) devrait donc être nécessaire. Outre cette méthode est utilisée pour que les variables, mesurées à différentes échelles, ont des valeurs comparables et mesurables. De plus, cette technique consiste à normaliser les variables pour amener les données sur l'échelle de 0 à 1 en soustrayant le minimum et en divisant par le maximum de toutes les observations. Cette étape de prétraitement est importante pour préparer les données pour l'algorithme.

la normalisation effectuée selon la relation suivante :

$$S.QoS_q = \frac{S.QoS_q - Min_q}{Max_q - Min_q}$$

### 4.5.2 Recherche des services

Cette étape est basée sur la recherche des services qui répondent le plus, à la requête spécifiée par l'utilisateur, qui est basée sur les qualités de services publiés dans le cloud, avec celles définies dans la requête. Pour optimiser la recherche, la tâche recherche des services interagit avec les requêtes afin d'adapter les résultats de la découverte aux attentes des utilisateurs.

### 4.5.3 Exécuter l'algorithme NSGA 2

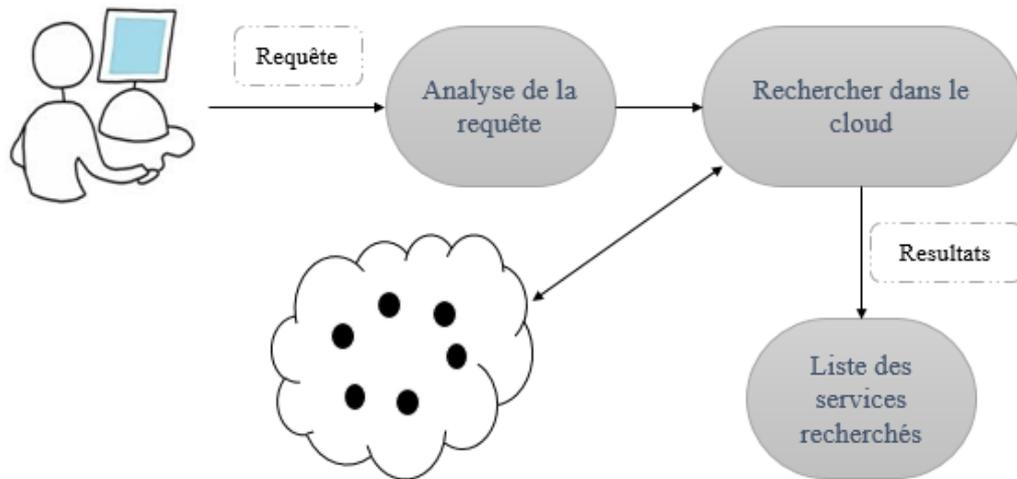


Figure 4.4 - Recherche des services.

Cette étape consiste sur l'exécution de l'algorithme d'optimisation proposé NSGA-ii. D'abord nous allons décrire les principales notions concernant cet algorithme.

✓ Définition 1 (solution optimale de Pareto) :

La relation de dominance signifie que s'il y a deux solutions  $x_1$  et  $x_2$  pour le problème de minimisation, si le fitness remplit la condition :

$\forall m \in (1, 2, \dots, M) : f_m(x_1) \leq f_m(x_2) \wedge \exists m \in (1, 2, \dots, M) : f_m(x_1) < f_m(x_2)$ , alors  $x_1$  domine  $x_2$ . Si  $x_1$  n'est dominé par aucune autre solution, alors  $x_1$  est une solution non dominée, également connue sous le nom de solution optimale de Pareto.

✓ Définition 2 (ensemble non dominé et front de Pareto) :

L'ensemble  $F \{1\}$  composé des solutions optimales de Pareto est appelé l'ensemble non dominé. La surface formée par les solutions optimales de Pareto est appelée front de Pareto. Les solutions du front de Pareto ne peuvent pas être dominées par les autres à l'intérieur ou à l'extérieur de sa surface.

❖ **Le déroulement du NSGA-2**

NSGA-2 comporte trois étapes importantes : le tri non dominé, le calcul de la distance de crowding et la stratégie élitiste.

