République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. Mira de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de MASTER EN INFORMATIQUE

Intelligence artificielle & Administration sécurité réseau

Thème

Proposition d'un système multi-agents pour la gestion du cloud computing et évaluation de ses performances

Présenté par :
Lydia BECHROUNE & Zidane AGHOUILES

Évalué en septembre 2020 par le jury composé de :

Encadrante : Dr ADEL-AISSANOU Karima U. A/Mira Béjaia.
Co-Encadrante : Dr BOULAHROUZE Djamila U. A/Mira Béjaia.
Présidente : Dr ELBOUHISSI Houda U. A/Mira Béjaia.
Examinatrice : Mme BOUKERRAM Samira U. A/Mira Béjaia.

Béjaia, Session normale septembre 2020.

⁕ *Remerciements* ⁕

Nos sentiments de reconnaissance vont en premier lieu à l'endroit de nos encadrantes, Mme. D. BOULAHROUZ et Mme K. ADEL, elles n'ont pas hésité à nous soutenir dès le départ et à accepter d'endosser la charge de promoteur, malgré leurs nombreuses occupations. Leurs encouragements nous ont beaucoup édifiés.

On remercie tous les membres du jury Mme H. ELBOUHISSI et Mme BOUKERRAM pour avoir bien voulu donner de leur temps pour lire et évaluer ce travail.

On remercie toutes les personnes qui, d'une quelconque manière, nous ont apporté leur amitié, leur attention, leurs encouragements, leur appui et leur assistance pour qu'on puisse mener à terme ce travail : nos familles respectives nos amis et proches, et tous ceux qui nous ont encouragés et soutenus. On ne saurait citer chacun par son nom. Que tous trouvent ici l'expression de nos franches et profondes reconnaissances !

AGHOUILES Zidane & BECHROUNE Lydia

⁂ *Dédicaces* ⁂

Je dédie ce travail :
À mes deux parents qui n'ont jamais cessé de prier pour moi, aucun hommage ne pourrait être à la hauteur de l'éducation et de l'amour qu'ils m'ont donné, qui a fait de moi ce que je suis aujourd'hui.
Ma tendre mère qui a toujours su trouver les mots pour m'encourager,
Mon cher père qui a toujours su me montrer le chemin et qui n'a jamais arrêté de croire en moi.
À mes sœurs qui ne m'ont jamais privé de leur conseils et de leur soutien,
À Leur maris et à tous mes neveux et nièces.
À tous mes amies, et à tous les gens que j'ai connu qui ont su me marquer avec leur bienveillance et leur support.
À mon binôme Zidane pour son travail rigoureux son soutien moral, sa patience et sa compréhension,
À toute sa famille également.

BECHROUNE Lydia

⁕ *Dédicaces* ⁕

Je dédie ce travail :
À mes très chers parents qui ont toujours été là pour moi et qui m'ont données un magnifique modèle de labeur et de persévérance.
À mes très chers frères et sœurs qui m'ont toujours soutenu dans n'importe quelle situation.
À Thinhinane qui ma aider soutenue dans mon projet à chaque instant.
À ma binome lydia bachroune pour avoir était une personne de travail exceptionnelle, Je dédié aussi à ma nièce nawara que j'aime tellement.
Je vous dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et vos innombrables sacrifices.
Que ce modeste travail, soit pour vous une petit compensation et reconnaissance envers ce que vous avez fait d'incroyable pour nous. Qu'allah, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.

M. ***AGHOUILES Zidane***

Table des matières

Table des figures

Introduction générale

Le cloud computing a émergé au cours de la dernière décennie pour être largement adopté aujourd'hui dans plusieurs domaines de l'informatique. Il consiste à proposer les ressources sur le marché sous forme de services qui peuvent être consommés de manière souple et transparente.

Il est aujourd'hui une des technologies la plus adoptée, il offre une panoplie d'avantages et de services avec des architectures rentables, il est vu comme étant une interconnexion et une coopération de ressources de l'informatique, située dans des entités et structures internes, externes ou mixtes dont les modes d'accès sont basés sur les protocoles et standard Internet.

Le cloud computing fournit des services à grande élasticité de haute performance et un stockage de données évolutif à un grand nombre croissant d'utilisateurs. De nombreuses stratégies sont utilisées pour fournir des ressources aux utilisateurs et elles sont regroupées sous le terme «allocation des ressources »[49].

Afin de superviser tous les types d'activités liées au cloud, notamment le déploiement de ressources, la gestion des utilisateurs et des fournisseurs, le suivi de l'utilisation, l'intégration des données et même la récupération après sinistre. il est nécessaire de gérer le cloud pour contrôler l'infrastructure, les plateformes, les applications et les données.

Les systèmes multi-agents représentent également un paradigme de calcul et de gestion distribué sous forme d'un ensemble d'agents qui interagissent dans un environnement pour résoudre un problème commun en utilisant les ressources et les connaissances de chaque agent. Les systèmes multi-agents sont souvent distribués et l'agent dispose de fonctionnalités pro-actives et réactives très utilisées.

La théorie des agents et des systèmes multi-agents (SMA) peut fournir un nouveau modèle de gestion des systèmes Cloud computing basé sur la répartition des responsabilités, la flexibilité et l'autonomie. La gestion des fonctions du noyau d'un système Cloud via un modèle basé sur des agents permet aux plateformes résultantes d'être beaucoup plus efficaces, évolutives et adaptables qu'elles ne le sont actuellement. Cependant, joindre les deux modèles de calcul (SMA et Cloud Computing) est un grand défi, étant donné la différence entre les deux modèles. Néanmoins le cloud computing est considéré comme un système ouvert et que

l'application des SMA dans les systèmes ouverts est un défi reconnu dans lequel il y a déjà eu un taux de réussite notable, l'union des deux modèles est un défi réalisable dans le cadre de ce travail.

L'objectif de notre travail est de concevoir un système de gestion du cloud basé agent pour fournir des solutions de cloud computing fondées sur la conception et le développement d'agents logiciels qui peuvent améliorer l'utilisation des ressources cloud, la gestion et la découverte des services, la composition de services et la gestion des clients impatients. En effet les systèmes multi-agents s'adaptent bien à la conception d'un système d'allocation des ressources où chaque membre doit gérer et échanger ses connaissances et collaborer avec les autres afin de réaliser ses buts. En plus, dans de tels environnements ouverts, dynamiques et complexes, il y a un besoin de distribuer les données, le contrôle ainsi que l'expertise ce qui rend l'utilisation des systèmes multi-agents bénéfiques.

Ce mémoire est structuré autour d'une introduction générale de quatre chapitres, conclusion générale et d'une bibliographie :

Dans le premier chapitre intitulé «Principes du cloud computing », nous présentons différentes notions relatives au cloud computing nous commençons par un aperçu sur la chronologie du cloud computing suivie de quelques définitions ainsi qu'une vue sur les différentes techniques d'évaluation des performances du cloud computing.

Nous concluons ce chapitre par la présentation de quelques travaux effectués pour la modélisation et l'évaluation des performances du Cloud Computing via les systèmes d'attente.

Dans le deuxième chapitre intitulé «Agents et système multi-agents »nous exposons brièvement les concepts clés d'agents et de systèmes multi-agents, puis nous présentons certains avantages de l'utilisation des SMA dans différents domaines ainsi que quelques plateformes SMA permettant de les développer. Nous concluons ce chapitre par la présentation de quelques travaux effectués sur le cloud computing basé sur les systèmes multi-agents.

Dans le troisième chapitre, intitulé «Architecture du modèle proposé »nous décrivons la contribution que nous apportons pour résoudre le problème de gestion du cloud computing et des clients impatients. Dans un premier temps, nous décrivons le modèle puis nous détaillons les composants de l'architecture proposée ainsi que l'architecture interne de chaque agent. Par la suite, nous exposons différents scénarios relatifs au fonctionnement du système proposé.

Dans le quatrième et dernier chapitre intitulé «Conception et réalisation »nous présentons différents diagrammes décrivant le fonctionnement de notre système, ainsi que les outils

et environnements de développement qui nous ont offert la possibilité d'implémenter notre travail.

Chapitre 1

Principes du cloud computing

1.1 Introduction

La dépendance de l'être humain à la technologie augmente de jour en jour, la demande d'accès aux ressources informatiques à tout instant devient nécessaire, ce qui a amené les chercheurs vers un nouveau paradigme nommé cloud computing, En effet le cloud computing est un concept émergeant en informatique ces dernières décennies, c'est le fruit du développement récent des technologies de l'information, il constitue une véritable révolution, en engendrant de nouvelles possibilités de mutualisation de services, de stockage, de traitement des données et d'économies pour les entreprises et les particuliers tout en adoptant une approche élastique.

L'objectif de ce chapitre est de présenter les principaux concepts du Cloud Computing et les différentes méthodes existantes pour l'évaluation de ses performances.

1.1.1 Chronologie

L'apparition du cloud computing remonte aux années soixante, où le professeur John McCarthy un informaticien visionnaire a suggéré que l'informatique serait un jour vendu comme un utilitaire qui consisterait à fournir des services informatiques à des utilisateurs.

Au début des années 80, des systèmes d'exploitation réseau ont été lancés pour permettre aux ordinateurs de communiquer entre eux [61, 60].

Les technologies fondamentales du cloud ont atteint un certain niveau de maturité dans les années 90 avec l'apparition d'Internet et par le lancement du World Wide Web en 1991, qui rend les données accessibles à tous et partout.

La création du cloud moderne a été réalisée par le lancement d'Amazon Web Services (AWS) en 2002, ces principaux avantages était l'élasticité et l'évolutivité.

Le cloud a allégé le fardeau de la maintenance des serveurs, des investissements initiaux sur les ressources de calcul et de la mise à l'échelle des services Web en fonction de la demande

- en louant des ressources qui étaient concentrées dans de grandes installations et maintenues par des fournisseurs de confiance. [61, 60].

Plus tard, en 2006, Amazon Elastic Cloud (EC2) a été mis à la disposition du grand public en tant que service Web commercial, en introduisant pour la première fois le terme «Cloud computing».

Ce n'est qu'en 2009 que l'institut NIST (National Institute of Standards and Technology) a fourni une définition standard du cloud computing, et ce n'est qu'en 2011 qu'il a mis à disposition une architecture de référence du cloud computing. bien que les technologies du cloud computing ont permis la mise au point de solutions élastique, flexibles et une informatique distribuée, cette technologie ne cesse d'évoluer.

1.2 Description du cloud computing

Dans la littérature, il existe plusieurs définitions concernant le concept du Cloud Computing,

Les auteurs de [43] définissent le cloud computing comme un modèle qui offre aux utilisateurs du réseau un accès à la demande, à un ensemble de ressources informatiques partagées et configurables (par exemple, réseaux serveurs, stockage, applications et services) qui peuvent être rapidement mises à la disposition du client et publiées avec un effort de gestion minimal et sans interaction directe avec le prestataire de service.

Selon [11] le cloud est présenté comme étant une approche permettant de disposer d'applications, de puissance de calcul, de moyens de stockage, etc... en tant que services. Ceux-ci seront mutualisés, dématérialisés (indépendants de toutes contingences matérielles, logicielles et de communication), contractualisés en (termes de performances, niveau de sécurité et coûts), évolutifs en (volume, fonction et caractéristiques) et en libre-service.

le cloud computing est vu par [13] comme un réseau informatique distribué composé de serveurs virtualisés dont les ressources peuvent être approvisionnées de manière dynamique et reposant sur un contrat de service SLA (Service Level Agreement) entre un fournisseur et un client.

Tandis que dans [36] les auteurs considèrent le cloud computing comme une technologie qui fournit des services informatiques, stockage des données, calcul et mise en réseau à plusieurs utilisateurs à la fois, à l'emplacement et selon la quantité de travail qu'ils souhaitent consommer, avec des coûts basés uniquement sur les ressources utilisées.

On peut déduire que le cloud computing n'est pas en soi une technologie nouvelle, il provient de l'aboutissement de plusieurs technologies existantes antérieurement : internet et la virtualisation, le tout appuyé sur un réseau fiable et à haut débit.
La figure 1.1 illustre le concept du cloud computing.

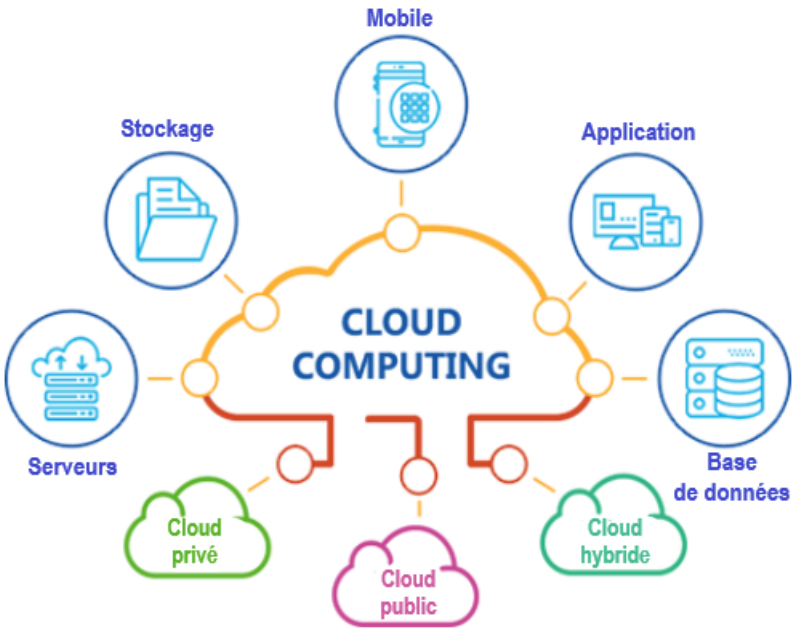

FIGURE 1.1 – Concepts du cloud computing [45].

1.2.1 Caractéristiques principales du cloud computing

La technologie du Cloud Computing a été émergée avec des caractéristiques qui la différencie des autres technologies. D'après [43], les cinq caractéristiques essentielles correspondantes au cloud computing sont :

Un accès en libre-service à la demande

Un consommateur peut commander des ressources informatiques, telles que le stockage sur le serveur et sur le réseau, selon les besoins, de manière automatique sans l'interaction directe avec chaque prestataire de services.

Large accès au réseau

Les ressources sont disponibles et accessibles sur le réseau via des plates-formes clients hétérogènes (téléphones portables, tablettes, ordinateurs portables, etc.).

Une mutualisation de ressources

Des ressources matériels et logiciels tels que la mémoire, capacité de stockage et la puissance de calcul sont mises à la disposition des clients sur un modèle multi-locataires, avec une attribution dynamique des ressources physiques et virtuelles en fonction de la demande.

Une élasticité rapide

En fonction de la demande, les ressources peuvent être ajustées rapidement de manière automatique et déployées de telle façon à ce que les ressources fournies soient conformes à la demande du client.

Un service mesuré en permanence

Les systèmes cloud contrôlent et optimisent automatiquement l'utilisation des ressources et celle-ci peuvent être surveillée, contrôlée et signalée, offrant une transparence pour le fournisseur et le consommateur du service utilisé. Cela garantit un niveau de disponibilité adapté aux besoins spécifiques des clients.

1.2.2 Modèles de service du cloud computing

Différents types de services couvrent la majorité des besoins des consommateurs, de ce fait plusieurs modèles peuvent être utilisés sur un cloud selon les ressources allouées aux clients. D'après la définition donnée dans [43] du Cloud on peut distinguer trois modèles de services tels que :

IaaS (Infrastructure as a Service)
Les services d'infrastructures (Infrastructure as a Service : IaaS) offrent des ressources informatiques générales à la demande telles que des serveurs virtualisés ou diverses formes de stockage en tant que ressources mesurées.
Les clients de services achètent des ressources au lieu de configurer un ensemble de serveurs, logiciels, ou d'espaces dans les data center eux-mêmes. De plus, les utilisateurs de ce type de service sont facturés en fonction des ressources consommées[30].

PaaS (Platform as a Service)
Les fournisseurs de service PaaS(platform as a service : PaaS) proposent une infrastructure logicielle gérée de niveau supérieur, où les clients peuvent construire et déployer des classes d'applications particulières et de services en utilisant des outils, environnements et langages de programmation qui sont pris en charge par le fournisseur. Le consommateur ne gère et ne contrôle pas l'infrastructure cloud sous-jacente, y compris le réseau, les serveurs, les systèmes d'exploitation ou stockage, mais contrôle les applications déployées[43, 30].

SaaS (Software as a Service)
Les services Applications (Software as a Service : SaaS) offrent des applications déjà créées qui s'exécutent dans une infrastructure Cloud. La plupart des services Cloud Computing en tant que logiciels sont des applications web qui peuvent accéder à travers différents périphériques clients afin d'envoyer des données et de recevoir des résultats. De plus, les clients qui utilisent ces services n'ont pas le droit de gérer ou de contrôler l'infrastructure sous-jacente du Cloud ou la plate-forme d'application[43].

1.2.3 Concepts liés aux services cloud

Dans ce qui suit nous définissons les trois concepts principaux liés au service du cloud computing

Qualité de service (Quality of Services QoS)

La qualité de service désigne la capacité d'un service à répondre par ses caractéristiques aux différents besoins de ses utilisateurs ou consommateurs. Dans une Infrastructure matérielle, la QoS a pour but de garantir de bonnes performances aux applications en termes de disponibilité, débit, délais de transmission, taux de perte, etc [24].

Accord de niveau de service (Service Level Agreement SLA)
Un SLA est la formalisation d'un accord négocié entre deux parties. C'est un contrat qui définit la QoS requise entre un prestataire et un client. Il précise de manière claire les objectifs de performance et de qualité. Le fournisseur du service s'engage ainsi par contrat sur une disponibilité de l'outil vis-à-vis des utilisateurs[24].

Objectifs de niveau de services (Service Level Objectives SLOs)
Les SLOs définissent des caractéristiques des services qui seront fournis à un client en termes qualitatifs. Ils sont considérés comme des moyens permettant de mesurer la QoS dans l'objectif d'éviter le malentendu entre les deux parties. Ils peuvent être composés d'une ou plusieurs mesures de QoS. Par exemple, un SLO de disponibilité peut dépendre de plusieurs composants, chacun d'entre eux pourra avoir une mesure de QoS de disponibilité propre[24].

1.2.4 Modèles de déploiement du cloud computing

Le cloud computing offre quatre modèles de déploiement différentier en fonction de l'utilisation de ses services soit par des entreprises ou autre type d'utilisateurs.

Cloud privé
L'infrastructure cloud est provisionnée pour une utilisation exclusive par une seule organisation, entièrement dédiée et accessible via des réseaux sécurisés, hébergé chez un tiers, mutualisé entre les différentes entités d'une seule et même entreprise, il peut exister dans ou hors des locaux[43].

Cloud public
L'infrastructure cloud est externe à l'organisation, accessible via Internet, et mise à disposition pour une utilisation ouverte pour le grand public. Il peut être détenu, géré et exploité par une entreprise, un établissement universitaire ou un organisme gouvernemental, ou une combinaison de ceux-ci [43].

