

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université HADJ LAKHDAR – BATNA**  
Faculté des sciences  
Département d'Informatique

N°d'ordre :.....  
Série :.....



## Mémoire

Présenté en vue de l'obtention du diplôme de Magister en Informatique

Option : Systèmes Informatiques de Communication (SIC)

# Une approche basée agent mobile pour le cloud computing

Par :

**Mr. Elwessabi Ali Ahmed Yahya**

Soutenu le :21 /01 /2014

### Devant le jury :

Dr. BILAMI Azeddine,	Pr.	Université de Batna	Président
Dr. KAZAR Okba,	Pr.	Université de Biskra	Rapporteur
Dr. ZIDANI Abdelmadjid,	Pr.	Université de Batna	Examineur
Dr. Benmohamed Mohamed,	Pr.	Université de Constantine	Examineur

# *Remerciements*

*Je loue et remercie tout d'abord Allah.*

*Je tiens à remercier mon encadreur Mr. KAZAR Okba, Professeur à l'université Med Khider de Biskra, de m'avoir proposé le sujet du travail, pour ses précieux conseils, sa disponibilité et son soutien tout au long de mon travail.*

*Mes remerciements vont à Mr. BILAMI Azeddine, Professeur à l'université de Batna, qui a accepté de présider le jury ainsi que les membres de jury :*

*Mr. ZIDANI Abdelmadjid, Professeur à l'université de Batna ;*

*Mr. Benmohamed Mohamed, Professeur à l'université de Constantine, qui ont accepté d'être des examinateurs.*

*Je suis honoré à l'attention et au temps qu'ils ont consacré à ce travail.*

*Ce mémoire, notamment dans sa phase terminale, a été réalisé grâce à mes chers parents, pour leur patience et leur soutien moral et spirituel pendant les moments difficiles.*

*Sans leur conseil et sans leur amour rien du tout cela ne s'aurait réalisé.*

*Je ne peu oublier de remercier tout mes collègues de la promo de post-graduation de Batna pour leurs esprit de groupe qui a été présent tout au long de notre formation.*

*Je tiens énormément à remercier mes amis et mes collègues pour leur soutien pendant toute la période de ce mémoire.*

## Résumé

Le Cloud Computing est apparu comme un nouveau paradigme informatique qui vise à fournir des environnements informatiques fiables, personnalisés et dynamiques axée vers une meilleure qualité de service et disponibilité de l'infrastructure sans fardeau financier beaucoup. Cette mémoire, explore le cloud computing sous différents aspects, réviser l'état de réalisations artistiques ainsi que ses enjeux et explore le champ potentiel de recherche. Cette mémoire analyse également les possibilités de fusion des agents mobiles dans le cloud computing, puisque ces deux technologies sont prometteuses et commercialement utile donc l'idée est de résoudre les problèmes liés à le cloud computing grâce à la technologie d'agent mobile.

Mots clés: système multi-agent, agent mobile, le cloud computing.

## Abstract

The Cloud Computing has emerged as a new computing paradigm which aims to provide reliable, customized, dynamic computing environments focused towards better quality of service and IT infrastructure availability without much financial burden. This memory explores the cloud computing epitome from various aspects, reviews state of art implementations as well as its inherent challenges and explores the potential scope for research. This work also analyses the possibilities of amalgamating mobile agents in cloud computing, since both these technologies are promising and commercially useful thus the idea is to resolve challenges pertaining in cloud computing by harnessing mobile agent technology.

**Key words:** System multi-agent, mobile agent, cloud computing.

## ملخص

برزت الحوسبة السحابية كنموذج الحوسبة الجديدة التي تهدف إلى توفير موثوق بها، حسب الطلب، وبيئات الحوسبة ديناميكية تركز نحو تحسين نوعية الخدمات والبنية التحتية لتكنولوجيا المعلومات توفر دون عبء مالي من ذلك بكثير. هذه الورقة، يستكشف مثال الحوسبة السحابية من مختلف الجوانب، استعراض حالة التنفيذ الفن فضلا عن التحديات الكامنة ويستكشف النطاق المحتمل للبحوث. يحل هذا العمل أيضا إمكانيات دمج وكلاء المحمول في الحوسبة السحابية، لأن كلا من هذه التقنيات واعدة ومفيدة تجاريا وبالتالي فإن الفكرة هي أن حل التحديات المتعلقة في مجال الحوسبة السحابية من خلال تسخير التكنولوجيا وكيل المتنقلة.

الكلمات الأساسية: نظام متعدد وكيل , وكيل المحمول والحوسبة السحابية

# *Tables des matières*

Introduction Générale.....	1
Chapitre I les agents mobiles.....	3
I.1. Introduction .....	3
I.2. Définition.....	3
I.3. Caractéristiques.....	4
I.3.1. L'autonomie .....	4
I.3.2. La situation.....	5
I.3.3. la flexibilité.....	5
I.4. Classification des agents.....	5
I.4.1. Les agents cognitifs.....	5
I.4.2. Les agents réactifs.....	6
I.5. Agent et objet .....	7
I.6. Les systèmes Multi-Agents.....	8
I.6.1. Définition .....	8
I.6.2. Caractéristiques des systèmes Multi-Agents.....	10
I.6.3. Interaction dans un système Multi-Agents.....	10
I.6.3.1. Définition de l'interaction .....	10
I.6.3.2. La coopération .....	10
I.6.3.3. La négociation.....	11
I.6.3.4. La coordination.....	11
I.6.4. Communication dans les systèmes Multi-Agents.....	12
I.6.5. La simulation Multi-Agents.....	14
I.6.5.1. Méthodologie de simulation Multi-Agents [Dro 03].....	14
I.6.5.2. Caractéristiques de la simulation Multi-Agents .....	14
I.6.6. Domaines d'utilisation des systèmes Multi-Agents .....	15
I.7. Les agents mobiles .....	15
I.7.1. Introduction .....	15
I.7.2. Les caractéristiques d'un Agent mobile .....	16
I.7.3. Application a base d'agents mobiles .....	18
I.7.3.1. Les normes .....	18
I.7.3.2. Autre approche .....	19

I.7.4.	Exemples de plateformes d'agents mobiles .....	20
I.8.	La conclusion .....	21
II.	Chapitre II Cloud computing : état de l'art.....	22
II.1.	Introduction .....	22
II.2.	HISTORIQUE .....	22
II.2.1.	L'INFORMATIQUE UTILITAIRE DE JOHN MCCARTHY .....	22
II.2.2.	LES SERVICES BUREAU .....	23
II.2.3.	LES APPLICATION SERVICE PROVIDERS .....	23
II.2.4.	LA VIRTUALISATION .....	23
II.3.	DEFINITIONS.....	24
II.4.	Architecteurs du cloud computing.....	28
II.5.	LES DIFFERENTES COUCHES.....	30
II.5.1.	L'INFRASTRUCTURE COMME UN SERVICE (IAAS) .....	31
II.5.2.	PLATEFORME COMME UN SERVICE (PAAS) .....	32
II.5.3.	LE LOGICIEL COMME UN SERVICE (SAAS) .....	33
II.6.	Types de Cloud Computing .....	34
II.6.1.	Le cloud prive.....	35
II.6.2.	Le cloud public.....	36
II.6.3.	Le cloud hybride .....	36
II.7.	DANS L'ERE DU CLOUD COMPUTING.....	36
II.7.1.	POUR QUEL TYPE D'APPLICATION ?.....	37
II.7.2.	BENEFICES / AVANTAGES .....	40
II.7.3.	LE MARCHE DU CLOUD COMPUTING .....	41
II.7.3.1.	LES ACTEURS DU CLOUD .....	42
II.7.3.1.1.	EDITEURS.....	42
II.7.3.1.2.	FOURNISSEURS .....	43
II.7.3.1.3.	PURE PLAYER.....	43
II.7.3.1.4.	OPEN SOURCE .....	43
II.8.	LE PRIX DU CLOUD.....	44
II.9.	Exemple des Solutions IaaS open source de Cloud Computing .....	45
II.9.1.	Eucalyptus.....	45
II.9.2.	OpenNebula.....	46
II.9.3.	Nimbus.....	47
II.9.4.	Xen Cloud Platform.....	49
II.9.5.	AbiCloud .....	50
II.10.	La conclusion .....	51
III.	Chapitre III la securite du cloud computing .....	52

<b>III.1.</b>	<b>INTRODUCTION .....</b>	<b>52</b>
<b>III.1.1.</b>	<b>Caractéristiques et obstacles.....</b>	<b>53</b>
<b>III.2.</b>	<b>SÉCURITÉ DANS UN ENVIRONNEMENT CLOUD .....</b>	<b>55</b>
<b>III.2.1.</b>	<b>Protection des données .....</b>	<b>56</b>
<b>III.2.2.</b>	<b>Sécurité des applications .....</b>	<b>56</b>
<b>III.2.3.</b>	<b>Sécurité réseau .....</b>	<b>57</b>
<b>III.2.4.</b>	<b>Sécurité de la virtualisation .....</b>	<b>57</b>
<b>III.2.5.</b>	<b>Gestion des identités .....</b>	<b>57</b>
<b>III.3.</b>	<b>TRAVAUX CONNEXES A TROUVER DES SOLUTIONS DANS LE CLOUD COMPUTING.....</b>	<b>58</b>
<b>III.3.1.</b>	<b>La Sécurité de base.....</b>	<b>59</b>
<b>III.3.2.</b>	<b>Sécurité des données.....</b>	<b>60</b>
<b>III.3.3.</b>	<b>Protection du réseau.....</b>	<b>60</b>
<b>III.3.4.</b>	<b>Protection contre les attaques aux différents niveaux.....</b>	<b>60</b>
<b>III.3.5.</b>	<b>Présentation de Priyank Singh et Ranjita Singh et Mukul Manmohan</b>	<b>61</b>
<b>III.3.5.1.</b>	<b>Idées de base.....</b>	<b>61</b>
<b>III.3.5.2.</b>	<b>Architecture du système.....</b>	<b>62</b>
<b>III.3.5.3.</b>	<b>METHODE PROPOSEE.....</b>	<b>63</b>
<b>III.4.</b>	<b>LA CONCLUSION.....</b>	<b>66</b>
<b>IV.</b>	<b>Chapitre IV Proposition d'un environnement decloud computing .....</b>	<b>67</b>
<b>IV.1.</b>	<b>Introduction .....</b>	<b>67</b>
<b>IV.2.</b>	<b>Objectif et motivation du travail.....</b>	<b>67</b>
<b>IV.2.1.</b>	<b>Objectif .....</b>	<b>67</b>
<b>IV.2.2.</b>	<b>motivation.....</b>	<b>68</b>
<b>IV.3.</b>	<b>Description de l'architecture proposée.....</b>	<b>69</b>
<b>IV.4.</b>	<b>L'architecteur proposée.....</b>	<b>72</b>
<b>IV.4.1.</b>	<b>l'architecteur générale .....</b>	<b>72</b>
<b>IV.4.1.1.</b>	<b>Couche Interface.....</b>	<b>73</b>
<b>IV.4.1.2.</b>	<b>Couche Médiation.....</b>	<b>73</b>
<b>IV.4.1.3.</b>	<b>Couche Agents Mobiles .....</b>	<b>74</b>
<b>IV.4.1.4.</b>	<b>Couche cloud computing.....</b>	<b>75</b>
<b>IV.5.</b>	<b>Spécification des agents.....</b>	<b>76</b>
<b>IV.5.1.</b>	<b>Agent Interface .....</b>	<b>76</b>
<b>IV.5.2.</b>	<b>Agent médiateur .....</b>	<b>78</b>
<b>IV.5.3.</b>	<b>L'agent analyseur .....</b>	<b>79</b>
<b>IV.5.4.</b>	<b>Agent transfère.....</b>	<b>80</b>
<b>IV.5.5.</b>	<b>Contrôleur des tâches.....</b>	<b>81</b>

IV.5.6.	Agent sécurité.....	82
IV.5.7.	Agent exécuteur .....	83
IV.6.	Exemple de scénario .....	85
IV.7.	Fonctionnement du système.....	86
IV.7.1.	La migration des agents .....	87
IV.7.2.	La communication entre les agents.....	87
IV.7.3.	L'interaction entre les agents.....	88
IV.8.	Les diagrammes en AUML.....	88
IV.8.1.	Diagramme de cas d'utilisation .....	88
IV.8.2.	Les diagrammes des séquences.....	89
IV.9.	La Conclusion .....	93
V.4.1.	Chapitre V Etude De Cas Et Implémentation .....	94
V.1.	Introduction .....	94
V.2.	Environnement de développement.....	94
V.2.1.	Choix du langage de programmation .....	94
V.2.2.	La plateforme J2ME .....	97
V.2.2.1.	L'architecture J2ME se découpe donc en plusieurs couches :.....	97
V.2.2.2.	Les configurations.....	98
V.2.2.2.1.	La configuration CDC (Connected Device Configuration).....	98
V.2.2.2.2.	CLDC (Connected Limited Device Configuration) .....	98
V.2.2.3.	Les profiles .....	99
V.2.2.3.1.	Foundation profile .....	99
V.2.2.3.2.	Le Personal Basis Profile (PBP) .....	99
V.2.2.3.3.	Le Personal Profile (PP).....	100
V.2.3.	Description générale de la plate form JADE.....	100
V.2.4.	NetBeans IDE 7.0.....	100
V.2.5.	La plate-forme d'agent mobile .....	101
V.2.5.1.	Description générale de la plate-forme Aglets .....	101
V.2.5.1.1.	Définition .....	102
V.2.5.1.2.	Les éléments de base d'Aglets.....	102
V.2.5.1.3.	Cycle de vie d'un Aglet .....	103
V.2.5.1.4.	Les Serveurs d'Aglets .....	104
V.3.	Présentation de l'étude de cas.....	104
V.4.	Description général de l'application .....	105
V.5.	L'implémentation des agents.....	107
V.5.1.	Agent interface.....	107
V.5.2.	Agent médiateur .....	107

<b>V.5.3.</b>	<b>L'agent Recherche.....</b>	<b>108</b>
<b>V.5.4.</b>	<b>Agent Analyseur .....</b>	<b>109</b>
<b>V.5.5.</b>	<b>agent Ressource.....</b>	<b>109</b>
<b>V.5.6.</b>	<b>Agent contrôleur .....</b>	<b>110</b>
<b>V.5.7.</b>	<b>Agent sécurité.....</b>	<b>110</b>
<b>V.6.</b>	<b>Présentation de Résultat de notre système Résultats .....</b>	<b>111</b>
<b>V.7.</b>	<b>La conclusion .....</b>	<b>118</b>
	<b>Conclusion Générale .....</b>	<b>119</b>
	<b>Bibliographie.....</b>	<b>121</b>

# Liste des figures

## Chapitre I Les Agents Mobiles et leurs spécificités

Figure 1.1: structure général d'un agent cognitif.....	6
Figure 1.2: structure général d'un agent réactif.....	7
Figure 1.3: Architecture contrôle centralisé d'un SMA.....	13
Figure 1.4: Architecture distribuée d'un SMA.....	13

## Chapitre II cloud computing : état de l'art

Figure 2.1: le niveau de charge d'une application1.....	26
Figure 2.2: le niveau de charge d'une application2.....	27
Figure 2.3: l'architecture d'un environnement de cloud computing.....	30
Figure 2.4: Les différentes couches du cloud computing.....	30
Figure 2.5: les types du cloud computing.....	35
Figure 2.6: l'architecteur d'Eucalyptus.....	46
Figure 2.7: l'architecteur d'OpenNebula.....	47
Figure 2.8: l'architecteur de Nimbus.....	48
Figure 2.9: l'architecteur de Xen Cloud Platform.....	49
Figure 2.10: l'architecteur d'AbiCloud.....	50

## Chapitre III Approches existantes à base d'agent mobile pour le cloud computing

Figure 3.1 : Classification des problèmes de sécurité.....	56
Figure 3.2: L'architecture de machine virtuelle Xen.....	61
Figure 3.3 : Les emplacements des MB2 et MB3 dans le système.....	63
Figure 3.4 : L'authentification.....	64
Figure 3.5 : vérifier l'intégrité.....	64

## Chapitre IV Proposition d'un environnement de cloud computing

Figure 4.1 Un environnement de cloud computing.....	69
Figure 4.2 Représentation du cloud computing basé agent mobile.....	71
Figure 4.3 Représentation des couches du notre système.....	72
Figure 4.4 Représentation de la couche interface.....	73
Figure 4.5 Représentation de la couche médiation.....	74
Figure 4.6 Représentation de la couche agent mobile.....	75
Figure 4.7 Représentation de la couche cloud computing.....	75
Figure 4.8 Architecture de l'agent interface.....	77

Figure 4.9 Architecture de l'agent médiateur.....	78
Figure 4.10 Architecture de l'agent analyseur.....	79
Figure 4.11 Architecture de l'agent mobile.....	80
Figure 4.12 Architecture de l'agent contrôleur des tâches. ....	81
Figure 4.13 Architecture de l'agent sécurité. ....	82
Figure 4.14 Architecture de l'agent exécuter.....	83
Figure 4.15 Représentation une architecture détaille. ....	85
Figure 4.16 Diagramme de cas d'utilisation de système ....	89
Figure 4.17 Diagramme de séquence « Inscription serveur».....	90
Figure 4.18 Diagramme de séquence « authentification» ....	91
Figure 4.19 Diagramme de séquence « lancer requêtes» ....	92
Figure 4.20 Diagramme de séquence « lancer services» ....	93

## Chapitre V Etude de cas et Implémentation

Figure 5.1 : l'IDE netbeanse7.0 .....	101
Figure 5.2: Relation entre un Aglet et son Proxy.Contexte .....	102
Figure 5. 3: contexte de l'aglet.....	102
Figure 5.4 : L'architecture logicielle du prototype.....	106
Figure 5.5: Pseudo code de l'agent interface.....	107
Figure 5.6: Pseudo code de l'agent médiateur. ....	108
Figure 5.7: Pseudo code de l'agent mobile. ....	108
Figure 5.8: Pseudo code de l'agent analyseur. ....	109
Figure 5.9: Pseudo code de l'agent ressource.....	109
Figure 5.10: Pseudo code de l'agent contrôleur.....	110
Figure 5.11: Pseudo code de l'agent sécurité. ....	111
Figure 5.12: lancement de l'agent interface client.....	111
Figure 5.13: lancement de l'agent médiateur.....	112
Figure 5.14: lancement de l'agent analyseur.....	112
Figure 5.15: lancement de l'agent contrôleur. ....	113
Figure 5.16: lancement de l'agent sécurité.....	113
Figure 5.17: lancement de l'agent ressource. ....	114
Figure 5.18 : Illustre l'interface de lancement d'une requête.....	114
Figure 5.19 : arriver une requête a l'agent médiateur .....	115
Figure V.21 : lancer un service après vérification la sécurité.....	116
Figure V.22 : le résultat à partir de l'agent ressource.....	116
Figure V.23 : arriver le résultat vers l'agent analyseur pour analyser .....	117
Figure V.24 : arriver le résultat vers le client .....	117





# Introduction Générale

Le cloud computing fournit des services élastiques, de haute performance et de stockage de données évolutives à un large nombre croissant et quotidien des utilisateurs.

Cloud computing agrandi le domaine des systèmes informatiques distribués en offrant des services Internet de pointe qui complète des fonctionnalités de l'informatique distribuée fournies par le Web, grille réseaux et peer-to-peer. En fait, les systèmes de cloud computing offrent l'infrastructure à grande échelle pour le calcul de haute performance qui s'adapte dynamiquement à l'utilisateur et les besoin de l'application.

Aujourd'hui le cloud computing est principalement utilisé pour le traitement des charges de travail informatiques très intensives et pour fournir de très grandes installations de stockage de données, ces deux objectifs sont combinés avec le troisième but de réduire potentiellement les coûts de gestion et d'utilisation.

Dans le même temps, les systèmes multi-agents (SMA) représentent un autre paradigme de l'informatique distribuée basée sur de multiples agents en interaction qui sont capables d'un comportement intelligent, les systèmes multi-agents sont souvent utilisés pour résoudre des problèmes en utilisant une approche décentralisée où plusieurs agents contribuent à la solution en coopérant l'un avec l'autre.

La technologie d'agent mobile a suscité un intérêt renouvelé dans le cadre d'environnements qui fournissent des ressources informatiques distribuées aux utilisateurs. Dans de tels environnements, les ressources sont disponibles sur différentes machines, et l'application est exécutée sur toute machine qui est en mesure de fournir les services demandées. Ceci, , peut exiger que l'application (son état et code) soit déplacée vers les services où elle peut être exécutée. C'est exactement le problème auquel la technologie d'agent mobile tente de répondre.

Bien que plusieurs différences existent entre Cloud computing et de systèmes multi-agents, ils sont deux modèles distribués de calcul, donc plusieurs problèmes communs peuvent être identifiés et plusieurs avantages peuvent être obtenus par l'intégration de l'utilisation de systèmes de cloud computing et multi-agents.

Les activités de recherche dans le domaine du cloud computing sont principalement axées sur l'utilisation efficace de l'infrastructure informatique, la prestation de services de stockage de données, les techniques de virtualisation évolutifs, et l'efficacité énergétique.

Nous pouvons dire que dans le cloud computing le principal objectif de la recherche est sur le l'utilisation efficace de l'infrastructure à coûts réduits.

Au contraire, les activités de recherche dans le domaine des agents sont plus porté sur les aspects intelligents d'agents et sur leur utilisation pour développer des applications complexes.

Malgré ces différences, le cloud computing et multi-agents systèmes partagent plusieurs points communs et des sujets de recherche dans ces deux domaines ont plusieurs chevauchements qui doivent être étudiées.

En particulier, le cloud computing peut offrir une très puissante, prévisible, fiable et évolutive infrastructure pour l'exécution de systèmes multi-agents qui est mise en œuvre complexe d'applications basées sur des agents

De l'autre côté, les agents mobile et statique peuvent être utilisés comme éléments de base pour la mise en œuvre de l'intelligence dans les systèmes de cloud computing ce qui les rend plus adaptable, flexible et autonome en ressource gestion, la fourniture de services et la gestion à grande échelle applications.

Dans le cadre de ce mémoire, nous nous sommes fixées comme objectif la réalisation d'une architecture de cloud computing , il s'agit d'une application basée sur les agents mobiles à travers l'architecture proposée.

Notre travail consiste à comprendre le fonctionnement de deux architectures à savoir le paradigme d'agents mobile ainsi que le concept du cloud computing.

Pour cela, nous avons entrepris notre étude selon les cinq chapitres suivants :

*Chapitre(01)* : Il est consacré à une étude sur la technologie d'agent mobile et les systèmes multi agents (généralité, définition, cycle de vie, domaines d'application et avantages, les différentes plates-formes d'agents existantes ...).

*Chapitre(02)* : nous étudierons quelques notions sur le cloud computing (architecture, principe, caractéristiques, .....).

*Chapitre(03)* : Ce chapitre est consacré à la présentation de quelques architectures basées agent mobile, qui sont conçus pour les cloud computing.

*Chapitre(04)* : nous entamerons la conception d'architecture conceptuelle proposée, ensuite nous détaillerons notre architecture basée agents mobiles pour la cloud computing.

*Chapitre(05)* : Enfin, ce chapitre s'intéressera à la réalisation de notre proposition d'architecture de système cloud computing.

# Chapitre I

---

## Les Agents Mobiles

---

### I.1. Introduction

Le domaine des systèmes Multi-agents est un domaine relativement jeune. Il est apparu pendant les années 80 suite à l'évolution considérable des applications informatiques qui sont devenues de plus en plus distribuées et diffusées dans de multiples objets et fonctionnalités qui sont amenées à coopérer.

De plus la taille, la complexité et l'évolutivité croissante de ces applications font qu'une vision centralisée rigide a atteint ses limites.

Une autre motivation de leur l'apparition était la nécessité de trouver des techniques de modélisation et de simulation performantes dans le domaine des sciences du vivant où l'utilisation des équations différentielles pour décrire l'évolution d'un écosystème est difficile.

L'approche Multi-agents est le fruit des recherches en intelligence artificielle. En effet, l'évolution des domaines d'application de l'IA s'est étendue aux problèmes complexes et hétérogènes. L'approche classique de l'IA qui se base sur une centralisation de l'expertise dans une entité unique a montré les limites.

Plusieurs recherches ont abouti à la naissance de la deuxième génération des systèmes, c'est l'approche de l'intelligence artificielle distribuée.

Elle a pour objectif de remédier aux insuffisances de l'IA classique en introduisant l'idée de partager l'expertise sur un ensemble d'entités qui travaillent et agissent dans un environnement, possédant chacune des capacités de résolution limitées du problème et interagissant avec les autres pour atteindre un certain objectif, d'où l'apparition des nouveaux concepts dans l'IA tel que la coopération, la coordination, la négociation et l'émergence.

### I.2. Définition

Etymologiquement, le mot agent est inspiré du verbe agir, donc littéralement, un agent est une entité qui agit [Dro 93].

La notion d'agent comme tous les concepts fondamentaux est relativement vague. On peut distinguer plusieurs définitions correspondant chacune à un courant de recherche dans le domaine de ce qui touche au nébuleux agent.

La définition la plus générale et la plus fréquemment utilisée est celle de Jacques Ferber 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"* [Fer 95].

A partir de cette définition, on comprend que l'agent est une entité située dans un environnement comprenant des objets passifs, et d'autres agents où chacun cherche à satisfaire ses propres objectifs, pour cela il exécute un ensemble de tâches qu'il voit adéquates suivant sa situation courantes et suivant son répertoire d'action et le but qu'il désire atteindre.

Le processus de fonctionnement de l'agent comprend donc trois phases successives :

- une phase de perception qui permet d'élaborer une idée sur l'état actuel de l'environnement via un ensemble de capteurs.
- Une phase de délibération qui permet de décider quelle action à exécuter suivant l'état de l'environnement et l'état interne de l'agent, c'est à ce niveau que le comportement de l'agent est décrit donc c'est la phase la plus importante.
- Et finalement la phase d'action c'est l'exécution de l'action par l'actionneur correspondant, l'exécution qui va apporter des modifications sur l'état de l'environnement et sur l'état de l'agent.

### **I.3. Caractéristiques**

Les différentes caractéristiques de l'entité agent peuvent être résumées dans la définition suivante : « un agent est un système informatique situé dans un environnement, et qui agit de façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu » [Jar 03].

#### **I.3.1. L'autonomie**

C'est la caractéristique la plus intéressante de l'agent. En effet, l'agent est capable d'agir sans aucune intervention et il contrôle son comportement ainsi que son état interne, autrement dit, l'agent est dit autonome dans le sens où le concepteur du système ne pilote pas son comportement c'est-à-dire l'agent décide lui même quelle action à entreprendre parmi celles qui sont possibles.

L'autonomie de l'agent rend son comportement indéterministe, c'est-à-dire aucun autre agent ne peut observer ou modifier sa partie décisionnelle et donc prédire ses comportements. L'agent choisit son comportement en fonction de ces perceptions, il se comportera différemment en fonction de ces dernières. L'agent peut être vu comme une boîte noire, on ne peut observer que les comportements externes.

### **I.3.2. La situation**

Les agents sont situés dans un environnement contenant également des entités passives, manipulées par les agents (par exemple : des ressources, des données, des objets physiques...) et communément appelées objets.

L'agent est capable d'agir sur son environnement qu'il peut percevoir grâce à ses entrées sensorielles.

L'agent doit s'adapter sans cesse aux changements de son entourage qui pourraient modifier de façon pertinente son comportement à tous les niveaux (objectif, plan, action...etc.).

### **I.3.3. la flexibilité**

L'agent est capable de réaliser des actions d'une façon autonome et réflexible afin d'atteindre les objectifs qui lui ont été fixés.

La réflexibilité signifie dans ce cas:

- la réactivité : l'agent est capable de percevoir les changements dans son environnement, et doit élaborer une réponse dans les temps requis (changer son état interne, exécuter une tâche particulière...etc.).
- la pro-activité : l'agent doit exhiber un comportement proactif, opportuniste et orienté objectifs, c'est-à-dire il n'agit pas uniquement en réponse à son environnement mais, il est également capable de prendre l'initiative au "bon" moment.
- la sociabilité : l'agent doit être capable d'interagir avec les autres agents (logiciels et humains) quand la situation l'exige afin de compléter ses tâches ou aider ces agents à accomplir les leurs.