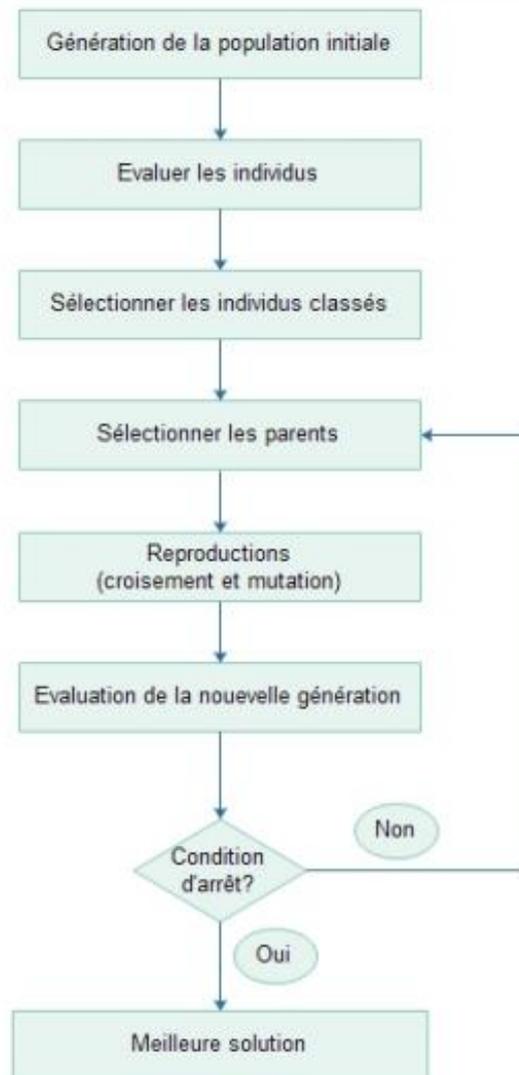


Figure 4.5 - organigramme du processus NSGA-II.

### Processus de l'algorithme NSGA 2

— Les étapes de l'algorithme NSGA-II : Comme les algorithmes évolutionnaires, l'algorithme NSGA-II suit une suite d'étapes présentées comme suit avec quelques modifications :

1. Génération de la population initiale : Initialisez aléatoirement la population initiale et la valeur de chaque paramètre dans l'algorithme NSGA2, puis définissez le numéro d'itération sur 1; puis lancez le calcul d'itération de l'algorithme.
2. Evaluer les individus : Calculer la fonction de fitness des individus. De notre travail nous avons utilisé le score de chaque service composite (individu).

$$I_q = \sum_{i=1}^4 QoS_{ij} ; j = 1..3 \quad (4.3)$$

Où **I** représente un service de type **q**.

3. Sélectionner les individus classés : Dans cette étape nous utilisons la notion de dominance afin de définir les fronts selon la définition de Pareto.

4. Sélectionner les parents

5. Reproductions : cette consiste à exécuter deux opérations.

— Croisement : dans cette opération, nous appliquons le croisement en utilisant l'opération traditionnelle inspirée depuis le phénomène biologique et utilisée dans l'algorithme génétique basique. Afin de reproduire des nouvelles solutions en utilisant les individus existants ,dans notre travail ,nous avons choisi deux individus de façon aléatoire et on a choisi un point pour faire le croisement.

— Mutation : dans cette opération, nous changeons un génome (type de service) de façon aléatoire avec d'autre individu depuis la population courante, en prenant en considération la même condition mentionnée dans l'opérateur de croisement.

6. Evaluation de la nouvelle population : dans cette étape nous utilisons la fonction fitness pour évaluer la nouvelle génération, ensuite nous calculons similarité entre la requête de l'utilisateur et les résultats obtenus. A la fin de cette étape nous choisissons la population qui a le meilleur score pour refaire l'algorithme.

7. Condition d'arrêt : dans les algorithmes évolutionnaires il existe deux critères d'arrêt nous avons utilisé le nombre d'itération.

#### 4.5.4 Choisir la meilleure solution

Dans cette partie, nous allons choisir la meilleure solution obtenu par le filtrage et la sélection des  $N$  meilleur solution par la sélection de  $N/2$  meilleur solutions de l'étape  $i - 1$  avec  $N/2$  meilleur solution de l'étape  $i$ .

#### 4.6 Conclusion

Dans ce chapitre, nous avons présenté la conception de notre système proposé qui consiste à résoudre le problème de composition de services dans le cloud à cause de la diversité des services et des objectives de Qos de ce dernier. Nous avons proposé une solution pour l'utilisateur, cette solution fournit à ce dernier un service composite selon l'aspect fonctionnel et non-fonctionnel de service.