Cloud communautaire
L'infrastructure cloud est provisionnée pour une utilisation exclusive par une communauté spécifique de consommateurs d'organisations partageant des préoccupations communes (mission, exigences de sécurité, politique et considérations de conformité). Il peut être détenu, géré et exploité par une ou plusieurs organisations de la communauté, par un tiers ou une combinaison de ceux-ci, et il peut exister dans ou hors des locaux.

Cloud hybride

L'infrastructure cloud est une composition de deux ou plusieurs infrastructures cloud distinctes (privées, communautaires ou publiques) qui restent des entités uniques, mais sont liées par une technologie standardisée ou propriétaire qui permet la portabilité des données et des applications[43].

1.2.5 Limites du cloud computing

Chaque nouvelle technologie arrivée porte des avantages et malheureusement suivie par quelques limites rencontrées par des utilisateurs du Cloud. Ces limites sont présentées comme suit :

- **La gestion d'énergie** : afin de définir un plan d'utilisation des ressources, le fournisseur doit définir une stratégie pour la gestion d'énergie (consommation d'électricité) [44].
- **Confidentialité et sécurité** : dans n'importe quelle technologie, la sécurité pose toujours des problèmes. Le problème concerne les attaques lors des opérations du transfert de données. Dans le cloud, ce problème est posé dans les cas des cloud publiques [44].
- **Dépendance** : au cas où l'entreprise (ou client final) souhaite des fonctionnalités très spécifiques, il est peut-être difficile de convaincre le fournisseur de proposer ces fonctionnalités. Le client final doit choisir un fournisseur en qu'il a la confiance.
- **Risques d'enfermement propriétaire** : De nombreux services cloud ont la particularité d'utiliser des technologies ou environnements non standards, rendant ainsi complexe voire impossible la migration vers un hébergeur ou éditeur concurrent[44].

1.3 Évaluation des performances du cloud computing

L'évaluation de performances consiste à calculer les paramètres de performances d'un système que l'on souhaite obtenir sous forme de grandeurs quantitatives qui peuvent se présenter sous différents ordres, en fonction des systèmes considérés. Elles se résument en trois étapes comme nous indique la figure 1.2 :

1. Comprendre le fonctionnement du système.
2. Élaborer un modèle plus fidèle aux caractéristiques et fonctionnement du système.
3. Évaluer les performances du système selon le formalisme du modèle.

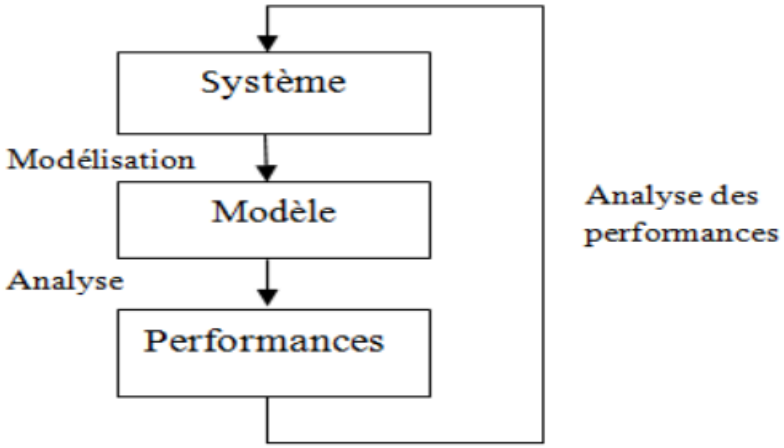

FIGURE 1.2 – Étapes d'évaluation des performances

1.3.1 Types d'analyse de performances

— Qualitative : Elle consiste à définir les propriétés structurelles et comportementales du système, telle que l'absence de blocage.
— Quantitative : Elle consiste à calculer les paramètres de performances du système. Elle n'a de sens que si une analyse qualitative a été préalablement menée. Il existe trois grands types de méthodes d'analyse quantitative des performances : la simulation, les méthodes analytiques et les mesures.

1.3.1.1 Simulation

La simulation est une technique qui consiste à construire un modèle d'une situation réelle, puis à faire des expériences sur ce modèle.

La démarche de simulation passe donc par trois étapes distinctes : l'étape de modélisation, qui consiste à construire le modèle du phénomène à étudier, l'étape d'expérimentation, qui consiste à soumettre ce modèle à un certain type de variations, et l'étape de validation, qui consiste à confronter les données expérimentales obtenues avec le modèle à la réalité [17]. Il y a plusieurs outils de simulation de systèmes distribués notamment le framework CloudSim, qui permet de la simulation de l'environnement du Cloud computing.

1.3.1.2 Méthodes analytiques

Dans les méthodes analytiques, l'évaluation de performance est alors réalisée, par la détermination des équations analytiques, qui est suivi d'une résolution numérique [41]. Ces

méthodes peuvent être en mesure d'analyser l'impact d'un grand espace de paramètres sur les performances des services, sont très efficaces, moins coûteuses et beaucoup plus rapides que la simulation.

Dans ce qui suit nous allons voir quelques méthodes analytiques pour l'évaluation des performances du cloud computing

— **Chaîne de Makov**
Une chaîne de Markov est une des techniques de base utilisée pour l'évaluation des performances des systèmes dans de nombreux domaines, L'avantage de la chaîne de Markov est la probabilité de condition qui repose sur l'état dynamique actuel de la chaîne [35, 53].

— **Les réseaux de petri**
Un réseau de petri est un graphique bipartite utilisé dans l'évaluation des performances dépend des types stochastiques : un réseau de petri stochastiques (SPN) et un réseau de petri stochastiques généralisés (GSPN). Il est évalué en régime permanent ou en transitoire par simulation ou par des techniques numériques. Les paramètres de performance des pétrins sont mesurés en termes de CTMC[35].

— **Arbre de défaillance**
Un arbre de défaillance est une méthode utilisée pour l'analyse de fiabilité contenant plusieurs nœuds, les nœuds racine sont certainement des événements de haut niveau et les feuilles sont des événements de base. Les événements de base sont connectés aux événements supérieurs avec des nœuds intermédiaires appelés portes. Gates représente généralement les opérations booléennes (AND, OR et NOT). L'événement supérieur définit l'état de l'ensemble du système[35].

Dans le cadre de notre travail nous allons utiliser les systèmes de files d'attente pour l'évaluation des performances de notre système.

1.3.2 Les systèmes d'attente

La théorie de files d'attente est une technique de la recherche opérationnelle qui permet de modéliser un système admettant un phénomène d'attente, d'évaluer ses performances et de déterminer ses caractéristiques[62].

Une file d'attente peut être décrite comme un système composé d'un certain nombre fini ou infini de places d'attente d'un ou plusieurs serveurs et d'un ensemble de clients arrivant à des instants aléatoires. Quand les serveurs sont tous occupés, les clients doivent alors patienter dans un espace d'attente jusqu'à ce qu'un serveur soit disponible[50].

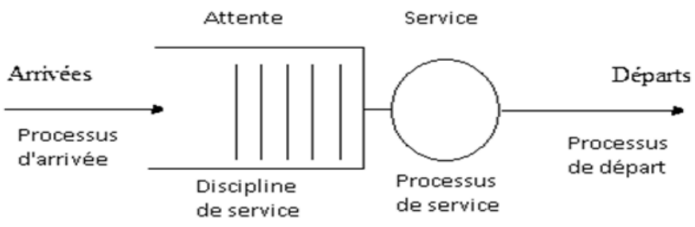

FIGURE 1.3 – Représentation d'une file d'attente

La notation de base d'un système d'attente est décrite en utilisant la notation de Kindal $A/B/c/k/P/D$, elle permet de ramener la description textuelle des différents éléments constituant un modèle d'attente simple à une formule symbolique :

A distribution d'inter-arrivé.
B distribution des durées de service.
c nombre de serveurs.
K Capacité du système.
P taille de la population.
D Discipline de service.

Lorsque les trois derniers éléments de la notation de Kendall ne sont pas précisés, il est sous-entendu que $K = +\infty$, $P = +\infty$, et $D = FCFS$(premier arrivé, premier servi) [50].

Fondamentalement, le modèle d'attente est caractérisé par

$L = E(X) =$ nombre moyen de client dans le système
$L_q =$ nombre moyen de clients dans la file d'attente
$W =$ temps de séjours moyen d'un client dans le système
$W_q =$ temps d'attente moyen d'un client
$W_q^* =$ temps d'attente moyen d'un client qui est obligé d'attendre

Ces valeurs permettent de juger le comportement opérationnel d'un système d'attente[50].

1.3.3 Quelques systèmes d'attente pour la modélisation et l'évaluation des performances du Cloud Computing

Dans [21] les auteurs ont proposé un modèle d'attente $M/M/C$ afin de réduire le temps d'attente, la longueur de la file d'attente, et améliorer la performance du réseau et la qualité

de service dans un environnement de cloud computing, en augmentant le nombre de serveurs C=1 , C=2, puis C=3 dans le cloud computing. Ils constatent à partir des résultats que lorsqu'il y a plus de serveurs, la file est réduite, le temps d'attente diminue, par conséquent, la qualité de service peut être obtenue efficacement.

Les auteurs de [55] ont proposé une architecture cloud qui consiste à connecter une file d'attente $M/M/1$ et une autre file d'attente de service Erlang $M/E_K/1$ dans le but d'améliorer le temps d'attente des utilisateurs et la gestion efficace des ressources. La simulation avec l'outil SHARP a montré que le système proposé permet une réduction du temps d'attente de 40% et une amélioration de la disponibilité des ressources.

Dans [54] les auteurs ont présenté un modèle de mise en file d'attente $M/M^Y/1$ pour concevoir une infrastructure pour différents types de clients demandant un service basé sur le cloud public, dans le but de faire une analyse des mesures de performances et améliorer la qualité de service telle que le temps d'attente, les résultats de la simulation avec l'outil SHARP ont montré une amélioration des performances du système d'attente dans le cloud ainsi qu'une variation considérable entre le temps d'attente total et le nombre total d'utilisateurs dans le système et cela pour différents taux d'arrivée et pour différentes tailles de lots.

Les auteurs de [52] ont modélisé le cloud computing par un modèle de file d'attente $M/G/m/m+r$ pour évaluer ses mesures de performances. Après simulation ils l'ont comparé avec une file d'attente simple, les résultats montre une amélioration des performances, et une diminution de la taille de la fille et du temps d'attente.

1.4 Conclusion

Au cours de ce chapitre, nous avons fourni quelques notions sur le Cloud Computing, en présentant ses types, ses services (IaaS, PaaS, SaaS), ses caractéristiques ainsi que ses inconvénients. Enfin, nous avons présenté les différentes méthodes d'évaluation de ces performance, nous avons mis l'accent sur la méthode d'évaluation avec les systèmes d'attente en présentant quelques travaux déjà faits dans la littérature.

On a pu constater que le cloud computing est un système complexe, de ce fait dans le chapitre suivant nous allons introduire la technologie des agents et des systèmes multi-agents qui est adaptés pour le traitement des systèmes complexes dans le but d'améliorer le domaine du Cloud Computing. Ces notions vont nous servir de base pour notre travail.

Chapitre 2

Agents et système multi-agents

2.1 Introduction

Les caractéristiques et les attentes des applications informatiques ont considérablement changé ces dernières années, soulevant ainsi un nombre important de défis à relever. Les concepteurs d'applications doivent maintenant faire face à la décentralisation, à la distribution et au besoin de fédérer des systèmes hétérogènes. De plus, ils doivent être en mesure de fournir des solutions robustes et capables de s'adapter dans des environnements qui peuvent être aussi imprévisibles et versatiles que l'Internet. Face à ces nouveaux enjeux, les techniques classiques ne parviennent qu'à proposer des réponses limitées.

C'est dans ce contexte que les systèmes multi-agents (SMA) ont été proposés comme nouveau paradigme de conception. Ils se situent à l'intersection de plusieurs domaines scientifiques notamment l'intelligence artificielle distribuée (IAD) et la vie artificielle. Ils amènent une décentralisation des données (connaissance) et du contrôle (comportement) : la résolution des problèmes ou la simulation des systèmes complexes ne résulte pas d'un calcul effectué par un programme monolithique, mais des interactions entre plusieurs entités autonomes[58].

2.2 Théories orientés agents

Pour présenter de manière structurée les théories orientées agents, nous adoptons dans cette section, une décomposition basée sur trois vues principales : vue agent, vue environnement et la vue interaction.

2.2.1 Vue agent

La notion d'agent, comme tous les concepts fondamentaux, est relativement vague. On peut distinguer plusieurs définitions parmi lesquelles nous citons :

La définition de [63] qui défini un agent comme un système informatique , situé dans

un environnement, et qui agit d'une façon autonome pour atteindre les objectifs (buts) pour lesquels il a été conçu.

Selon [51] un agent est toute entité percevant son environnement grâce à des capteurs et agissant sur celui-ci via des effecteurs.

La définition la plus générale et la plus fréquemment utilisée est celle de [22] qui définit l'agent comme une entité autonome physique ou abstraite qui est capable d'agir sur elle-même et sur son environnement, et qui dans un univers multi-agents peut communiquer avec d'autres agents et dont le comportement est la conséquence de ses observations, de ses connaissances, et des interactions avec les autres agents.

2.2.1.1 Modèles d'agents

Les agents sont principalement divisés en quatre modèles qui sont :

Modèle d'agents réactifs

Les agents réactifs sont des agents simples répondant d'une façon opportune aux modifications de leurs environnements qui résultent des stimuli externes. Leurs actions sont programmés et non pas choisies, ils agissent sans nécessité de compréhension de leurs univers ni de leurs buts et ne disposent pas de capacités d'apprentissage.

Les sociétés d'agents réactifs sont caractérisées par le nombre important des agents qui sont capables ensemble de produire des actions évoluées et d'exhiber un comportement global intelligent. Or, les individus pris séparément ne sont souvent pas considérés comme intelligents. ils ne possèdent qu'une représentation faible de leur environnement et des buts globaux[18]

Modèle d'agents cognitifs Les agents cognitifs quant à eux possèdent une représentation explicite de leur environnement, des autres agents et d'eux-mêmes. Ils sont parfois dits "intentionnels", leur caractéristique fondamentale est la volonté de communiquer, de coopérer et de mémoriser le passé. Ils possèdent des buts à atteindre à l'aide d'un plan explicite.

Les sociétés d'agents cognitifs contiennent communément un petit groupe d'individus qui sont des agents intelligents où chacun possédera une base de connaissances comprenant l'ensemble des informations et des savoir-faire nécessaires à la réalisation de sa tâche et à la gestion des interactions avec son environnement et les autres agents[7].

Un agent cognitif raisonne sur ses croyances qui représentent sa compréhension du monde dans lequel il évolue ainsi que sur ses désirs et intentions pour accomplir ses buts [31].

Modèle d'agents hybrides

Les agents hybrides sont des agents ayant à la fois des capacités cognitives et réactives. Ils conjuguent la rapidité de réponse des agents réactifs ainsi que les capacités de raisonnement des agents cognitifs. Ce modèle a été proposé par plusieurs auteurs afin

de palier aux problèmes liés au temps de décision et au temps d'action relativement élevé pour les agents réactifs[58].

Modèle d'agents mobile Ces agents sont souvent qualifiés d'autonomes. Ils ont la capacité de se transporter entre les noeuds d'un réseau ou de plusieurs réseaux avec une totale liberté aménageant ainsi leur parcours initial et décidant des tâches à effectuer. Même en dehors de leur environnement, ils sont capables de communiquer avec d'autres agents et de se déplacer d'un hôte à l'autre.
Ils peuvent ainsi exécuter des tâches malgré l'arrêt de leur machine initiale[20].

2.2.2 Vue environnement

L'environnement désigne le monde réel ou bien l'univers virtuel dans lequel les agents évoluent. Il fournit un support commun aux actions des agents, permettant ainsi l'interaction dans le système, et il constitue une source d'information à laquelle les agents peuvent accéder au travers de leur perception[58].

La description d'un environnement dépend fortement de la nature d'un problème, il peut être accessible ou inaccessible, déterministe ou non déterministe, statique ou dynamique, discret ou continu [38].

2.2.3 Vue interaction

L'interaction est l'un des éléments les plus importants des systèmes multi-agents, elle consiste à mettre en relation dynamique deux ou plusieurs agents par le biais d'un ensemble d'actions régies par un protocole interaction. Dans un système multi-agents, les différents agents doivent être en mesure d'interagir et de se comprendre mutuellement afin de pouvoir se coordonner et éventuellement coopérer.

L'auteur de [22] définit l'interaction comme une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble de relations réciproques. Les interactions sont non seulement la conséquence d'actions effectuées par plusieurs agents en même temps, mais aussi l'élément nécessaire à la constitution d'organisations sociales.

En général, les interactions sont mises en œuvre par un transfert d'informations entre agents ou entre l'environnement et les agents, soit par perception, soit par communication. L'interaction entre les agents apparaît sous plusieurs modes, qui sont la coopération, la coordination et la négociation[20].

— **Communication :**
Les agents sont souvent confrontés à des situations où ils ont besoin d'interagir avec d'autres agents pour atteindre leurs buts locaux ou globaux.
La communication permet de mettre en œuvre cette interaction, elle est définie comme

une forme d'action locale d'un agent vers d'autres agents, elle peut être sélective sur un nombre restreint d'agents ou diffusée à l'ensemble des agents, elle nécessite un langage de communication compréhensible et commun à tous les agents[20].

— **Coopération :** Chaque agent possède un ensemble de compétences qui lui permettent de résoudre les différents problèmes, mais il existe des situations où ses capacités et ses compétences ne suffisent pas à accomplir certaines tâches. Par conséquent, il a besoin de l'aide des autres agents du système, cela implique qu'il y a une coopération pour faire évoluer le système vers ses objectifs.
Donc la coopération consiste à faire participer plusieurs agents pour satisfaire un but individuel ou commun, l'ajout ou la suppression d'un agent influe considérablement sur la performance du groupe[20].

— **Coordination :** La coordination entre les agents est présente lorsque les agents utilisent des ressources communes ou résolvent des problèmes qui ne sont pas complètement indépendants mais liés et complémentaires. Les agents du système doivent accomplir en plus de leurs tâches de résolution des problèmes individuels, des tâches supplémentaires (appelées tâches de coordination) qui améliorent le fonctionnement du système. À ce stade, coordonner les activités des agents devient un aspect essentiel afin de synchroniser leurs activités, de régler les conflits, et d'éviter des problèmes dans le comportement global du système[20, 7].

— **Négociation :** La négociation correspond à la collaboration entre agents en univers compétitif. Les buts des agents dans un système multi-agents peuvent être incompatibles et leurs demandes sont parfois contradictoires, il peut y avoir des conflits d'intérêts ou de buts, comme par exemple l'accès à des ressources rares ou la proposition de plusieurs solutions différentes à un seul problème. De telles situations conflictuelles si elles ne sont pas résolues peuvent empêcher l'avancement désiré du système.
Afin de résoudre ces conflits et trouver un compromis attractif pour tous les agents participants, ces derniers négocient entre eux en faisant des concessions ou en cherchant des alternatives [20, 7].

2.3 Systèmes multi-agents (SMA)

Un Système Multi-Agents (SMA) est un paradigme de calcul et de gestion distribué sous forme d'un ensemble d'agents qui interagissent dans un environnement pour résoudre un problème commun en utilisant les ressources et les connaissances de chaque agent [48].