## **I.4. Classification des agents**

Les agents sont généralement classés selon deux courants de pensée : l'école cognitive et l'école réactive.

### **I.4.1. Les agents cognitifs**

Ils sont parfois dits "intentionnels", leur caractéristique fondamentale est la volonté de communiquer et de coopérer et la mémorisation du 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 possèdera 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.

Le système est régi par des règles sociales prédéfinies (c'est-à-dire lors de situations conflictuelles les agents seront amenés à négocier) [Bla 05].

Un agent cognitif raisonne sur ses croyances qui représentent sa compréhension du monde dans lequel il évolue. Il raisonne aussi sur ses désirs et intentions en relation avec ses croyances et capacités afin de prendre des décisions auxquelles sont associés les plans qu'il va accomplir pour agir dans le monde [Jar 02].

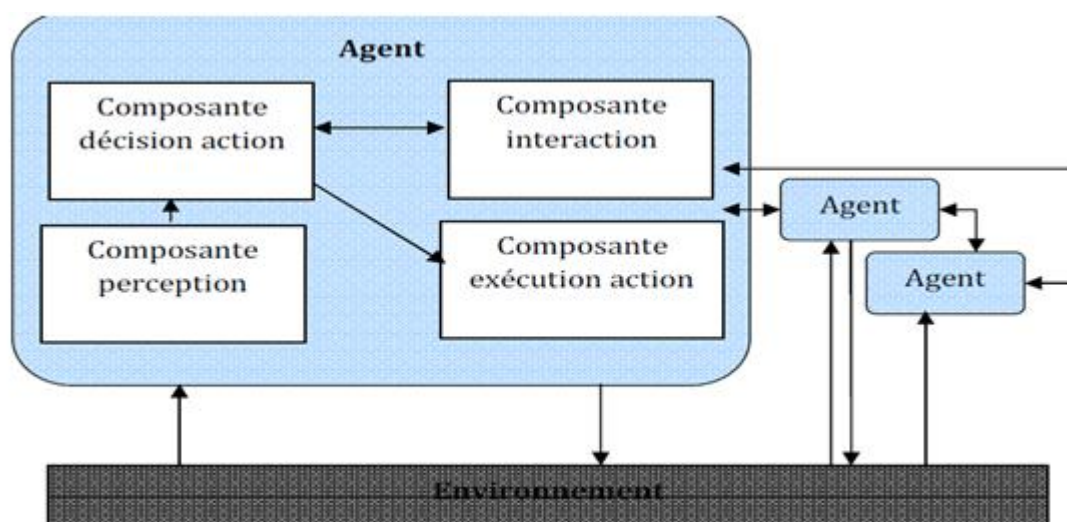


Figure 1.1: structure général d'un agent cognitif.

#### I.4.2. Les agents réactifs

L'école « réactive » prétend au contraire qu'il n'est pas nécessaire que les agents soient intelligents individuellement pour que le système ait un comportement global intelligent. Les agents réactifs sont des agents simples qualifiés de non intelligents, ils répondent d'une façon opportune aux modifications de leurs environnements résultants des stimuli externes (leurs actions sont provoquées et non pas choisies), ils agissent en fonction de ces dernières sans nécessité de compréhension de leurs univers ni de leurs buts. Les sociétés d'agents réactifs sont caractérisées par le nombre important des agents qui sont capables ensemble de produire des actions évoluées qu'on appelle l'émergence de l'intelligence, mais dont

les individus pris séparément ne possèdent qu'une représentation faible de leur environnement et des buts globaux [Bla 05].

En fait, il n'existe pas réellement de frontière entre réactif et cognitif qui nous permet de déterminer la nature d'un agent (réactif ou cognitif). Il serait plus juste de dire qu'un agent est plus réactif ou bien plus cognitif qu'un autre, c'est une caractéristique relative. Considérons une fourmilière comme un système Multi-agents. Une fourmi par exemple est un agent plutôt réactif, elle a très peu d'intelligence, et son comportement consiste à réagir à des stimuli. Elle peut néanmoins communiquer avec les autres agents par des messages très simples (les phénomènes laissés dans l'environnement).

A l'opposé, considérons comme un système Multi-agents une entreprise. Les employés sont vus comme des agents cognitifs. En effet ceux-ci possèdent des grandes capacités cognitives et ne se contentent pas de réagir à des stimuli : ils sont capables de faire des planifications, d'appliquer des stratégies complexes et leurs actions dépendent fortement de leurs éducations et de leurs expériences passées. On parlera donc d'agents cognitifs.

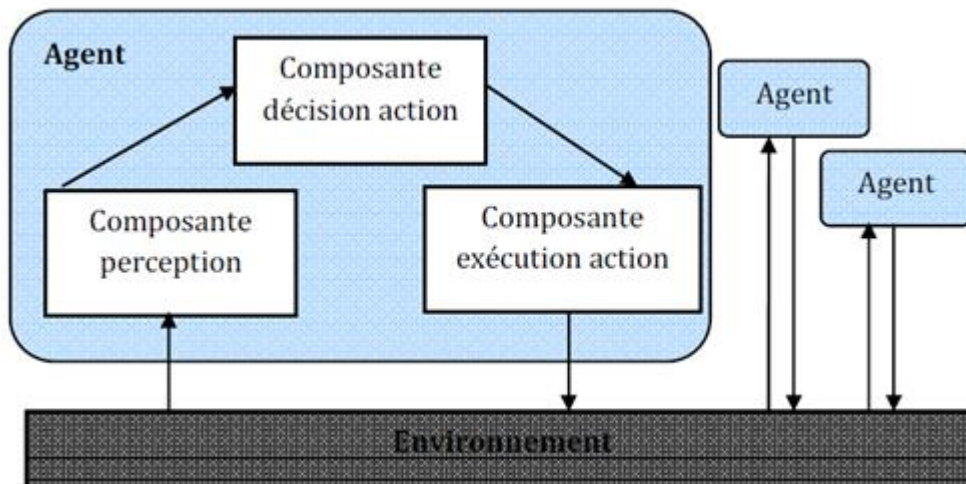


Figure 2: structure général d'un agent réactif.

## I.5. Agent et objet

Selon la nature du problème à traiter on peut déterminer l'approche à utiliser Agent ou objet. Ces deux concepts même s'ils se croisent en plusieurs points communs au niveau de la structure (avoir un état interne, un ensemble de méthodes modifiant cet état et une capacité de communication). Plusieurs autres points les différencient concernant le mécanisme d'exécution. Les objets n'ont ni but ni recherche de satisfaction et le mécanisme de communication se résume à un simple appel de méthodes. La principale différence entre objet et agent revient à l'autonomie de ce dernier.

En effet, un objet est défini par un ensemble de services (ses méthodes) qu'il ne peut refuser aucune demande d'exécution si un autre objet le lui demande. On dit que l'objet n'a pas de contrôle sur son comportement.

Contrairement aux objets, les agents peuvent recevoir des messages qui ne sont pas uniquement des demandes d'exécutions mais des informations, des plans....

De même, les agents tentent de satisfaire des objectifs ce qui nécessite plus d'autonomie pendant l'interaction avec les autres. En effet, un agent peut accepter d'effectuer un travail demandé par un autre, comme il peut le refuser s'il est occupé ou ne sais pas le faire par exemple ou bien négocier son exécution s'il n'est pas dans son meilleur intérêt.

Une autre distinction importante est le comportement flexible (réactivité, pro-activité et sociabilité) de l'agent qui est complètement absent pour l'objet.

Effectivement, si on souhaite représenter une bibliothèque on va utiliser une conception orienté objet en considérant chaque livre comme un objet qui contient une structure contenant certaines variables (le titre par exemple) et un ensemble de méthodes permettant leur manipulation, il est très clair qu'un livre n'est pas une entité autonome ni réactive ni sociale : un livre n'est pas un agent.

Cependant, si on désire simuler les déplacements des voyageurs dans une gare de trains, on va utiliser l'approche agent où chaque voyageur (agent) a un point d'entrée à la gare (en sortant d'un train par exemple) et un point de sortie qui représente son but (prendre un train ou quitter la gare) ainsi, chaque agent possède un certain nombre de paramètres (son degré de stress dans une foule de personnes) qui influent sur son comportement, ses interactions et ses déplacements dans la gare qui dépendent également du nombre des trains ou l'architecture de la gare...etc.

## **I.6. Les systèmes Multi-Agents**

### **I.6.1. Définition**

Usuellement, un système est un ensemble organisé d'éléments concourant à la réalisation d'une tâche donnée. En suivant cette définition on peut définir immédiatement le système Multi Agents comme étant un ensemble organisé d'agents se chargeant de réaliser un but commun.

Les systèmes Multi Agents sont des systèmes distribués conçus et implantés idéalement comme un ensemble d'agents interagissant, le plus souvent, selon des modes de coopération, de concurrence et de coexistence [Jar 03].

Jacques Ferber [Fer 95] définit les systèmes Multi-Agents comme des systèmes composés des éléments suivants:

- Un environnement: c'est-à-dire un espace disposant généralement d'une métrique.
- Un ensemble d'objets: situés dans cet environnement cela signifie que pour tout objet il est possible à un moment donné d'associer une position. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être créés, détruits, manipulés et perçus par les agents.
- Un ensemble d'agents: ce sont des objets particuliers, ils représentent les entités actives du système.
- Un ensemble de relations: qui unissent les objets entre eux.
- Un ensemble d'opérations: ce sont les différents types de manipulation qu'appliquent les agents sur les objets du système et qui sont en générale: perception production, consommation, transformation...etc.
- Un ensemble d'opérateurs: chargés de représenter l'application de ces opérations et la création du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

Ces systèmes possèdent les avantages traditionnels de la résolution distribuée comme la modularité, la vitesse, et la fiabilité (due à la redondance), Ils héritent aussi des bénéfices envisageables de l'intelligence artificielle (IA) comme:

- Le traitement symbolique (aux niveaux des connaissances).
- La facilité de maintenance.
- La réutilisation et la portabilité.

Mais surtout, ils ont l'avantage de faire intervenir des schémas d'interaction sophistiqués qui incluent:

- La coopération (travailler ensemble à la résolution d'un but commun).
- La coordination (organiser la résolution d'un problème de telle sorte que les interactions nuisibles soient évitées ou que les interactions bénéfiques soient exploitées).
- La négociation (parvenir à un accord acceptable pour toutes les parties concernées).

Bien que les systèmes Multi Agents offrent de nombreux avantages, ils doivent aussi relever beaucoup de défis et dépasser les problèmes inhérents à la conception et à l'implémentation qui sont nombreux, on peut citer par exemple:

- Comment reconnaître et réconcilier les points de vue disparates et les intentions conflictuelles dans un ensemble d'agents essayant de coordonner leurs actions?
- Comment éviter ou diminuer un comportement nuisible du système global, comme les comportements chaotiques ou oscillatoires?

## **I.6.2. Caractéristiques des systèmes Multi-Agents**

Un système Multi Agents possède généralement les caractéristiques suivantes:

- Il n'y a pas de contrôle global du système.
- Les données sont décentralisées.
- Le calcul est asynchrone.
- Chaque agent a des informations ou des capacités de résolution limitées de problème, ainsi chaque agent a un point de vue partiel.

## **I.6.3. Interaction dans un système Multi-Agents**

### **I.6.3.1. Définition de l'interaction**

Un système Multi Agents est un système intelligent constitué d'une multitude d'agents, chacun d'eux est chargé de réaliser ses propres buts afin d'atteindre le but global du système. La cohérence du système et son intelligence ne viennent pas de l'intelligence de ses agents, mais de leurs interactions.

Jacques Ferber 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*"[Fer 95].

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.

### **I.6.3.2. La coopération**

Parmi les caractéristiques fondamentales d'un système Multi Agents, on trouve la distribution du travail entre les différents agents qui le constituent, ou chacun d'eux se charge de réaliser ses propres buts (qui sont un sous-problème du problème global).

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 (ou bien il ne dispose pas des moyens nécessaires) donc il aura besoin de l'intervention d'un autre agent du système qui va l'aider à résoudre le problème, c'est-à-dire qu'il y'a une coopération pour faire évoluer le système vers ses objectifs.

La coopération consiste, donc, à faire participer plusieurs agents pour satisfaire un but individuel ou commun.

Jacques Ferber définit la coopération entre plusieurs agents comme "*une situation dans laquelle, soit l'ajout d'un nouvel agent permet d'accroître les performances du groupe, soit l'action des agents sert à éviter ou à résoudre des conflits potentiels ou actuels*" [Fer 95].

La situation de coopération apparaît quand les actions de chaque agent satisfont au moins l'une des conditions suivantes:

- Les agents ont un but en commun et leurs actions tendent à réaliser ce but.
- Les agents effectuent les actions qui réalisent non seulement leurs propres buts mais aussi ceux des autres.

### **I.6.3.3. La négociation**

Les buts des agents dans un système Multi Agents peuvent être incompatibles et leurs demandes sont parfois contradictoires: dans une situation de jeu, chacun cherche à être gagnant mais la réalisation de ce but pour un des joueurs rend impossible la réalisation du but des autres joueurs; Les ressources dont les agents ont besoin peuvent être rares et l'utilisation d'une ressource par un des agents peut empêcher un autre agent de réaliser son but...etc. Ces situations et d'autres empêchent l'avancement désirable du système. il faut, donc, trouver un moyen qui permette à chaque agent de poursuivre son travail: les agents doivent négocier la solution.

On définit la négociation comme le processus d'améliorer les accords (en réduisant les inconsistances et l'incertitude) sur des points de vue communs ou des plans d'action grâce à l'échange structuré d'informations pertinentes.

Contrairement à la coopération qui suppose la sociabilité des agents, la négociation correspond à la collaboration entre agents en univers compétitif. La négociation est un mécanisme puissant pour gérer les dépendances inter agents ou le processus par lequel un groupe d'agents arrive à une décision mutuelle acceptable sur un sujet donné, elle représente un axe fondamental qui distingue un agent d'un objet.

Pour mener à bien le processus de négociation, il est nécessaire de suivre un protocole qui facilite la convergence vers la solution, la négociation est caractérisée en général par un protocole minimal d'actions qui est: proposer, évaluer, accepter ou refuser une solution.

### **I.6.3.4. La coordination**

Quand 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.

Jacques Ferber donne la définition suivante: "*la coordination d'actions dans un système Multi-Agents est définie comme l'ensemble des tâches effectuées par les agents pour réaliser les autres actions (actions effectives) dans les meilleures conditions*"[Fer 95].

La coordination entre les agents d'un système apparaît sous deux formes distinctes, elle sert d'une part à éviter les problèmes, et à améliorer le fonctionnement du système d'autre part.

Sa première forme consiste à bien coordonner les plans de fonctionnement des agents pour assurer une meilleure gestion des ressources (notamment celles qui sont rares) et éviter les conflits d'accès, c'est la coordination due à la gêne.

Sa deuxième forme est la coordination due à l'aide, comme le signifie son nom, les agents doivent synchroniser leurs actions, et échanger les résultats qui sont nécessaires au fonctionnement des autres.

#### **I.6.4. Communication dans les systèmes Multi-Agents**

La communication représente la base de réalisation de tous les modes d'interaction qu'on a vue précédemment, soit la négociation ou bien la coordination...

Elle est définie comme une forme d'action locale d'un agent vers d'autres agents. Les questions abordées par un modèle de communication peuvent être résumées par l'interrogation suivante: qui communique, quoi, à qui, quand, pourquoi, et comment?

Pourquoi les agents communiquent-ils ? La communication doit permettre la mise en œuvre de l'interaction et par conséquent la coopération et la coordination d'actions.

Quand est ce que les agents communiquent-ils ? 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 difficulté réside dans l'identification de ces situations.

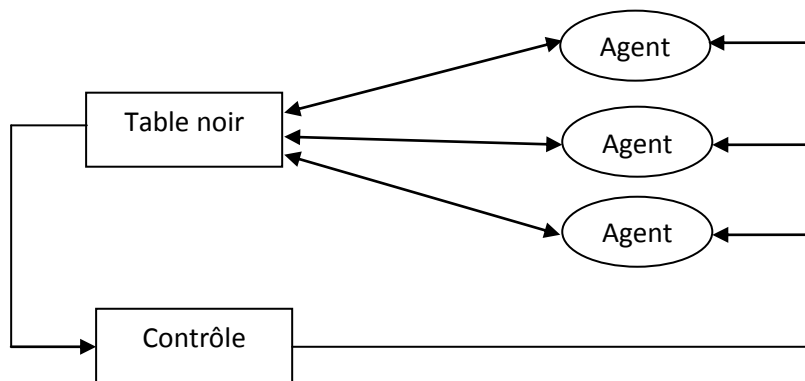
Avec qui les agents communiquent-ils ? Les communications peuvent être sélectives sur un nombre restreint d'agents ou diffusées à l'ensemble d'agents.

Comment les agents communiquent-ils ? La mise en œuvre de la communication nécessite un langage de communication compréhensible et commun à tous les agents. Il faut identifier les différents types de communication et définir les moyens permettant non seulement l'envoi et la réception de données mais aussi le transfert de connaissances avec une sémantique appropriée à chaque type de message.

Il existe principalement deux modes de communication :

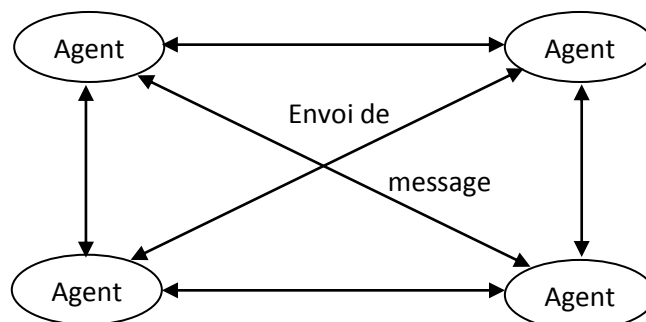
- Communication par envoi de messages.
- Communication par tableau noir.

Dans les systèmes fonctionnant par partage de ressources, les différents composants ne sont pas directement liés entre eux. Ils communiquent au travers d'une zone de données commune appelée tableau noir (Blackboard), dans laquelle sont stockées les connaissances du système.



**Figure 3: Architecture contrôle centralisé d'un SMA.**

En revanche, dans les systèmes où la communication se fait par envoi de messages, les connaissances sont distribuées entre les différents agents. Chacun d'eux communique directement avec les autres par envoi de messages.



**Figure 4: Architecture distribuée d'un SMA.**

## **I.6.5. La simulation Multi-Agents**

La simulation Multi-Agents est bien adaptée aux systèmes complexes constitués de plusieurs entités, elle consiste à séparer ces entités qui seront représentées par des agents et définir leurs différents comportements et interactions, puis faire évoluer les agents en fonction du temps, et étudier et analyser la progression du système global suite aux actions produites par les agents et leurs interactions.

Un système de simulation Multi-Agents est un ensemble d'agents et de règles d'interaction entre eux. Chaque agent est capable d'évaluer individuellement sa situation et de prendre des décisions à partir d'un ensemble de règles. Il peut exécuter divers comportements relatifs au système représenté, Les agents peuvent ensuite être capables d'évoluer, permettant à des comportements non anticipés d'apparaître.

### **I.6.5.1. Méthodologie de simulation Multi-Agents [Dro 03].**

La modélisation d'un phénomène dans une perspective Multi-Agents se traduit par Quatre étapes, mais avant tout il est indispensable de prendre une image approfondie du système qu'on vient simuler et d'en comprendre le fonctionnement en détail puis:

- Une décomposition du phénomène en un ensemble d'éléments discrets autonomes dont les interactions reproduisent le phénomène. Il est à noter que ce préliminaire nécessite une vision déjà distribuée du phénomène à modéliser (l'un des objets de la simulation est justement de déterminer les éléments ou les niveaux pertinents à étudier).
- La modélisation de chacun de ces éléments par un agent: Dans cette phase intervient un choix nécessaire quant à la théorie à employer pour définir les connaissances de l'agent, ses capacités fonctionnelles, ses comportements et les modes d'interaction qu'il adoptera à l'encontre des autres agents, ce choix peut être délicat. C'est au cours de cette étape que le modèle théorique de comportement individuel proposé par le domaine de recherche dans lequel s'effectue la modélisation est traduit en un modèle informatique.
- La définition de l'espace ou l'environnement dans lequel évoluent ces agents et des lois qui le gouvernent. Sa définition permettant d'affiner la description des actions possibles des agents, ainsi que celle de leurs moyens de communication, on définira souvent agents et environnement de manière concomitante.

### **I.6.5.2. Caractéristiques de la simulation Multi-Agents**

Les principaux atouts de la simulation Multi-Agents résident dans sa modularité et son incrémentalité. La modularité autorise un ajout ou un retrait aisé d'un ou plusieurs agents.

L'incrémentalité signifie qu'on peut facilement améliorer et affiner les agents qui composent le système.

Dans une approche Multi-Agents, le comportement de chacun des agents est décrit par un algorithme. Par conséquent, le système est plus précis et mieux détaillé qu'une description plus globale incluant toute une population.

### **I.6.6. Domaines d'utilisation 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 échantillon: les systèmes financiers et commerciaux comme le commerce et les banques électroniques, les systèmes embarqués, la gestion des base de données, la gestion des réseaux de communication, la recherche d'information et le contrôle industriel.

L'utilisation des systèmes Multi-Agents varie d'un domaine à un autre, mais elle tourne généralement autour de trois axes fondamentaux:

- Un système d'agents intelligents dédié à résoudre un certain nombre de problèmes d'une manière distribuée (utilisé dans le domaine de l'intelligence artificielle).
- Un système d'agents physiques consacré à réaliser un ensemble de tâches et d'actions réel représentant le but du système. On peut donner comme exemple le système des robots spatiaux exploratoires, les systèmes des robots footballeurs...etc.
- Un système Multi-Agents pour la simulation: c'est un système constitué des agents virtuels (agents logiciels) qui simulent des actions physiques, biologiques ou sociales, citons comme exemple la simulation des épidémies, la simulation des processus industriels...etc. Le fonctionnement de ce genre de systèmes est couplé aux outils d'études (comptage, statistiques, calculs des fonctions...) afin d'analyser le phénomène simulé.

## **I.7. Les agents mobiles**

### **I.7.1. Introduction**

Les agents mobiles sont une catégorie particulière des agents logiciels dont la caractéristique prédominante est leur capacité de se transporter entre les nœuds d'un réseau ou de plusieurs réseaux. Les agents mobiles (ou transportables) sont une extension directe du paradigme client/serveur. Dans ce paradigme, les entités communicantes ont des rôles fixes et bien définis. Un serveur offre des services alors que le client procède à leur utilisation. La communication qui prend place entre le client et le serveur est basée sur un échange de messages, obligeant les programmeurs à gérer divers aspects liés à la communication comme les adresses réseaux et les points de synchronisation.

## **I.7.2. Les caractéristiques d'un Agent mobile**

Le terme de mobilité est souvent associé à des agents mobiles. Elle est considérée, dans la majorité des cas, comme une propriété orthogonale des agents qui ne sont pas tous, forcément, mobiles.

Un agent est une entité qui peut résider sur un hôte et communiquer avec son environnement en utilisant les concepts conventionnels, comme l'appel des méthodes distantes (RPC) ou par un message de notification.

Ces agents sont communément nommés agents stationnaires ou agents immobiles.

Ils agissent seulement sur leur environnement d'origine où ils sont initialement construits. Toutefois, s'ils ont besoin d'informations ou d'interagir avec des agents résidants sur d'autres machines, ils utilisent des mécanismes de communication directe, basée sur des protocoles spécifiques basés sur le modèle client/serveur.

Contrairement aux agents stationnaires, nous trouvons des agents dits agents mobiles. Ces derniers, même en dehors de leur environnement, 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.

Ces agents sont souvent qualifiés d'autonomes. Une propriété qui leur permet de se déplacer avec une totale liberté aménageant ainsi leur parcours initial et décidant des tâches à effectuer. Cependant, le paradigme d'agent mobile a soulevé, à ses débuts, plusieurs critiques liées à la sécurité considérée comme un obstacle à sa généralisation. Fort heureusement, les technologies ont évolués entraînant un intérêt pour les chercheurs et industriels au développement d'applications basées sur des agents mobiles dans ces domaines d'accès sur les télécommunications, PDA ou les bus logiciels... il ne faut pas faire de confusion entre la mobilité de code et la mobilité d'utilisation. Celle-ci est plus rigoureusement nommée nomade. Pourtant, à ce jour, aucune solution industrielle disponible sur le marché n'utilise des agents mobiles [Milo 99] comme base de son architecture. Leur utilisation reste plus expérimentale ou spécifique à certains projets initiés par des indépendants ou des équipes de recherche.

Notre choix d'utiliser les agents mobiles n'est pas motivé par la technologie ou l'effet de mode, mais par les avantages qu'ils apportent à la construction d'un système distribué.

À ce titre, nous citons sept raisons [Lange 99] optimisant leur utilisation :

- Diminution de la charge du réseau : les systèmes distribués s'appuient souvent sur les protocoles de transmission impliquant différentes interactions, ce qui augmente le trafic réseau. Les agents mobiles permettent aux utilisateurs d'empaqueter une conversation puis de l'envoyer à une destination pour que l'interaction se fasse localement. Cette raison est très perceptible quand la taille des messages est nettement supérieure à celle des agents.

- **Maîtrise de la latence du réseau** : les systèmes en temps réel, tels que des robots dans des processus de fabrication, doivent répondre instantanément aux changements de leur environnement. Le contrôle d'un tel système par un réseau d'une taille substantielle implique une latence significative, ce qui est inacceptable pour des systèmes critiques. Les agents mobiles peuvent se déplacer vers une machine afin d'agir localement et pouvoir exécuter certaines directives.
- **Encapsulation des protocoles** : quand les données sont échangées dans un système distribué, chaque hôte dispose d'une implémentation pour envoyer les données et/ou interpréter les données reçues. Seulement, les protocoles évoluent pour s'adapter à de nouveaux besoins, pour être plus efficaces ou pallier à des problèmes de sécurité. Or, il est difficile voire quasi impossible de procéder à cette amélioration des protocoles. En conséquence, ces derniers peuvent poser problème. Les agents mobiles peuvent se déplacer sur un hôte distant pour ouvrir ou établir un canal basé sur des protocoles propriétaires. Dans ce cadre, l'agent mobile possède les moyens d'accéder au message qu'il transporte. Le format du message évolue, le moyen de consommer ce message aussi. Ce n'est pas le receveur qui change, c'est l'agent qui le transporte.
- **Exécution asynchrone et autonome** : les tâches d'un dispositif mobile se fondent souvent sur des connexions réseau coûteuses ou fragiles et exigent une connexion ouverte sans interruption. Ce qui n'est pas économiquement ou techniquement faisable. Pour résoudre ce problème, ces tâches sont incluses dans les agents mobiles, qui sont alors expédiés dans le réseau. Après avoir été envoyés, les agents deviennent indépendants du processus qui les a créés et peuvent fonctionner d'une façon asynchrone voire autonome. Le dispositif mobile peut se reconnecter après un temps nécessaire au traitement pour récupérer l'agent.
- **Adaptation dynamique** : les agents mobiles présentent leur environnement d'exécution et réagissent de façon autonome aux changements. Les agents mobiles multiples, ou plutôt un groupe d'agents mobiles a la capacité unique de travailler en communauté, de se répartir parmi les hôtes dans le réseau afin de maintenir la configuration optimale et résoudre un problème particulier. Cette collaboration s'établit de la manière suivante s'il n'y a pas d'agent sur un hôte1 alors l'agent de hôte2 exportera un agent dessus et réciproquement. Nous pouvons aussi dire que si un hôte s'arrête, les agents mobiles se notifient pour redéfinir leurs chemins et ainsi éliminer l'hôte en question de leurs parcours.