Nous avons proposé dans ce chapitre une architecture basée sur l'optimisation multi objectifs assuré par l'algorithme NSGA 2. Comme nous avons vu dans ce chapitre la relation de dominance nous encourage à confirmer les meilleurs choix grâce aux paramètres Qos.

# Chapitre 5

## Implémentation et mise en œuvre

## Chapitre 5

# Implémentation Et Mise En Œuvre

### 5.1 Introduction

Après la présentation de l'architecture de notre système dans le chapitre précédent, nous allons dans ce qui suit, décrire les différents aspects techniques liés à l'implémentation et le déploiement de notre système. Dans le processus du développement, l'implémentation d'un logiciel vient après un enchaînement de plusieurs étapes et son but principal est de réaliser un produit capable de résoudre les problèmes posés en utilisant des outils et des algorithmes. Dans le but de réaliser et valider les idées proposées dans les chapitres précédant, le présent chapitre montre les outils et les configurations utilisées afin de développer ce système. Dans ce chapitre nous allons présenter les outils et les plateformes utilisés pour développer notre système. Ensuite, nous présentons quelques interfaces qui décrivent notre système. Puis, nous présentons une discussion sur les résultats.

### 5.2 Outils et Plateformes Utilisées

Pour mener à bien notre travail, nous avons fait appel à des outils et de langages de développement permettant la réalisation et la mise en œuvre de notre application.

Dans cette section, nous allons présenter les plateformes utilisées afin de réaliser notre système. Les approches proposées dans notre travail sont implémentées sur un ordinateur :

- Processeur Intel Core I7-3632QM CPU @ 2.20 GHz.
- RAM de 8 Go.
- Système d'exploitation : Windows 8, 64 bits processeur x64.

Le simulateur de sélection et de composition de services sensible à la QoS a été développé sur la plateforme JetBrains PyCharm Community Edition 2017.3.3 x64 utilisant le langage Python. Ce dernier est un langage orienté objet, simple, portable, et possède une bibliothèque riche de classes comprenant diverses fonctions.

### 5.3 Environnement de développement

Notre choix du langage de programmation s'est porté sur le langage python.

#### 5.3.1 Python

Python est un langage de programmation interprété, multi paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Ruby. Le langage Python est placé sous une licence libre et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux, de Windows à Unix. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

### 5.3.2 Plateforme PyCharm

PyCharm est un environnement de développement intégré utilisé pour programmer en Python. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web. Développé par l'entreprise tchèque JetBrains, c'est un logiciel multi-plateforme qui fonctionne sous Windows, Mac OS X et Linux.

### 5.3.3 Tkinter

Tkinter (de l'anglais Tool kit interface) est la bibliothèque graphique libre d'origine pour le langage Python, permettant la création d'interfaces graphiques. Elle vient d'une adaptation de la bibliothèque graphique Tk écrite pour Tcl.

## 5.4 Pseudo code de l'algorithme proposé

Voici le pseudocode de l'Algorithme NSGA-2 :

---

### Algorithme NSGA-II

---

Génération de population initial (N) ;

Evaluation des individus () ;

Trie et définition des fronts () ;

Répéter

    Sélection des parents () ;

    Croisement () ;

```
Mutation () ;  
Evaluation de nouvelle population () ;  
Jusqu' à test d'arrêt atteint ;
```

---

Le déroulement de l'algorithme NSGA II représenté dans le pseudo code ci-dessus consiste sur les étapes suivantes :

Premièrement la génération de population initial de taille N ensuit, l'évaluation des individus selon la fonction de fitness. En suit l'étape de tri nom dominé, c'est le processus de tri basé sur des critères de nom dominance de la population initialisée. En suite sélection des parents pour la reproduction des individus l'opération génétique qui se base sur le croisement et la mutation. Après, c'est l'étape d'évaluation qui va faire évoluer la nouvelle population puis on vérifiera si la condition d'arrêt est atteinte.

### **5.5 Présentation de l'interfaces du système**

La figure 5.1, représente la composant interface système dont nous l'avons mentionnée dans le chapitre précédent.

Ce composant représente le lieu entre le monde. De plus, cette composant se produit l'interconnexion entre le système et le client, elle permet à l'utilisation de sélectionner ses exigences concernant les qualités de service (QoS) et lancer la recherche et la composition des service voulait.