Donc selon [22] on peut définir un système multi-agents comme un système composé des éléments suivants :

Environnement E : Un espace disposant généralement d'une métrique.

Ensemble d'objets O : Ces objets sont situés, pour tout objet, il est possible à un moment

donné, d'associer une position dans E ces objets sont passifs, ils peuvent être perçus, crées, détruits et modifiés par les agents.

Ensemble d'agents A : Représentent les entités actives du système.

Ensemble de relations R : Unissent les objets et les agents entre eux.

Ensemble d'opérations Op : Permettant aux agents A de percevoir, produire, consommer, transformer et manipuler les objets de O.

Opérateurs : Chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

La Figure 2.1 représente de manière imagée un système multi-agents.

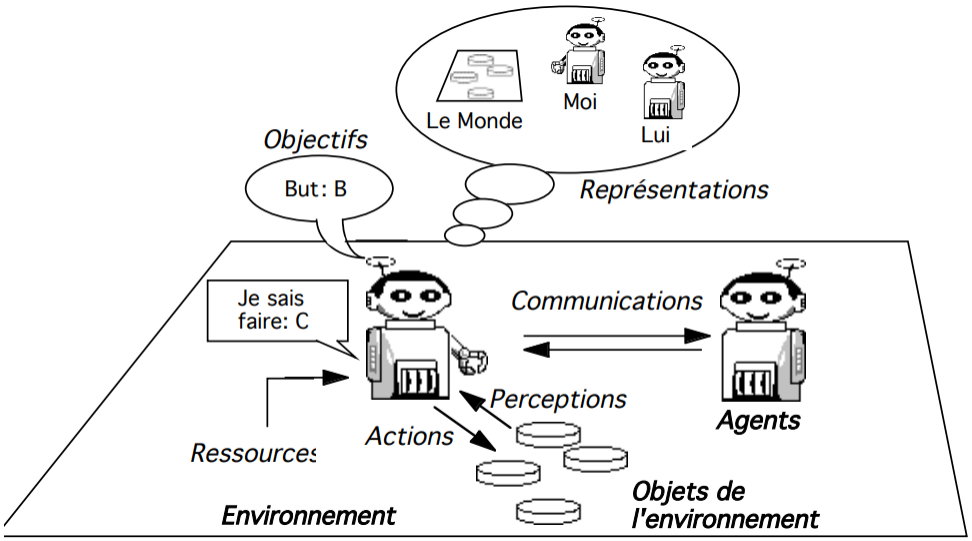

FIGURE 2.1 – Représentation imagée d'un système multi-agents [22].

2.3.1 Caractéristiques des systèmes multi-agents

Les systèmes multi-agents (SMA) sont des systèmes idéaux pour représenter des problèmes possédant de multiples méthodes de résolution et de multiples perspectives. Ces systèmes possèdent les avantages traditionnels de la résolution distribuée et concurrente de problèmes comme la modularité, la vitesse, et la fiabilité.

Ils héritent aussi des bénéfices envisageables de l'Intelligence Artificielle comme, la facilité de maintenance, la réutilisation et la portabilité, mais surtout, ils ont l'avantage de faire intervenir des schémas d'interaction sophistiquée[14]. Ils sont généralement caractérisés par[57] :

Chaque agent a des informations ou des capacités de résolution limitées de problème, ainsi chaque agent a un point de vue partiel.

Il n'y a aucun contrôle global du système multi-agents.
Les données sont décentralisées.
Le calcul est asynchrone.

2.3.2 Domaines d'application des systèmes multi-agents

La jeunesse des systèmes multi-agents n'a pas empêché sa présence dans de nombreux domaines, grâce aux avantages qu'ils offrent et les caractéristiques dont ils disposent. Citons comme exemples : les systèmes financiers et commerciaux comme le commerce et les banques électroniques, les systèmes embarqués, la gestion des bases de données, la gestion des réseaux de communication, la recherche d'information et le contrôle industriel.

Les systèmes Multi-agents ont attiré l'attention à cause de leur grande souplesse d'utilisation puisqu'il n'y a aucune limite fixée sur les comportements des agents ni sur leur manière de s'exprimer, mais aussi à cause de leur pouvoir de représentation d'un univers intuitif où chaque composante d'un système peut être modélisée par un agent. Il existe cependant quelques grandes catégories d'applications des SMA : [42].

— **La résolution de problèmes** Elle concerne toutes les situations dans lesquelles des agents logiciels, purement informatiques, accomplissent des tâches utiles aux êtres humains [42].
— **La robotique distribuée** La robotique distribuée porte sur la réalisation d'un ensemble de robots qui coopèrent pour accomplir une mission et utilise des agents concrets qui se déplacent dans un environnement réel [42].
— **La simulation multi-agents** La simulation est la démarche scientifique qui consiste à réaliser une reproduction artificielle, appelée modèle d'un phénomène réel que l'on désire étudier, à observer le comportement de cette reproduction lorsqu'on en fait varier certains paramètres et à induire ce qui se passerait dans la réalité sous l'influence de variations analogues. La simulation multi-agents offre un outil puissant pour expliquer les changements au sein d'un système naturel ou artificiel en fonction des changements des individus qui le composent[42].

2.3.3 Plateformes d'implémentation des Systèmes multi-agents

Il existe une multitude de plateformes multi-agents dédiées à différents modèles d'agent. Les plateformes fournissent une couche d'abstraction permettant d'implémenter facilement les concepts des systèmes multi-agents. D'un autre côté, elle permet aussi le déploiement de ces systèmes. Ainsi, elles constituent un réceptacle au sein duquel les agents peuvent s'exécuter et évoluer. En effet, les plateformes sont un environnement permettant de gérer le cycle de vie des agents[14].

MACE En 1987, est apparu le premier environnement de conception de distinctes architectures d'agents utilisables pour une variété de domaines. Pour cet environnement, un agent est représenté par un objet actif communiquant par envoi de messages. Chaque agent possède un moteur représentant sa partie active qui détermine les actions à exécuter et la façon dont les messages doivent être interprétés. L'environnement assure une bonne communication inter-agents, il gère les allocations agents-processus (en étiquetant chaque agent à son processus), en plus d'un ensemble d'agents système[25].

MadKit (Multi-agent development Kit Est une plateforme mise en œuvre en 1998 qui a pour particularité d'être un modèle organisationnel plutôt qu'une architecture d'agents bien précise. Développée avec java, elle dispose d'un outil de modélisation et de conception pour les développeurs de systèmes multi-agents. Elle offre à l'utilisateur un environnement de développement graphique permettant une élaboration plus commode et plus conviviale des applications[28].

JADE (Java Agent DEvelopmen) Est une plateforme multi-agents développée en Java par CSELT (groupe de recherche de Gruppo Telecom, Italie) permettant la construction de systèmes multi-agents. Cette plateforme veille à réaliser des applications conformes à la norme FIPA(Foundation for Intelligent Physical Agents). JADE comprend principalement deux composantes : une plateforme agents compatible FIPA et un paquet logiciel pour le développement des agents Java[8].

Du fait que nous avons choisi cette plateforme pour l'implémentation de notre modèle, une description plus approfondie sera présentée dans le quatrième chapitre.

2.4 Le Cloud computing et les systèmes multi-agents

Bien que l'on puisse dans un premier temps considérer ces deux systèmes distribués (Cloud computing et systèmes multi-agents) comme incompatibles, une analyse détaillée démontre qu'ils sont en fait non seulement complémentaires, mais partagent une synergie considérable entre eux[9].

La théorie des agents et des systèmes multi-agents (SMA) peut fournir un nouveau modèle de gestion des systèmes cloud computing basé sur la répartition des responsabilités, la flexibilité et l'autonomie. La gestion des fonctions du noyau d'un système cloud computing via un modèle basé sur des agents permet aux plateformes qui en résultent d'être beaucoup plus efficaces, évolutives et adaptables qu'elles ne le sont actuellement. Par conséquent l'utilisation du système multi-agents dans le cadre de la conception de systèmes cloud computing apporte à ce paradigme de nouvelles caractéristiques telles que l'apprentissage ou l'intelligence, ce qui permet de développer des environnements de calcul beaucoup plus avancés dans tous les aspects (services intelligents, interopérabilité entre plateformes, efficacité de mutualisation des ressources, etc.)[37].

En effet au cours des dernières années, de plus en plus de chercheurs ont participé à la recherche sur la technologie des agents et le cloud computing. Des efforts ont été faits principalement pour réduire la frontière entre les deux technologies. La combinaison des agents et du cloud computing repose sur les défis auxquels sont confrontées les deux communautés et sur la nécessité de développer des systèmes plus intelligents.
La technologie des agents dont le but est de traiter des systèmes complexes a révélé des opportunités pour améliorer le domaine du Cloud Computing[37]. Par exemple dans l'article [6] l'auteur a proposé un système multi-agents pour aider à déterminer les meilleures ressources et créer une méthode de négociation entre les fournisseurs et les utilisateurs du cloud pour exploiter tout le potentiel du cloud computing, il a amélioré le rôle de l'agent courtier de tel sorte qu'il puisse aider à déterminer les meilleures ressources et créer une méthode de négociation entre les fournisseurs et les utilisateurs et de surveiller les travaux de l'utilisateur pendant leur traitement. Le système a été implémenté et évalué sur la plateforme JADE, la simulation montre que le système proposé réduit le temps de traitement des taches en détectant le plus tôt des failles durant le traitement et en trouvant des alternatives.

Les SMA ont également été utilisés dans le cloud pour fournir un service de gestion d'ordonnancement des taches. En effet dans le papier [29] les auteurs ont proposé un système multi-agents pour résoudre le problème d'ordonnancement des taches dans le cloud, basé sur une architecture d'agent BDI(Belief-Desire-Intention) qui permet de raisonner, de collaborer et de prendre des décisions d'adaptation dans le but d'optimiser l'équilibre de charge, le délai d'exécution, le temps d'attente et de réponse et la qualité de service.
Les résultats de la simulation montre que l'emploi d'un seul agent BDI améliore l'exécution des taches, offrent un meilleur coût d'exécution, de délai et améliore la qualité de service et celle-ci devient plus garantie avec l'emploi de plusieurs agents BDI.

Ainsi dans [3] les auteurs ont proposé une combinaison de la technologie Agent avec le Cloud en présentant une approche distributive pour faciliter la découverte intelligente de services cloud et prendre en charge la complexité de l'environnement.
Cependant, les résultats de la simulation montrent que l'extension de l'approche multi-agents en intégrant la confiance dans le processus de découverte de services cloud améliore la qualité de service en terme de temps de réponse et d'évolutivité.

Dans l'article [4] les auteurs ont contribué à intégrer les systèmes multi-agents dans le cloud computing, en proposant le système DRPM (Dynamic Resources Provisioning and Monitoring), permettant de gérer les ressources du fournisseur de cloud tout en tenant compte des exigences de qualité de service (QoS) des clients telles que déterminées par l'accord de niveau de service (SLA).
Le système DRPM proposé a été évalué à l'aide de l'outil CloudSim. Les résultats montrent que le système DRPM permet au fournisseur de cloud d'augmenter l'utilisation des ressources et de réduire la consommation d'énergie tout en évitant les violations des accords SLA.

2.5 Conclusion

Nous avons présenté dans ce chapitre une vision générale sur les agents et les systèmes multi-agents. Ces systèmes qui ont porté résolution aux problèmes de l'IA classique sont organisés dans des sociétés d'agents qui interagissent, communiquent et coopèrent entre eux pour accomplir une tâche bien déterminée. Nous avons vu que les systèmes multi-agents sont considérés comme un concept bien adapté pour développer des applications cloud computing, par la suite nous avons présenté une synthèse concernant quelques travaux réalisés pour la gestion du cloud computing en se basant sur les systèmes multi-agents.

Ainsi, dans le prochain chapitre, nous allons présenter notre contribution afin d'améliorer la gestion du cloud computing à travers la proposition de notre architecture à base d'agents.

Chapitre 3

Architecture du modèle proposé

3.1 Introduction

Nous avons présenté dans les chapitres précédents les différents concepts du cloud computing et des systèmes multi-agents ainsi qu'un aperçu de quelques travaux sur le cloud computing et la technologie des agents[20].

Dans ce chapitre, nous allons décrire le problème traité, puis montrer le modèle proposé pour la résolution de ce problème et l'utilité de cette solution.

Par la suite, nous allons proposer une architecture globale du modèle en montrant les différentes couches de notre système pour simplifier le déroulement de l'approche proposée. Puis, nous allons expliquer l'architecture détaillée de chaque composant (Agents) leurs rôles et leurs fonctionnement, enfin nous terminerons ce chapitre par une conclusion.

3.2 Position du problème

En raison de la nature dynamique et virtualisée de l'environnement Cloud Computing, de la diversité des demandes des utilisateurs, satisfaire les exigences des utilisateurs n'est pas une tâche simple à réaliser. En effet le nombre accru d'utilisateurs cloud et le flux intense des demandes provoquent un engorgement dans le cloud, de ce fait quelques clients sont amenés à quitter le système avant d'être servis par défaut de disponibilité de ressources et d'un long temps d'attente[12].

L'allocation efficace des ressources dans le cloud doit prendre en compte plusieurs contraintes à savoir, satisfaire les exigences des utilisateurs et assurer la qualitée de service (QoS) [19].

Le cloud computing fait face à divers problèmes que nous avant décrit précédemment, or avec l'avancée technologique et les techniques récentes de l'IAD (Intelligence Artificielle Distribuée), des solutions de plus en plus efficaces pour la gestion du cloud computing ont émergé.

C'est dans cette optique que nous proposons un modèle multi-agents pour la gestion efficace et l'allocation des ressources dans le cloud computing.

Notre solution repose principalement sur un protocole de négociation entre les différents agents du système dont le but principal est de trouver le meilleur compromis quant à l'allocation des ressources aux différents utilisateurs cloud, et gérer l'impatience de ces derniers.

Il faut noter qu'avec une telle solution nous assurons une amélioration de qualité de service, une meilleure gestion des clients et une satisfaction des exigences de l'utilisateur cloud en terme de temps et d'allocation de ressources dans le cloud computing.

Dans ce qui suit, nous allons détailler les principaux aspects de notre solution.

3.3 Description du modèle

Notre modèle est composé de plusieurs serveurs, d'une file d'attente ordinaire de capacité limitée, d'un ensemble de clients visant à demander des services en exploitant les serveurs cloud libres. En plus il y a une place d'attente VIP (Very Important Person) pour les clients désirant passer en priorité, comme le montre la Figure 3.1. On suppose que l'arrivée des clients n'est pas simultanée et que la file d'attente suit la discipline de service FCFS (First Come, First Served).

À l'arrivée d'un client, s'il y a un serveur libre alors le client commence immédiatement son service.

Sinon, les serveurs étant tous occupés, si la file d'attente est vide alors le client peut soit rejoindre la file d'attente ou quitter le système s'il est impatient.

En revanche si la file d'attente n'est pas vide et que le client est impatient, il peut soit négocier la place d'attente VIP si elle est libre pour être traité en priorité et passer en VIP ou quitter le système. S'il n'est pas impatient, il peut rejoindre la file d'attente ordinaire, si elle n'est pas encore saturée, et attendre la disponibilité d'un serveur. Si par contre la file d'attente est pleine alors le client est automatiquement refusé.

Les clients dans la file d'attente réussissent à accéder aux serveurs uniquement lorsqu'ils sont disponibles, selon la discipline de service FCFS. En cas d'impatience, ils peuvent quitter définitivement le système et revenir en tant que nouveaux clients pour bénéficier d'une offre VIP. Il est à noter que lorsqu'un serveur devient libre, le client dans la place d'attente VIP et les clients dans la file d'attente ordinaire peuvent entrer en concurrence pour accéder au serveur. Dans ce cas, le client dans la place d'attente VIP a la priorité sur tous les clients dans la file d'attente ordinaire.

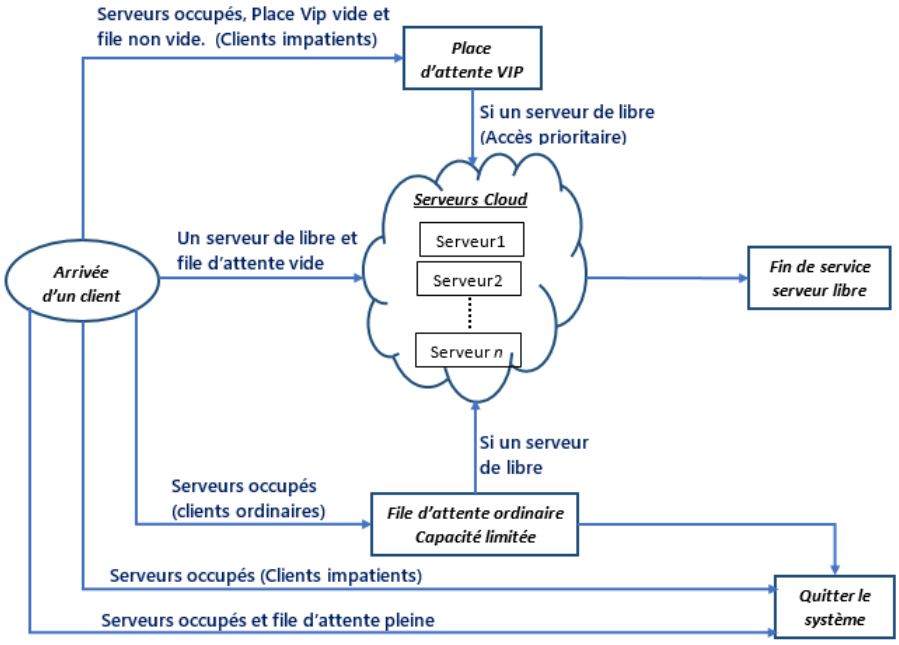

FIGURE 3.1 – Description du système modélisé.

3.4 Architecture du modèle agent

Dans cette section, nous allons présenter l'architecture générale de notre système multi-agents à savoir les différentes couches qui le composent ainsi que l'architecture détaillée de chaque composant, leurs rôles, leurs fonctionnements et les différentes interactions entre eux.

3.4.1 Architecture générale

Nous proposons un système multi-agents pour la gestion du cloud computing composé de trois principaux agents, agent utilisateur cloud, agent fournisseur cloud et agent de courtage chargé de mener des activités de négociation entre les deux premiers.

Dans ce qui suit, nous définissons brièvement les rôles de ces trois agents :

— Agent Utilisateur Cloud : Il représente l'utilisateur dans l'environnement cloud, il participe dans le système et vise à satisfaire au mieux les demandes de son client.

— Agent Fournisseur Cloud : Il représente le fournisseur de services dans l'environnement cloud, il participe dans le système et vise à obtenir un bénéfice maximal pour le fournisseur.

— Agent Courtier : Le rôle principal de l'agent courtier est la gestion des ressources (serveurs cloud) et leur allocations aux utilisateurs. Il permet d'initier des négociations de services VIP pour réduire la perte des clients impatients.