- Hétérogénéité naturelle : les traitements en réseau sont fondamentalement hétérogènes, du point de vue matériel et logiciel. Les agents mobiles fournissent des conditions optimales pour l'intégration des systèmes sans faille. Ceci est dû à l'indépendance de la couche du transport et celle du traitement, ces deux dernières ne dépendent que de l'environnement d'exécution.
- Robustesse et insensibilité aux défaillances : la capacité des agents mobiles à réagir dynamiquement aux situations et aux événements défavorables, facilite la construction de systèmes distribués robustes et insensibles aux défaillances. Si un hôte est en train de s'arrêter, tous les agents s'exécutant sur ce dernier sont avertis pour se déplacer et continuer leur opération vers un autre hôte. Les frontières de cette communauté ne sont pas inamovibles et le périmètre d'une communauté évolue au cours du temps en fonction, par exemple, des défaillances ou des nouvelles ressources à prendre en compte.

### **I.7.3. Application a base d'agents mobiles**

Depuis l'apparition de la notion de mobilité, de nombreuses plates-formes ont été développées afin de faciliter la programmation d'applications structurées en termes d'agents mobiles.

Plusieurs papiers de synthèse [Fug 98],[Thor 97]existent sur les principales plateformes utilisées de nos jours. Le plus souvent, lorsque l'on parle de plates-formes, nous regroupons deux niveaux élémentaires. [CROS 05]

Le premier est le langage de programmation qui offre les primitives nécessaires à l'exécution, à la communication et à la mobilité. Il constitue le niveau de base.

Le deuxième étant les services de haut niveau liés aux besoins ou l'administration, comme les annuaires, permettant aux agents la réalisation de leurs tâches.

#### **I.7.3.1. Les normes**

L'émergence des nombreuses plates-formes expérimentales a rendu essentielle la proposition d'une harmonisation grâce à la standardisation des différents concepts communs pouvant être identifiés.

Cette normalisation devrait permettre à terme de rendre compatibles les différents systèmes.

On peut trouver à l'heure actuelle deux normes principales : il s'agit de la norme FIPA (Foundation for Intelligent Physical Agents) [Pos 01] et de la norme MASIF (Mobile Agent System Interoperability Facility) [Mil 99].

Lorsque deux normes existent au sein d'une même technologie, elles ont une tendance naturelle à s'opposer alors que, dans notre cas, elles penchent plus vers la complémentarité due à la différence de leurs domaines d'origine.

- MASIF : la norme MASIF a été spécifiée par l'Object Management Group (OMG) qui se préoccupe généralement de l'hétérogénéité entre les systèmes, comme dans le cas de CORBA. Dans cette optique, le but, dans la norme MASIF, est de décrire les notions élémentaires permettant l'échange des agents entre différentes plates-formes. Pour ce faire, elle standardise la manière de gérer le code des agents, leur identification, la migration et l'adressage local.
  
- FIPA : en revanche, la communauté à l'origine de FIPA étant celle des systèmes multi agents, plus proche de l'intelligence artificielle, elle va se situer à un niveau plus élevé, c'est-à-dire le niveau applicatif en décrivant les éléments nécessaires à la réalisation d'une application et principalement en détaillant la communication entre les agents. Le but est de décrire un ACL (Agents Communication Language), les ontologies et les protocoles de négociation permettant ainsi de définir parfaitement les interactions entre les agents.

Les deux standards MASIF et FIPA sont considérés comme des religions de la communauté de logiciels d'agents. [Mag 99]

Leurs différences résident dans le fait que MASIF vise à permettre aux agents mobiles de migrer entre les systèmes d'agents du même profil par l'intermédiaire des interfaces normalisées de CORBA, contrairement à FIPA qui permet l'interopérabilité d'agents intelligents par l'intermédiaire de la communication normalisée des agents et des langues.

Ces deux normes tendent plus vers la complémentarité que vers la divergence. C'est déjà le cas de FIPA qui a inscrit dans son planning l'intégration des règles de MASIF sur la gestion de la migration.

### **I.7.3.2. Autre approche**

Au premier abord, les applications à base d'agents mobiles nous laissent penser à des plates-formes de services web. Puisque pour les deux, nous parlons de composants logiciels écrits dans divers langages de programmation et surtout des standards employés. Les services Web (en anglais Web services) [Tsal 02] représentent un mécanisme de communication entre applications distantes à travers le réseau Internet indépendant de tout langage de programmation et de toute plate-forme d'exécution. Ils utilisent le protocole HTTP, comme moyen de transport pour que les communications s'effectuent sur un support universel et standard, également un échange basé sur le langage XML pour décrire les appels et les données échangées.

Grâce aux services Web, les applications peuvent être vues comme un ensemble de services métiers, structurés et correctement décrits, dialoguant selon un standard international plutôt qu'un ensemble d'objets et de méthodes entremêlés. Cela facilite la maintenance de l'application. L'interopérabilité fournie par les échanges en XML, autorise à modifier un composant (service Web) pour le remplacer par un autre, éventuellement développé par un tiers. Qui plus est, les services Web permettent de réduire la complexité d'une application car le développeur peut se focaliser sur un service, indépendamment du reste de l'application.

Il nous semble nécessaire de dire que, sur certains aspects et propriétés, les agents mobiles et les services Web ont des similitudes. Cependant, leurs aspects divergent sur le rôle des traitements par rapport aux données. Le traitement d'un Web service est immobile alors que les données en entrée lui arrivent en XML. En revanche un agent mobile fait entrer des traitements alors que les données sont immobiles. Celles-ci peuvent d'ailleurs être aussi au format XML.

Un service Web implémente le métier et reçoit les données afin de les traiter alors qu'un agent mobile se déplace avec le métier et éventuellement les données [Zha 07].

#### **I.7.4. Exemples de plateformes d'agents mobiles**

Nos recherches se sont focalisées sur les différentes plates-formes mobiles existantes non pas pour les comparer mais pour mettre en avant certaines de leurs caractéristiques afin de savoir quelle implémentation choisir pour répondre à nos besoins ou celle qu'il faut adapter l'appliquer à notre approche.

Notre volonté n'est pas de réinventer la roue mais d'utiliser des études précédentes et de les valoriser.

Certaines plates-formes ne sont plus d'actualité, soit elles ont été abandonnées, soit elles sont pauvres en terme d'informations ou compliquées à utiliser. D'autres sont anciennes mais restent prometteuses et pertinentes malgré le critère d'âge, d'ailleurs elles sont toujours proposées dans différentes études récentes [CROS 05].

Malheureusement, nous ne pouvons pas toutes les décrire. Donc nous avons choisi les plates-formes suivantes :

- Agile Applets (Aglets) : simpliste, utilisant le langage KQML et respectant la norme MASIF émulée sans CORBA depuis 2003. D'autres plateformes respectent la même spécification comme Concordia, Telescript , Grasshopper ou MOA . Seulement elles ne sont plus maintenues.

- Java Agent DEvelopment Framework (JADE) : une plate-forme parmi tant d'autres, respectant le standard FIPA et toujours d'actualité. Le tableau ci-après (cf. Tableau 1) donne un état sur l'activité des plates-formes respectant le standard FIPA. En aucun cas il remet en cause le standard, il nous permet de motiver notre choix.
- Linda In Mobile Environnement (LIME) : elle s'appuie sur le modèle Linda, un modèle de coordination et de communication parmi plusieurs processus parallèles opérant sur des objets stockés dans une mémoire partagée, virtuelle et associative. Ce modèle est implémenté comme un langage de coordination.
- PLANGENT : est la plate-forme la plus proche de l'intelligence artificielle, en fait elle se concentre sur la question de l'intelligence comme un préalable aux fonctions des agents afin que ces derniers puissent comprendre l'utilisateur pour effectuer leurs tâches.
- Troms And COrnell Moving Agents (TACOMA) : C'est un projet universitaire qui se concentre sur l'utilisation des agents mobiles pour palier aux problèmes traditionnellement abordés par d'autres paradigmes tels que le modèle client/serveur. Par ailleurs, ce projet offre une abstraction portant sur la communication déléguée qui propose un mécanisme fondé sur le concept du rendez-vous et réalisé par l'intermédiaire d'un "Briefcase" échangé entre les deux agents.

## **I.8. La 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. Par la suite ; nous avons décrit la nouvelle technologie, c'est la technologie d'agents mobiles. Les agents mobiles sont des entités logicielles autonomes qui peuvent suspendre leur exécution sur une machine et migrer avec leur code, variables et états vers une autre machine où ils reprennent leur exécution.

Nous avons vu que les agents mobile sont considérés comme un concept bien adapté pour développer des applications cloud computing grâce à leurs propriétés de tolérance aux fautes, d'autonomie et d'adaptation. Dans le quatrième chapitre de ce mémoire, nous allons présenter une étude détaillée sur les applications cloud computing.

## Chapitre II

---

### Cloud computing : état de l'art

---

#### II.1. Introduction

La première question est qu'est-ce que le Cloud Computing ?

La traduction littérale « informatique dans les nuages » propose une informatique dématérialisée, pouvant être délivrée à la demande. Cette philosophie n'est pas nouvelle car elle est sans rappeler les concepts d'informatiques utilitaires proposés par John McCarthy en 1961. D'où est-on parti pour arriver à cette informatique dans les nuages ? De quoi le Cloud Computing est-il constitué ? Quelles sont ses différentes architectures ? Que va-t-il apporter dans l'informatique d'aujourd'hui et de demain ?

#### II.2. HISTORIQUE

Il n'y a pas de date-clé à laquelle nous puissions dire que le Cloud Computing est né ! La notion de Cloud fait référence à un nuage, tel que l'on a l'habitude de l'utiliser dans des schémas techniques lorsque l'on veut représenter Internet.

Un réseau comme Internet est constitué d'une multitude de systèmes fournissant des services et des informations. Le Cloud Computing est dans cette lignée : un ensemble de services et de données consommables.

##### II.2.1. L'INFORMATIQUE UTILITAIRE DE JOHN MCCARTHY

Cette notion de consommation a été proposée en 1961, lors d'une conférence au MIT (Massachusetts Institute of Technology), par John McCarthy aussi connu comme l'un des pionniers de l'intelligence artificielle (dont il proposa le nom en 1955) et pour avoir inventé du LISP en 1958 [John 61]. Lors de ce discours, John McCarthy suggéra que la technologie informatique partagée («time-sharing») pouvait construire un bel avenir dans lequel la puissance de calcul et même les applications spécifiques pouvaient être vendues comme un service public. Cette idée, très populaire dans les années 60, disparu au milieu des années 70 : à l'époque, les technologies matérielles, logicielles et réseaux n'étaient tout simplement pas prêtes. Le Cloud Computing met en œuvre l'idée d'informatique utilitaire du type service public, proposée par John McCarthy. Il peut aussi être comparé au cluster de calcul dans lequel un groupe d'ordinateurs se relie pour former un ordinateur virtuel unique permettant le calcul de haute performance (HPC), mais aussi à l'informatique en grille (Grid Computing) où des ordinateurs reliés et répartis géographiquement permettent la résolution d'un problème commun.

## **II.2.2. LES SERVICES BUREAU**

C'est dans cette philosophie, que depuis les années 70, on inventa la notion de «service bureau » pour qualifier une entreprise louant des lignes téléphoniques, répondeurs, services informatiques etc.

Généralement, les clients des « services bureau » n'ont ni l'ampleur ni l'expertise pour intégrer en interne ces services, c'est pourquoi ils passent par un prestataire. La combinaison de technologies, processus et expertise dans le domaine des entreprises est la valeur ajoutée des « services bureau », comme modèle économique basé sur leur capacité à produire des services et à les déployer en volume.

## **II.2.3. LES APPLICATION SERVICE PROVIDERS**

Les ASP, « Application Service Provider » ont aussi leur part dans l'historique du Cloud Computing.

Une ASP désigne une application fournie comme un service, c'est ce que l'on nomme maintenant SaaS pour « Software as a Service » dans la terminologie actuelle du Cloud Computing. Plutôt que d'installer le logiciel sur le poste client en ayant à assurer les phases d'installations et de maintenance sur chaque poste, les applications ASP sont hébergées et centralisées sur un serveur unique et accessible par les clients au travers de protocole standard.

C'est par exemple le cas avec des applications Web accessibles par http : il n'y a alors plus de déploiement ou de maintenance à effectuer sur le poste utilisateur, celui-ci n'a alors besoin que d'un simple navigateur Internet. Le déploiement, la configuration, la maintenance, la sauvegarde, etc. sont désormais de la responsabilité du fournisseur du service, le client est alors consommateur.

## **II.2.4. LA VIRTUALISATION**

Comme nous le verrons dans la suite, la virtualisation a été la première pierre vers l'ère du Cloud Computing. En effet, cette notion permet une gestion optimisée des ressources matérielles dans le but de pouvoir y exécuter plusieurs systèmes « virtuels » sur une seule ressource physique et fournir une couche supplémentaire d'abstraction du matériel.

Les premiers travaux peuvent être attribués à IBM, qui dans les années 60, travaillait déjà sur les mécanismes de virtualisation en développant dans les centres de recherche de Cambridge et de Grenoble, CMS (Conversation Monitor System), le tout premier hyper viseur.

C'est donc depuis presque 50 ans que l'idée d'une informatique à la demande est présente dans les esprits même si les technologies n'étaient jusqu'alors pas au rendez-vous pour pouvoir concrétiser cette idée.

Avec les différents progrès technologiques réalisés durant ces 50 dernières années, tant sur le plan matériel, logiciel et conceptuel, aux avancées des mécanismes de

sécurité, à l'élaboration de réseaux complexes mais standardisés comme Internet, et à l'expérience dans l'édition et la gestion de logiciels, services, infrastructures et stockage de données, nous sommes maintenant prêts à entrer dans l'ère du Cloud Computing, telle que rêvait par John McCarthy en 1961 [John 61].

### II.3. DEFINITIONS

Comme nous l'avons vu plus haut, le Cloud Computing est un nuage de services et de données. Plus précisément, c'est un paradigme, et à ce titre il est difficile de lui donner une définition exacte et de dire avec certitude s'il s'agit ou non de Cloud [LivF 09].

Il faut donc être vigilant, car de nombreux fournisseurs de services utilisent le mot « Cloud » à des fins marketings. A l'instar du domaine alimentaire, il n'y a pas pour l'instant de programme de certification comme les labels ou les « A.O.C. » (Appellation d'Origine Contrôlée).

Sur Internet, il n'y a pas de définition exacte du Cloud Computing et donc pas de certification pour dire si nous avons à faire à un « vrai Cloud ». Seule une conception personnelle de l'informatique permettra de faire la distinction entre une solution classique et le Cloud. Nous tenterons toutefois, au travers de ce mémoire, de donner toutes les clés pour comprendre le Cloud Computing.

Revenons donc à notre question initiale : « qu'est-ce que le Cloud Computing ? ».

Pour Wikipédia, il s'agit d' « un concept de déportation sur des serveurs distants et traitements informatiques traditionnellement localisés sur le poste client ». Pour le Syntec, cela consiste en « une interconnexion et une coopération de ressources informatiques, situées au sein d'une même entité ou dans diverses structures internes, externes ou mixtes, et dont le mode d'accès est basé sur les protocoles et standards Internet ».

Pour vulgariser, le Cloud Computing peut être perçu comme un système d'exploitation distribué sur des milliers de machines. Cet OS distribué, que l'on représente par ce fameux nuage, assure l'abstraction de l'infrastructure (matérielle, réseau, etc.) et a pour rôle d'héberger et d'exécuter des applications ou des services mais aussi de stocker des données.

L'idée de mettre ses applications et données dans un nuage unique, accessible par tout le monde et réparti sur des milliers de machines « abstraites » peut faire peur. Comme nous le verrons, les avancées en terme de sécurité, tant sur le plan technologique qu'intellectuel, nous permettent d'assurer une confiance optimale.

Le Cloud Computing n'impose aucune dépense en immobilisation puisque les services sont payés en fonction de l'utilisation. Cela permet aux entreprises de ne plus

se focaliser sur la gestion, la maintenance et l'exploitation de l'infrastructure ou des services applicatifs.

Les fortes avancées dans le domaine de la virtualisation ont rendu possible le Cloud Computing. Cette virtualisation permet d'optimiser les ressources matérielles en les partageants entre plusieurs environnements (« time-sharing »).

De même, il est possible de coupler l'application (et son système d'exploitation) et le matériel (encapsulé dans la machine virtuelle), cela assure également un « provisioning », c'est-à-dire la capacité de déploiement d'environnement, de manière automatique.

Le Cloud Computing couplé, aux technologies de virtualisation, permet la mise à disposition d'infrastructures et de plate-forme à la demande.

Mais le Cloud Computing ne concerne pas seulement l'infrastructure (IaaS), il bouleverse la plate-forme d'exécution (PaaS) et les applications (SaaS) : Comme nous le verrons plus loin, le Cloud est à la fois transversal et vertical.

Basiquement, le Cloud propose trois couches :

- L'infrastructure (IaaS : Infrastructure as a Service)
- La plate-forme (PaaS : Platform as a Service)
- L'application (SaaS : Software as a Service)

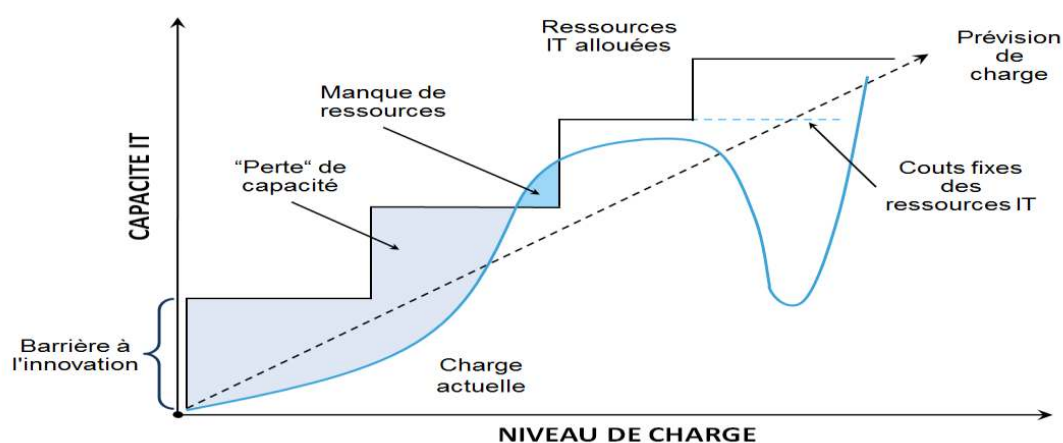
Comme l'a écrit François Tonic dans son livre blanc : « Tout SaaS est un service Cloud mais tout Cloud n'est pas un service SaaS », on peut parler de Cloud pour une application SaaS mais cela ne s'arrête pas à la couche applicative comme nous l'avons vu précédemment.

Étant donné que le Cloud Computing repose sur un ensemble de machines interconnectées et massivement distribuées, cela permet d'être tolérant face aux défaillances matérielles et logicielles (par des mécanismes de redondance, réplication, SLA). De plus, le Cloud Computing permet une élasticité, une flexibilité. Ainsi la puissance de calcul ou les capacités de stockage peuvent être très facilement augmentées par l'installation de nouveaux équipements au sein ou en dehors du Cloud, la charge sera alors répartie en fonction de ces nouveaux équipements. Il en va de même avec les applications et les données :

Elles pourront être hébergées à différents « endroits ». En cas de montée en charge, le Cloud créera plusieurs instances afin d'y répartir la charge et d'assurer une disponibilité maximale. Force de constater que dans un Datacenter classique, le taux moyen d'utilisation peine à dépasser les 25% (le but est de saturer la charge serveur pour une exploitation optimale), la moyenne étant de 10%. Les serveurs sont sous-utilisés et en cas de pic de charge, il faut acheter de nouveaux serveurs pour pallier à la demande. Cela signifie donc que le nouveau matériel restera en exploitation même inutilisé après les pics.

Ces infrastructures sont monolithiques dans le sens où elles ne permettent aucune exploitation dynamique et flexible, nécessitant une lourde administration et des coûts induits. Les schémas ci-dessous matérialisent le niveau de charge d'une application sur l'axe du temps et des capacités IT mises à disposition.

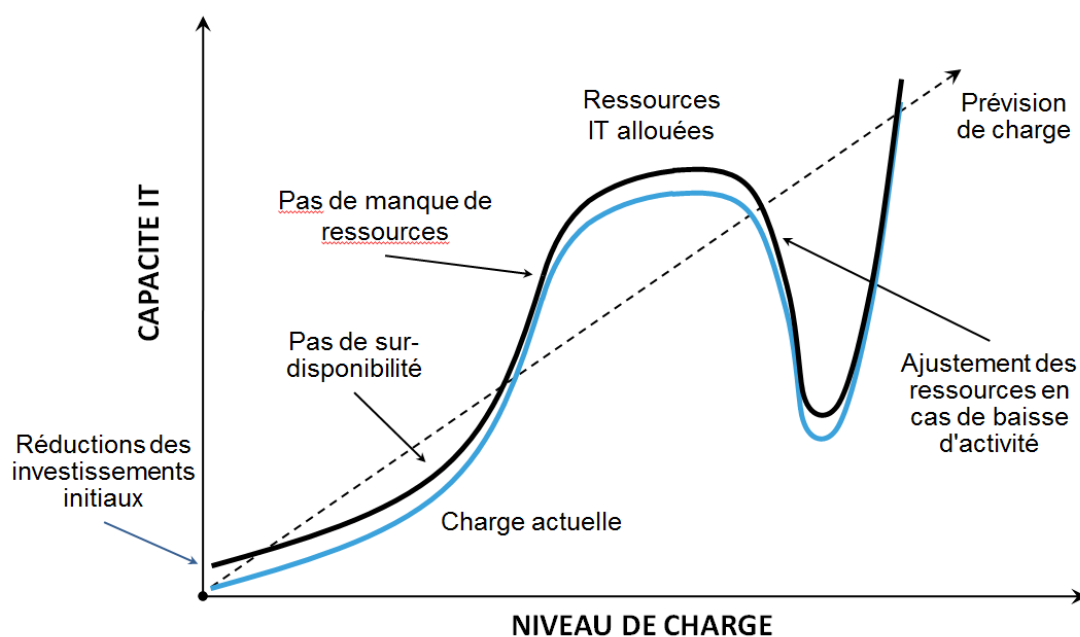
Dans le cas d'une solution hébergée dans un Datacenter classique, on retrouve tout d'abord une barrière à l'innovation en raison de la nécessité d'un investissement initial, lourd en termes d'achats de matériels, de licences, etc. La plupart du temps les serveurs sont sous utilisés et dès lors que l'application connaît un pic de charge, le manque de ressource provoque une saturation du service. Le seul moyen d'y remédier est l'achat et le déploiement de nouveaux serveurs entraînant alors de nouveaux coûts fixes. A contrario, en cas de baisse de l'activité, les ressources sous-utilisées seront alors plus importantes.



**Figure 5: le niveau de charge d'une application1.**

Pour rendre élastique et flexible, bref une infrastructure à la demande, le Cloud offre plusieurs approches. Le Cloud privé est la transformation de son infrastructure interne en un « Cloud » interne massivement virtualisé avec les bons outils d'automatisation allié à un catalogue de services. Une autre solution est de passer par ce que l'on appelle un Cloud public (de type Amazon EC2, Windows Azure, etc.) [LivC ],[Amaz] pour déporter tout ou partie de son infrastructure et de ces logiciels sur le nuage. Attention, toutes les offres de Cloud public ne sont pas forcément de l'IaaS. Par contre, Windows Azure est une plate-forme, alors qu'Amazon EC2 est une infrastructure.

Le Cloud privé nécessite un investissement, plus ou moins important selon la maturité de virtualisation de l'entreprise. En Cloud public, l'investissement se fera selon votre consommation, vos besoins réels.



**Figure 6: le niveau de charge d'une application2.**

Qui plus est, la mise en place d'applications traditionnelles est souvent onéreuse et très compliquée. Nous avons besoin de Datacenter puissants, avec des solutions de refroidissement, de la bande passante, du réseau, des serveurs, du stockage mais aussi de solutions de sécurité et de sauvegarde, le tout allié à une couche logicielle compliquée. Il faut souvent beaucoup d'experts pour installer, configurer et maintenir cet ensemble.

Avec le Cloud Computing, cela devient obsolète : au lieu de faire tourner vos applications vous-même, le tout est géré par un « Virtual Datacenter » partagé. Lorsque le besoin se fait ressentir de déployer une nouvelle application, il suffira de s'identifier sur le Cloud, de sélectionner dans un catalogue de services ou d'envoyer directement l'application souhaitée, puis de la personnaliser et la lancer. En quelques minutes, (voire quelques secondes), l'application sera fonctionnelle et pourra être consommée. C'est aussi valable pour une infrastructure ou une plateforme. Si une entreprise a besoin, de trois serveurs Windows et d'un serveur Linux, il suffira de sélectionner dans le catalogue et d'instancier cette infrastructure « à la demande ». En quelques minutes, celle-ci sera disponible et prête à l'emploi. C'est ici toute la puissance du Cloud Computing : une informatique à la demande.

Les applications sont, comme nous l'avons abordé plus tôt, hébergées à plusieurs « endroits » dans le Cloud (techniquement la machine virtuelle contenant l'application est virtualisée sur différentes machines physiques que forme le Cloud).

Cette duplication permet d'assurer une haute disponibilité mais aussi une grande élasticité (provisionnement automatique en cas de charge montante ou au contraire, le retrait de machine virtuelle en cas de baisse de charge).

En clair, on adapte la charge à la demande en mutualisant les ressources physiques dans des systèmes multi-tenants.

Autre principe, le Cloud Computing est une couche d'abstraction de l'infrastructure ou de la plateforme.

Comme nous le verrons ultérieurement, les fournisseurs et éditeurs doivent rester transparents sur les mécanismes internes permettant d'évaluer correctement une technologie. Cependant, la force du Cloud Computing est d'abstraire les mécanismes techniques et internes permettant une administration et un déploiement simplifiés.

Dès lors, on est bien dans une logique de consommation de service : la responsabilité de l'exploitation, du déploiement et de la maintenance de l'infrastructure, plateforme ou application, est à la charge du fournisseur du service Cloud. De plus, les phases d'élaboration sont raccourcies tant au niveau du développement, du déploiement que de l'exploitation. Pour finir, le Cloud Computing, de par son slogan « IT as a Service », propose un modèle de paiement « à l'usage », comme c'étaient le cas des « services bureau » des années 70.

Nous reviendrons sur tous les avantages mais aussi les inconvénients du Cloud Computing plus tard.

En clair, le Cloud Computing c'est :

- L'informatique « comme un service », c'est-à-dire à la demande
- Souvent basé sur de la virtualisation (surtout pour l'IaaS et le PaaS)
- Trois couches :
  - Infrastructure
  - Plateforme
  - Applicative
- Du « self-service » avec un paiement à la consommation (je paye ce que je consomme)
- Abstraction, mutualisation et allocation dynamique des ressources physiques.