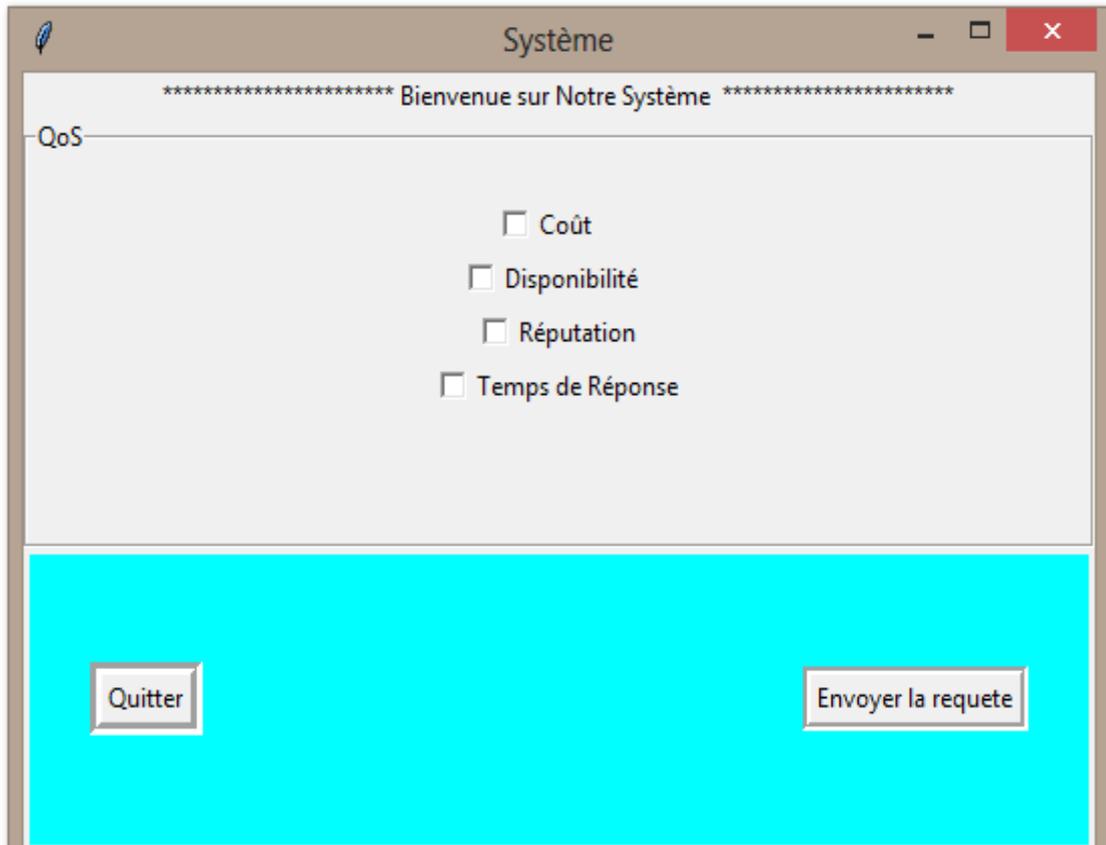


Figure 5.1 - interface client pour la recherche de services.

## 5.6 Résultat et discussion

Dans nos expériences, nous avons utilisé les paramètres mentionnés précédemment pour l'exécution de l'algorithme NSGA-II qui a été évaluée avec la relation de dominance et la distance de crowding. De plus, ces expériences ont utilisé différentes configurations pour les mêmes données mentionnées précédemment et les mêmes valeurs de demande. L'algorithme implémenté utilisait différentes générations entre les intervalles [1–150] (voir la colonne d'itération dans le tableau 5.1). Le tableau 5.1 suivant explique les valeurs moyennes pour chaque QoS liées à la meilleure combinaison obtenue à l'aide de notre algorithme.

Tableau 5.1 : Résultats des valeurs moyennes de QoS obtenues.

<i><b>Itération</b></i>	<i><b>Coût</b></i>	<i><b>Temps de réponse</b></i>	<i><b>Réputation</b></i>	<i><b>Disponibilité</b></i>
<b>1</b>	22.2	1,144.2	0.67	0.98
<b>20</b>	22.56	1,220.8	0.64	0.98
<b>50</b>	21.38	1,274.02	0.64	0.97
<b>100</b>	23.16	1,338.3	0.71	0.96
<b>150</b>	23.70	1,389.4	0.76	0.95

### **5.7 Conclusion**

La composition de services dans le Cloud est le problème épineux à cause de la diversité du nombre des services déployés dans le Cloud. Afin de déterminer l'efficacité d'un système, nous devons l'utiliser plusieurs fois pour dire que le système est fiable.