L'architecture générale de notre système est de nature hiérarchique qui s'articule autour de trois principales couches en interaction :

— Couche d'interaction utilisateur-cloud : c'est la couche intermédiaire entre le système cloud computing et l'utilisateur cloud humain.
— Couche de courtage : c'est la couche intermédiaire entre la première couche (Couche d'interaction utilisateur-cloud) et les fournisseurs cloud.
— Couche d'allocation de ressources : Cette couche contient l'ensemble des agents fournisseurs cloud.

La figure 3.2 illustre l'architecture générale de notre système multi-agents pour l'allocation des ressources dans le cloud computing.

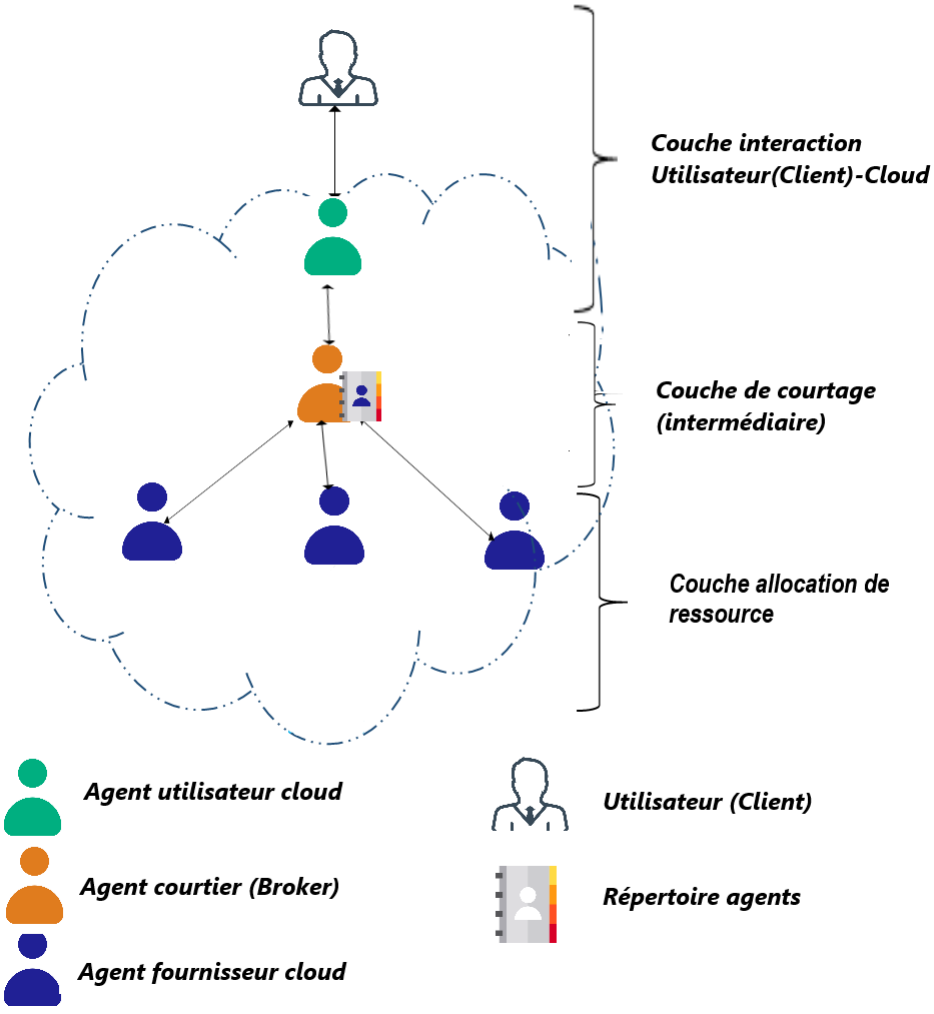

FIGURE 3.2 – Architecture multi-agents pour l'allocation des ressources dans le cloud computing

a. Couche d'interaction utilisateur-cloud

C'est la couche intermédiaire entre le client et le système cloud, elle permet aux utilisateurs cloud humains d'interroger le système. Elle offre une interface graphique qui

permet aux utilisateurs cloud humains d'interagir avec le système en introduisant leur demande de service souhaité, et permet d'afficher les résultats du traitement de leur demande.

Afin que le client puisse faire sa demande, il se doit d'interagir avec l'interface. L'agent utilisateur cloud (virtuel) est chargé de transmettre la demande de l'utilisateur humain à l'agent courtier afin de traiter sa demande. L'agent utilisateur cloud (virtuel), permet de trouver les ressources appropriées et de gérer la négociation entre l'utilisateur cloud humain et l'agent courtier.

La figure 3.3 illustre le schéma descriptif de la couche d'interaction utilisateur-cloud.

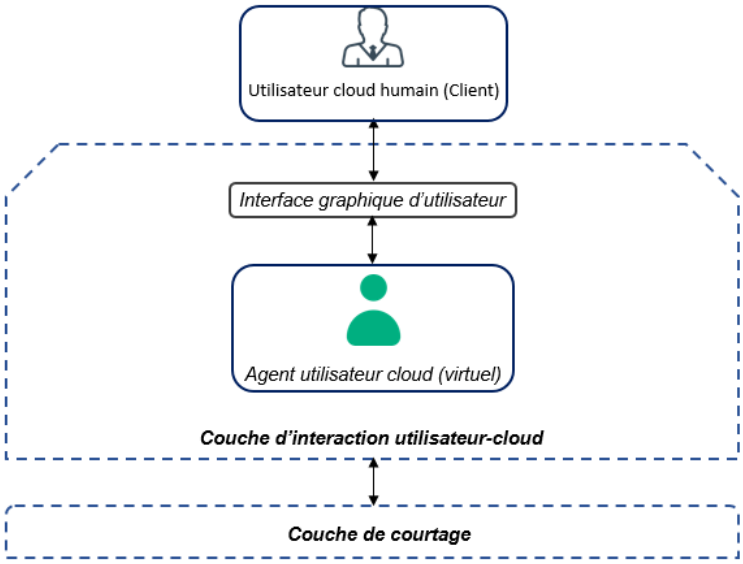

FIGURE 3.3 – Schéma descriptif de la couche interaction utilisateur-cloud

b. Couche de Courtage

C'est la couche intermédiaire reliant la couche d'interaction utilisateur-cloud et les fournisseurs de services cloud. Elle se compose d'un agent courtier détenant un répertoire des agents fournisseurs cloud. Il a pour objectif d'allouer les serveurs aux utilisateurs cloud. L'agent courtier mène des activités de négociation pour parvenir à un arrangement entre les fournisseurs cloud et les utilisateurs cloud, afin de répondre au problème d'impatience de certains utilisateurs.

La figure 3.4 illustre le schéma descriptif de la couche de courtage.

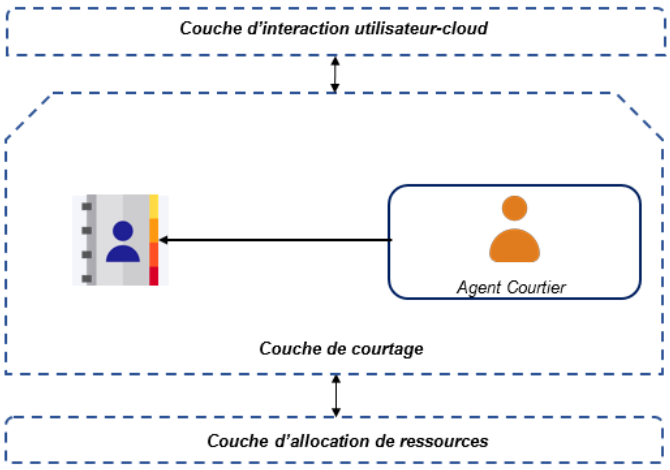

FIGURE 3.4 – Schéma descriptif de la couche de courtage

c. Couche allocation de ressource

Dans cette couche nous retrouvons l'ensemble des agents fournisseurs cloud. Ils s'occupent d'accomplir le service demandé et d'informer l'agent courtier de l'état de son serveur (libre ou occupé)..

La figure 3.5 illustre le schéma descriptif de la couche d'allocation de ressource.

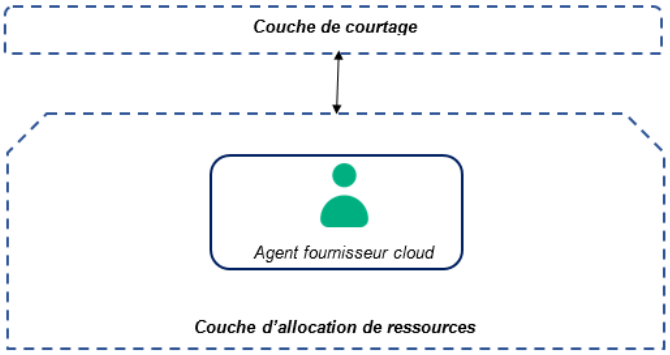

FIGURE 3.5 – Schéma descriptif de la couche d'allocation de ressources

3.4.2 Architecture détaillée

Dans cette partie, nous allons décrire chaque type d'agents composant notre système, à savoir l'agent utilisateur cloud, l'agent fournisseur cloud et l'agent courtier, tout en déterminant le rôle et le comportement de chacun dans notre système.

a. Agent Utilisateur cloud

Ces agents virtuels représentent les différents utilisateurs humains dans l'environnement cloud. Chaque agent virtuel représente un seul utilisateur humain à la fois.

L'agent utilisateur cloud (virtuel) fournit une interface graphique, qui permet aux utilisateurs humains d'interagir avec le système (saisir leurs besoins, suivre leurs demandes et visualiser les résultats). Il a pour mission de recueillir les requêtes de l'utilisateur, envoyer ces requêtes à l'agent courtier et d'afficher les résultats aux utilisateurs humains.

Chaque utilisateur cloud humain a la possibilité de demander un service auprès du cloud computing et de suivre l'état de sa demande. Ces informations vont être collectées et transmises par l'agent utilisateur cloud à l'agent courtier. Après l'envoi de la demande de service, l'agent utilisateur cloud se met en attente d'une réponse, si un serveur est libre alors il lui est alloué immédiatement. Sinon si aucun serveur n'est disponible l'agent courtier a la possibilité de lui envoyer une proposition de négociation pour l'allocation d'un serveur VIP si aucun client n'est en attente VIP, afin de passer en priorité et traiter son service dès qu'un serveur se libère, il a le choix d'accepter ou de refuser cette proposition et attendre dans une file d'attente ordinaire.

Dans notre système un client peut suivre et demander l'état de sa demande ou l'annuler et quitter le système à tout moment.

b. Agent Fournisseur cloud

Chaque fournisseur de services cloud est représenté par un agent fournisseur cloud. Il est chargé d'accomplir le service demandé. L'agent fournisseur est chargé aussi de transmettre les informations relatives à l'état du serveur à l'agent courtier.

c. Agent Courtier (Broker)

L'agent courtier représente la pièce maîtresse de notre système, il assure le rôle d'intermédiaire ou de médiateur entre les fournisseurs de services Cloud et les utilisateurs cloud.

Il est chargé d'allouer des serveurs aux agents utilisateurs cloud virtuels pour répondre aux requêtes des utilisateurs cloud humains. Il reçoit la demande de l'agent utilisateur cloud, et les informations relatives à l'état des serveurs par les agents fournisseurs. Il dresse un répertoire contenant les informations relatives à l'état des serveurs et leurs identifiants envoyées par les agents fournisseurs cloud.

Il a la possibilité de proposer aux agents utilisateurs cloud une négociation pour l'allocation d'un serveur VIP afin d'être traité en priorité si aucun utilisateur est en attente d'un serveur VIP. Il gère la file d'attente et il est chargé de mettre à jour l'état des différents serveurs.

3.5 Déroulement explicatif du fonctionnement de notre système

Dans ce qui suit nous allons présenter un scénario résumant au mieux le fonctionnement de notre système. Dans ce contexte nous utiliserons une file d'attente à priorité relative (sans préemption).

Scénario 1 : On suppose qu'un serveur est disponible et qu'aucun utilisateur n'est en attente dans la file d'attente, la figure 3.6 illustre les interactions entre les agents dans ce cas-là.

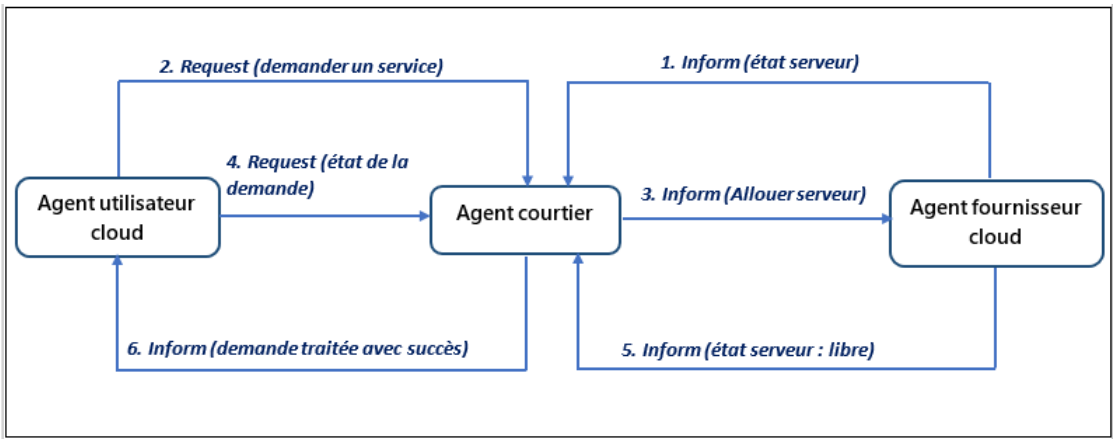

FIGURE 3.6 – Schéma explicatif du déroulement des échanges entre les agents de notre système dans le scénario 1

1 Initialement, chaque agent fournisseur informe l'agent courtier de l'état de son serveur (serveur disponible).

2 L'agent utilisateur cloud envoie une requête à l'agent courtier spécifiant le service souhaité.

3 L'agent courtier vérifie l'état des serveurs dans son répertoire et alloue un serveur libre à l'agent utilisateur, met à jour l'état du serveur dans son répertoire.

4 L'agent utilisateur cloud demande un suivi de l'état de sa demande à l'agent courtier.

5 Une fois le traitement de la demande achevé l'agent courtier est informé par l'agent fournisseur que le serveur est libre. L'agent courtier met à jour son répertoire.

6 L'agent courtier informe l'agent utilisateur cloud de l'état de sa demande. L'agent utilisateur cloud (virtuel) affiche à son tour ces informations sur l'interface graphique utilisateur.

Scénario 2 : Dans ce cas on suppose qu'aucun serveur n'est disponible, et qu'aucun utilisateur n'est en attente d'un serveur dans la file d'attente.

La figure 3.7 illustre les différents messages échangés entre les agents de notre système dans ce cas-là.

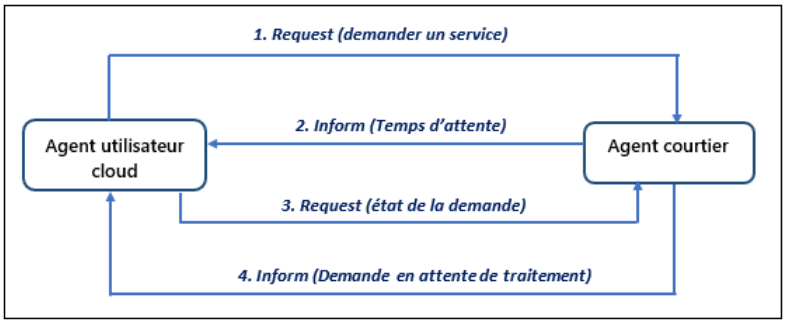

FIGURE 3.7 – Schéma explicatif du déroulement des échanges entre les agents de notre système dans le scénario 2

1 L'agent utilisateur cloud envoie une requête à l'agent courtier demandant un service.

2 L'agent courtier vérifie l'état des serveurs dans son répertoire ainsi que le nombre de clients dans la file d'attente. Dans ce scénario, tous les serveurs sont occupés et la file d'attente est vide. Il met l'agent utilisateur cloud dans la file d'attente et l'informe de la non-disponibilité d'un serveur et du temps d'attente estimé.

3 L'agent utilisateur cloud demande un suivi de l'état de sa demande à l'agent courtier.

4 L'agent courtier informe l'agent utilisateur cloud de l'état de sa demande qu'elle est en attente de traitement. L'agent utilisateur cloud (virtuel) affiche à son tour ces informations sur l'interface graphique utilisateur

Scénario 3 : Dans ce cas on suppose qu'aucun serveur n'est disponible, qu'il y au moins un utilisateur en attente d'être servi et qu'aucun utilisateur n'est en attente d'un serveur VIP (place d'attente VIP est vide). Dans ce scénario, on suppose que l'utilisateur accepte le VIP et sera servi dès la libération d'un serveur. L'échange de messages entre les différents agents de notre système dans ce cas-là est illustré dans la figure 3.8.

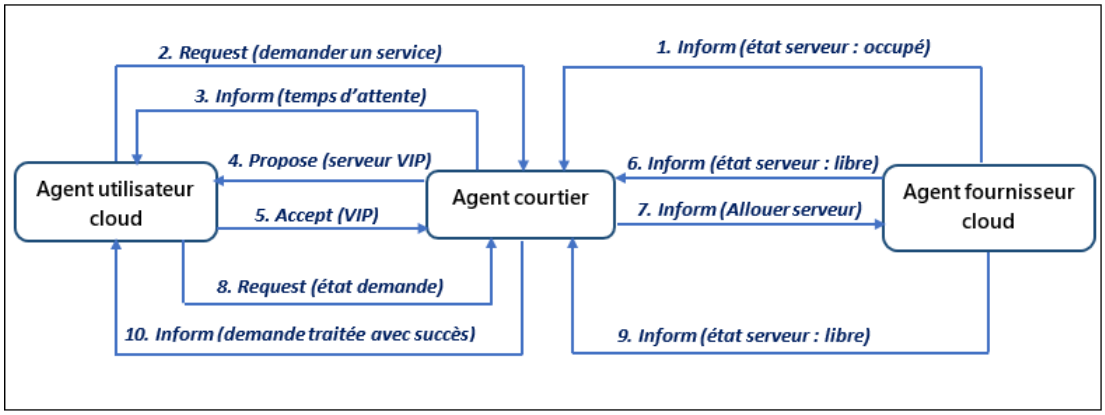

FIGURE 3.8 – Schéma explicatif du déroulement des échanges entre les agents de notre système dans le scénario 3

1 Initialement, l'agent fournisseur informe l'agent courtier de l'état de son serveur (serveur non disponible).

2 L'agent utilisateur cloud envoie une requête à l'agent courtier demandant un service.

3 L'agent courtier vérifie la disponibilité des serveurs, le nombre d'utilisateurs en attente de traitement dans la file d'attente et l'état de la place d'attente VIP qui, dans ce scénario, est libre. L'agent courtier informe l'agent utilisateur cloud de la non disponibilité d'un serveur.

4 L'agent courtier lance une négociation et propose à l'agent utilisateur cloud l'allocation d'un serveur VIP.

5 L'agent utilisateur cloud peut refuser la proposition et attendre dans la file d'attente ou l'accepter. On suppose que l'utilisateur cloud est impatient donc accepte la proposition, et informe l'agent courtier de sa décision.