## II.4. Architecteurs du cloud computing

De manière générale, l'architecture d'un environnement de cloud computing peut être divisée en quatre couches:

Le matériel couche, la couche d'infrastructure, la plate-forme couche et la couche d'application, comme le montre la figure 7. Nous décrivons chacun d'eux en détail:

- La couche matérielle: Cette couche est responsable de la gestion des ressources physiques du cloud computing y compris physique serveurs, routeurs, commutateurs, systèmes d'alimentation et de refroidissement. Dans la pratique, la couche matérielle est généralement mise en œuvre dans les centres de données. Un centre de données contient généralement des milliers des serveurs qui sont organisés dans des racks et interconnectés par le biais des commutateurs, des routeurs ou d'autres tissus. [LivS ], Les problèmes typiques à la couche matérielle incluent la configuration matérielle, la tolérance aux pannes, la gestion du trafic, alimentation et refroidissement gestion.
- La couche d'infrastructure: Aussi connu comme la virtualisation couche, la couche d'infrastructure crée un pool de stockage et de calcul de ressources en partitionnant les ressources physiques en utilisant les technologies de virtualisation telles que Xen, KVM et VMware. La couche d'infrastructure est un composante essentielle de la cloud computing, depuis de nombreuses clé caractéristiques, telles que l'affectation dynamique des ressources, sont mis à disposition par le biais des technologies de virtualisation.
- La couche de plate-forme: Construit au-dessus de l'infrastructure couche, la couche de plate-forme se compose de systèmes d'exploitation et des cadres d'application. Le but de la couche plate-forme est de minimiser le fardeau de déploiement d'applications directement dans des conteneurs VM. Par exemple, Google App Engine fonctionne à la couche de plate-forme pour fournir un soutien pour mettre en œuvre l'API logique de stockage, bases de données et des activités de web classique applications.
- La couche application: Au plus haut niveau de la hiérarchie, la couche application se compose des applications de cloud computing réels. Différente des applications traditionnelles, les applications cloud peuvent tirer parti de la fonctionnalité de mise à l'échelle automatique pour atteindre de meilleures performances, la disponibilité et faible coût d'exploitation.  
Par rapport aux traditionnels services environnements d'hébergement comme les fermes de serveurs dédiés, l'architecture de cloud calcul est plus modulaire. Chaque couche est faiblement couplée avec les couches au-dessus et au-dessous, permettant à chaque couche évoluer séparément. Ceci est similaire à la conception de l'OSI modèle pour les protocoles réseau. La modularité architecturale le cloud computing permet de soutenir un large éventail d'applications tout en réduisant les besoins de gestion et d'entretien les frais généraux.

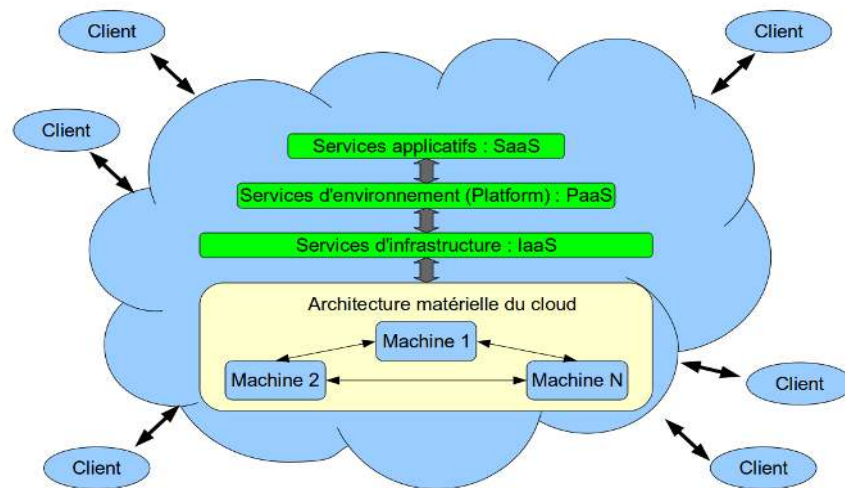


Figure 7: l'architecture d'un environnement de cloud computing.

## II.5. LES DIFFERENTES COUCHES

Que sont plus exactement ces couches, quels sont leurs modes de fonctionnements et leurs périmètres ?

Les couches les plus basses du Cloud Computing fournissent un service complet jusqu'à l'utilisateur.

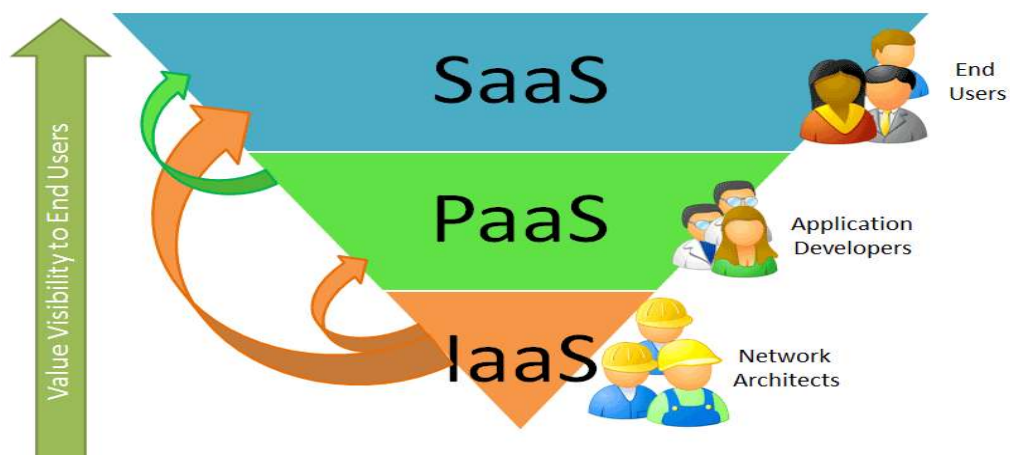


Figure 8: Les différentes couches du cloud computing.

### II.5.1. L'INFRASTRUCTURE COMME UN SERVICE (IAAS)

L'infrastructure fournit des capacités de calcul et de stockage ainsi qu'une connectivité réseau. Les serveurs, systèmes de stockage, commutateurs, routeurs et autres équipements, sont mis à disposition pour gérer une charge de travail demandée par les applications.

L'infrastructure comme un service ou IaaS, permet de disposer d'une infrastructure à la demande, pouvant héberger et exécuter des applications, des services ou encore stocker des données.

Concrètement, cela se caractérise par une infrastructure physique souvent mise à disposition par un fournisseur de services. On y trouvera une solution de virtualisation permettant la création de « Datacenter virtuels ».

Grâce aux solutions de virtualisation existantes, il est possible de créer très facilement des machines virtuelles connectées sur des réseaux, eux aussi virtuels, et qui seront exécutés sur les hyperviseurs des machines physiques.

Cette virtualisation nous donne une grande flexibilité parce qu'elle permet d'abstraire la couche matérielle sur laquelle les applications qui vont pouvoir être déployées et redéployées sans être liées à un serveur spécifique.

La virtualisation répond de manière dynamique là où les serveurs physiques fournissent un ensemble de ressources allouées selon les besoins, et où la relation entre les applications et les ressources de calcul, de stockage et de réseau, pourront s'adapter de manière automatique pour répondre à la charge de travail et aux exigences demandées.

Amazon AWS propose, par exemple, le service EC2. Il s'agit d'une offre IaaS reposant sur un hyperviseur Xen, permettant à un client de demander des ressources informatiques pour y héberger ses applications.

Sur Amazon EC2, il est possible de demander une ou plusieurs machines virtualisés sur les hyperviseurs d'Amazon. Cela permet de bénéficier d'une infrastructure flexible liée à une facturation en fonction de l'utilisation des ressources exploitées.

EC2 est compatible Windows et Linux pour les systèmes virtualisés.

Et de plus en plus d'éditeurs proposent leurs logiciels « EC2 ready » sous forme d'images virtuelles préconfigurées.

Pour créer son propre IaaS, plusieurs solutions existent.

VMware propose vSphere et vCloud Director, qui permettent la création d'un « Datacenter virtuel » dans lequel on pourra héberger des « vApp », une (ou plusieurs) application fonctionnant sur un ensemble de machines virtuelles connectées sur leur propre réseau virtuel [LivS ].

Ces « Datacenters » seront virtualisés sur une infrastructure physique partagée, mise à disposition par un fournisseur de services comme Terremark, Verizon ou encore Colt.

Microsoft fait de même avec Hyper-V Cloud qui définit les règles d'architectures que les constructeurs utilisent pour proposer des infrastructures clés en main.

## **II.5.2. PLATEFORME COMME UN SERVICE (PAAS)**

La plate-forme comme un service est la plate-forme d'exécution, de déploiement et de développement des applications. [LivF 09]

Il y a deux types de PaaS :

- Celui qui fournit une plate-forme intégrant le système d'exploitation (OS), la couche middleware et celle applicative qui sont fournies ensuite au client comme un service.
- Un service métier encapsulé et présenté via une API. Le client interagit avec cette plate-forme grâce à une API pour construire un service de plus haut niveau : la plate-forme se gère et s'adapte elle-même pour fournir le niveau de service attendu.

Le PaaS met à disposition des environnements prêts à l'emploi, fonctionnels et performants. Les offres PaaS commencent à être matures.

Parmi les solutions, citons Windows Azure de Microsoft, AppEngine de Google, Force.com de Salesforce. Chaque PaaS propose des modèles de développement différents. Google AppEngine se limite à Java et Python alors que Windows Azure travaille avec les langages .Net, Python, PHP, Ruby, Java.

Pour un développeur .NET, il s'agit de développer son application de manière locale, qui une fois prête, pourra être déployée : une simple commande génère le paquet et le déploiement se fait automatiquement sur le Cloud Windows Azure par exemple. L'application est alors immédiatement disponible et fonctionnelle.

Cependant, les PaaS ne sont pas seulement des environnements prêts à l'emploi pour les déploiements d'applications.

Ils peuvent aussi être des services applicatifs accessibles au travers d'API : Citons par exemple les services de stockage de Windows Azure Storage pour le Cloud de Microsoft, ou encore S3 (Simple Storage) chez Amazon.

Grâce aux PaaS, le déploiement d'applications dans différents environnements est très facile (test, pré-production et production sans se soucier de l'infrastructure et de la plate-forme dans lesquelles vont s'exécuter l'application ou le stockage de données).

### **II.5.3. LE LOGICIEL COMME UN SERVICE (SAAS)**

La dernière couche du SaaS est celle applicative mettant à disposition des applications complètes fournies à la demande. On y trouvera différents types d'application allant du CRM, à la gestion des ressources humaines, comptabilité, outils collaboratifs, messagerie, BI et d'autres applications métiers. [Clou 09]

Il n'y a donc aucun pré requis sur le poste client si ce n'est d'avoir un accès réseau au Cloud (en général Internet). Le déploiement, la maintenance, la supervision du bon fonctionnement de l'application et la sauvegarde des données, sont alors de la responsabilité du fournisseur de services.

Sur cette couche, des acteurs tels que Salesforce.com, proposent des applications à la demande de type CRM et des outils de collaboration. Cela permet à un client de bénéficier d'une application de manière instantanée et à la demande sans aucun frais en immobilisation (serveur ou licence), aucun déploiement ou maintenance à assurer. Il suffit de créer un compte et de profiter immédiatement d'un service de collaboration ou d'un CRM prêt à l'emploi.

Prenons l'exemple simple d'une solution de messagerie que l'on souhaite déployer dans une nouvelle entreprise. Le schéma classique intègre l'acquisition ou la location de serveurs et l'installation du système d'exploitation sur lequel la solution sera déployée (comme Lotus, Postfix ou Microsoft Exchange). Suite au déploiement, il y a une phase de configuration puis de test avant de pouvoir passer ce service en production.

Il s'agit ensuite de gérer la maintenance de cette messagerie, la supervision de son bon fonctionnement, l'application des mises à jour notamment concernant la sécurité, la réalisation de sauvegarde des données, l'élaboration d'une politique de gouvernance et d'un plan de reprise sur activité en cas d'incident. Tout cela a un coût, tant sur le plan financier, matériel et humain : en effet, cela nécessite une équipe d'experts qualifiés pour assurer la bonne exploitation de cette solution. Il n'est pas rare d'ailleurs, que les entreprises fassent appel à des sociétés extérieures, notamment sur le secteur de l'infogérance. Que l'on soit au niveau de l'infrastructure, de la plate-forme ou ici du service ou de l'application, les problématiques restent les mêmes : il faut des moyens financiers, matériels et humains pour maintenir un niveau de service demandé par les utilisateurs.

Avec le Cloud, nous avons des solutions SaaS comme « App » de Google ou encore «Exchange Online » de Microsoft qui mettent à disposition un service de messagerie à la demande : Il suffit qu'un administrateur crée son organisation sur l'un de ces services et il peut l'utiliser instantanément et avec très peu de configuration dans son entreprise.

Toute la charge repose sur le fournisseur de service SaaS qui assurera le déploiement de la solution de manière dynamique et instantanée, supervisera le bon fonctionnement, répartira la charge sur ces différentes ressources physiques, assurera la sécurité, les mises à jour et les plans de reprise sur incident. La plupart des services Cloud garantissent la disponibilité du service et des données par différents contrats SLA (nous reviendrons sur cette notion un peu plus tard).

De ce fait, le client est consommateur d'un service lui offrant la possibilité de s'abstraire de toutes contraintes financières, matérielles et humaines, dans le déploiement et l'exploitation de différentes solutions applicatives.

Dans cette notion de « consommation », la facturation est liée à l'utilisation du service. Par exemple sur Exchange Online, le coût est lié au nombre d'utilisateurs dans l'organisation.

De plus, Exchange Online fournit un quota de stockage pour l'ensemble des boîtes aux lettres de l'organisation (sur base de 25 Go par utilisateur). Il est ensuite possible d'acheter de la place supplémentaire si besoin (par tranche de Giga à répartir dans l'organisation).

De nombreuses organisations ont fait le choix de solutions SaaS, s'assurant ainsi une disponibilité de services souvent supérieure à ce qu'une équipe peut fournir en interne (surtout pour les PME et institutions) et pour un coût adapté à leurs besoins.

## **II.6. Types de Cloud Computing**

Comme nous l'avons vu, le Cloud Computing repose sur des ressources physiques. La question est « où sont ces ressources physiques ? » (Serveurs, commutateurs, routeur, solutions de stockage, etc.).[LiviUT]

La réponse « dans le nuage » n'est pas vraiment acceptable. Du point de vue consommateur, l'abstraction est telle qu'on ne peut déterminer sur quelles ressources physiques l'application est hébergée. De par son côté dynamique, les ressources physiques hébergeant une application et des données dans un Cloud ne sont jamais fixes et évoluent dans le temps.

En théorie, le Cloud Computing n'impose aucune dépense en immobilisation. Comme c'est le cas avec la solution de messagerie décrite, on exploite généralement les ressources physiques d'un fournisseur de Cloud.

Cependant cette technologie de Cloud Computing peut très bien se retrouver sur l'infrastructure physique d'une entreprise : n'étant plus mutualisé, le Cloud reste privé on parlera alors de Cloud public, de Cloud privé et de Cloud hybride.

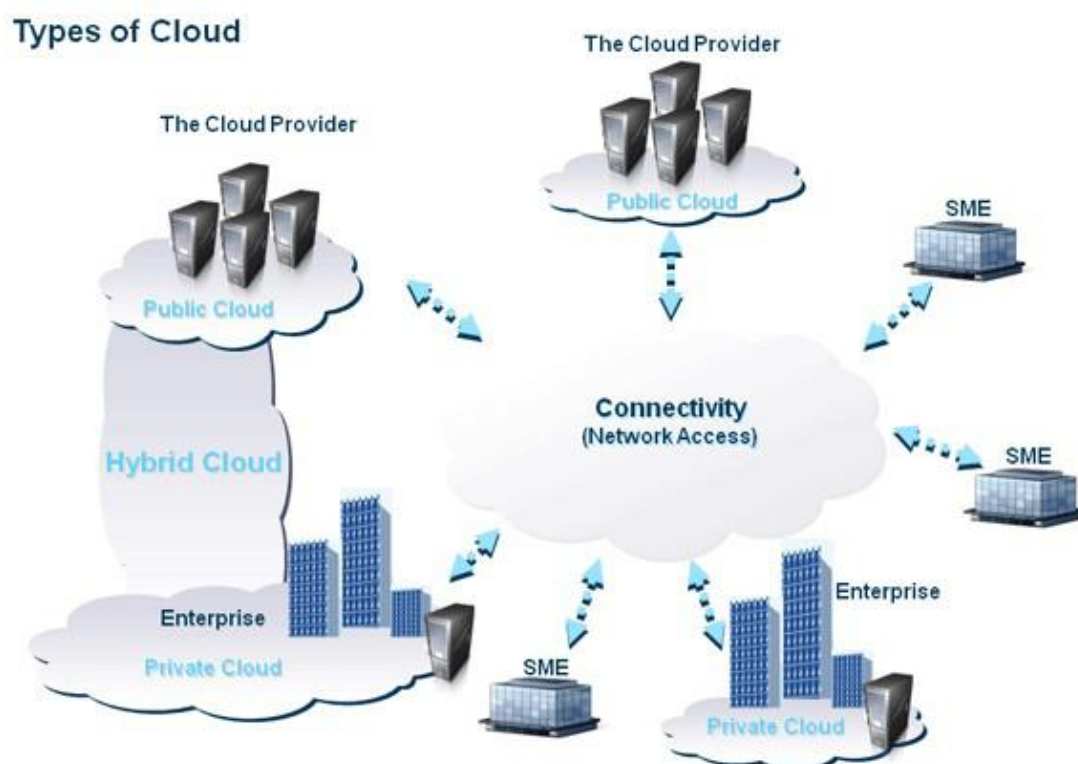


Figure 9: les types du cloud computing.

### II.6.1. Le cloud privé

Ces ressources physiques peuvent être hébergées dans une infrastructure propre à l'entreprise et étant sous son contrôle, à sa charge donc de contrôler le déploiement des applications. [LiviUT]

Nous reviendrons sur cette notion de Cloud privé, mais nous pouvons d'ores et déjà nous demander si « un Cloud privé est réellement un Cloud ? ». En effet, dans le sens où, comme nous l'avons dit plus haut, un Cloud ne doit pas imposer de dépenses en immobilisations, l'infrastructure physique dans un Cloud privé est à la charge de l'entreprise. Le Cloud privé peut aussi désigner un Cloud déployé sur une infrastructure physique dédiée et mise à disposition d'un fournisseur de services. Ainsi une entreprise peut louer à un fournisseur de services, un nombre conséquent de serveurs qui lui sont entièrement dédiés et sur lesquels une solution de Cloud sera déployée pour gérer dynamiquement l'application, la plate-forme ou l'infrastructure (virtuelle).

### **II.6.2. Le cloud public**

Pour certaines personnes, tel que Werner Vogels (Directeur des technologies d'Amazon), le Cloud ne peut être que public. [LivIUT]

Un Cloud public est un service IaaS, PaaS ou SaaS proposé et hébergé par un tiers.

Amazon, Google et Microsoft propose un Cloud public dans lequel n'importe quel particulier ou n'importe quelle entreprise peut y héberger ses applications, ses services ou ses données. Pour les consommateurs, il n'y a donc aucun investissement initial fixe et aucune limite de capacité.

Les fournisseurs de Cloud public facturent à l'utilisation et garantissent une disponibilité de services au travers des contrats SLA (« Service Level Agreement » : document qui définit la qualité de service requise entre un prestataire et un client).

### **II.6.3. Le cloud hybride**

Un Cloud Hybride est l'utilisation de plusieurs Clouds, publics ou privés. On peut ainsi déporter nos applications vers un Cloud public qui consommera des données stockées et exposées dans un Cloud privé, ou bien faire communiquer deux applications hébergées dans deux Clouds privés distincts, ou encore consommer plusieurs services hébergés dans des Cloud publics différents. Dans tous ces cas de figure, nous avons affaire à la notion de Cloud hybride.

## **II.7. DANS L'ERE DU CLOUD COMPUTING**

L'ère du Cloud Computing s'est accélérée en 2006 lorsqu'Amazon, l'une des plus grosses entreprises de commerce électronique, proposa d'exploiter leurs fermes de serveurs pour héberger des applications et des données. Il s'agissait pour eux de rentabiliser l'énorme infrastructure prévue en absorbant les charges en période de fêtes de Noël et en louant leurs ressources de calcul et de stockage. [LivM 09]

La crise économique a également joué un rôle clé dans la croissance du Cloud Computing. En effet, l'un de ses avantages est l'économie réalisée en disposant d'une infrastructure dont le coût est relatif à l'utilisation.

Le Cloud Computing permet de s'affranchir des investissements souvent très coûteux en termes de déploiement et d'exploitation des infrastructures, plates-formes et applications. Selon Brookings Institution, les économies réalisées pour les dépenses informatiques sont de l'ordre de 25 à 50 % avec le Cloud Computing. Il est important de comprendre en quoi le Cloud Computing répondra aux problématiques d'aujourd'hui et de demain, ses avantages et ses inconvénients, d'éclaircir les zones

d'ombres sur les sujets de l'interopérabilité, de la sécurité ou encore comment dans un avenir proche vont évoluer les métiers.

### II.7.1. POUR QUEL TYPE D'APPLICATION ?

Le Cloud Computing propose un découpage en trois couches :

- l'infrastructure,
- la plate-forme
- les applications.

Cela répondra à différentes problématiques. À la question « Le Cloud Computing: pour quel type d'application ? » la réponse pourrait potentiellement être toutes suivant leurs spécificités.

Pour une entreprise ou un particulier dont le métier n'est pas l'informatique, l'aspect infrastructure ou plate-forme ne sont pas au cœur de leurs préoccupations.

Une entreprise cherchera d'abord à se concentrer sur son métier et aura donc recours à différents types d'applications, comme par exemple un service de messagerie, un ERP pour tenir la comptabilité, la gestion des stocks ou du personnel, ou encore un service de stockage sécurisé leur garantissant l'intégrité et la disponibilité de leurs données. Quelle que soit l'architecture matérielle et logicielle qu'il doit se trouver derrière, seul le service rendu est important. [LivM 09]

Inconsciemment, on utilise le Cloud Computing depuis des années, ne serait-ce qu'en utilisant un service de messagerie comme par exemple des services comme Gmail de Google ou Hotmail de Microsoft, qui propose un service de boîtes aux lettres à la demande après une simple inscription.

De plus en plus, les entreprises conçoivent leurs logiciels sous forme de services (SaaS) permettant ainsi à quiconque d'en bénéficier de manière instantanée, sans pré requis et sans investissement lourd.

Prenons l'exemple de Salesforces qui propose des outils de CRM et de collaboration, ou de Google App qui offre un service de messagerie, d'agenda, de gestion documentaire ou encore d'échange et de partage d'informations. Nous trouvons aussi Microsoft Online Services qui met à disposition leurs produits serveurs tel qu'Exchange, SharePoint, Live meeting ou encore Office Communicator, le tout sous forme de services.

En fin de compte, n'importe quel type d'application peut être proposé à la demande dans la mesure où cette dernière ait été conçue pour être délivrée ainsi. Dans le cas de Microsoft Online Services, les produits serveurs, tels qu'Exchange Server ou SharePoint, sont très modulaires et permettent aux équipes de mettre en place des outils d'automatisation du déploiement et de configuration de ces produits.

Grâce à cela, il a été possible de délivrer ces produits sous forme de services, sans impacter l'intégralité de leurs architectures et de leurs fonctionnements internes.

Bien sûr, lorsque l'on évoque des applications de type SaaS, nous pensons tout d'abord aux applications Web. L'évolution des technologies RIA (Rich Internet Application) tels qu'Adobe Flash, Microsoft Silverlight ou, comme on le voit apparaître de plus en plus, le standard HTML5, permet une expérience utilisateur riche proche de ce que l'on peut trouver au travers d'un logiciel lourd. Même si cela permet de répondre à la majorité des besoins des applications, certaines d'entre elles devront dépasser le contexte Web.

De plus, une application SaaS peut poser des problèmes en cas de connexion indisponible. Étant donné que ces services sont délivrés via Internet, l'utilisateur a besoin d'une connexion constante pour pouvoir les consommer. Même si le développement d'Internet nomade ne cesse de progresser (hotspot Wifi, Internet mobile, etc.), il faut être en mesure de garantir à l'utilisateur une disponibilité de ses services et de ses données même hors connexion.

Dans cette lignée, nous pouvons citer l'idée du « S+S » pour « Software + Service » en bon français « le logiciel + le service ». Dans cette philosophie, le processus métier et le stockage des données sont fournis par le service. Quant au logiciel, il est déployé sur le poste utilisateur permettant de profiter de toutes les ressources locales du périphérique mais aussi d'assurer le fonctionnement en cas de non connexion.

Pour reprendre l'exemple de Microsoft Online Services qui dans son offre « Office 365 » fournit les services tels qu'Exchange Online, SharePoint Online, etc., et la gamme logicielle Office où nous retrouvons Outlook. On retrouve pour le service de messagerie le S+S avec le client lourd Outlook : « Outlook + Exchange Online = Software + Service » mais aussi un mode SaaS si on se limite à « Outlook Web Access » le web mail fourni en native sur Exchange (Server et Online).

Le client Outlook pourra consulter sa boîte aux lettres, son agenda et ses contacts, mais aussi rédiger des courriels même lorsque la connectivité au service est rompue, lors d'un trajet en avion par exemple. C'est ce logiciel qui assure localement l'enregistrement des données et les modifications effectuées hors connexion ; il se chargera également de les synchroniser avec le service lorsque la connectivité sera rétablie.

En fin de compte, chaque application, proposée en SaaS et/ou S+S, pourra se retrouver sur un Cloud. Mais qu'en est-il de l'infrastructure ou des plates-formes ?

Le Cloud Computing ne se destine pas qu'aux utilisateurs finaux. Pour les équipes de développement informatique, il s'agit par exemple de disposer d'une plateforme permettant d'héberger leurs applications en production. Ils auront aussi besoin d'une autre plate-forme sensiblement identique pour tester des nouvelles versions en pré-production.

Le PaaS peut répondre à cette problématique en mettant à disposition des plates-formes applicatives accessibles à la demande.

Un développeur Java pourra envisager de développer et de déployer une application Java sur Google AppEngine, d'une plate-forme vFabric de VMware et bientôt sur Force.com.

Attention, cela ne se fait pas par un claquement de doigt ou simple portage du code. Il faut adapter ce dernier pour son application au Cloud. Souvent, chaque PaaS expose son propre modèle de développement. Aujourd'hui, dans le monde « PaaS Java », le modèle SpringSource, racheté par VMware, s'impose avec des accords technologiques pour que les fournisseurs de PaaS déploient SpringSource (Framework Java très populaire).

En bref, si le modèle de développement d'un PaaS n'est pas compliqué, il impose rigueur et adaptation du code. Même chose pour la couche d'accès aux données qui nécessite une adaptation. Pour un développeur PHP ou .NET, le choix s'orientera plutôt sur la plateforme Windows Azure.

Comme pour les utilisateurs finaux, les entreprises dont le métier est de développer des applications, ne doivent pas se soucier des problématiques d'infrastructure et de plate-forme. Le Cloud Computing permettra de disposer de manière instantanée et sans aucun investissement de départ, d'une plate-forme dynamique et adaptable à la charge des applications et des données qu'elle héberge. Bien sûr, il faudra s'assurer que la plate-forme PaaS permette l'hébergement de l'application ou du service désiré, en fonction des technologies utilisées.

Quoi qu'il en soit, la couche la plus basse du Cloud Computing est l'infrastructure. Imaginons une infrastructure où les applications et les technologies utilisées sont tellement spécifiques à l'entreprise, qu'elles ne trouvent pas d'application équivalente ou de plate-forme capable de les héberger.

Le Cloud Computing, au travers de l'IaaS, pourra aider cette entreprise en lui proposant une infrastructure à la demande, lui permettant de s'affranchir des investissements matériels, de configuration et d'exploitation de ces ressources physiques.

Par exemple, prenons une entreprise ayant développé une application maison utilisant une technologie peu connue ou, en tout cas, non proposée dans les offres PaaS actuelles. L'architecture est composée de plusieurs serveurs chacun ayant un rôle défini, une offre IaaS, comme celle de Windows Azure (avec la notion de vmRole, d'Amazon EC2 ou de « vCloud Datacenter Service » des partenaires VMware), permet de construire cette infrastructure en très peu de temps et sans investir. Une fois le nombre de serveurs défini, la configuration réseau et les systèmes d'exploitation sélectionnés, le service IaaS déploiera et configurera cette infrastructure de manière automatique et rapidement. L'entreprise pourra alors exploiter cette infrastructure et y déployer ses applications, services et données.

N'importe quel projet informatique peut prétendre à être déployé dans un Cloud, reste encore à déterminer les avantages, les contraintes et le prix face aux solutions classiques.

## **II.7.2. BENEFCES / AVANTAGES**

Les avantages sont multiples :

- aucun investissement préalable,
- aucun prérequis demandé,
- un déploiement facile,
- une gestion simplifiée,
- un service d'une grande disponibilité et adaptable,
- un paiement relatif aux consommations.

Pour l'utilisateur, comme une start-up par exemple, l'avantage principal est de bénéficier d'un service sur mesure, sans investissement et capable d'absorber n'importe quel pic de charge.