A travers ce chapitre, nous avons réalisé un système de composition de services dans le cloud cet objectif est concrètement atteint par l'utilisation d'un algorithme évolutionnaire multi objectif NSGA II. Au cours de ce chapitre, nous avons présenté les outils et les plateformes utilisées afin d'implémenter le système de la composition de services dans le Cloud par les algorithmes évolutionnaire. Ensuite, nous avons présenté les configurations utilisées dans notre approche, cela peut aider l'application de fournir des résultats fiables. Par conséquent, ceci peut rendre les clients satisfait de ces exigences. Puis, nous avons montré l'interface système de notre application suivie par une explication suivi par un tableau qui discute les résultats collectés.

# **CONCLUSION GENERALE**

## **Chapitre 6**

### **Conclusion Générale**

#### **6.1 Conclusion**

A cause de l'élargissement des services web sur internet, il est rare pour un utilisateur de découvrir un service qui réponde à ses besoins. Pour cela, plusieurs services peuvent interagir et échanger dynamiquement des informations. La composition de services vise la construction automatique de services complexes, à base de services atomiques, ceci en passant par le processus de sélection. En effet la sélection de services consiste à choisir le service le plus approprié pour réaliser une fonctionnalité donnée en tenant compte de plusieurs propriétés non fonctionnelles des services et les exigences de l'utilisateur et ses préférences.

Pour résoudre le problème posé, nous avons proposé une approche de sélection et de composition de services cloud sensible à la QoS, cette approche permet d'une part la sélection de services fiables, et d'une autre part la garantie du respect des exigences globales de l'utilisateur.

Afin de valider l'approche proposée, nous avons développé un système écrit en langage PYTHON. Le système basé sur l'algorithme évolutionnaires multi objectifs NSGA II, montre une meilleure performance en termes de garantir la fiabilité des objectifs de qualités de services maximiser (disponibilité, réputation) et minimiser (cout, temps de réponse).

#### **6.2 Perspectives**

En guise de perspectives, nous envisageons de :

- Prendre en compte d'autres modèles de composition (parallèle, conditionnelle, etc.).
- La prise en charge de la réparation. En effet quand un service est inaccessible ou défaillant, le système doit pouvoir le remplacer par un autre service au moment de l'exécution. (Proposer un mécanisme de surveillance).
- Implémentation de notre approche dans un environnement réel, en se basant sur une plate-forme d'expérimentation.

## Références

- [1] M. MEHDARY, B.L. EL HABIB and A. TRAGHA, *Proposition d'un meta-model fractionner en module pour une approche hybride du cloud computing*, **2012**.
- [2] P. Mell and T. Grance, *The NIST definition of cloud computing*, <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>, **2011**.
- [3] P.Nerkar & S.Rathod, *Weather Analytics Using Machine Learning Techniques On Openstack Cloud*, [www.ibm.com/cloud-computing/us/en/whatis-cloud-computing.html](http://www.ibm.com/cloud-computing/us/en/whatis-cloud-computing.html).
- [4] G. IT, *Gartner IT glossary*, Technology Research, **2013**.
- [5] H. MATAALLAH, *Vers un nouveau modèle de stockage et d'accès aux données dans les Big Data et les Cloud Computing*, Thèse de doctorat, **2019**.
- [6] Vaquero, L, Rodero-Merino, L, Caceres, J and Lindner M. *A break in the clouds : towards a cloud definition*. ACM SIGCOMM computer communications review. Vol 39, Issue 1, pp 50-55, **2009**.
- [7] R. L. K. R. D. Vines and R. Krutz, *Cloud security: A comprehensive guide to secure cloud computing*: Wiley Publishing, Inc, **2010**.
- [8] A. Jula, E. Sundararajan and Z. Othman, *Cloud computing service composition: A systematic literature review*, Expert systems with applications, vol. 41, pp. 3809-3824, **2014**.
- [9] Höfer and G. Karagiannis, *Cloud computing services: taxonomy and comparison*, Journal of Internet Services and Applications, vol. 2, no. 2, pp. 81-94, **2011**.
- [10] [Enligne].Available:<https://cloud.google.com/developers/articles/buildig-high-availabilityapplications-ongoogle-compute-engine/>.
- [11] A. K. Y. SETTOUTI, *Indexation et sélection des services Cloud Computin IaaS pour des réseaux de capteurs*, Thèse de doctorat, **2018**.
- [12] Z. Kartit, *Contribution à la sécurité du Cloud Computing: Application des algorithmes de chiffrement pour sécuriser les données dans le Cloud Storage*, Thèse de doctorat, **2016**.
- [13] A. Merizig, *Approche de composition de services web dans le Cloud Computing basée sur la coopération des agents*, PhD thesis, Université Mohamed Khider-Biskra, **2018**.
- [14] B. Beldjilali, *Gestion de Confiance dans le Cloud Computing*, PhD thesis, Université d'Oran, **2016**.
- [15] S. Sadi. *Techniques de Checkpointing pour la Tolérance aux Fautes dans le Cloud Computing*, PhD thesis, Université d'Oran , Algérie, **2017**.
- [16] K. Gai and S. Li. *Towards Cloud Computing : A Literature Review on Cloud Computing and Its Development Trends*, Fourth International Conference on Multimedia Information Networking and Security, p 142–146, IEEE, **2012**.