6 l'agent fournisseur informe l'agent courtier de la libération de son serveur.

7 Après libération d'un serveur, l'agent courtier l'alloue en tant que serveur VIP pour le traitement de la demande de l'utilisateur cloud et met à jour son répertoire.

8 L'agent utilisateur cloud peut suivre l'état de sa demande, pour ce faire il envoie une requête à l'agent courtier pour demander l'état de sa demande.

9 Une fois le traitement términé l'agent fournisseur cloud libère le serveur alloué et informe l'agent utilisateur cloud de la disponibilité du serveur.

10 L'agent courtier informe l'agent utilisateur cloud du bon traitement de sa demande. l'agent utilisateur cloud(virtuel) va à son tour afficher les résultats aux clients sur l'interface graphique.

Scénario 4 : Dans ce cas on suppose qu'aucun serveur n'est disponible, qu'il y au moins un utilisateur en attente d'être servi et que la place d'attente VIP est occupée. Dans ce scénario, on suppose que l'utilisateur est impatient annule sa demande et quitte le système.
La figure 3.9 illustre les différents messages échangés entre les agents de notre système dans ce cas là.

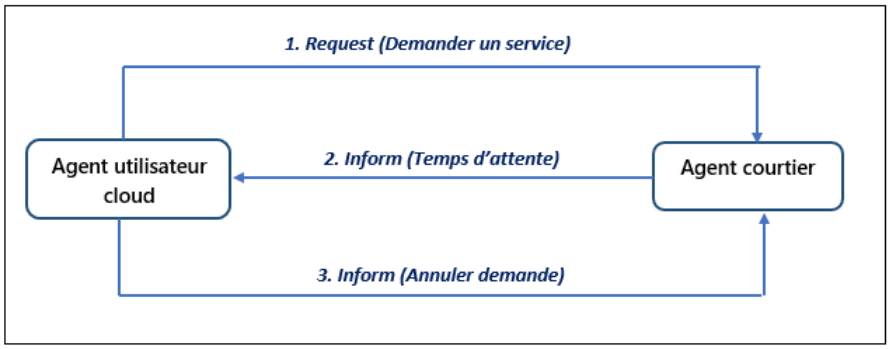

FIGURE 3.9 – Schéma explicatif du déroulement des échanges entre les agents de notre système dans le scénario 4

1 L'agent utilisateur cloud envoie une requête à l'agent courtier pour demander un service.

2 L'agent courtier vérifie la disponibilité des serveurs, le nombre d'utilisateurs en attente de traitement dans la file d'attente et l'état de la place d'attente VIP (occupée).
Il informe l'agent utilisateur cloud du temps d'attente estimé avant qu'un serveur se libère et que tous les clients le précédant dans la file d'attente soient traités.

3 L'agent utilisateur cloud informe l'agent courtier de l'annulation de sa demande et quitte le système.

Concernant le traitement des utilisateurs dans la file d'attente, ils sont traités en suivant la discipline de service FCFS(first come first served). Chaque utilisateur a la possibilité d'annuler sa demande de service à tout moment et de quitter le système en cas d'impatience.

3.6 Conclusion

Dans ce chapitre, nous avons décrit le problème posé et nous avons donné l'architecture du modèle proposé pour sa résolution.

Nous avons ensuite détaillé le système multi-agents modélisant la gestion des ressources du cloud computing où nous avons décrit le comportement de chacun des types d'agents

composant ce système. Enfin, nous avons présenté quelques scénarios d'exécution pour mieux expliquer l'opération de notre modèle.

Dans le chapitre suivant, nous nous intéressons à la concrétisation de notre système proposé et cela par l'implémentation du modèle présenté sur la plateforme Jade.

Chapitre 4

Conception et réalisation

4.1 Introduction

Dans ce chapitre, nous allons exprimer notre démarche de conception à l'aide du langage UML et AgentUML, et plus particulièrement le diagramme de cas d'utilisation pour exprimer les fonctionnalités du système et le diagramme de séquence pour exprimer le comportement du système proposé et les diagrammes de classes agents pour exprimer l'aspect statique et le rôle des agents du système. Par la suite, nous allons définir le langage de programmation et l'environnement de développement utilisé. Nous concrétisons le système proposé par l'implémentation de l'approche présentée sur la plateforme Jade.

4.2 Conception

4.2.1 UML (Unified Modeling Language)

Le Langage de Modélisation Unifié, est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet. Il comporte ainsi plusieurs types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information [46].

4.2.2 Démarche de modélisation

Nous avons adopté un processus de modélisation itératif centré sur l'architecture, Piloté par des cas d'utilisation et orienté vers la diminution des risques, comme illustré sur la figure 4.1.

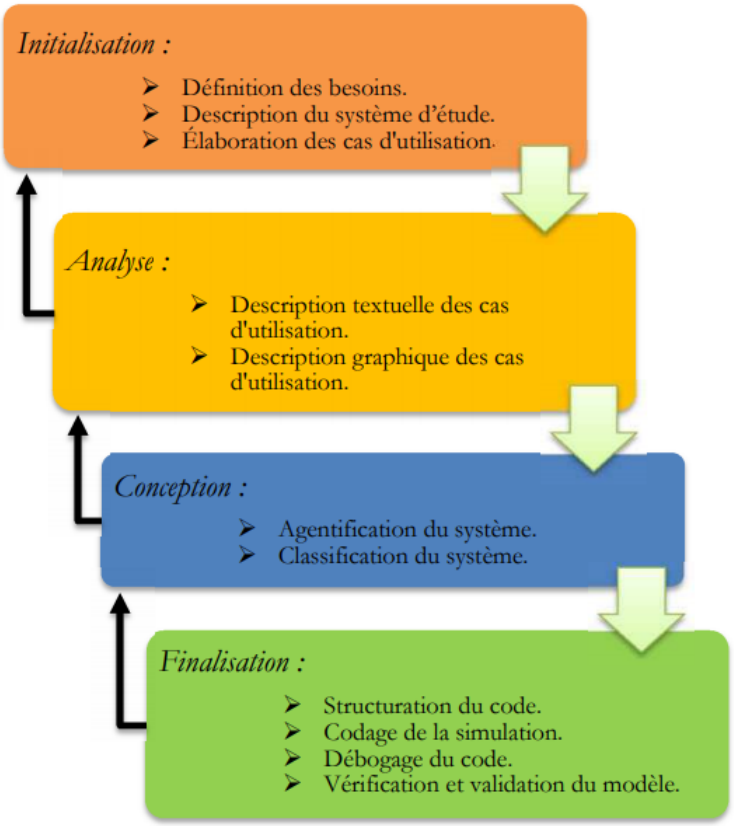

FIGURE 4.1 – Processus de modélisation itératif [1]

4.2.3 Agent UML

AgentUML propose des extensions pour la représentation des agents, il se base sur la méthode UML (Unified Modeling Language). Comme nous l'avons déjà vu, par rapport aux objets, les agents ont des activités autonomes et des buts. C'est cette différence qui entraîne l'insuffisance d'UML pour modéliser les agents et les systèmes multi-agents. L'idée générale d'AUML est donc de combler les déficiences d'UML pour la modélisation des systèmes agents [15].

4.2.4 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système. Il représente les fonctionnalités primordiales aux utilisateurs et dont le système doit disposer.

4.2.5 Diagrammes d'interactions

Les diagrammes d'interactions capturent le comportement d'un cas d'utilisation. Leur but est de montrer l'échange de messages entre les agents dans le temps en utilisant les pro-

tocoles de communication. Ces diagrammes sont composés de deux axes : un vertical représentant le temps, et un horizontal représentant différents agents ou différents rôles d'agents. Plus trois connecteurs AND, OR et XOR, illustré sur la figure 4.2

Dans les diagrammes de séquences AUML il est possible d'envoyer un message à plusieurs agents ou rôles au même temps[5].

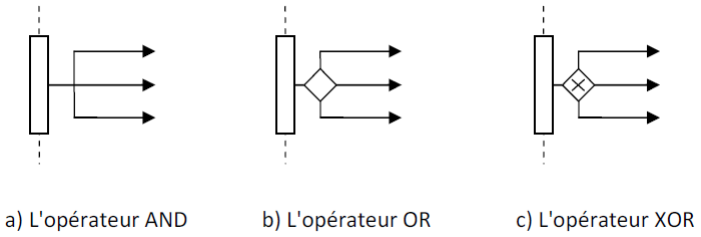

FIGURE 4.2 – Types de connecteurs dans AUML

4.2.6 Diagrammes de classe agents

Le diagramme de classes agents décrit la sémantique statique des agents. Une classe d'agent représente un agent ou un groupe d'agents pouvant jouer un rôle ou avoir un comportement déterminé [15], il comporte :

- **Nom de la classe agent/ role1, role2, ... :** Un agent d'une classe donnée peut avoir plusieurs rôles.
- **Description des états :** Définition de variables d'instance qui reflètent l'état de l'agent
- **Actions :** Désigne les actions que l'agent peut effectuer, il existe deux types : pro-active les actions effectuer par l'agent lui-même et réactive les actions résultant des messages qu'il reçoit.
- **Méthodes :** Elles sont définies comme dans UML classique.
- **Envoi et réception de messages :** Description des messages émis et reçus par l'agent en précisant les protocoles.

4.2.7 Conception et modéliasation

— **Diagramme de cas d'utilisation de l'utilisateur cloud humain**
Ce diagramme illustre un seul type d'acteur qui est l'utilisateur cloud humain. Un utilisateur cloud doit s'authentifier ou s'inscrire pour pouvoir accéder au système ainsi, il pourra demander un service et suivre l'état de sa demande ou annuler sa demande s'il le souhaite.

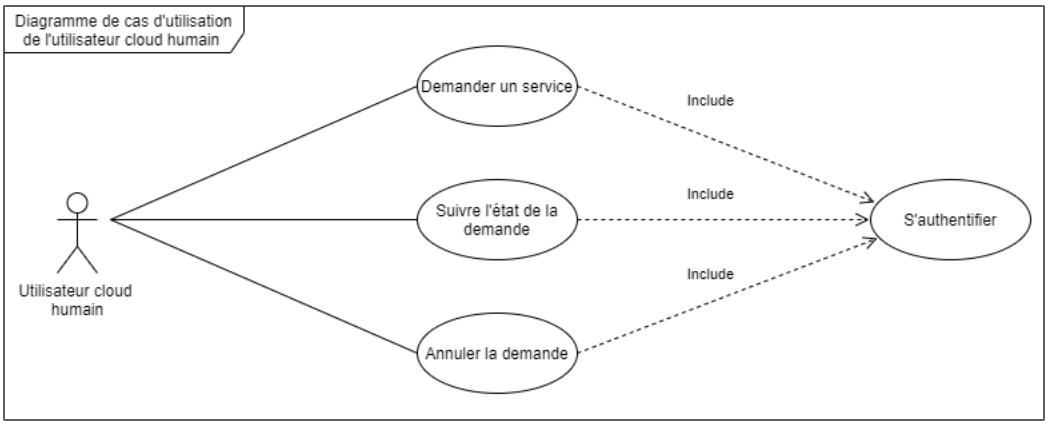

FIGURE 4.3 – Diagramme de cas d'utilisation de l'utilisateur cloud humain

— **Diagramme de cas d'utilisation de l'agent utilisateur cloud virtuel**
Ce diagramme aussi illustre un seul type d'acteur qui est l'agent utilisateur cloud virtuel. L'agent utilisateur cloud virtuel représente l'utilisateur cloud humain dans le système. Il est chargé d'envoyer la demande de l'utilisateur humain à l'agent courtier, il peut suivre ainsi l'état de la demande et afficher les informations relatives à cet état sur l'interface graphique pour l'utilisateur cloud humain.

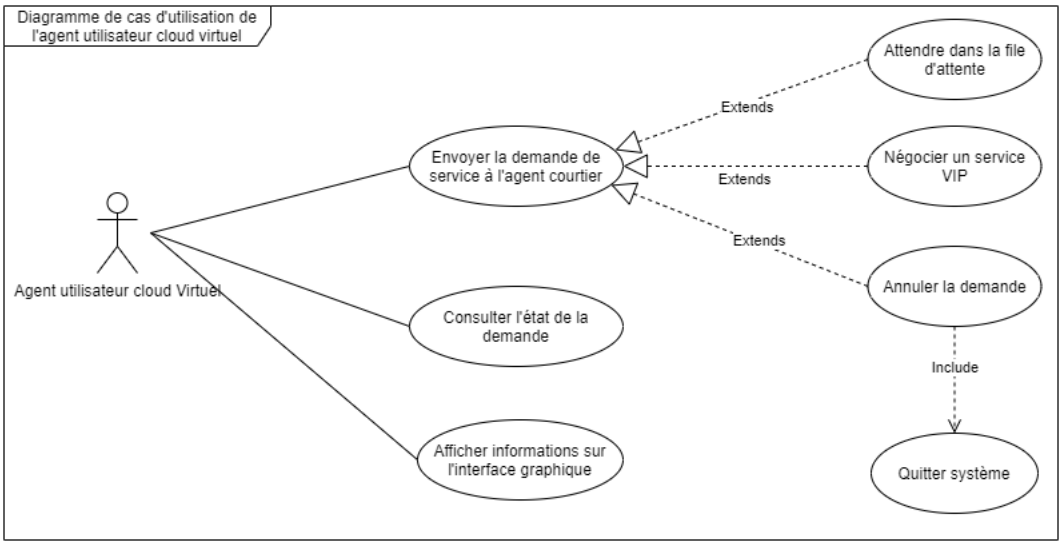

FIGURE 4.4 – Diagramme de cas d'utilisation de l'agent utilisateur cloud virtuel

— **Diagramme de cas d'utilisation de l'agent courtier et de l'agent fournisseur cloud**
Ce diagramme illustre deux types d'acteurs à savoir l'agent courtier qui est chargé d'allouer des serveurs (VIP ou normal) aux utilisateurs cloud, il incite l'utilisateur cloud à négocier pour bénéficier d'un serveur VIP, il gère la file d'attente des utilisateurs et mets à jour l'état des serveurs. L'agent fournisseur envoie les informations relatives à l'état des serveurs à l'agent courtier.

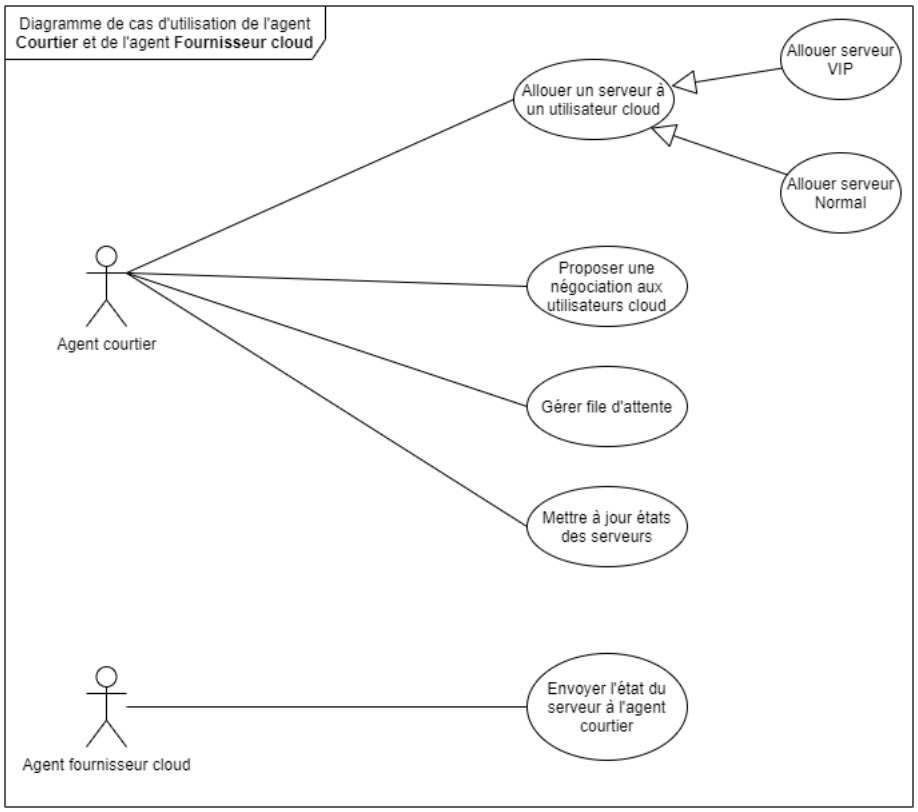

FIGURE 4.5 – Diagramme de cas d'utilisation de l'agent courtier et de l'agent fournisseur cloud

— **Diagramme de séquence**

Différents types de messages sont échangés entre les agents de notre système. Nous présentons ci-dessous les différentes interactions entre les agents, que nous modélisons par un diagramme de séquence AUML. Le diagramme d'interaction illustré par la figure 4.6 montre les protocoles de négociation qui régit les interactions entre les différents agents de notre système à savoir l'agent utilisateur cloud, l'agent courtier et l'agent fournisseur cloud. Ce diagramme regroupe les différents scénarios possibles lors d'une exécution du système.