La société Kobojo propose des jeux et des applications pour les réseaux sociaux. Créée en septembre 2008, l'entreprise a, en moins d'un mois, connu une croissance de 5 000 à 700 000 utilisateurs par jour. Aujourd'hui, c'est plus de 2,5 téraoctets téléchargés quotidiennement, avec des pics de charge très importants entre 18 et 23 heures.

Difficile pour une petite start-up d'investir dans une infrastructure capable d'évoluer en si peu de temps pour absorber une telle charge. Le Cloud Computing leur a permis de disposer d'une plate-forme et d'une infrastructure leur permettant d'héberger leurs applications sans investissement financier, avantage non négligeable lors de la création d'une structure de ce type.

De plus, ils ont pu répondre dynamiquement à la charge grâce à la souplesse proposée par le Cloud Computing. Aujourd'hui Kobojo compte plus de 50 millions d'utilisateurs.

Le Cloud Computing permet aussi de s'affranchir des contraintes de gestion des déploiements et de l'exploitation d'une telle architecture.

Les start-ups peuvent désormais se concentrer sur leur métier, le Cloud Computing étant à leur côté pour pallier les problématiques d'exploitation d'une plateforme hautement disponible.

L'impact financier est important : n'importe qui peut bénéficier d'une infrastructure technique dernier cri, capable d'absorber n'importe quel type de charge et sans aucun investissement préalable. Il n'y a donc plus de dépense en immobilisation et en cas de sous-utilisations, le Cloud Computing adaptera les ressources mises à disposition de manière proactive en fonction de la charge. D'un point de vue financier, la facture sera relative à l'utilisation du service.

En clair, le Cloud Computing permet de limiter les coûts d'exploitation en mutualisant les ressources physiques, garantit une haute disponibilité des services et des données et adapte l'infrastructure technique au volume d'activité de l'entreprise.

Pour finir, les développeurs, administrateurs, chef de projets, décideurs et les utilisateurs ne sont pas les seuls gagnants dans l'utilisation du Cloud Computing, la planète y trouve aussi une manière de réduire l'empreinte écologique de notre technologie actuelle et future.

Dans cette lignée, le « Green-IT » est un concept de plus en plus présent dans les investissements puisqu'il faut tenir compte des contraintes et des coûts en énergie des matériels informatiques.

On peut aussi citer la technologie DPM pour « Distributed Power Manager » de VMWare qui permet d'arrêter ou de redémarrer les hyper viseurs (ressources matérielles) d'un « vSphere » en fonction de la charge. Autre innovation estampillée « green », la conception des conteneurs de ressources physiques du Cloud Windows Azure qui tente à réduire au maximum l'empreinte écologique via un refroidissement assuré naturellement par un système de « free cooling ». Cela améliore le PUE « Power Unit Effectiveness » (rapport entre l'énergie apportée aux Datacenters et celle réellement utilisée par les serveurs).

### **II.7.3. LE MARCHE DU CLOUD COMPUTING**

Comme le prédisait le cabinet Gartner, l'année 2010 fut celle du Cloud Computing et la tendance n'est pas prête de s'inverser. Avec une croissance d'environ 25 %, le marché du Cloud Computing représentait plus de 68 milliards de dollars en 2010 d'après leur étude «Forecast: Public Cloud Services, Worldwide and Regions, Industry Sectors, 2009-2014 », une tendance qui va se poursuivre jusqu'en 2014 où le marché atteindra près de 149 milliards de dollars. [Web 01]

### **II.7.3.1. LES ACTEURS DU CLOUD**

Ce marché est partagé entre plusieurs acteurs : les éditeurs, les fournisseurs et les « pures players ». D'après l'étude de Markless International « Approches d'hébergement avec le Cloud Computing & la virtualisation- Référentiel de pratiques », les entreprises interrogées mentionnent avoir recours à plus de 60 prestataires. De plus en plus d'éditeurs portent leurs produits en mode SaaS et les éditeurs SaaS entrent comme « Pure Player ».

#### **II.7.3.1.1. EDITEURS**

On retrouve ici des éditeurs proposant des solutions Cloud. Un éditeur n'est pas forcément un fournisseur de services, autrement dit son périmètre n'est pas de fournir un service Cloud, mais plutôt de fournir une technologie capable d'héberger une solution Cloud. Cependant la frontière est mince car bon nombre d'éditeurs sont fournisseurs de leurs propres produits.

VMware est un éditeur de produits de virtualisation. Comme beaucoup d'autres éditeurs, VMware s'est lancé depuis 2008 à la conquête du Cloud Computing. Aujourd'hui, il édite des produits pour la couche IaaS comme « vSphere » et « vCloud Director », « vFabric » pour la couche PaaS ou encore Zimbra racheté cette année pour développer ses produits SaaS.

Interrogé lors du VMWorld 2010 à Copenhague, Jean-Pierre Brulard (Vice-président VMWare Europe) insiste sur le fait que VMware ne sera jamais un fournisseur de service laissant ainsi la place à d'autres entreprises pour fournir des solutions Cloud basées sur les technologies de VMware au travers du programme « vCloud Datacenter Service ».

Microsoft aussi fourni des produits de Cloud Computing comme « Windows Azure Appliance », dont la sortie est prévue en 2011. « Windows Azure Appliance » est la version produit de Windows Azure que l'on va pouvoir installer directement sur les infrastructures d'une entreprise. Tout comme avec VMware, on retrouvera des clouds « Windows Azure » proposés par différents fournisseurs de services, directement auprès de Microsoft, par des fournisseurs de services tiers ou bien installés sur les infrastructures internes d'une entreprise.

Les éditeurs fournissent donc une technologie de Cloud Computing qui peut être hébergée sur les infrastructures physiques d'une société de services et être proposée comme un Cloud public, ou bien directement installée sur les infrastructures internes d'une entreprise laissant place à un Cloud privé.

#### **II.7.3.1.2. FOURNISSEURS**

Les fournisseurs de services de Cloud Computing sont des hébergeurs tels que l'on a l'habitude de les retrouver depuis plusieurs années sur Internet. Ils mettent à disposition des infrastructures physiques proposant une plate-forme de Cloud.

On peut citer Microsoft avec sa plate-forme d'IaaS, de PaaS et de SaaS au travers de « Windows Azure » et « Office 365 », Google avec son service SaaS « Google App » et son PaaS « Google App Engine ». On retrouve aussi le géant Amazon avec ses services de IaaS et PaaS comme « Elastic Compute Cloud (EC2) », « Elastic MapReduce » ou encore « Simple Storage Service (S3) ».

#### **II.7.3.1.3. PURE PLAYER**

Enfin les Pures Player, en français les « purs joueurs », qui jouent d'emblée la carte du service en ligne. C'est le cas par exemple de Salesforces créé en 1999 par Marc Benioff qui est considéré comme l'un des pionniers du modèle SaaS. On peut aussi nommer YouSaaS qui propose un bureau virtuel d'applications professionnelles, ou encore « entreprise-facile.com » créé en avril 2007 qui édite des services en ligne pour les entrepreneurs (devis, pour livraison, facture, activités commerciales, tableaux de bord, partage d'informations etc.).

Se pose alors la question du potentiel des « pures players » à supplanter les éditeurs

Comme le rappelle François Tonic dans son livre blanc « Le Cloud Computing » : « lorsque l'on s'attaque frontalement à un géant comme SAP sur des progiciels, difficile d'imaginer un combat équitable. Sur de petits projets ou des projets précis dans une grande entreprise, le pure player à sa place. Mais l'éditeur traditionnel, quand il a vu la menace, a réagi soit en tissant des alliances avec le pure player, soit en commercialisant sa propre solution en ligne ».

Une autre question apparait quant à la pérennité d'un tel service puisqu'il s'agit la plupart du temps de petites structures récentes d'où la difficulté de garantir que le service existera toujours dans plusieurs années.

#### **II.7.3.1.4. OPEN SOURCE**

On retrouve aussi l'initiative open source dans le Cloud Computing. C'est notamment le cas avec Ubuntu qui propose sa distribution « Ubuntu Entreprise Cloud » mettant à disposition un IaaS pouvant être déployé sur ses propres infrastructures. Il y a aussi des acteurs tels que Novell, Sun, Eucalyptus ou encore avec AppScale.

Pour voir des solutions open source de Cloud Computing ou des notions de Cloud ouvert (Open Cloud), il faut d'abord définir les standards et rendre disponibles des spécifications. Tout comme il était très difficile de définir des standards pour l'interopérabilité du Web services l' « Open Cloud » n'est pas prêt d'exister.

Bien que des initiatives soient prises, que des consortiums soient créés, il faut encore que les grands acteurs puissent définir les normes permettant la migration d'applications d'une plate-forme à une autre. En effet, à ce jour, l'Open Source n'est pas encore au niveau de pouvoir offrir une réelle alternative aux grands acteurs actuels du marché.

Quoi qu'il en soit l'open source ne pourra proposer que des produits de Cloud Computing et ne jamais fournir ce service puisqu'il nécessite d'énormes moyens rien que pour les Datacenters sans poser la question du modèle économique.

## **II.8. LE PRIX DU CLOUD.**

La tarification du Cloud Computing est un sujet assez complexe à cause de la difficulté pour comparer les différentes offres. Il est par exemple très compliqué de pouvoir comparer une offre Amazon EC2 avec Windows Azure. Bien que ces deux services facturent à l'heure de calcul, nous pouvons nous demander quelles sont les puissances des instances hébergeant mon application (processeur, mémoire, accès disque etc.). De plus, une heure de calcul sur un Quad-Core 2Ghz n'aura pas la même valeur que sur un Mono-Core 1Ghz. [Web 02]

On parlera de type d'instances pour définir ses capacités techniques.

En général la tarification Cloud comprend :

- Le temps de calcul, en général à l'heure
- la quantité de stockage mesuré en Gigaoctet
- le nombre de transactions représentant le nombre d'accès aux services de stockage
- la bande passante en entrée comme en sortie

Bien entendu, le prix évoluera en fonction du nombre d'instances demandées et de leur type (Small, Large, Extra-Large, etc.).

Au-delà de ces chiffres, il faut tenir compte du niveau de qualité offert et des différentes fonctionnalités proposées. Chez Amazon, par exemple, la répartition des charges (load-balancing) entre les différentes instances n'est pas comprise dans le prix standard, c'est une option entraînant un coût supplémentaire alors que ce n'est pas le cas sur Windows Azure.

Les fournisseurs proposent aussi des forfaits incluant un certain quota d'utilisation des ressources. Tout comme avec la téléphonie mobile, il est possible de souscrire un contrat définissant la consommation mensuelle, tout dépassement pouvant être facturé à la carte comme du hors-forfait.

## II.9. Exemple des Solutions IaaS open source de Cloud Computing

Le Cloud Computing représente un nouveau défi dans le monde informatique. Plusieurs solutions sont proposées : des solutions propriétaires et des solutions open sources. La mise en place d'un environnement du Cloud Computing pour des buts de recherche nécessite initialement le choix d'une solution open source et puis son installation. La décision est souvent très difficile à prendre puisque chaque solution possède ses propres caractéristiques.

Une comparaison des solutions du Cloud Computing représente donc un bon point de départ. Voici un panorama des outils open-sources de Cloud Computing offrant une infrastructure comme service et permettant une gestion simplifiée d'architectures matérielles complexes.

### II.9.1. Eucalyptus

- Présentation

Eucalyptus est un outil open source issue d'un projet de recherche de l'université de Californie. Cette solution est la plus connue, car elle est intégrée dans les distributions Ubuntu Server et Debian. [Web 03]

Eucalyptus est écrit en C, Java et Python et permet de créer des Clouds IaaS de type privé ou hybride. Il supporte les machines virtuelles Linux ainsi que les hyperviseurs Xen et KVM. Son avantage majeur est le fait qu'il est compatible avec Amazon EC2.

Il possède également une version entreprise (payante) de la société Eucalyptus Systems qui apporte des fonctionnalités supplémentaires comme le support de VMware.

- Architecture

Eucalyptus comporte cinq composants principaux :

- ✓ Cloud Controller : c'est l'unique point d'entrée (Front end) pour tous les utilisateurs et les administrateurs d'Eucalyptus. Il est responsable de la gestion de tout le système.
- ✓ Cluster Controller : il conserve les informations relatives aux ressources physiques.
- ✓ Node Controller : il interagit avec l'hyperviseur pour gérer les machines virtuelles. Lors d'une demande d'allocation de ressources physiques par le Cloud Controller, c'est le Cluster Controller qui alloue les ressources en sollicitant les Nodes Controllers.
- ✓ Walrus : c'est le composant qui gère l'accès aux services de stockage.

- ✓ Storage Controller : ce composant fonctionne avec le composant Walrus et permet de stocker les images des machines virtuelles et les données des utilisateurs.

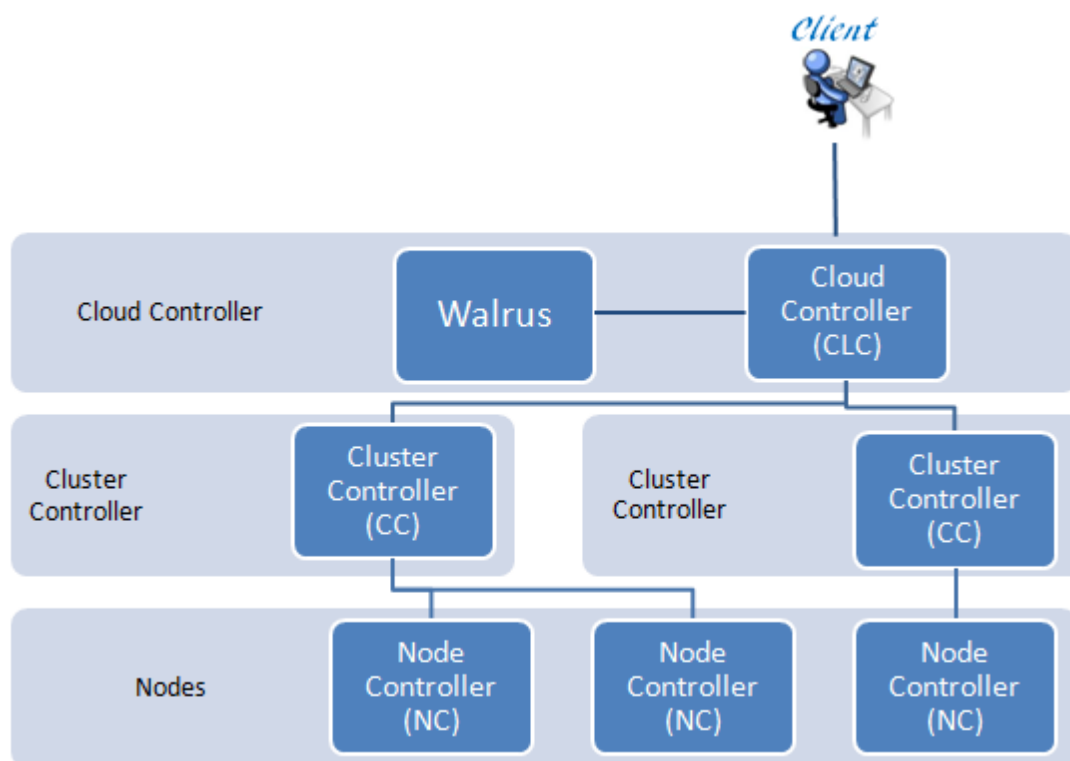


Figure 10: l'architecteur d'Eucalyptus.

## II.9.2. OpenNebula

- Présentation

Il s'agit d'une plateforme purement open-source permettant de déployer des Clouds privés, hybrides et publics. [Web 05]

Elle est écrite en C++, Ruby et Shell et elle supporte les hyperviseurs Xen, KVM et VMware. Le support de Virtualbox est prévu à partir de la version 4.0 de VirtualBox. Sa puissance consiste dans ses connecteurs vers des fournisseurs d'IaaS sur les Clouds publics tels que : Amazon EC2 Web Service, Nimbus WSRF, ElasticHosts REST, etc.

OpenNebula est soutenu par le projet européen RESERVOIR , qui propose une architecture complète pour la gestion de Datacenter et la création de services Cloud.

- Architecture

L'architecture interne d'OpenNebula peut être divisée en trois couches :

- ✓ Tools : c'est l'ensemble des outils de gestion pour OpenNebula ;

- ✓ Core : il se compose d'un ensemble de composants pour contrôler les machines virtuelles, le stockage et le réseau virtuel ;
- ✓ Drivers : l'interaction entre OpenNebula et l'infrastructure de Cloud est effectuée par des pilotes spécifiques qui sont les drivers.

Les machines Front end et Node sont reliés entre eux à travers un réseau privé.

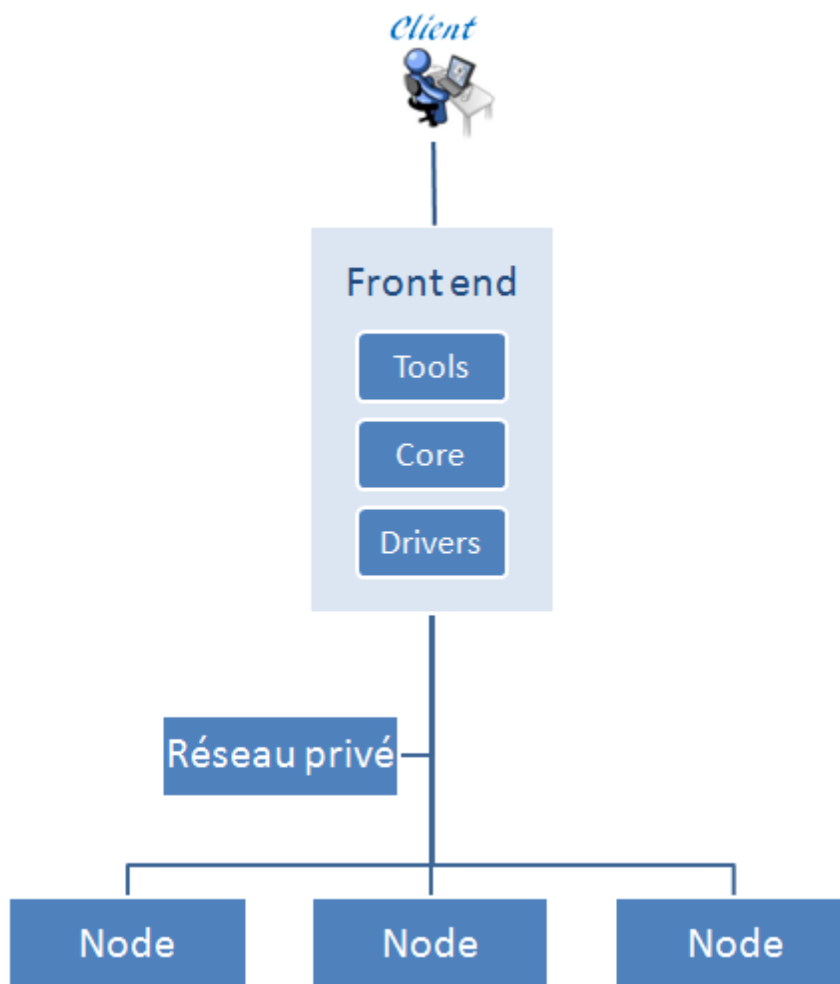


Figure 2.7: l'architecteur d'OpenNebula.

### II.9.3. Nimbus

- Présentation  
Issu du monde de la recherche, Nimbus permet de déployer un Cloud de type IaaS.[Web 04] Cette plateforme supporte les hyperviseurs Xen et KVM, et peut s'interfacer avec le Cloud Amazon EC2. Elle est associée au projet Cumulus, qui permet de déployer des services de stockage en Cloud, compatible avec le Cloud Amazon S3.

- Architecture

Nimbus comprend plusieurs composants qui peuvent être regroupés selon trois contextes comme le montre la figure 12. Ces composants sont reliés entre eux à travers un réseau privé.

- ✓ Client-supported modules : il est utilisé pour gérer les clients du Cloud. Il comprend le *context client module*, le *Cloud client module*, la *reference client module* et l'*EC2 client module*.
- ✓ Service-supported modules : il fournit tous les services du Cloud. Il comprend le *context broker module*, le *Web service resource Framework module*.
- ✓ Background resource management modules : son rôle est de gérer les ressources physiques du Cloud. Il comprend différents modules à savoir : *workspace service manager module*, *IaaS gateway module*, *EC2 module*, *workspace pilot module*, *workspace resource management module* et *workspace controller*.

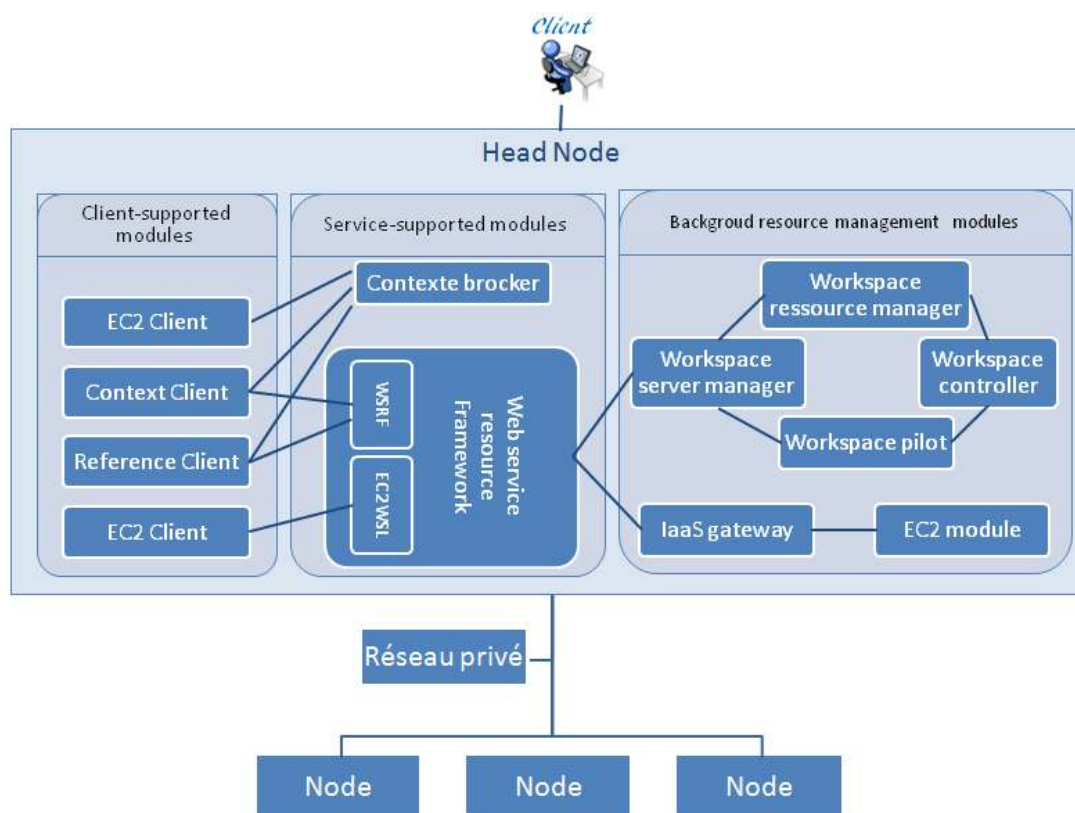


Figure 12: l'architecteur de Nimbus.

## II.9.4. Xen Cloud Platform

- Présentation

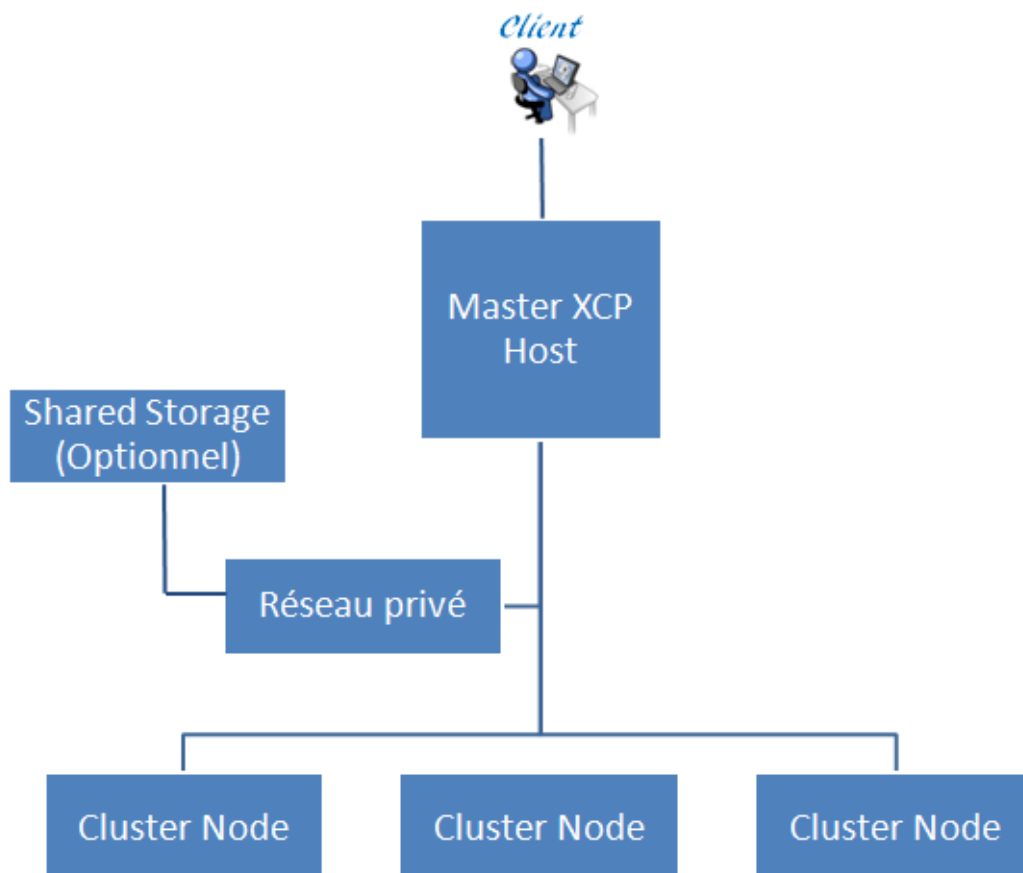
Xen Cloud Platform (XCP) est une plateforme open-source du Cloud Computing développée par la communauté Xen et distribuée sous licence GPL. Elle a pour but de proposer une plateforme open-source gratuite pour construire et faire dialoguer des services du Cloud Computing. [Web 06]

- Architecture

Les composants principaux de Xen Cloud Platform sont :

- ✓ XCP Host : il consiste en un système d'exploitation Xen.
- ✓ Master XCP Host : il gère les XCP hosts.
- ✓ Shared storage : c'est un composant optionnel où sont stockées les machines virtuelles. Ce composant permet aux administrateurs de déplacer des machines virtuelles d'un XCP host à un autre.