- [17] R. F. Roggio, T. Bilyayeva, and J. R. Comer. *Everyday Cloud Computing with SaaS*. International Conference on Software Engineering Research and Practice SERP12, **2012**.
- [18] S. P. Mirashe and N. V. Kalyankar. *Cloud Computing*. Journal of Computing, 2(3) 78–82, **2010**.
- [19] B. Armbrust, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin. *A review of Cloud Computing*. *Communications of the ACM*, 53(4) 50–58, **2010**.
- [20] M. Armbrust, A. D Joseph, R. H Katz, and D. A Patterson. *Above the Clouds : A Berkeley View of Cloud Computing*. Technical Report, EECS Department, University of California, Berkeley, **2009**.
- [21] A. Bekkouche, *Composition des Services Web Sémantiques À base d'Algorithmes Génétiques*, Thèse de doctorat, **2012**.
- [22] M. A. Yagoub, *Une approche basée agent pour la sécurité dans le Cloud Computing*, Thèse de doctorat, Université Mohamed Khider de Biskra, **2019**.
- [3] R. Abdellah and M. Imane, *Composition des Services Web*, Mémoire de, Master, **2016**.
- [23] N. Arenaza, *Composition semi-automatique de Services Web*, mémoire de Master, Ecole Polytechnique Fédérale de Lausanne, **2006**.
- [24] F. JoséMoo-Mena, *Modélisation des architectures logicielles dynamiques : application à la gestion de la qualité de service des applications à base de services web*. Réseaux et télécommunications. Institut National Polytechnique de Toulouse, **2007**.
- [25] U. G. Shehu, G. Epiphaniou and G. A. Safdar, *A survey of QoS-aware web service composition techniques*, *International Journal of Computer Applications*, **2014**.
- [26] A. Merizig, O. Kazar and M. López-Sánchez, *A multi-agent system approach for service deployment in the cloud*, *International Journal of Communication Networks and Distributed Systems*, vol. 23, pp. 69, **2019**.
- [27] S. K. Gavvala, C. Jatoh, G. R. Gangadharan and R. Buyya, *QoS-aware cloud service composition using eagle strategy*, *Journal of Future Generation Computer Systems*, vol. 90, pp. 273-290, **2019**.
- [28] M. Yuan, Z. Zhou, X. Cai, C. Sun and W. Gu, *Service composition model and method in cloud manufacturing*, *Journal of Robotics and Computer-Integrated Manufacturing*, vol. 61, pp. 101840, **2020**.
- [29] C. Jatoh, G. R. Gangadharan and U. Fiore, *Optimal fitness aware cloud service composition using modified invasive weed optimization*, *Journal of Swarm and Evolutionary Computation*, vol. 44, pp. 1073-1091, **2019**.
- [30] H. Mezni and M. Sellami, “*Multi-cloud service composition using formal concept analysis*,” *Journal of Systems and Software*, vol. 134, pp. 138-152, **2017**.
- [31] S. R. Boussalia, A. Chaoui, A. Hurault, M. Ouederni and P. Queinsec, “*Multi-objective quantum inspired Cuckoo search algorithm and multi-objective bat*

inspired algorithm for the web service composition problem,” International Journal of Intelligent Systems Technologies and Applications, vol. 15, no. 2, pp. 95-126, **2016**.

- [32] F. Dahan, K. El Hindi and A. Ghoneim, “*An adapted ant-inspired algorithm for enhancing Web service composition*,” International Journal on Semantic Web and Information Systems (IJSWIS), vol. 13, no. 4, pp. 181-197, **2017**.