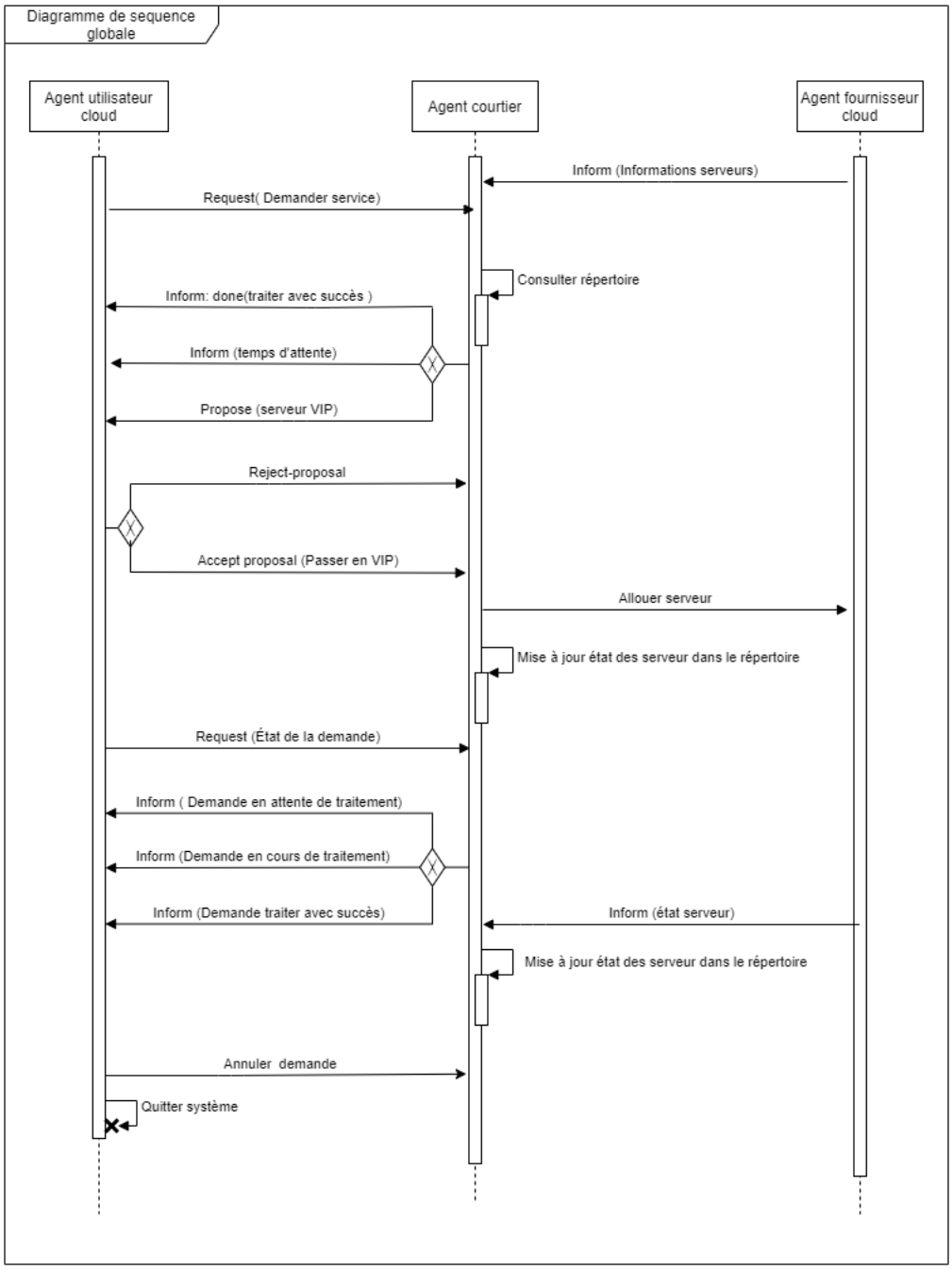

FIGURE 4.6 – Diagramme de séquence globale

- **Diagramme de classe agent utilisateur cloud** L'agent utilisateur cloud peut recevoir deux types de messages, le premier type contient des informations relatives à sa demande de service. Quant au second type, c'est une proposition de négociation d'un serveur VIP. Concernant le type des messages qu'il peut envoyer, c'est un REQUEST pour demander un service ou suivre l'état de sa demande, un ACCEPT-PROPOSAL pour accepter la proposition de négociation de serveur VIP ou un REJECT-PROPOSAL

pour refuser, et un INFORM pour informer l'agent courtier de l'annulation de sa demande.

La structure de l'agent utilisateur cloud est présentée dans la figure 4.7.

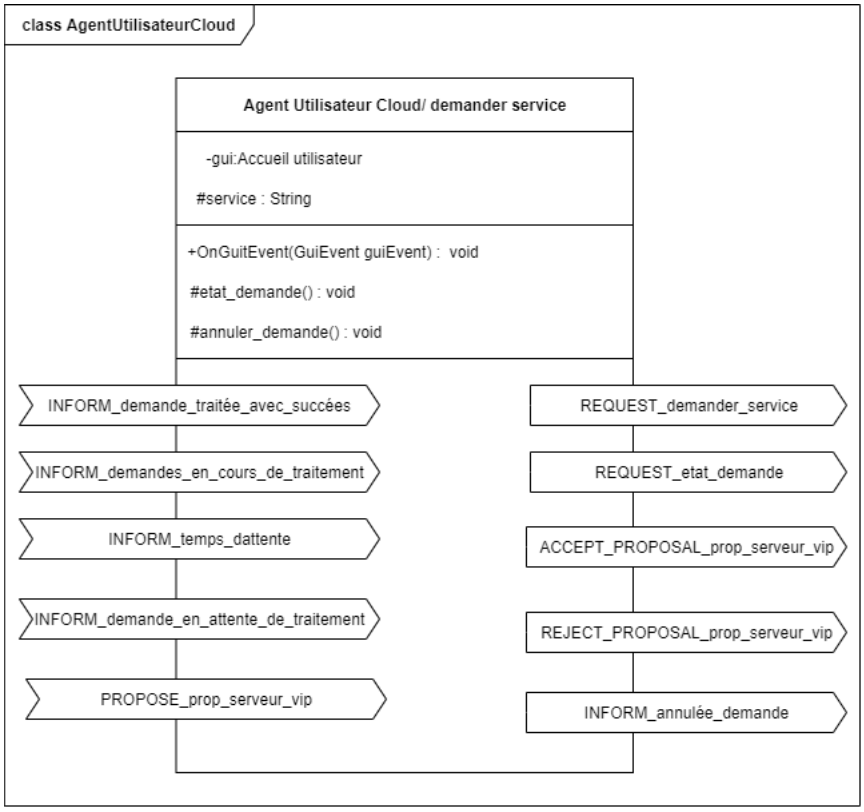

FIGURE 4.7 – Diagramme de classe agent utilisateur cloud

- **Diagramme de classe agent Fournisseur cloud**

  Cet agent envoie un seul type de message à l'agent courtier, un INFORM pour faire passer l'état du serveur. Il peut recevoir un seul type de message qui l'informe de l'allocation de son serveur de la part de l'agent courtier. La structure de l'agent fournisseur cloud est présentée dans la figure 4.8.

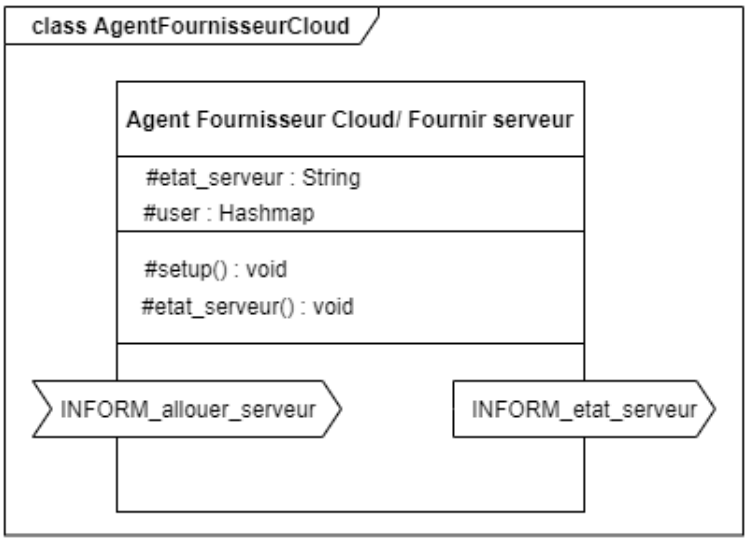

FIGURE 4.8 – Diagramme de classe agent fournisseur cloud

- **Diagramme de classe agent courtier**
  Cet agent envoie deux types de messages. Le premier type sert à informer les agents fournisseur et utilisateur cloud de l'allocation d'un serveur et de l'état de la demande, le second sert a proposer un serveur VIP à l'agent utilisateur cloud.
  Il peut recevoir quatre types de messages un ACCEPT-PTOPOSAL pour confirmer l'acceptation de la proposition du serveur VIP de la part de l'agent utilisateur cloud ou un REJECT-PROPOSAL pour refuser, un INFROM de la part de l'agent fournisseur pour lui communiquer l'état de son serveur, ou de la part de l'agent utilisateur cloud pour l'informer de l'annulation de son service. La structure de l'agent courtier est présentée dans la figure 4.9.

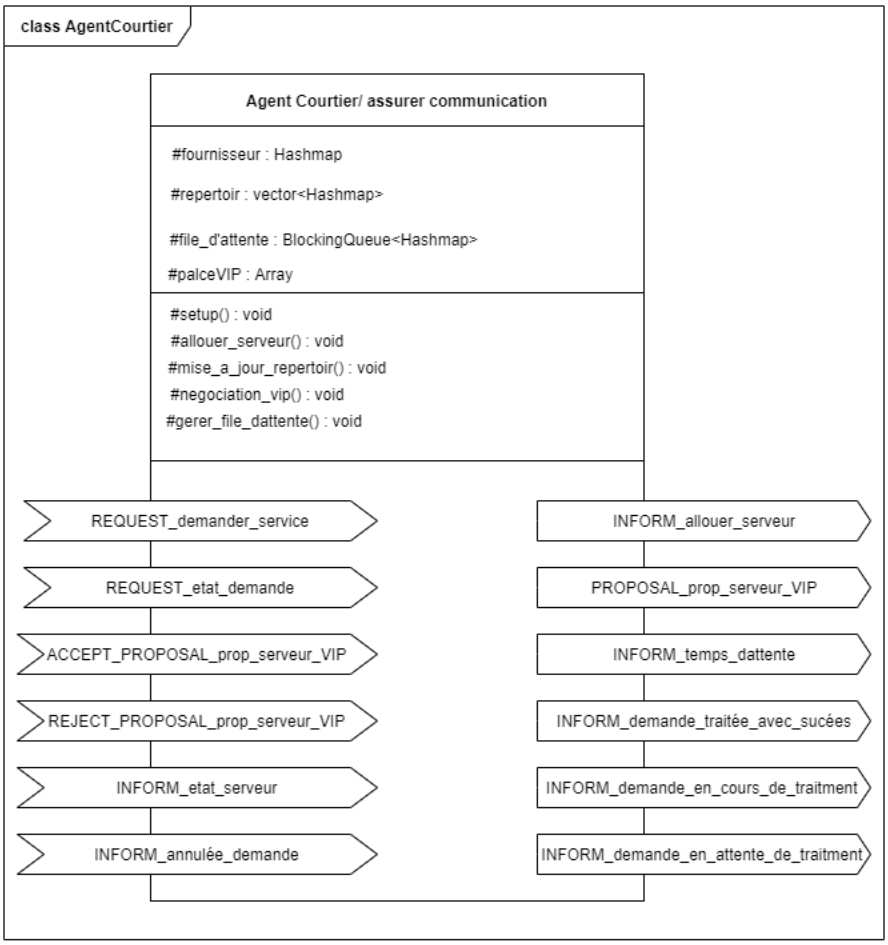

FIGURE 4.9 – Diagramme de classe agent courtier

4.3 Réalisation

La phase de réalisation de tous systèmes informatiques correspond d'une part à choisir les outils de développement conformes aux spécificités du système, d'autre part, la traduction du modèle conceptuel en utilisant les outils choisis sous forme d'un logiciel opérationnel.

4.3.1 Outils de développement et environnement utilisé

Comme un outil de développement, nous avons utilisé l'IDE Eclipse (version : Photon Release (4.8.0)) basé sur le langage java, nous avons utilisé JADE (version 4.5) comme plate-forme de développement SMA, JADE est basé sur un langage de communication des agents appelés FIPA-ACL. Pour un bon affichage d'informations au cours de l'exécution nous avons utilisé le framework JavaFx(version 8.0.8).

4.3.2 Environnement de développement ECLIPSE

Eclipse est un environnement de modélisation et de développement générique, open source et extensible, il permet le développement de diverses applications mettant en œuvre n'importe quel langage de programmation et exécutables sûrs de nombreux systèmes d'exploitation. Il offre un ensemble de modules et de bibliothèques servant à la gestion des ressources, et offre de façon standard un environnement de développement JAVA et des outils de développement des modules d'extension. Eclipse a été conçu de manière à pouvoir facilement étendre ses fonctions à l'aide de modules d'extension tout en conservant une interface graphique cohérente. [23].

4.3.3 Le Langage de programmation Java

Le langage java est un langage de programmation orienté objet dérivé du C, mais plus simple à utiliser, créé par James Gosling et Patrick Naughton, employé de Sun Microsystems en 1995.

Il a la particularité d'être portable et multi-plateformes, il permet le développement d'applications portables hautes performances sur une large gamme de plateformes informatiques, et sur plusieurs systèmes d'exploitation dotés, en standard, de bibliothèques de classes très riches comprenant la gestion des interfaces graphiques, la programmation multithread, la gestion des exceptions, les accès aux fichiers et au réseau. L'utilisation de ces bibliothèques facilite grandement les tâches lors de la construction d'application complexe [39].

4.3.4 framework JavaFx

JavaFX est un framework open source basé sur Java pour le développement d'applications clients enrichis. Il est également considéré comme le successeur de Swing dans le domaine de la technologie de développement d'interface utilisateur graphique (GUI) sur la plateforme Java. La bibliothèque JavaFX est disponible en tant qu'interface de programmation d'applications (API) Java publique. JavaFX contient plusieurs fonctionnalités qui en font un choix privilégié pour le développement d'applications clients enrichis.

4.3.5 Plateforme Jade

JADE (Java Agent DEvelopment Framework) est défini dans le chapitre 02. Son objectif est de simplifier le développement tout en assurant la conformité aux normes grâce à un ensemble complet de services système et d'agents. [32], il possède des modules principaux(nécessaire aux normes FIPA) :

- DF (Directory Facilitator) : fournie un service de page jaune à la platefrome.
- ACC (Agent Communication Channel) : gère la communication entre les agents.
- AMS (Agent Management System) : supervise l'enregistrement des agents,leur authentification, leur accès et l'utilisation du système.
- RMA (Remote Management Agent) : permet de contrôler le cycle de vie de la plateforme et tous les agents le composant.

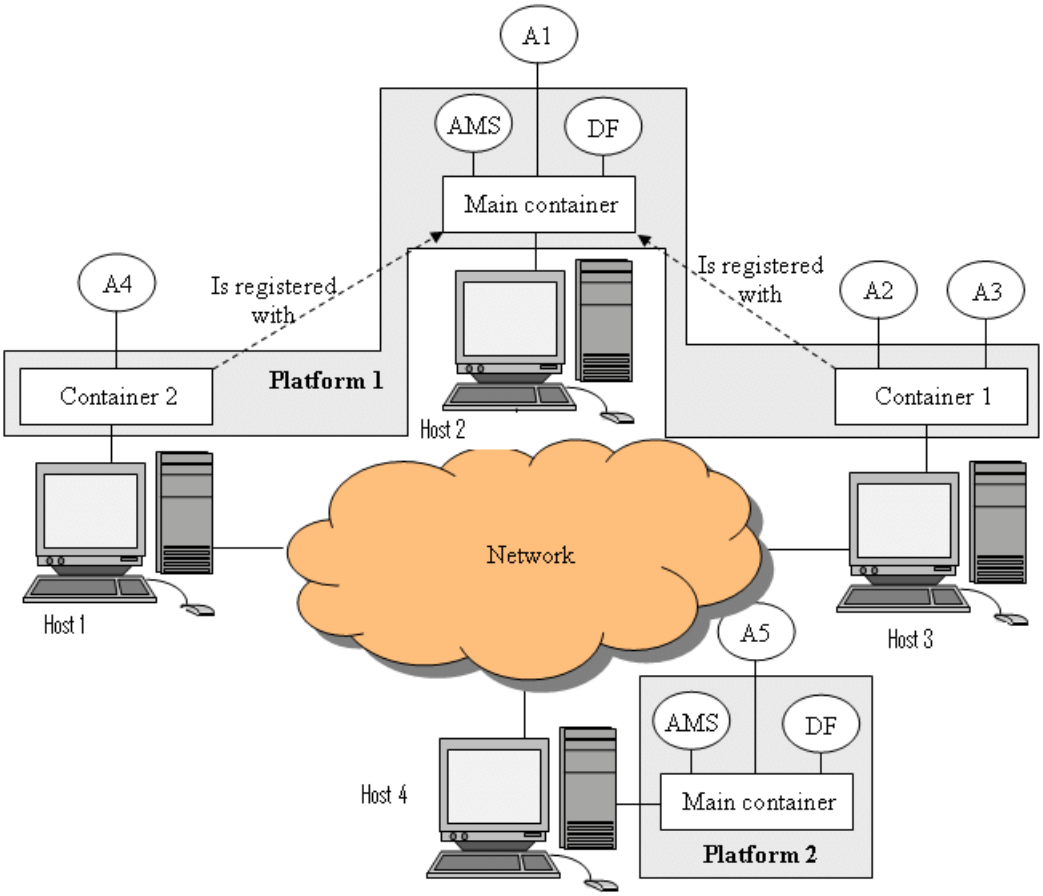

FIGURE 4.10 – Architecture de la plateforme JADE [26].

4.3.6 Déploiement des agents du système proposé

Comme décrit précédemment, notre application est basée sur les agents : Agent utilisateur cloud, Agent courtier, agent fournisseur cloud et les trois agents qui se lancent lors du démarrage de la plateforme JADE(AMS, DF et RMA), la figure 4.11 illustre ces différents agents

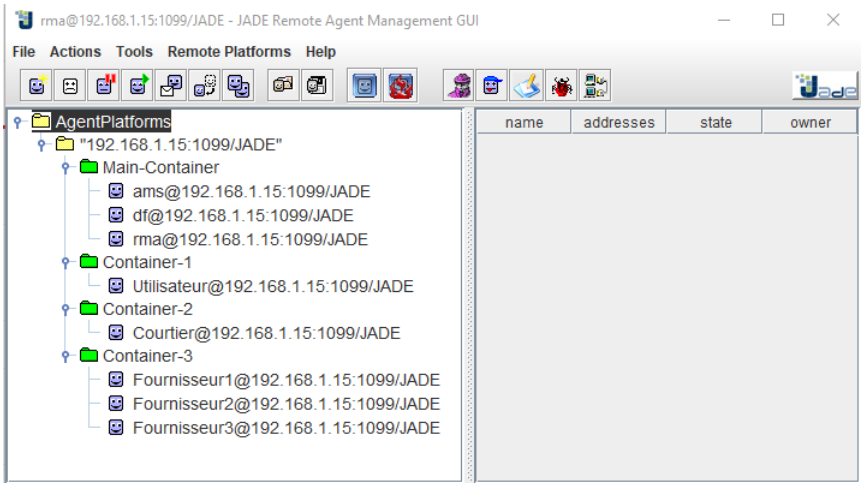

FIGURE 4.11 – Déploiement des agents du système sur la plateforme JADE.

Comme le montre la figure 4.11 les agents de notre système sont déployés sur trois conteneurs agents différents afin d'assurer leurs exécutions parallèles, mais aussi le développement d'un système distribué.

4.3.7 Description des interfaces du système proposé

Une interface graphique est associée à l'utilisateur cloud humain, la figure 4.12 montre l'interface d'authentification qui permet à l'utilisateur cloud humain d'accéder au système.