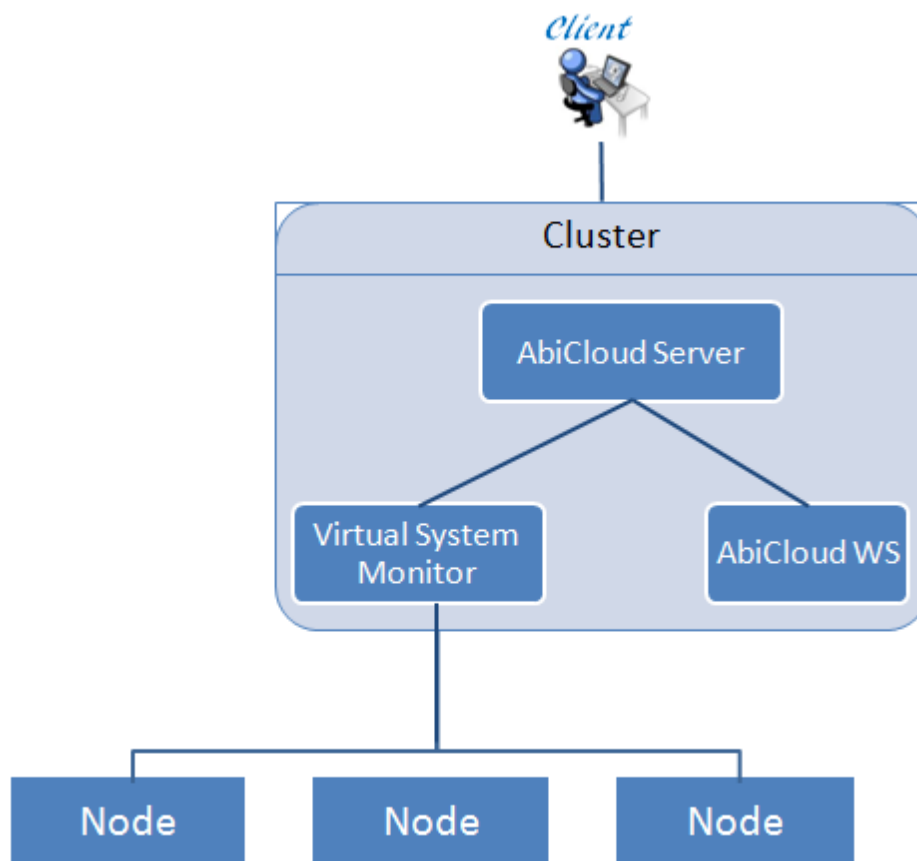
La figure 13 montre l'architecture de Xen Cloud Platform.



**Figure 13: l'architecture de Xen Cloud Platform.**

## II.9.5. AbiCloud

- **Présentation**  
AbiCloud, principalement développé par Abiquo, est la plateforme du Cloud Computing permettant de créer et de gérer des Cloud publiques, privés et hybrides. [Web 07]
- **Architecture**  
AbiCloud se compose principalement de trois éléments (figure) :
  - ✓ AbiCloud server : il consiste en un système d'exploitation Xen.
  - ✓ AbiCloud WS (AWS) : il est responsable de la gestion des machines virtuelles.
  - ✓ Virtual System Monitor (VSM) WS : c'est le composant qui permet de suivre toute l'infrastructure virtuelle du Cloud.



**Figure 14: l'architecture d'AbiCloud.**

## **II.10. La conclusion**

Nous avons présenté dans ce chapitre une idée générale sur le Cloud Computing, son architecture et ses différents services.

Le cloud computing n'est pas une nouvelle technologie en soi, il utilise des technologies existantes sous une forme nouvelle et un concept marketing innovant.

Le cloud est disponible sous différentes formes, chacune s'insérant à un niveau différent de l'architecture d'un système d'information, il apporte de nombreux avantages mais comporte un certain nombre de risques qui peuvent être importants selon le domaine d'activité de l'entreprise.

Actuellement, le marché du cloud est dans une phase d'expansion très importante et de nombreuses entreprises tentent l'aventure en tant que fournisseur de cloud même si quelques acteurs majeurs occupent la plus grande partie du marché.

Dans ce contexte de concurrence, un facteur fera sûrement la différence dans les années à venir : la sécurité. Ainsi la première entreprise à proposer un niveau de sécurité largement supérieur aux autres aura pris un avantage indéniable sur ses concurrents.

## Chapitre III

---

### La Sécurité du cloud computing

---

#### III.1. INTRODUCTION

Le Cloud Computing est un type d'infrastructure informatique qui consiste en une collection des nœuds de calcul interconnectés, des serveurs et autres matériels ainsi que des services et des applications logicielles qui sont provisionnés de façon dynamique entre les utilisateurs concurrents.

Les services sont offerts sur Internet ou réseaux privés, ou leur combinaison.

Les services de cloud computing sont accessibles via ces réseaux en fonction de leur disponibilité, la performance, la capacité et la qualité de service (QoS).

L'objectif est de fournir des services fiables, sécurisés, durables et évolutives Plateformes et infrastructures pour les utilisateurs finaux.

Ces systèmes ont des objectifs de la prestation informatique et stockage illimitée et cacher la complexité de la grande informatique distribuée auprès des utilisateurs.

Selon le type de service fourni, il existe trois types de services de cloud computing également appelés comme les modèles de prestation :

Infrastructure as a service (IaaS), Platform as a service (PaaS) et Software as a service (SaaS).

- IaaS traite de la fourniture des installations informatiques, de stockage ou de tout autre matériel de ressource.  
Amazon est l'un des fournisseurs de cloud IaaS, lorsque ces services sont EC2 et S3.
- PaaS offre des plates-formes en termes de système d'exploitation et le logiciel système qui peut être utilisé pour créer des applications personnalisées par les utilisateurs. L'utilisateur peut configurer et développer son application sur la plate-forme spécifique.  
Microsoft Azure est un exemple de PaaS.
- SaaS traite en utilisant n'importe quelle application ou un service via le nuage. Google Apps est l'un des exemples qui permet la collaboration sur diverses applications, comme la gestion d'événements, gestion de projet, etc. via internet.

Un état de l'art des techniques qui contribuent au cloud computing sont les suivants:

- **Virtualisation:** Il a été le concept sous-jacent vers une telle hausse considérable du cloud computing dans l'ère moderne.  
Le terme se réfère à offrir un environnement qui est capable de rendre tous les services, soutenu par un matériel qui peut être observé sur un ordinateur personnel, pour les utilisateurs finaux.  
Les trois formes actuelles de virtualisation classés comme: la virtualisation des serveurs, la virtualisation de stockage et La virtualisation du réseau.
- **Service Web et SOA:** services Web fournis des services sur le Web en utilisant des technologies comme XML, Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP), et Universal Description Discovery and Integration (UDDI).  
L'organisation de service à l'intérieur d'un cloud est gérée sous la forme de Service Oriented Architecture (SOA) et par conséquent, nous pouvons définir SOA comme quelque chose qui fait appel à de multiples services à effectuer une tâche spécifique.
- **Application Programming Interface (API):** sans API il est difficile d'imaginer l'existence d'un cloud computing.  
L'ensemble de services de cloud computing dépendent des API et de permettre le déploiement et la configuration à travers eux.
- **Web 2.0 / Mash-up:** le Web 2.0 a été défini comme une technologie qui nous permet de créer des pages Web et permet aux utilisateurs d'interagir et de collaborer en tant que créateurs de contenu généré par l'utilisateur dans un environnement virtuel de communauté.  
Il permet l'utilisation de la technologie de World Wide Web vers une approche de plate-forme plus créative et collaborative.  
Mash-up est un web application qui combine les données de plus d'une source dans un outil de stockage intégré unique.

### **III.1.1. Caractéristiques et obstacles**

Le cloud computing a commencé sa base dans le milieu de l'année 2007 et se développe rapidement jusqu'à ce jour.

Il a des caractéristiques différentes qui rendent les utilisateurs veulent passer à l'environnement de cloud computing. Certaines de ces caractéristiques sont discutées ci-dessous :

- **Élasticité et extensibilité:** les ressources en cloud peuvent être provisionnées ou dé-provisionnées par l'augmentation ou la diminution de la demande des utilisateurs.  
La puissance de calcul, de mémoire et d'autres installations peuvent être agrandie ou réduite selon l'exigence de l'utilisateur.
- **Facilité d'utilisation:** il n'est pas nécessaire de posséder et d'entretenir le matériel, les logiciels et autres ressources par le cloud utilisateur.  
Les services de cloud computing sont directement accessibles à l'aide d'un navigateur Web.  
Pas de ressources supplémentaires sont nécessaires pour faire fonctionner et exécuter des services de cloud computing, Un bureau simple avec connexion Internet normale est suffisant.
- **Lieu et Localisation indépendant:** Étant donné que le service de cloud computing peut être consulté via le navigateur Web, il peut être consulté à partir de n'importe où géographiquement et depuis n'importe quel appareil qui prend en charge l'interface web.  
Un service de cloud computing peut être consulté comme un service Web.
- **Coût réduit:** pour faire une entrée dans une entreprise, le coût nécessaire pour les infrastructures est réduit en déplaçant vers le cloud. Comme la puissance de calcul, de stockage et d'autres ressources sont utilisées de cloud; coût d'achat ainsi que de gérer eux est grandement affectée.
- **Multi-location:** un seul serveur de données des ressources, de l'informatique et d'autres sont partagés entre plusieurs utilisateurs en utilisant la virtualisation et de l'isolement.  
Cette fonctionnalité qualifiée de multi-location, permet une utilisation efficace des ressources.
- **Fiabilité:** de multiples ressources sont disponibles, comme la puissance de calcul, de stockage, etc.  
Pour fournir des services aux utilisateurs.  
De plus, les données peuvent être stockées à plusieurs endroits par le fournisseur. Cette redondance en termes de stockage de données et autres ressources permet la fourniture de reprise après un sinistre et permet d'obtenir la fiabilité et la disponibilité des données ainsi que des services.

Avec de tels avantages pour l'utilisation d'applications de cloud computing et de services, il ya des obstacles qui agissent comme une barrière dans sa croissance. Ils comprennent:

- Verrouiller : verrouiller pourrait être pour les données et les fournisseurs.

Données verrouille: les données stockées sur un cloud ne peuvent pas être facilement repris, si un utilisateur souhaite modifier un fournisseur de cloud. Elle peut être due à un manque d'API standardisée. Il en résulte un problème de données verrouille.

Vendeur verrouille: un fournisseur de cloud donne des services en termes d'API. API fait pour un fournisseur de cloud n'est pas utile pour un autre cloud fournisseurs. Si le changement de fournisseur est nécessaire, API doit également être modifiée. Ce problème est nommé comme vendeur verrouille.

- Disponibilité de service: Pour un utilisateur de cloud, le service devrait être disponible à tout moment. Chaque fois qu'un utilisateur demande un service cloud, le fournisseur doit assigner SLA (Service Level Agreement). Celle-ci définit les termes et conditions et spécifications pour les services de cloud computing.

Il comprend également le pourcentage de temps de service est disponible.

Un utilisateur de cloud s'attend à un service de haute disponible avec temps d'indisponibilité nul ou minime. Un fournisseur de cloud et son service correspondant sont sélectionnés en fonction de la disponibilité des services et les besoins des entreprises.

- Protection des données: Pour diverses organisations, les préoccupations concernant la sécurité, la confidentialité, la conformité et le contrôle sur leurs données sont un obstacle dans le déplacement vers le cloud. Les préoccupations particulières comprennent.

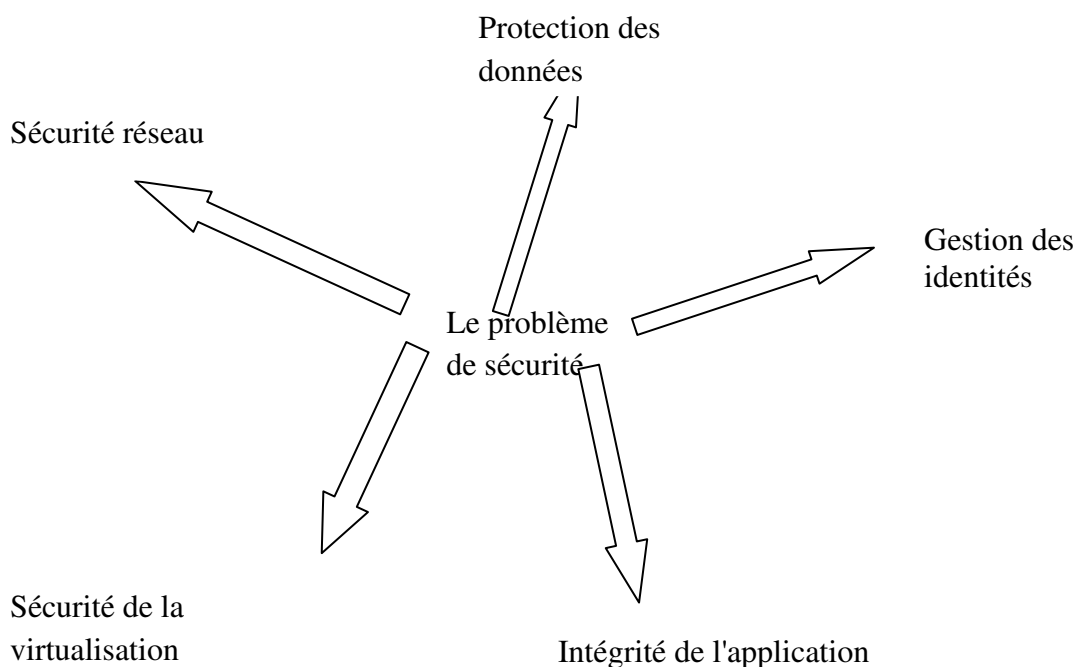
### **III.2. SÉCURITÉ DANS UN ENVIRONNEMENT CLOUD**

Dans le paradigme du cloud computing, un fournisseur de cloud crée, déploie et gère les ressources, les applications et les services.

Multi-location et la virtualisation sont les principales caractéristiques pour faire une utilisation efficace des ressources existantes et leur application. Un grand nombre d'utilisateurs sont desservie par un fournisseur de cloud par ce concept de partage des ressources.

La protection des données, la communication, la gestion des ressources pour l'isolement, la virtualisation, etc., sont quelques-unes des problèmes de sécurité qui se pose en raison de la multi-location et de la virtualisation dans l'environnement cloud [Math 09].

Les principaux types de menaces de sécurité dans le cadre de l'application de cloud sont brièvement décrits ci-dessous.



**Figure 15 : Classification des problèmes de sécurité**

### **III.2.1. Protection des données**

L'infrastructure de cloud computing est partagée entre plusieurs utilisateurs à n'importe quel moment. Les données utilisateur sont stockées et traitées dans l'environnement partagé dans un cloud qui est sous le contrôle du fournisseur. Les données utilisateur peuvent être altérées par entité malveillante autre dans le cloud. Le manque de transparence sur l'emplacement de stockage des données dans l'environnement cloud, problème de réglementation en raison de traverser la frontière de stockage, etc. rend l'exigence de la intimité et protection des données dans l'environnement de cloud plus important.

### **III.2.2. Sécurité des applications**

Le logiciel d'application fonctionnant sur ou en cours de développement pour les plates-formes de cloud computing pose des différents problèmes de sécurité.

La flexibilité, l'ouverture et la disponibilité publique de l'infrastructure cloud sont des menaces pour la sécurité des applications.

Les vulnérabilités existantes, telles que la présence de trappes, des problèmes de débordement, le code de mauvaise qualité, etc. sont des friandises pour les différentes attaques.

Multi-locataire environnement de plates-formes cloud, le manque de contrôle direct sur l'environnement et l'accès aux données par le fournisseur de la plate-forme cloud; sont les principaux enjeux pour utiliser d'une cloud application.

Préserver l'intégrité des applications en cours d'exécution sur des machines distantes est un problème ouvert.

### **III.2.3. Sécurité réseau**

Un cloud computing peut être de type public ou privé en fonction de l'accessibilité des services.

Les Service et les applications sont accessibles à partir d'emplacements distants dans un environnement de cloud computing.

La disponibilité continue des services de cloud computing, sans interruption, déni de service et autres attaques sont des problèmes de sécurité des réseaux. Distribuée par déni de service, attaque Signature d'emballage etc. crée des risques de transmission de données dans le réseau cloud.

La technologie de virtualisation a de graves répercussions sur la sécurité du réseau.

Réseau invisible créé par des serveurs virtuels, il est difficile de surveiller le trafic réseau et les performances.

Contrôles standard de sécurité réseau ne sont pas suffisantes pour contrôler le trafic VM et leur contrôle des tâches.

### **II.2.4. Sécurité de la virtualisation**

La technologie de virtualisation introduit de nouvelles attaques avec l'hyper viseur et les composants de gestion.

La multi-location en infrastructures de Cloud Computing pour le partage des ressources physiques entre les machines virtuelles (Virtual Machine), peut donner lieu à l'humaine dans le milieu du l'attaque (man in the middle attack) au moment de l'autorisation pour un service.

Parce que les machines virtuelles peuvent rapidement être ramenés aux cas précédents, et se déplacer facilement entre les serveurs physiques, il est difficile d'atteindre et de maintenir une sécurité constante.

### **III.2.5. Gestion des identités**

Les identités sont générées pour accéder à un service en cloud par le fournisseur de services de cloud computing. Chaque utilisateur utilise son identité pour accéder à un service en cloud.

L'accès non autorisé aux ressources et applications cloud est un enjeu majeur. Une entité malveillante peut usurper l'identité d'un utilisateur légitime et accéder à un service en cloud.

Beaucoup de ces entités malveillantes acquérir les ressources cloud mené à une non-disponibilité d'un service pour l'utilisateur actuel.

En outre, il peut arriver que l'utilisateur passe sa frontière au moment de l'utilisation du service dans l'environnement cloud. Cela pourrait être en termes d'accès à la zone protégée dans la mémoire ou d'effectuer toute autre opération qui ne sont pas maintenus dans la liste de contrôle d'accès pour une ressource spécifique et de l'application. Ce système de gestion des identités pour fournir l'authentification et l'autorisation est un problème à la fois du fournisseur et l'utilisateur dans un environnement de cloud computing.

Les problèmes de sécurité sont un domaine actif de recherche et d'expérimentation. Beaucoup de recherches sont en cours pour répondre aux problèmes tels que la sécurité des réseaux, protection des données, la virtualisation et l'isolement des ressources.

Répondre à ces problèmes, il faut obtenir la confiance de l'utilisateur pour les applications de cloud computing et des services.

L'obtention de la confiance des utilisateurs peut être réalisée en créant la confiance de la ressource de cloud et les applications, ce qui est un problème crucial dans le cloud computing.

### **III.3. TRAVAUX CONNEXES A TROUVER DES SOLUTIONS DANS LE CLOUD COMPUTING**

La principale préoccupation dans les environnements cloud est d'assurer la sécurité de la multi-location et l'isolement, donnant aux clients plus le confort en plus d'idée de cloud.

La sécurité à différents niveaux tels que le niveau du réseau, niveau de l'hôte et le niveau d'application est nécessaire de garder le cloud fonctionnent en permanence.

Conformément à ces différents niveaux, différents types de failles de sécurité peuvent se produire qui ont été classés dans cette section.

### III.3.1. La Sécurité de base

Web 2.0 est une technologie clé pour permettre l'utilisation de Software as a Service (SaaS) soulage les utilisateurs des tâches telles que la maintenance et l'installation de logiciels [Saur 09].

Les attaques par injection SQL, sont celui dans lequel un code malveillant est inséré dans un code standard SQL. Donc les attaquants accèdent sans autorisation à une base de données et sont capable d'accéder aux informations sensibles.

Parfois, les données d'entrée du hacker est mal comprise par le site web que les données de l'utilisateur et lui permet d'être accessible par le serveur SQL et cela permet à l'attaquant d'ont un savoir-faire sur le fonctionnement du site et de faire changements dans cela.

Diverses techniques telles que: éviter l'utilisation de générer dynamiquement SQL dans le code, l'utilisation de filtrage techniques pour assainir l'entrée de l'utilisateur, etc. sont utilisés pour vérifier l'attaques par injection SQL.

Cross Site Scripting (XSS) attaques, qui injectent scripts malveillants dans le contenu Web sont devenus très populaires depuis la création du Web 2.0.

Il existe deux méthodes pour injecter le code malveillant dans la page Web affichée à l'utilisateur: Stocké XSS et réfléchi XSS.

Une autre classe d'attaques très populaires au SaaS est appelée comme Man in the Middle attacks (MITM).

Dans une telle attaque, une entité tente de s'immiscer dans une conversation en cours entre un expéditeur et un client d'injecter de fausses informations et d'avoir connaissance des données importantes transférées entre eux.

Divers outils de mise en œuvre des technologies de chiffrement fort comme: dsniff, Cain, Ettercap, Wsniff, etc Airjack ont été développé afin de fournir protéger contre eux.

Les différents niveaux de sécurité sont nécessaires pour garantir l'application correcte du cloud computing telles que: la sécurité d'accès du serveur, la sécurité d'accès internet, la sécurité d'accès a la base de données, la sécurité de la confidentialité des données et la sécurité d'accès au programme.

### **III.3.2. Sécurité des données**

Un cloud sécurisé, c'est aussi assurer la protection des données dans le cloud. Certaines techniques pour assurer la sécurité des données sont analysées comme suit.

*Siani Pearson* examine les politiques et les procédures d'évaluation des méthodes de l'intimité et des outils. [Pear 09]

Protection des renseignements personnels en matière de respect des lois et la confiance des utilisateurs, les fuites de données pour les données sensibles sont fournis.

*Ji Hu Klein* a donné un point de repère pour sécuriser les données en transit dans le cloud. La protection des données lors de la migration est traitée via indice de référence pour les frais généraux de cryptage et de sécurité [Ji Hi 09].

Plus de cryptage est souhaitable pour la sécurité forte, mais elle nécessite plus de calculs. Ainsi, un point de repère donne l'équilibre de la sécurité et les frais généraux de cryptage [Tets 09].

Un modèle de sécurité basé sur l'agent pour contrôler les données de canal caché est présenté. Cela peut résoudre le problème de la fuite de données dans l'environnement cloud.

*Descher* discuté la question de la confidentialité des données en conservant le contrôle à l'utilisateur et augmenter la confiance. [Desc 09]

### **III.3.3. Protection du réseau**

Les niveaux de sécurité de réseau d'infrastructure peuvent être renforcés par la mise en œuvre de Domain Name System Security pour le développement de déni de service de la prévention et de la tactique de sécurité du routeur. [Dian 09]

### **III.3.4. Protection contre les attaques aux différents niveaux**

*Jensen* donnent les fondements de problème de sécurité techniques qui consistent en des services de sécurité Web en utilisant XML et les messages SOAP et la sécurité de couche transport utilisant le protocole SSL. [Jens 09]

Diverses attaques sont considérées comme emballage signature XML, basé sur un navigateur, l'injection malware cloud, l'usurpation des métadonnées et des inondations.

*Arshad* proposent une méthode pour assurer la qualité de service pour les charges de travail de calcul intensif en terme d'attaque de sécurité, l'algorithme de chiffrement et d'authentification. Efficace de prévention d'intrusion et de détection au moment de l'acquisition de ressources sont fournies [Arsh 09].

VM attaque spécifique, la protection porte dérobée, l'intégrité du système d'exploitation client, etc. sont considérées comme des exigences de sécurité.

### III.2.2. Présentation de Priyank Singh et Ranjita Singh et Mukul Manmohan [Pri 11]

#### III.3.5.1. Idées de base

Cette approche a été proposée par Priyank Singh Hada et Ranjita Singh et Mukul Manmohan à International Journal of Computer Applications [Pri 11]

Dans cette structure, ils proposent un trust modèle basé sur des agents de sécurité, qui sont des simples agents mobiles qui assurent la sécurité au niveau de la machine virtuelle et le point d'entrée du cloud réseau à cloud clients et les fournisseurs de services pour la gestion de leurs ressources et des données en toute sécurité et efficacement. Ces agents mobiles, non seulement fournissent des mesures de sécurité, mais aussi assurent la comptabilité et le suivi des activités dans la machine virtuelle si son état normal ou malveillant, de sorte que le client est tenu au courant de ses données. En cas de conditions alarmantes, le client en est informé et peut prendre les mesures nécessaires requis.

Les agents de sécurité sont postés à trois endroits dans l'environnement de cloud computing qui sont l'entrée de gamme, le Domain 0 et Domain U de Xen.

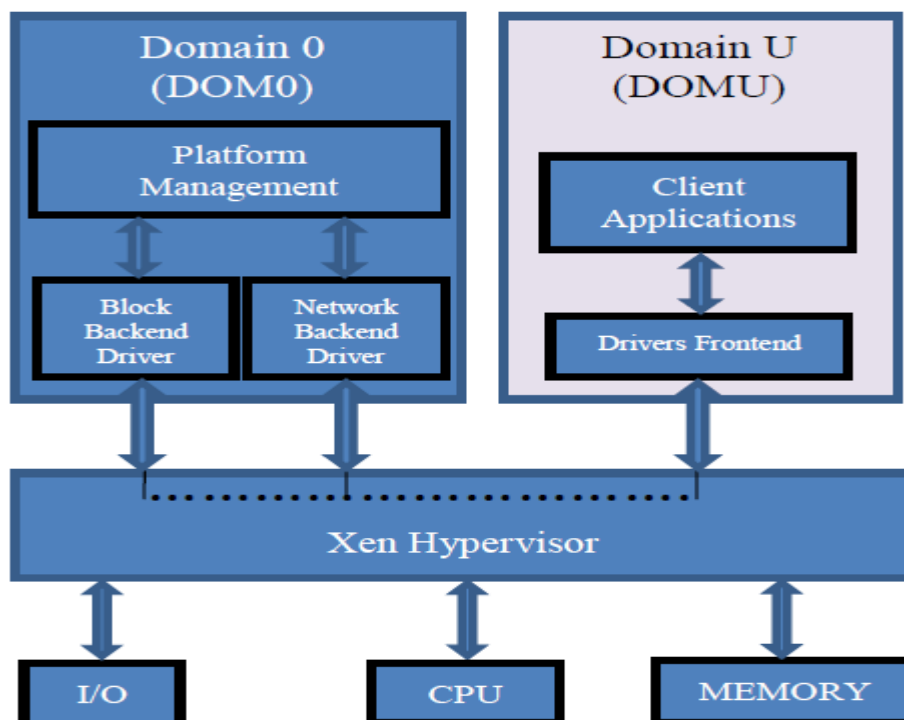


Figure 16 : L'architecture de machine virtuelle Xen

### III.3.5.2. Architecture du système

Dans le modèle proposé, ils implémentent la sécurité à trois points de contrôle dans l'environnement cloud.

Le premier point de contrôle se trouve au niveau de la communication dans le réseau, la seconde à l'hyper viseur dans la machine virtuelle et finalement dans chaque cas particulier de machine virtuelle affectée à chaque utilisateur.

Pour réaliser ce scénario trois agents mobiles sont utilisés dans le modèle agent mobile1 (MB1), agent mobile2 (MB2) et agent mobile3 (MB3).

Agent MB1 est placé au point d'entrée de cloud et il est directement sous le contrôle du client, agent MB1 est utilisé pour l'établissement des communications sécurisées entre le fournisseur et le client du cloud pour échanger des données en toute sécurité et de gestion des ressources cloud pour le compte du client.

Un autre agent de MB2 est employé à DOM0 machine virtuelle. En fait MB2 agent mobile traque toutes les actions effectuées par le fournisseur de services et fournisseur d'infrastructure dans DOM0, et signale tout comportement anormal de MB1.

MB2 vérifie également l'intégrité des pilotes et des applications exécutées sur DOM0.

Enfin, l'agent MB3 est déployé à chaque instance de machine virtuelle du client et le suivi de toutes les activités de DOMU machine virtuelle et surveille les activités malveillantes de partage de machines virtuelles et aussi DOM0.

Dans le cas d'un comportement anormal de la machine, il envoie une notification à MB1, qui donne le contrôle de données vers client.

Ainsi DOMU est protégée contre l'environnement partagé ainsi que DOM0 machine virtuelle.