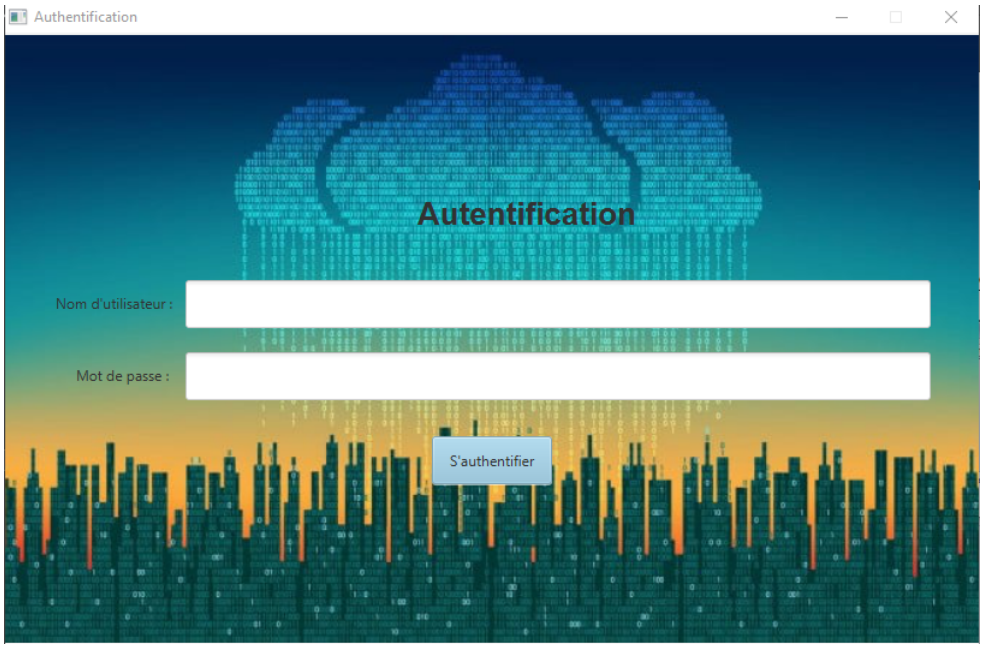

FIGURE 4.12 – Interface d'authentification.

Après l'authentification d'un utilisateur cloud humain, l'interface d'accueil pour un utilisateur cloud humain est illustré par la figure 4.13.

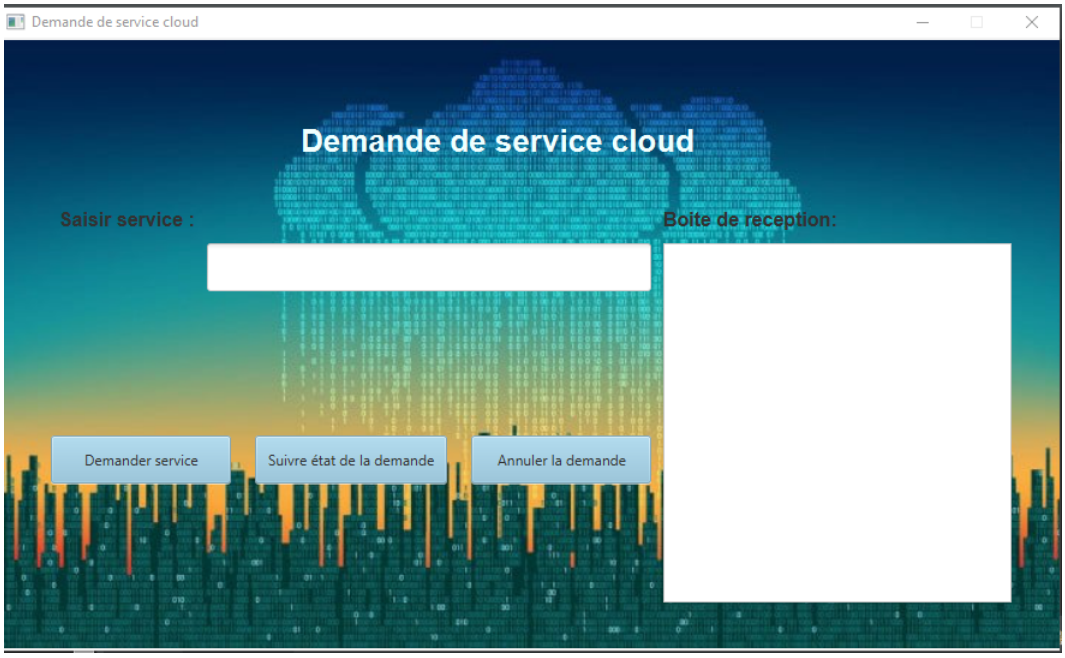

FIGURE 4.13 – Interface d'accueil pour l'utilisateur cloud humain

4.3.8 Simulation et Évaluation de la solution proposée

Pour évaluer notre application nous avons considéré différents scénarios d'exécution, entre autre :

- Scénario 01 : un utilisateur cloud humain demande un service avec un serveur libre, un serveur lui est directement allouer.
- Scénario 02 : un utilisateur cloud humain demande un service avec un serveur occupé et une file d'attente vide, il est redirigé directement vers la file d'attente, étant impatient, il annule sa demande et quitte le système.
- Scénario 03 : un utilisateur cloud humain demande un service avec un serveur occupé, une file d'attente non-vide et la place VIP libre, un serveur VIP lui est proposé étant impatient, il accepte cette proposition.
- Scénario 04 : un utilisateur cloud humain demande un service avec un serveur occupé, une file d'attente non-vide et la place VIP libre, un serveur VIP lui est proposé, mais refuse cette proposition et rejoint la file d'attente.

Afin de représenter les traces d'une communication entre agents, la plateforme JADE dispose de deux agents spécifiques, l'agent Dummy et l'agent Sniffer. Dummy est un outil pour inspecter les échanges de messages entre agents, et offre une interface graphique pour l'édition, l'écriture et l'envoi des messages ACL vers les agents, permet de recevoir et lire les messages des autres agents.

Sniffer trace l'échange de messages et donne une interface graphique pour afficher les

échanges de messages entre les différents groupes d'agents en utilisant une notation proche d'UML.

Grâce à l'agent sniffer, nous présentons dans ce qui suit quatre scénarios différents selon l'impatiente ou non d'un utilisateur, la disponibilité du serveur, l'état de la file d'attente, et de la place d'attente VIP. Nous nous sommes limités à récupérer le nom du service saisie par l'utilisateur cloud humain, et l'état du serveur communiquer par l'agent fournisseur cloud. La file d'attente et la place d'attente VIP sont gérer par l'agent courtier.

— **Scénario 01**

La figure 4.14 illustre les messages échangés entre les trois types d'agent, On suppose qu'au moins un serveur est libre. Les agents fournisseurs cloud envoient un message de type INFORM à l'agent courtier pour l'informer de l'état de leur serveur, celui-ci le sauvegarde dans son répertoire. L'agent utilisateur cloud envoie un message de type REQUEST à l'agent courtier pour demander un service, l'agent courtier vérifie l'états des serveurs dans son répertoire, et alloue un serveur à l'agent courtier pour le traitement de son service. L'agent utilisateur cloud demande à suivre l'état de sa demande et envoie un message de type REQUEST à l'agent courtier celui-ci répond par un message de type INFORM pour l'informer que son service est traité avec succès.

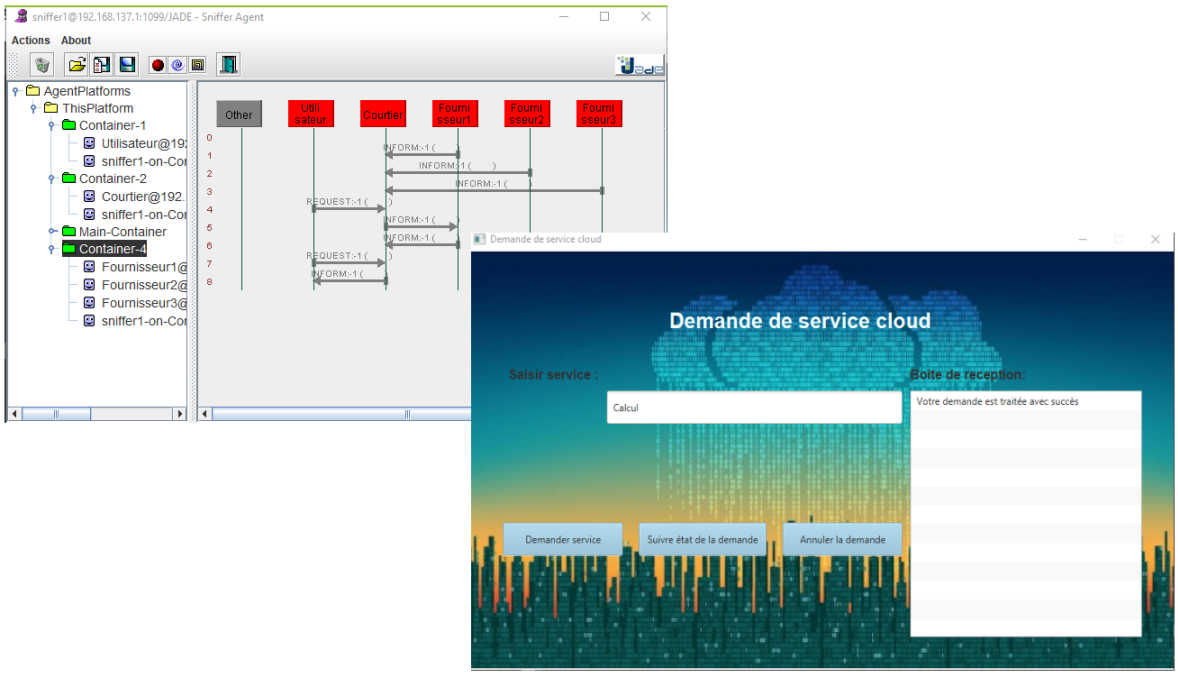

FIGURE 4.14 – fragment de la conversation scénario 01

— **Scénario 02**

La figure 4.15 illustre les messages échanges entre les trois types d'agent, on suppose que tous les serveurs sont occupés, la file d'attente est vide. Les agents fournisseurs cloud envoient un message de type INFORM à l'agent courtier pour l'informer de l'état

de leur serveur, celui-ci le sauvegarde dans son répertoire. L'agent utilisateur cloud envoie un message de type REQUEST à l'agent courtier pour demander un service, l'agent courtier vérifie l'états des serveurs dans son répertoire et envoie un INFORM a l'agent utilisateur cloud pour l'informer du temps d'attente estimé avant le traitement de sa demande. L'agent utilisateur cloud étant impatient Annule sa demande et quitte le système.

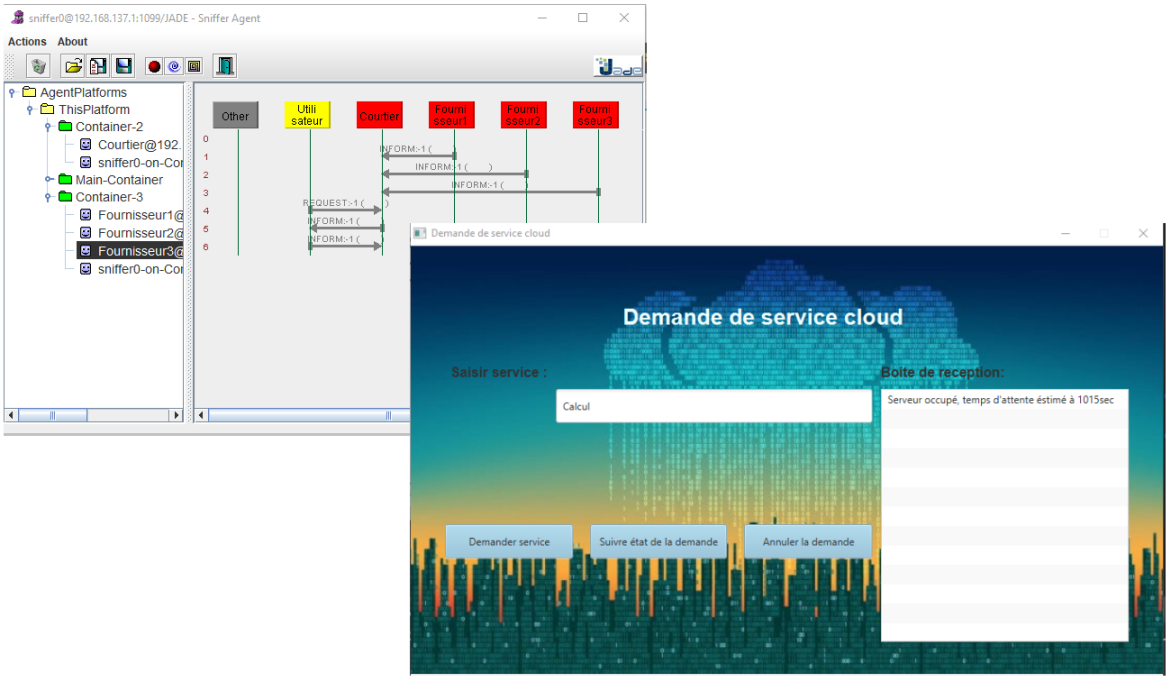

FIGURE 4.15 – fragment de la conversation scénario 02

— **Scénario 03**

La figure 4.17 illustre les échanges de messages entre les trois types d'agents de notre système.On suppose que tous les serveurs sont occupés, la file d'attente n'est pas vide et que la place VIP est libre. L'agent courtier envoie un message de type INFORM à l'agent utilisateur pour l'informer du temps d'attente pour que son service soit traité ensuite il lui envoie un autre message de type PROPOSE pour lui proposer un l'allocation d'un serveur VIP, l'agent utilisateur cloud refuse la proposition et décide d'attendre dans la file d'attente il envoie un message de type REJECT-PROPOSAL à l'agent courtier, celui-ci l'informe qu'il est redirigé vers la file d'attente. Après écoulement du temps d'attente et libération d'un serveur l'agent courtier alloue un serveur normal à l'agent utilisateur cloud.

L'agent utilisateur cloud envoie un message de type REQUEST à l'agent courtier pour demander l'état de sa demande, celui-ci répond par un message de type INFORM pour l'informer que son service est en cours de traitement

La figure 4.16 illustre la boite de dialogue proposant l'allocation d'un serveur VIP pour l'utilisateur cloud.

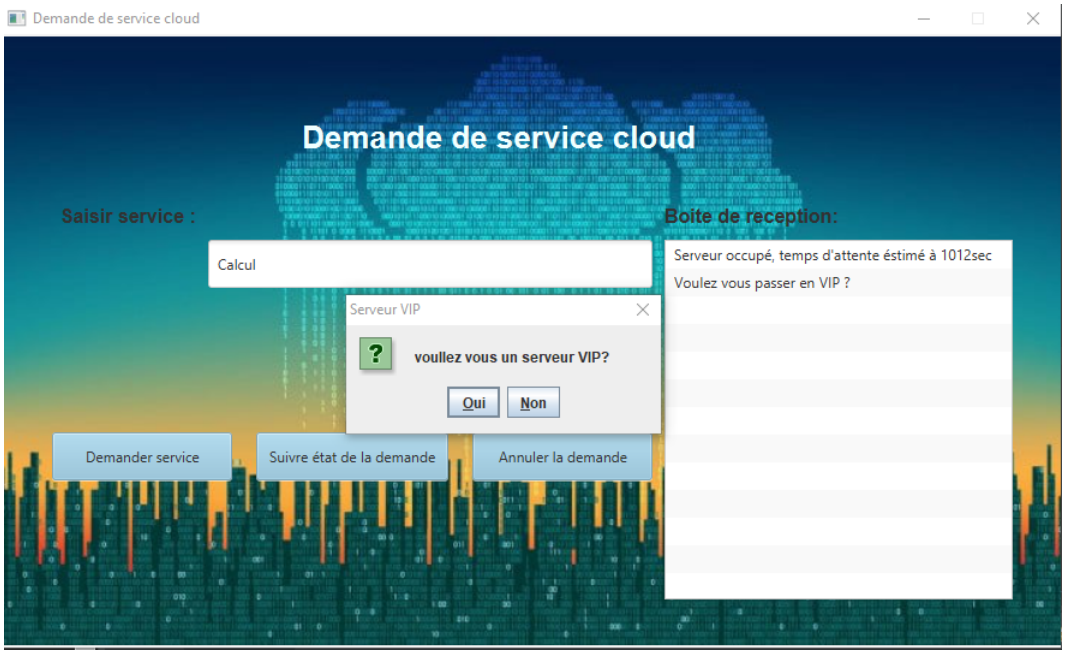

FIGURE 4.16 – interface proposition serveur VIP

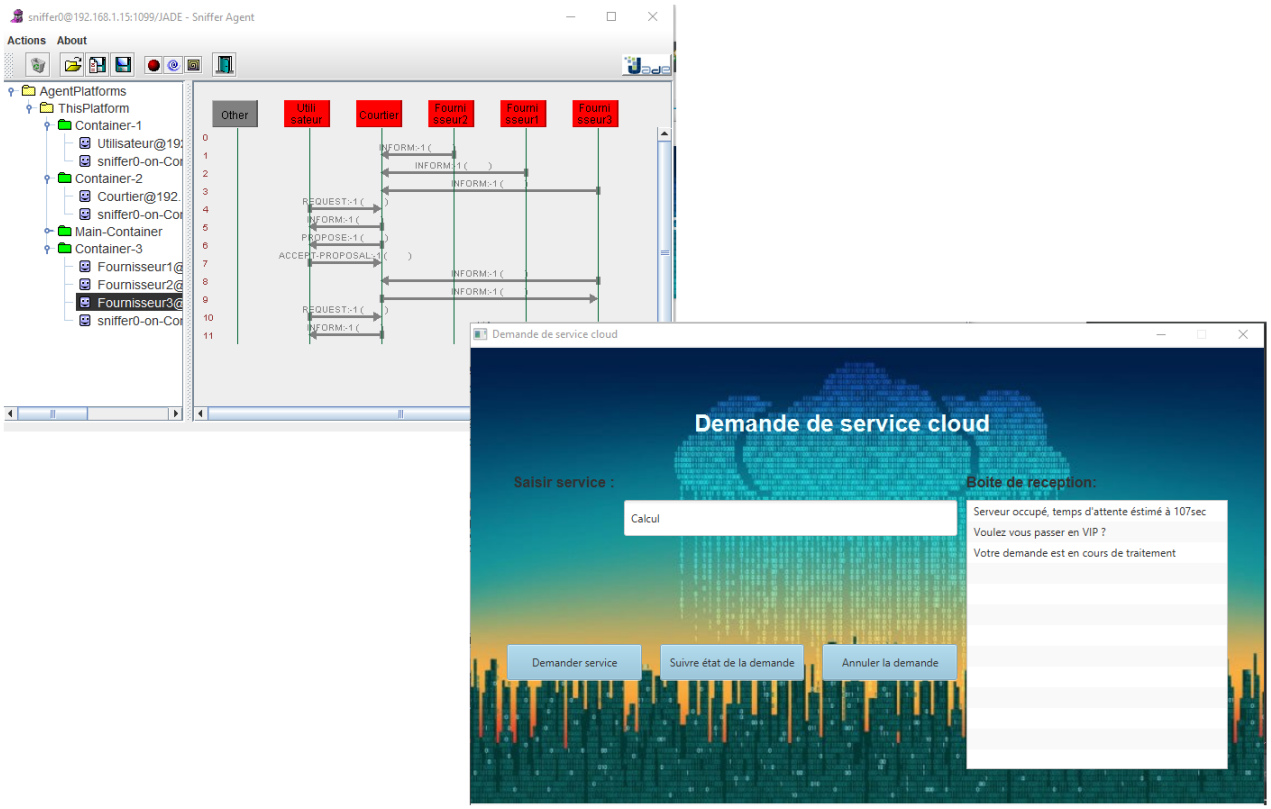

FIGURE 4.17 – fragment de la conversation scénario 03

— **Scénario 04**

La figure 4.18 illustre les échanges de messages entre les trois agents du système dans le cas où le serveur est occupé, la file d'attente n'est pas vide et que la place VIP est libre, dans ce cas l'agent courtier envoie un message de type INFORM à l'agent utilisateur pour l'informer du temps d'attente pour que son service soit traité ensuite il lui envoie un autre message de type PROPOSE pour lui proposer un serveur VIP, l'agent utilisateur cloud refuse la proposition et décide d'attendre dans la file d'attente il envoie un message de type REJECT-PROPOSAL à l'agent courtier, celui-ci l'informe qu'il est rediriger ver la file d'attente après écoulement du temps d'attente et libération d'un serveur l'agent courtier alloue un serveur normal à l'agent utilisateur cloud.
L'agent utilisateur cloud demande de suivre l'état de ça demande et envoie un message de type REQUEST à l'agent courtier celui-ci répond par un message de type INFORM pour l'informer que son service est en cours de traitement.