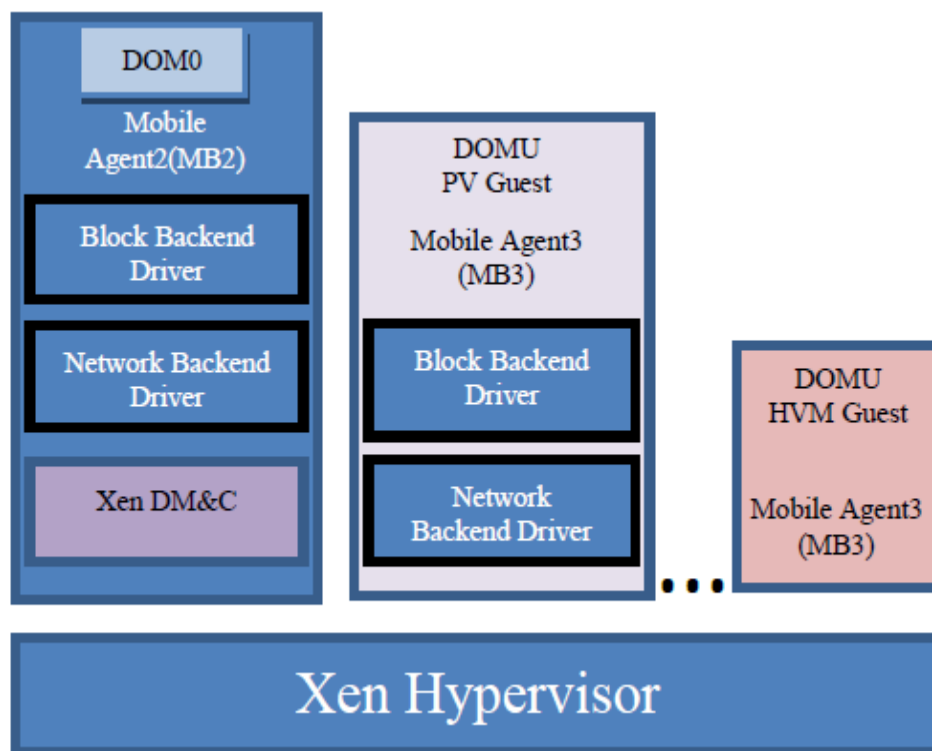
Cette communication entre les agents est entièrement cryptée et sécurisée de scénarios d'attaque.

Une clé de session est générée pour la communication entre les agents.

Il contrôle également le flux de données dans le cloud.

Dans la figure ci-dessous, les emplacements des MB2 et MB3 agents mobiles sont montrés avec leurs labels respectifs.

L'hyper viseur contrôle la communication entre les applications clientes et les machines virtuelles.



**Figure 17 : Les emplacements des MB2 et MB3 dans le système**

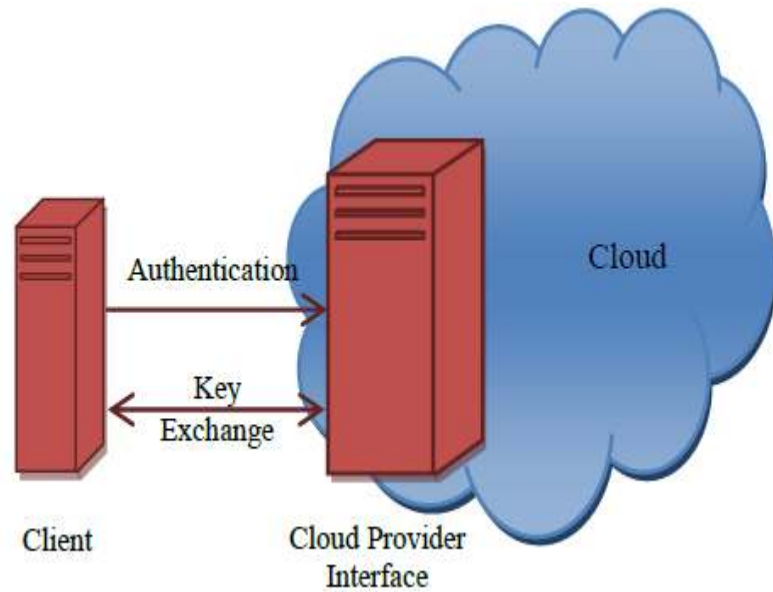
### III.3.5.3. METHODE PROPOSEE

Dans cette section, nous discutons le modèle proposé.

Environnement de cloud ici est considéré comme ayant trois composantes qui sont client, fournisseur de services cloud et de pool de ressources qui est sous le contrôle du fournisseur d'infrastructure.

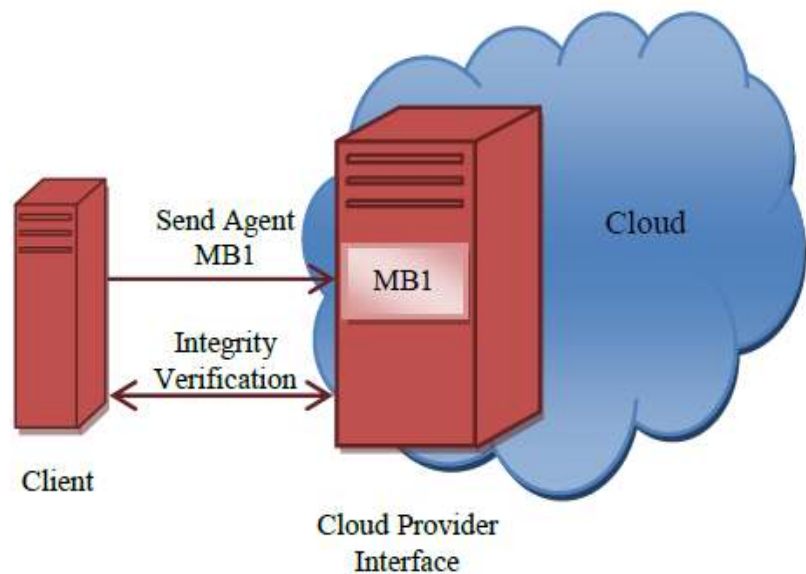
L'ensemble de la procédure peut être résumée dans les étapes suivantes

- Fournisseur de services Cloud et le client doivent s'authentifier mutuellement. Ceci peut être réalisé par mot de passe, nom d'utilisateur ou un autre mécanisme. Clé SSL est établie pour une communication sécurisée.



**Figure 18 : L'authentification**

- Après la procédure d'authentification ci-dessus, MB1 est transféré du client vers le site du fournisseur de service. Client et le serveur doit vérifier l'authenticité et l'intégrité des agents mobiles.



**Figure 19 : vérifier l'intégrité**

- MB1 est activé et établit une nouvelle clé de session avec le côté client. Cette clé est gardée secrète du fournisseur de

service qui est ensuite utilisé pour sécuriser les communications et cache de données du fournisseur de services cloud.

- MB1 demande de ressources du fournisseur de services cloud sur le nom du client selon la condition et la charge de service. MB1 contrôle également l'utilisation des ressources et de poster un chèque fournisseur de cloud computing pour les faux usages.
- Le fournisseur de services cloud alloue VM et d'autres ressources en fonction de la demande. Un nouvel agent MB2 mobile est généré et envoyé à la plate-forme lorsque les nouvelles ressources sont fournies. MB2 agent mobile est installé dans DOM0 et contrôle l'authenticité et l'intégrité de la plate-forme, les logiciels et les pilotes. Il est utilisé pour contrôler le comportement des DOM0 et attestation de plate-forme. Il enregistre tous les pilotes et le matériel d'attestation et d'authentification.
- Un agent de MB3 plus mobile est installé dans chaque machine virtuelle allouée au client. MB3, s'inscrit dans MB1 agent mobile. MB3 génère une nouvelle paire de clés de communication avec MB1. MB3 exécute la tâche de sécuriser la communication entre les machines virtuelles et MB1 (côté client) et le comportement du moniteur d'applications synchronisées. Si les ressources sont allouées à plus d'un hyper viseur, d'MB2 et MB3 distincts sont installés à chaque DOM0 et DOMU.
- Enfin, lorsque l'utilisateur ferme la connexion, MB3 informe MB1 pour retirer les ressources allouées et la déconnexion de l'association.

### **III.4. LA CONCLUSION**

Bien que le Cloud computing peut être vu comme un phénomène nouveau qui va révolutionner la façon dont nous utilisons Internet, il ya beaucoup à faire preuve de prudence.

Il ya beaucoup de nouvelles technologies émergentes à un rythme rapide, chacun des progrès technologiques et le potentiel de rendre la vie de l'homme est plus facile.

Cependant, il faut être très prudent pour comprendre les risques de sécurité et les défis dans l'utilisation de ces technologies. Le cloud computing n'est pas une exception.

Dans Ce chapitre nous avons présenté l'état de l'art utile sur des travaux développés dans la sécurité du cloud computing, Pour cela nous avons présenté une architecture basée

Agents mobiles pour utilisées le cloud computing en sécurités.

Le cloud computing a le potentiel pour devenir un précurseur dans la promotion d'une sécurité, virtuel et économiquement viable comme solution informatique à l'avenir.

## Chapitre IV

### Proposition d'un environnement de cloud computing

---

#### IV.1. Introduction

Le Cloud computing fournit des services élastiques, de haute performance et de stockage de données évolutives pour un grand quotidien augmenter le nombre d'utilisateurs.

Le Cloud computing agrandi le domaine des systèmes informatiques distribués en offrant des services Internet avancés complètent et complète des fonctionnalités de l'informatique distribuée fournis par le Web, grid computing et Peer-to-Peer.

En fait, les systèmes de cloud computing offrent une infrastructure à grande échelle pour l'informatique a des performances élevées qui snt dynamiquement s'adapte à l'utilisateur et les besoins d'application

Aujourd'hui nuages sont principalement utilisés pour la manipulation très charges de calcul intensif et de fournir de très grandes des installations de stockage de données. Ces deux buts sont combinés avec le troisième objectif de réduire potentiellement les coûts de gestion et d'utilisation. Un autre paradigme informatique distribué basée sur des agents multiples qui sont capables d'interagir avec comportement intelligent.

Les systèmes multi-agents sont souvent utilisés pour résoudre des problèmes en utilisant une approche décentralisée où plusieurs agents contribuent à la solution en coopérant l'un avec l'autre.

Bien que plusieurs différences existent entre Cloud computing et des systèmes multi agents, ils sont les deux des modèles distribuée de calcul, donc plusieurs problèmes communs peuvent être identifiés et plusieurs avantages peuvent être obtenus par l'intégration de l'utilisation des systèmes de cloud computing et multi-agents.

Nous avons présenté dans les chapitres précédents un état de l'art sur les systèmes du cloud computing, la technologie des agents et des agents mobiles

Dans ce chapitre, nous proposons notre architecture qui va permettre d'assurer la sécurité dans le cloud computing basée sur la technologie d'agents mobiles.

#### IV.2. Objectif et motivation du travail

##### IV.2.1. Objectif

Notre architecture est basée sur les agents mobiles qui ont comme but de gardé la sécurité dans un cloud computing.

Ce travail vise essentiellement la proposition d'une architecture qui permet de répondre aux besoins d'utilisateur à travers d'accéder à un cloud computing par des agents mobiles, en nous appuyant sur la capacité de la mobilité des agents.

Dans ce chapitre nous allons voir la conception de notre système ; nous commençons par la motivation de choix d'approche basée agent mobile pour la recherche d'information ensuite, nous présentons la description de l'architecture globale proposée et son fonctionnement ainsi que le rôle de chacun de ces composants pour garder la sécurité dans un cloud computing. Nous terminons ce chapitre par une conclusion.

#### **IV.2.2. motivation**

Le modèle client/serveur est certainement le modèle le plus utilisé pour la construction des applications. Ce modèle possède l'inconvénient d'augmenter le trafic sur le réseau et exige une connexion permanente. Pour ce la, nous proposons une nouvelle approche qui utilise les agents mobiles, l'une des applications plus importante de ce dernier est le cloud computing et ses services. Comme on a dit dans l'introduction de notre travail basé agents mobiles, il est conçue pour garder la sécurité dans un cloud computing, Avec le modèle agent mobile notre système est capable de faire un service ou exécuter un but de client, mobile et accessible puisque les agents mobiles est migre d'un site à autre pour atteindre des demandes des clients, adaptable puisque il répond au besoin demandé.

La technologie d'agent mobile a été largement utilisée dans plusieurs domaine, cette utilisation est motivée par les points suivants :

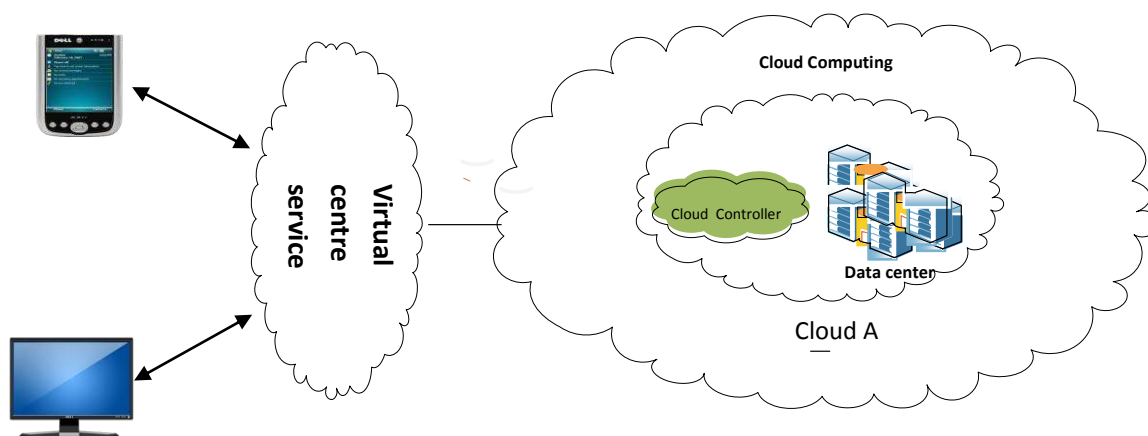
- La mobilité d'agent lui permet de déplacer la charge de calcul d'un site à un autre.
- les agents sont capables de communiquer et coopérer entre eux, ce qui accélère et facilite la recherche.
- La mobilité d'agent permet à un client d'interagir localement avec un serveur, et donc de réduire le trafic sur le réseau.
- L'exécution d'agents spécialisés offre d'avantage de souplesse que l'exécution et de robustesse que la transaction.
- L'asynchronisme et l'autonomie des agents leur permettent de réaliser une tâche tout en étant déconnecté du client, ce qui est particulièrement utile dans le cas de supports physiquement mobiles.
- Les agents sont capables d'exécuter un service d'une façon plus intelligente, où en transféré et obtenu les résultats selon des concepts.

Dans la suite de ce chapitre, nous proposons notre architecture qui va permettre d'exécuter un service ou accéder à des ressources dans le cloud computing avec toutes sécurités basée sur la technologie d'agents mobiles.

### IV.3. Description de l'architecture proposée

L'approche proposée est une approche basée agents mobiles conçue pour le cloud computing pour assurer la sécurité.

Notre approche désigne un mode de communication de données sécurisée entre le fournisseur de Service Cloud et les utilisateurs dans un cloud computing



**Figure 4.1 Un environnement de cloud computing.**

Tous les clients et les fournisseurs de services sont censés être honnête dans la phase d'enregistrement, Après que la phase d'enregistrement est terminée, aucun client ou serveur est sécurisé. Les clients ont besoin de vérifier eux-mêmes au cours de l'accès aux services de cloud computing en fournissant des données d'identification exactes pour accéder a des services via un serveur virtuel, Quand un utilisateur veut envoyer une demande au fournisseur de cloud computing, ils transmet les données d'authentification correctes (telles que nom d'utilisateur et mot de passe défini pour lui par le système) par l'intermédiaire d'un agent interface au serveur virtuel situé dans notre approche.

L'agent d'interface est un agent stationnaire qui s'exécute sur un périphérique de l'utilisateur et fournit une interface graphique pour interagir directement avec notre serveur virtuel de service.

L'agent d'interface est utilisé pour transmettre la requête du client à l'agent médiateur dans notre application, et afficher les résultats vers le client via l'agent médiateur.

L'agent analyseur offre les services Directory pour les clients comme l'autorisation, privilèges, Identifiant de l'utilisateur des mots de passe etc. pour authentifier et autoriser à utiliser les services de cloud computing.

Lorsque la demande transite par le l'agent médiateur ver l'agent analyseur est passé avec succès, la demande sera envoyée à l'agent médiateur pour traiter la demande.

L'agent analyseur analyse les types de service demandée et les types d'accès de services de l'utilisateur, Si l'analyse de cette information est positive la demande sera envoyer a l'agent médiateur pour traiter la demande et créer un agent mobile et passé la demande au contrôleur des taches d'un cloud spécifié.

Le rôle principal de l'agent médiateur est de gérer la création, l'enregistrement, l'événement et la suppression de chaque agent transfert.

L'agent médiateur n'obtient que des informations relationnelles de l'agent analyseur. Cette information détermine le cas de la création d'agent transfert et le service à fournir aux utilisateurs.

L'événement de la création, l'activité et la suppression de l'agent transfert sont régulièrement généré par l'agent médiateur en fonction de la décision de l'agent analyseur.

L'agent transfert peut être utilisé d'une part pour transférer la demande du client à partir de notre région du contrôleur des taches, et d'autre part doit transférer le résultat sélectionné par le contrôleur des taches à l'agent médiateur Lorsque cette demande d'information atteint le contrôleur des taches, elle est immédiatement transmise à l'agent de sécurité situé dans le cloud computing pour vérifier l'information d'accès et le degré de sécurité de la région d'où cette demande est venue. L'agent vérifie ensuite toute l'information fournie par le serveur virtuel et puis envoie le résultat au contrôleur des taches.

Si la réponse est positive la demande sera envoyé a l'agent exécuter qui a la responsabilité d'exécuter le travail et retourner la résultat au contrôleur des taches.

Le contrôleur des taches transmet les informations demandées par l'utilisateur par l'intermédiaire de l'agent médiateur par demandera l'agent transfert et l'agent médiateur de transmettre les informations demandées a l'agent interface après l'analyse de cette information par l'agent analyseur, puis à l'utilisateur à partir cette agent.

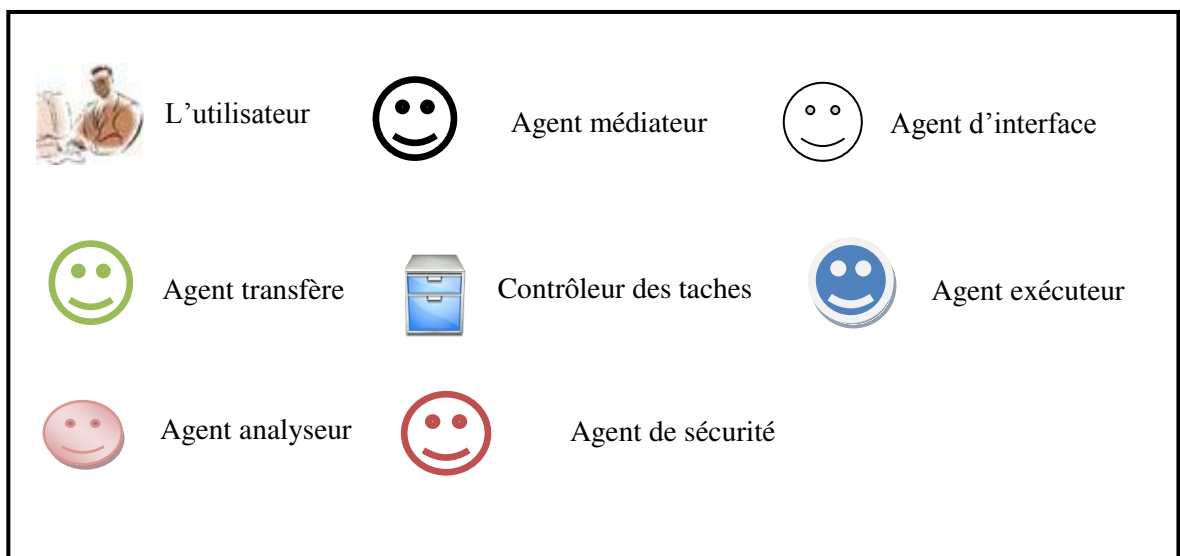
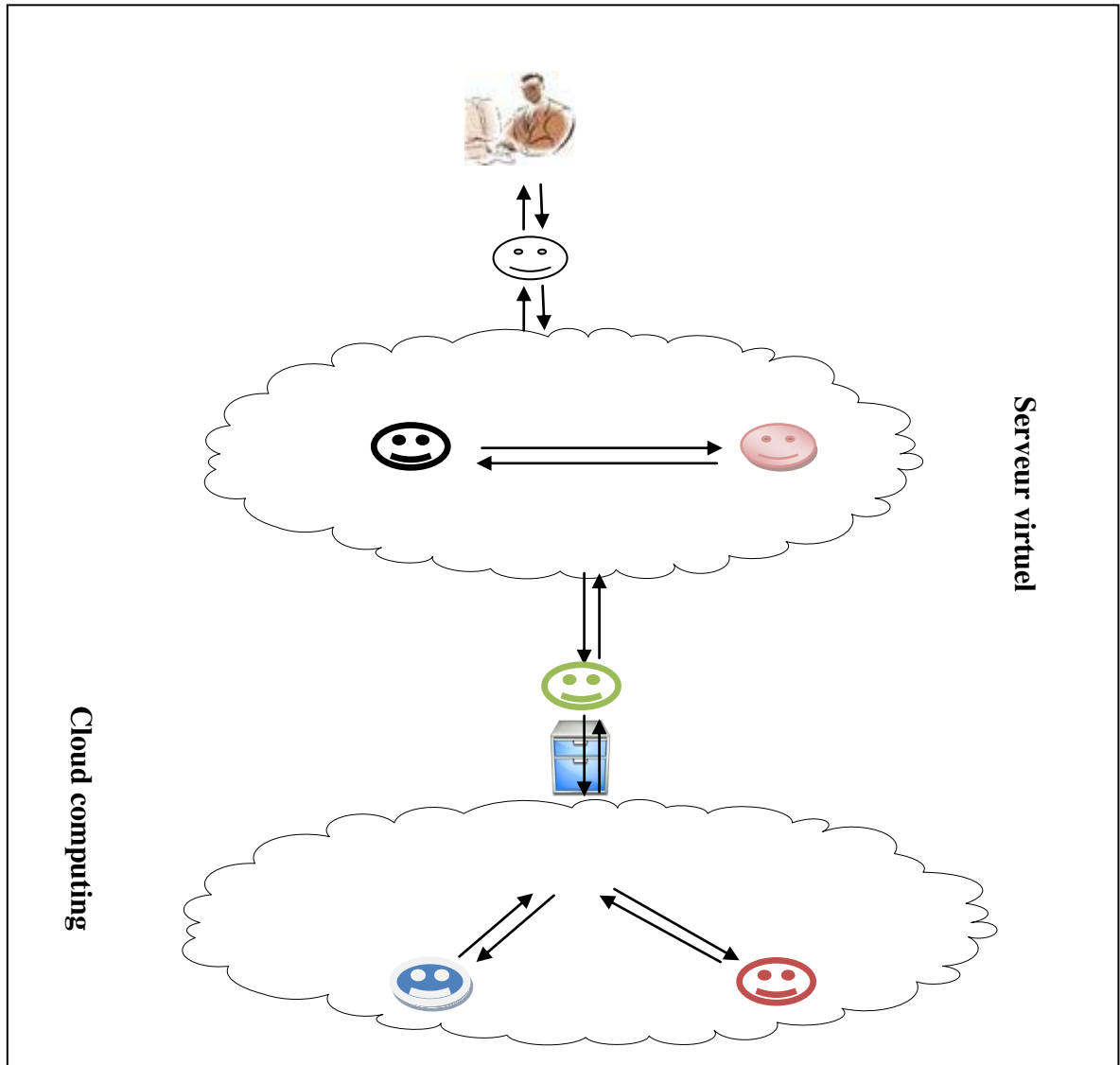


Figure 4.2 Représentation du cloud computing basé agent mobile.

## IV.2.2. L'architecteur proposée l'architecteur générale

L'architecture générale du notre système est illustrée à la figure 4.3, elle s'articule autour de quatre principales couches en interaction.

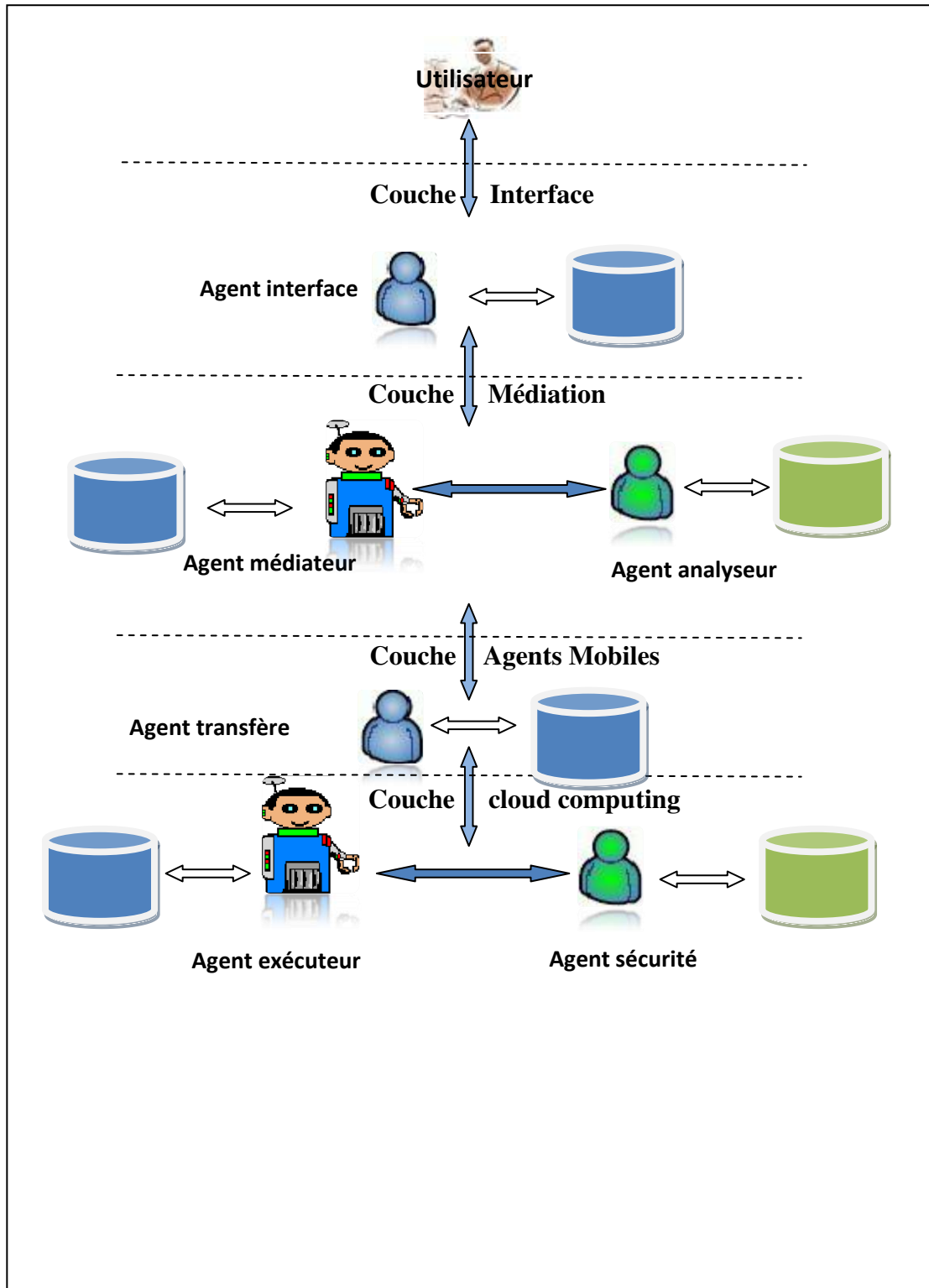


Figure 4. 3 Représentation des couches du notre système.

#### IV.4.1.1. Couche Interface

Cette couche contient l'application qui permet au client d'interroger le système.

Son rôle essentiel est de capturer le but de l'utilisateur afin de répondre le mieux possible à son besoin. Elle comprend des agents d'interface qui interagissent avec l'utilisateur pour l'aider à réaliser une tâche bien précise. Cette interaction se traduit par une transformation des demandes de l'utilisateur, transmise ensuite à l'agent médiateur.