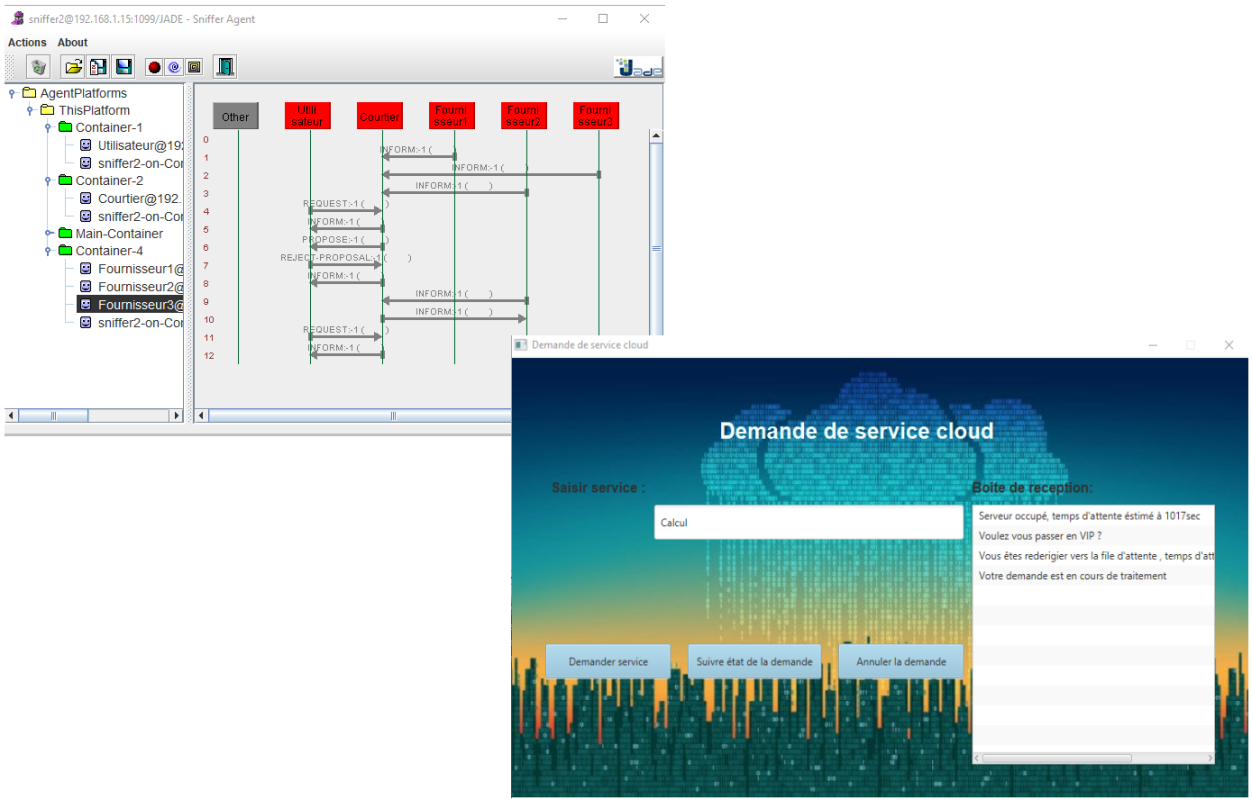

FIGURE 4.18 – fragment de la conversation scénario 04

4.4 Conclusion

Dans ce chapitre, nous avons détaillé la conception de notre application. En effet, nous avons commencé par les notions de base reliées à la modélisation. Nous avons exprimé notre démarche de conception à l'aide du langage UML et AgentUML, pour décrire l'aspect fonctionnel et comportemental du système proposé. De plus, nous avons présenté quelques concepts de base sur les outils et plateformes qui nous ont été indispensables pour le développement de notre application. Ensuite, nous avons illustré quelques interfaces du système implémenté sur la plateforme JADE. Nous avons testé quelques scénarios pour s'assurer du bon fonctionnement de l'application proposée.

Conclusion générale et perspectives

Tout au long de ce mémoire, nous avons présenté les différentes technologies nécessaires pour proposer une approche basée-agents pour la gestion cloud computing. Nous nous sommes intéressés à la technologie des systèmes multi-agents et à leur utilisation dans un cloud computing dont le but est de satisfaire les besoins des utilisateurs cloud.

En adoptant le paradigme agent, nous avons modélisé le processus d'allocation des ressources dans le cloud computing par un système multi-agent qui comporte trois agents : un agent utilisateur cloud qui représente l'utilisateur dans l'environnement cloud, un agent fournisseur cloud qui représente le fournisseur dans l'environnement cloud et un agent courtier qui joue le rôle de courtage entre les utilisateurs et les fournisseurs cloud.

Nous avons fait le point sur les différents concepts clés du cloud computing, ainsi qu'une étude générale sur les agents et les systèmes multi-agents. Ensuite, nous avons discuté un certain nombre de travaux et projets de recherche sur les techniques d'évaluation des performances du cloud computing via les systèmes d'attente ainsi que sur la conception de cloud computing basée sur les systèmes multi-agents.

Enfin, nous avons présenté notre architecture à base d'agents pour la gestion du cloud computing, où on a donné un intérêt particulier pour le rôle, les fonctions et les différentes interactions entre les agents.

Pour mettre en œuvre notre système, nous avons utilisé le langage de programmation java et la plateforme JADE pour implémenter les agents. Il est à noter que ce travail est loin d'être une solution complète. Pour ce faire, de nombreuses perspectives peuvent être envisageables pour apporter une amélioration à ce présent projet :

- Améliorer notre proposition par le traitement des clients impatients avec priorité absolue.
- Intégrer une seconde file d'attente orbite pour améliorer le temps d'attente de traitement de services.
- Ajouter une interface de gestion fournisseurs, pour la publication et la gestion des services.

- Évaluation des performances de notre proposition en utilisant les outils mathématiques : la théorie des files d'attente.

Bibliographie

[1] M. N. Abourraja. *Gestion multi-agents d'un terminal à conteneurs*. Thèse de doctorat, Université Le Havre, Normandie, France, 2018.

[2] D. Adams. *Une approche basée agents pour l'allocation des ressources dans le Cloud Computing*. San Val, 1995.

[3] A.O. Akinwunmi and E.A. Olajubu G.A. Aderounmu. A multi-agent system approach for trustworthy cloud service discovery. Novembre 2016.

[4] M. Al-Ayyoub, M. Daraghmeh, Y. Jararweh, and A. Qutaibah. Towards improving resource management in cloud systems using a multi-agent framework. *International Journal of Cloud Computing*, 5 :112–133, 02 2016.

[5] A. Ali. *L'approche multi-agents pour le pilotage des systèmes complexes, Appliquée aux Systèmes du Trafic Urbain*. Thèse de doctorat, Université Blaise Pascal – Clermont II, 2010.

[6] T. Alwadan. Cloud computing and multi-agent system : Monitoring and services. *Journal of Theoretical and Applied Information Technology*, 96(9), 05 2018.

[7] S. Azaiez. *Approche dirigée par les modèles pour le développement de systèmes multi-agents*. PhD thesis, Université de Savoie, France, 2017.

[8] Y. Baghdadi. *Contribution méthodologique à la conception des systèmes d'information coopératifs*. Thèse de doctorat, Univérsitté Toulouse 1, Toulouse, France, 1997.

[9] Z. Bakraouy, A. Baina, and M. Bellafkih. System multi agents for automatic negotiation of sla in cloud computing. In A. Abraham, A. Haqiq, A. Muda, and N. Gandhi, editors, *Innovations in Bio-Inspired Computing and Applications*, pages 234–244, Cham, 2018. Springer International Publishing.

[10] A. Benaouda. *Contribution à la conception et à l'implementation d'un language de spécification formelle dédié à la e-maintenance des systèmes de production par l'approche des systèmes multi-agents*. PhD thesis, Université Ferhat Abbas, Setif, 2006.

[11] G. Bendiab. *Sécurité des applications métiers au niveau du Cloud Computing : Contrôle d'accès au niveau des APIs du Cloud Computing.* PhD thesis, Université Abdelhamid Mehri – Constantine 2, 2015.

[12] D. Boukredera and K. Adel-Aissanou. Modeling and performance analysis of cognitive radio networks using stochastic timed colored petri nets. *Wireless Personal Communications*, 112 :1659–1687, Juin 2020.

[13] R. Buyya, J. Broberg, and M. Andrzej. *Cloud computing : Principles and paradigms*, volume 87. John Wiley & Sons, Hoboken, New Jersey, 2011.

[14] B. Chaib-draa, I. Jarras, and B. Moulin. Systèmes multi-agents : principes généraux et applications. Principes et architectures des systèmes multi-agents, 2001.

[15] S. Chouchane. *Conception et réalisation d'un système multi-agents pour les enchères en ligne.* Ingénieur d'état en informatique, Université Larbi Ben M'Hidi Algérie, 2009.

[16] T. Dominico. Clouds meet agents toward intelligent cloud services. mars 2012.

[17] F. Drizi and A. Dehdouh. Modélisation et evaluation des performances de la solution cloud computing de l'entreprise icosnet. Master's thesis, Université Abderrahmane Mira, Béjaia, 2015.

[18] A. Drogoul. *Thèse de Doctorat de L'université Paris VISpécialité : INFORMATIQUE-Présentée pour obtenir le grade de Docteur de l'Université Paris VIParALEXIS DROGOULDe La Simulation Multi-Agent A LaRésolution Collective de ProblèmesUne Étude De l'Émergence De Structures D'Organisation Dans LesSystèmes Multi-Agents.* PhD thesis, l'Université Paris VI, Novembre 1993.

[19] F. Mohamed El-kabir. *Une approche basée agents pour l'allocation des ressources dans le Cloud Computing.* PhD thesis, Université Mohamed Khider-Biskra, 2015.

[20] A.A.Y Elwessabi. *Une approche basée agent mobile pour le cloud computing.* PhD thesis, Université de Batna 2 - Mustafa Ben Boulaid, 2014.

[21] K. Ruth Evangelin and V. Vidhya. Performance measures of queuing models using cloud computing. *Asian Journal of Engineering and Applied Technology*, 4 :8–11, 2015.

[22] J. Ferber. *Les Systèmes multi-agents : vers une intelligence collective.* Inter editions edition, 1995.

[23] Eclipse Fondation. About the eclipse foundation. https ://www.eclipse.org/org/.

[24] A. Gadafi. *Gestion mémoire dans une infrastructure répartie.* PhD thesis, Institut National Polytechnique de Toulouse (INP Toulouse), 2013.

[25] L. Gasser, C. Braganza, and N. Herman. *Implementing Distributed AI Systems Using MACE*, page 445–450. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[26] C. Giovanni. Jade tutorial jade programming for beginners. http ://www.iro.umontreal.ca/, 2003.

[27] P. Grange and F. Behr. Le livre blanc du cloud computing. *Paris : Syntec Informatique*, 2010.

[28] O. Gutknecht, J. Ferber, and F. Michel. Madkit : une expérience d'architecture de plate-forme multi-agent générique. pages 223–236, Saint-Jean-la-Vêtre, France, 2000. Huitième journées francophones d'Intelligence Artificielle et systèmes multi-agents, JFIADSMA.

[29] A. Harrou and A. Guermoudi. Développement d'un agent bdi pour la prise de décision d'ordonnancement dans un environnement cloud computing. Master's thesis, Université Abou Bakr Belkaid, Tlemcen, Juin 2016.

[30] D. Hilley. Cloud computing : A taxonomy of platform and infrastructure-level offerings. Technical report, Georgia Institute of Technology, 2009.

[31] I. Jarras and B. Chaib-draa. Aperçu sur les systèmes multiagents. Technical Report 2002s-67, Centre interuniversitaire de recherche en analyse des organisations CIRANO, Montréal, July 2002.

[32] J.Macedo, Z.Kokkinogenis, and R. J. F. Rossetti. An integrated framework for multi-agent traffic simulation using sumo and jade. 2013.

[33] H. Joumaa. *Performance Analysis of a multi-agents system using visualization.* Thèse de doctorat, Université Joseph-Fourier - Grenoble I, 2010.

[34] N. Khanghahi and R. Ravanmehr. Cloud computing performance evaluation : issues and challenges. *Comput*, 5(1) :29–41, 2013.

[35] M. S. Kumar and B. Balamurugan. A review on performance evaluation techniques in cloud. In *2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*, pages 19–24, 2017.

[36] KE. Kushida, J. Murray, and J. Zysman. Cloud computing : From scarcity to abundance. *Journal of Industry, Competition and Trade*, 2015.

[37] F. De la Prieta and J.M. Corchado. *Cloud Computing and Multiagent Systems, a Promising Relationship*, pages 143–161. Springer International Publishing, 2016.

[38] Y. Lebrun. *Architecture multi-agent pour la gestion d'objets tangibles et virtuels sur Tables Interactives*. PhD thesis, Université de Valenciennes et du Hainaut-Cambrésis, École doctorale Sciences pour l'Ingénieur, France, 2017.

[39] C. Levointurier. *De la nécessité d'une vision holistique du code pour l'analyse statique et la correction automatique des Applications Web*. Thèse de doctorat, Université Rennes 1, 2011.

[40] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7) :30–40, July 1994.

[41] F. Martin. *Modélisation et évaluation de performances prévisionnelles d'architectures avioniques modulaires intégrées*. PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, 1999.

[42] H. Mazyad. *Une Approche Multi-agents à Architecture P2P pour l'Apprentissage Collaboratif*. PhD thesis, Université du littoral côte d'opale, 2013.

[43] P. Mell and T. Grance. Sp 800-145. the nist definition of cloud computing. Technical report, Gaithersburg, MD, United States, 2011.

[44] I. Menken. *Cloud Computing - The Complete Cornerstone Guide to Cloud Computing Best Practices Concepts, Terms, and Techniques for Successfully Planning, Implementing ... Enterprise IT Cloud Computing Technology*. Emereo Pty Ltd, London, GBR, 2008.

[45] K. Nawsher, A. Noraziah, H. Tutut, and I. Zakira. Cloud computing : Locally subclouds instead of globally one cloud. *International Journal of Cloud Applications and Computing*, 2 :68–85, 07 2012.

[46] James J Odell. *Advanced object-oriented analysis and design using UML*, volume 12. Cambridge University Press, 1998.

[47] F. De La Prieta and J. Corchado Rodríguez. *Cloud Computing and Multiagent Systems a Promising Relationship*, pages 143–161. Universidad de Salamanca Salamanca, Castilla y León, Spain, 2016.

[48] S. Rhazlane, N. Harbi, N. Kabachi, and H. Badir. Les systèmes multi-agents au service de la sécurité des données entreposées dans le cloud, Octobre 2018.

[49] S. Rhazlane, N. Harbi, N. Kabachi, and H. Badir. Les systèmes multi agents au service de la sécurité des données entreposées dans le cloud. octobe 2018.

[50] A. Ruegg. *Processus stochastiques : avec applications aux phénomènes d'attente et de fiabilité*. Méthodes mathématiques pour l'ingénieur. Presses polytechniques romandes, 1989.

[51] S.J. Russell. Rationality and intelligence. *Artificial Intelligence*, 94(1) :57 – 77, 1997.

[52] M. Safvati and M. Sharzehei. Analytical review on queuing theory in clouds enviroments. Third National Conference on New Approaches in Computer and Electrical Engineering Young Researchers and Elite Club, 2017.

[53] O. Salem. *Modélisation algébrique et évaluation de performances des mécanismes de gestion de la qualité de service dans les réseaux Ad hoc*. PhD thesis, université Toulouse 3, 2006.

[54] K. Santhi and R. Saravanan. Performance analysis of cloud computing bulk service using queueing models. *International Journal of Applied Engineering Research*, 12(17) :6487–6492, 2017.

[55] K. Santhi and R. Saravanan. Performance analysis of cloud computing using series of queues with erlang service. *International Journal of Internet Technology and Secured Transactions*, 9 :147, 01 2019.

[56] H. Saouli. *Découverte de services web via le cloud computing à base d'agent mobile*. PhD thesis, Université Mohamed Khider Biskra, 2015.

[57] K. Sycara, N.R. Jennings, and M Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1 :7–38, March. 1998.

[58] J. Tranier. *Vers une vision intégrale des systèmes multi-agents : Contribution à l'intégration des concepts d'agent, d'environnement, d'organisation et d'institution*. PhD thesis, Université Montpellier II - Sciences et Techniques du Languedoc, 2017.

[59] E. van Ommeren, S. Duivestein, de J. Vadoss, C. Reijnen, and E. Gunvaldson. *Collaboration in the Cloud : How Cross-boundary Collaboration is Transforming Business*. Microsoft and Sogeti, 2009.

[60] B. Varghese and R. Buyya. Next generation cloud computing : New trends and research directions. *Future Generation Computer Systems*, 79 :849–861, Fevrier 2018.

[61] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos. Challenges and opportunities in edge computing. In *IEEE International Conference on Smart Cloud*, pages 20–26. IEEE, 2016.

[62] A. Willig. A short introduction to queueing theory. *Technical University Berlin, Telecommunication Networks Group*, 21, 1999.

[63] M. Wooldridge and N.R. Jennings. Intelligent agents : theory and practice. *The Knowledge Engineering Review*, 10(2) :115–152, 1995.

Résumé

Le cloud computing consiste à proposer des services informatiques sur le marché qui peuvent être consommés de manière souple et transparente via Internet. En raison de la nature dynamique et virtualisée de l'environnement du cloud computing , de la diversité des demandes des utilisateurs, le nombre accru d'utilisateurs cloud et le flux intense des demandes provoquent un engorgement dans le cloud, de ce fait quelques clients sont amenés à quitter le système avant d'être servis par défaut de disponibilité de ressources et d'un long temps d'attente. Les technologies basées sur les agents sont devenues un outil puissant pour le contrôle distribué et l'apprentissage.

Dans ce contexte, nous proposons un système multi-agents pour la gestion du coud computing. Nous avons opté pour l'approche UML et AUML pour décrire les principales fonctionnalités de notre système. Nous avons implémenté le système sous Java et Jade, via lequel les utilisateurs coud peuvent demander un service, consulter l'état de leurs demandes ou l'annuler, en plus de recevoir les informations relatives à leur demande de service en cas d'impatience, ils peuvent passer en priorité et avoir accès à un serveur VIP.

Mots clés : Cloud computing, système multi-agents, Agent, file d'attente, JADE.