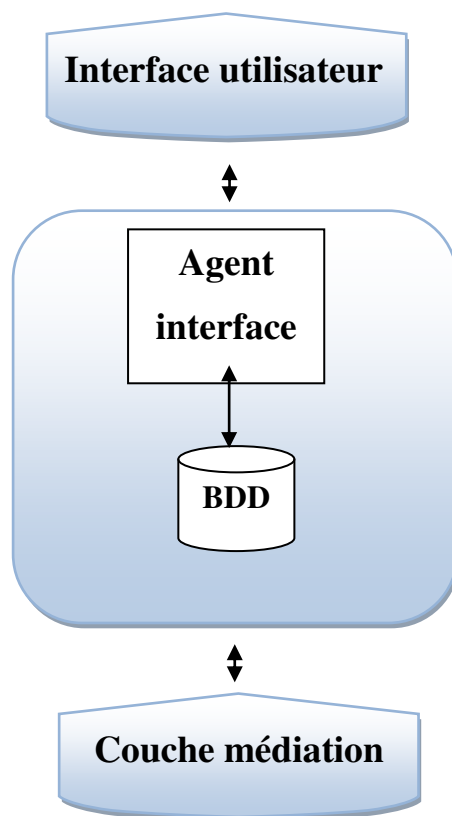
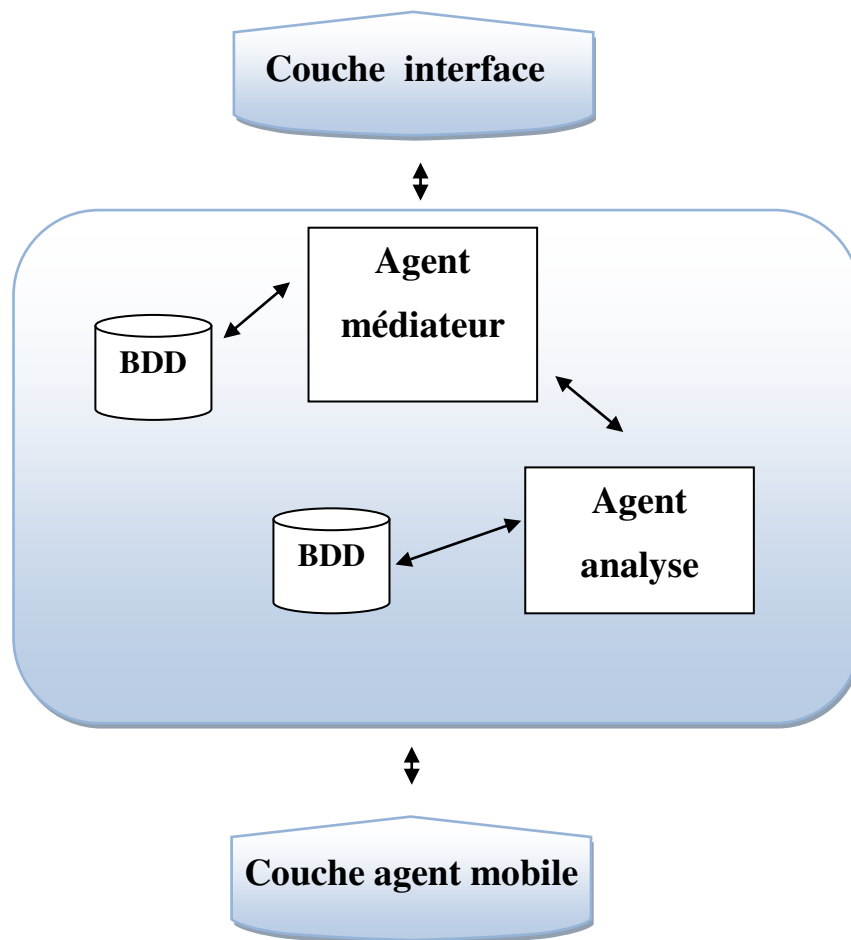


Figure 4.4 Représentation de la couche interface.

#### IV.4.1.2. Couche Médiation

Cette couche regroupe tous les éléments nécessaires à l'exécution de processus d'une requête écrite par l'agent interface, elle fournit aussi des services comme la création des agents mobiles en fonction des demandes des utilisateurs. Elle dispose aussi d'un répertoire de base de données qui stocke les adresses de tous les contrôleurs des tâches dans cloud computing accessible, et peut recevoir des informations sur des services disponibles, ou envoyer à d'autres agents mobile fournissant les services nécessaires pour réaliser son but.

À la fin de processus l'agent médiateur et l'agent analyseur analysent et rassemblent les informations trouvées sous forme d'un service qui représente les services demandés par l'utilisateur.



**Figure 4.5 Représentation de la couche médiation.**

#### **IV.4.1.3. Couche Agents Mobiles**

Cette couche contient l'ensemble des agents de transfert générés par l'agent médiateur. A ce niveau, et pour chaque demande de service, l'agent médiateur va activer un des agents transfert créés. Les agents transferts sont des agents mobiles envoyant au contrôleur des tâches de cloud computing pour trouver des services convenables à chaque demande de service reçue sur différentes cloud computing.

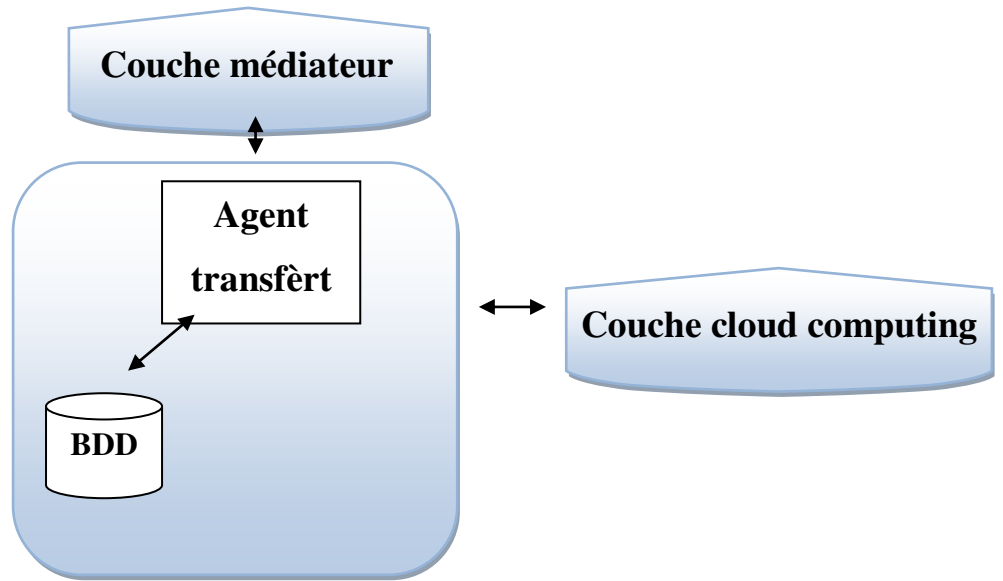


Figure 4.6 Représentation de la couche agent mobile.

#### IV.4.1.4. Couche cloud computing

C'est une classe capable de recevoir des demandes de service, de les traiter et de retourner les résultats. Cette couche regroupe l'ensemble des agents contrôleur des tâches ; des agents sécurités et des agents exécuter qui sont répartis sur le cloud computing pour gérer l'ensemble de service associés à lui.

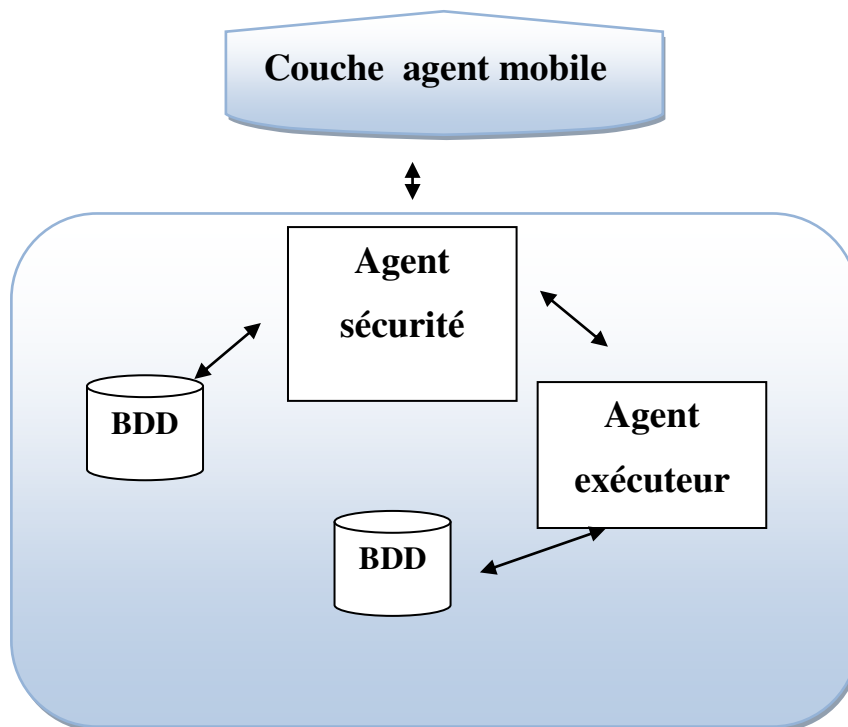


Figure 4.7 Représentation de la couche cloud computing.

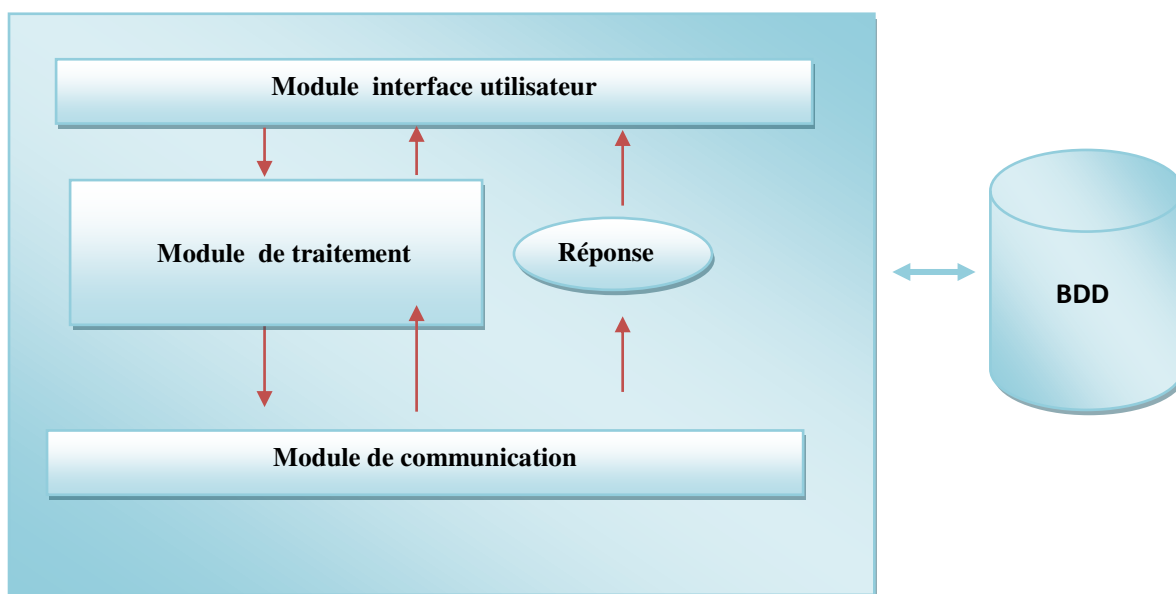
## IV.5. Spécification des agents

Nous allons décrire dans ce qui suit la structure de notre système et les fonctionnalités de ces différents agents et qui sont :

- Agent interface : c'est un agent qui simule notre client dans le cloud computing
- Agent médiateur : cet agent s'occupe de la gestion de notre système et il est utilisé pour traiter la demande de partie de l'utilisateur qui inclut la création et gère la création de l'agent transfère.
- Agent analyseur : cet agent s'assure de la sécurité de l'utilisateur par vérification l'information d'accès et l'identification de chaque client pour un service demande.
- Agent transfère : L'agent transfère peut être utilisé d'une part pour transférer la demande du client à partir de l'agent médiateur vers le cloud computing, et d'autre part transférer le résultat sélectionné par le cloud computing.
- Agent contrôleur des tâches: cet agent s'occupe de la gestion de système du cloud computing et il est utilisé pour traiter la demande de partie d'un serveur virtuel.
- Agent de sécurité : cet agent maintient la sécurité dans le cloud computing en Vérifient le degré de sécurité du service demander et vérifier d'où la demande arrivée.
- Agent exécuter : c'est un agent qui simule le cloud computing ou offreur de service dans le cloud computing.

### IV.5.1. Agent Interface

Cet agent peut être vu comme un simplificateur permettant aux utilisateurs d'interagir avec le système, l'agent d'interface est un agent stationnaire qui s'exécute sur un périphérique de l'utilisateur et fournit une interface graphique pour interagir directement avec notre système, Il est responsable principalement d'acquiescer toutes les requêtes des utilisateurs, envoyer ces requêtes aux agents adéquats et présenter les résultats aux utilisateurs.



**Figure 4.8 Architecture de l'agent interface.**

Cet agent est composé de

- L'interface utilisateur  
Ce module permet l'interaction avec l'utilisateur pour que celui-ci puisse interagir avec notre système.  
Cette interface fournit un ensemble de services ; l'utilisateur doit choisir parmi ces derniers. Une fois son choix fait, l'agent peut établir son profil qui est estimé dans le module d'acquisition des requêtes.  
Après le traitement le résultat est fourni à travers cette même interface.
- Module de traitement  
Le rôle de ce module est d'analyser les données des utilisateurs de la cloud computing, ce module permet aussi de collecter les informations à partir de l'interface graphique et après la collection de ces informations il est créé la requête structurée et l'envoie à l'agent médiateur.
- Le module de communication  
Le module de communication est responsable de l'interaction de l'agent avec les autres agents du système pour obtenir, envoyer, recevoir des informations. Par exemple envoyer à l'agent médiateur un message l'informant de l'existence d'une demande de service
- La base de connaissance  
Dans ce module est stocké toute la connaissance de l'agent :
  - Les informations transmises par l'agent interface et les services demandant au l'agent médiateur
  - Le profil de l'utilisateur actuel.
  - L'historique de l'étape de l'accès des utilisateurs au système et les informations sur les services offerts

#### IV.5.2. Agent médiateur

L'agent médiateur est un agent stationnaire intelligent qui traite la demande de service , Il joue le rôle d'interface entre l'utilisateur qui pose une requête ou demande de service ou ressource et l'ensemble des sources de cloud computing qui offre les services pour chaque demande l'agent médiateur va générer un agent mobile pour se déplacer vers le contrôleur des tâches de cloud computing afin de trouver les services adéquates à la demande acquise. Lorsque l'exécution des demandes est terminée, le médiateur rassemble les résultats trouvés par l'agent mobile sous forme d'un service qui représente le service demandée par l'utilisateur. L'agent médiateur à la faculté de contrôler les motivations des agents transferts qui arrivent chez lui ce qui permet d'augmenter le niveau de sécurité du système.

L'agent médiateur doit assurer les fonctionnalités suivantes :

- Il envoie à l'agent Analyseur l'information d'authentification et les requêtes des utilisateurs
- Créer et initialiser les agents mobiles.
- Demander la migration des agents mobiles vers les cloud computing.
- Il envoie à l'agent Analyseur les résultats des agents mobiles.
- Recevoir les réponses de l'agent Analyseur.

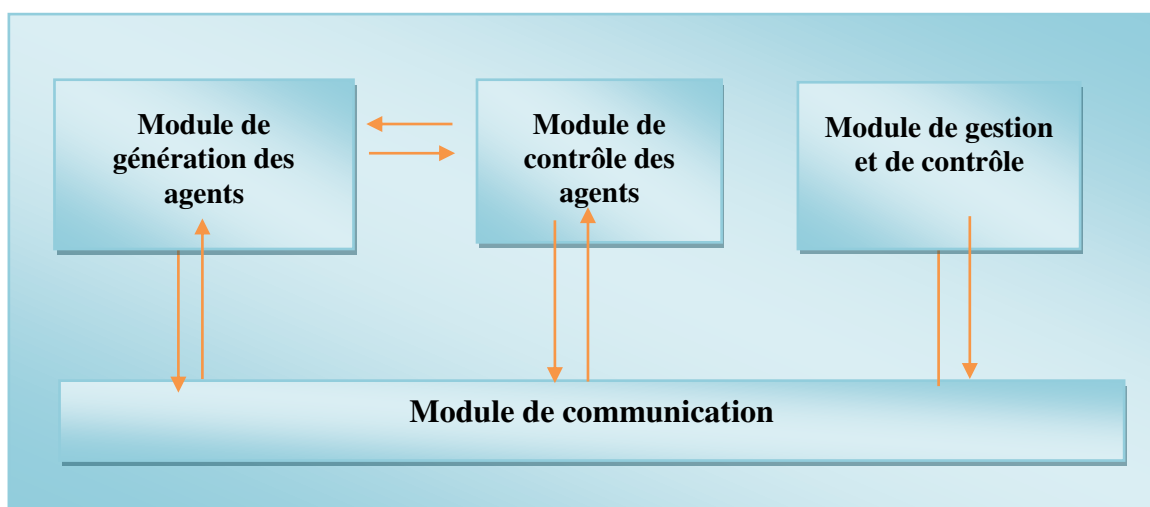


Figure 4.9 Architecture de l'agent médiateur.

Cet agent est composé de :

- **Module de communication**  
À travers ce module, l'agent médiateur communique avec les autres agents, c'est un module d'échanges des messages : la construction, l'envoi et aussi la réception des messages.
- **Module de gestion et de contrôle**  
Ce module est chargé de prendre en charge toutes les activités en relation à l'exécution d'une requête d'un utilisateur. Il gère aussi la liste des utilisateurs et leur cloud et la planification et l'ordonnancement des tâches assignées à ces cloud.
- **Module de génération des agents**  
Ce module permet de générer des agents mobiles et définir l'endroit où ils démarrent leurs exécutions. Une fois l'agent est créé, l'exécution de son code est déclenchée. Chaque agent créé à un identificateur unique ce qui va permettre de localiser l'agent.
- **Module de contrôle des agents**  
Ce module permet à l'agent médiateur de contrôler tous les agents créés. Un système à agents mobiles doit offrir des mécanismes permettant une exécution sécurisée des agents dans le système.

#### IV.5.3. L'agent analyseur

Cet agent communique avec l'agent médiateur pour analyser l'information d'authentification et pour analyser la requête et la réponse du système à la requête des utilisateurs. Donc son rôle est en fonction des demandes des services par l'utilisateur, l'agent analyseur sélectionne une liste de cloud provider qui possède des informations sur le service demandé, avec l'aide d'une base de données et lorsque l'exécution de la requête est terminée, l'analyseur rassemble les résultats trouvés par l'agent transfère et l'agent analyseur ensuite renvoie le résultat à l'agent médiateur.

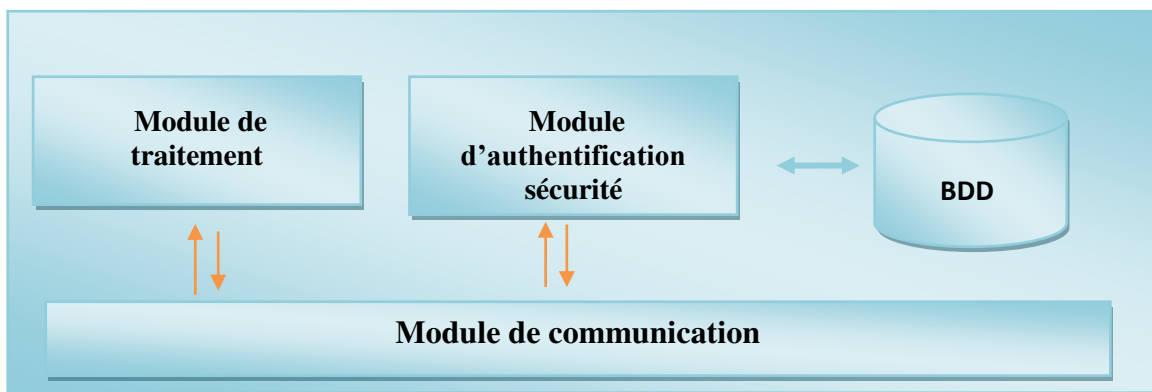


Figure 4.10 Architecture de l'agent analyseur.

Un agent analyseur est donc composé de:

- Le module de communication  
Le module de communication est responsable de l'interaction de l'agent analyseur avec l'agent médiateur pour recevoir des informations sur l'authentification de chaque utilisateur et pour analyser et envoyer les résultats à l'agent médiateur.  
Par exemple envoyer à l'agent médiateur le résultat obtenu par l'agent transfère d'une demande de service.
  
- Le module de traitement  
Ce module est responsable de l'analyse des informations ou des services demandant par l'agent médiateur et analyser l'authentification de chaque utilisateur et l'analyse des informations ou des services fournis par l'agent transfère.
  
- Le module d'authentification sécurité  
Le rôle de ce module est d'authentifier les utilisateurs de cloud computing et le cloud computing correspondant a chaque utilisateur  
Pour authentifier les utilisateurs et les cloud computing ce module se sert des informations stocké dans une base de données.
  
- La base de données  
Dans ce module est stocké toute la connaissance de l'agent analyseur, les informations sur les services, les utilisateurs et leur authentification et les fournisseurs des services « cloud computing » et l'historique des services offert par chacun.

#### IV.5.4. Agent transfère

L'agent transfère dans notre système représente l'agent mobile, Il est créé par l'agent médiateur qui lui indique la tâche à effectuer. Ces agents de transfère migrent vers un cloud computing spécifié, pour satisfaire la requête par l'interaction avec le contrôleur des taches située dans un cloud computing spécifié, et ensuite il retourne vers l'agent médiateur, pour passer la réponse à l'agent analyseur. Enfin, il se détruit lui-même.

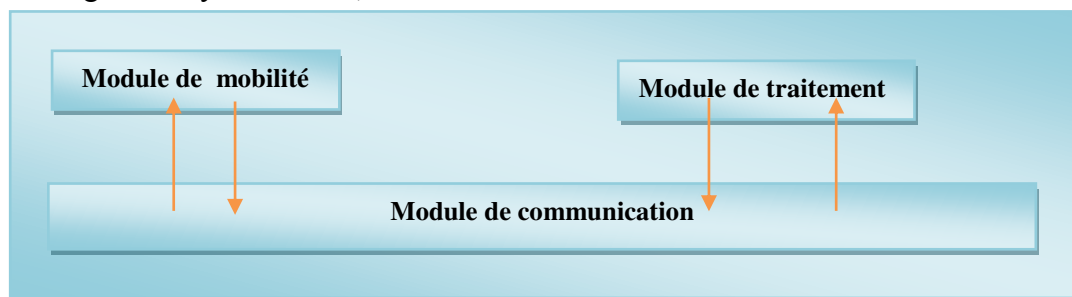


Figure 4.11 Architecture de l'agent mobile.

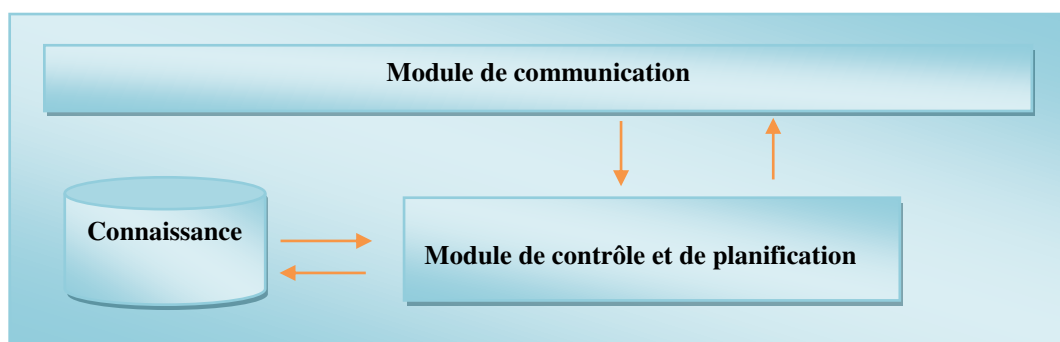
Un agent mobile est donc composé de :

- **Module de communication**  
Comme ses équivalents dans les autres agents, ce module est responsable du traitement des messages entrant et de l'élaboration des messages sortants. A travers ce module, l'agent mobile de transfère communique avec les autres agents et contrôleur des taches de cloud computing.
- **Module de gestion de mobilité**  
Ce module permet de gérer la gestion de migration d'un agent en cours d'exécution d'un cloud à un autre à travers le réseau.
- **Module de traitement**  
À partir de ce module, l'agent mobile doit éventuellement pouvoir exécuter un processus de traitement (lancement d'exécution de la requête spécifique au client). Ici, un avantage réside dans l'autonomie des agents et leur capacité à traiter les informations sur le cloud où elle se trouve. et dans ce module les résultats sont stockés et conservés par l'agent mobile pour être traités et puis transmis à l'agent médiateur.

#### IV.5.5. Contrôleur des taches

Le rôle de cet agent est de maintenir l'ordre dans le cloud computing, il fournit toutes les informations nécessaires pour chaque agent. Par exemple prévenir les agents sécurité d'une arrivée d'un agent transfère. Il contrôle et gère tous les autres agents du cloud telle que agent exécuteur et agent sécurité.

L'agent contrôleur des taches est un agent temporaire, chargé de réaliser le service demandé par le client.



**Figure 4.12 Architecture de l'agent contrôleur des taches.**

La structure de l'agent contrôleur des taches comprend les modules suivants

- **Le module de communication**  
Comme ses équivalents dans les autres agents, l'agent contrôleur des taches communique avec l'agent exécuteur concernant l'avancement d'exécution des tâches. Par exemple, un agent transmet son état au contrôleur pour l'informer

qu'il attend le résultat d'un autre agent pour achever sa tâche et qu'il faut lui signaler une fois que la tâche est prête.

- **Module de contrôle et de planification** : ce module responsable prendre en charge toutes les activités en relation à l'exécution d'un service au serveur virtuel.

Il gère aussi la liste des partenaires membre de cloud computing et la planification et l'ordonnancement des tâches assignées à ces partenaires.

- **Les connaissances** : c'est une base de données comportant la liste des partenaires des cloud computing avec leurs tâches correspondantes. Cette liste sera modifiée à chaque fois qu'un partenaire quitte cloud computing, ou la tâche d'un partenaire défaillant est réaffectée à un autre, etc. En plus les connaissances incluent toutes les informations concernant la planification dans cloud computing, c'est à dire, le plan global de l'agent contrôleur des tâches, les règles de résolutions des conflits entre agents de cloud et les contraintes à respecter pour la réalisation de chaque service.

#### IV.5.6. Agent sécurité

Le rôle de cet agent est de préserver la sécurité, l'intégrité des données et l'authentification des partenaires de la cloud computing

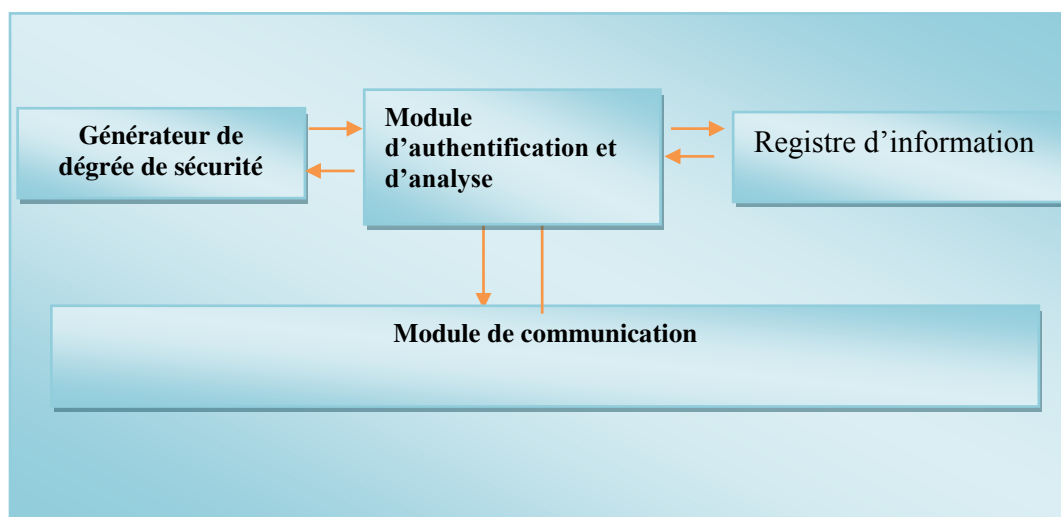


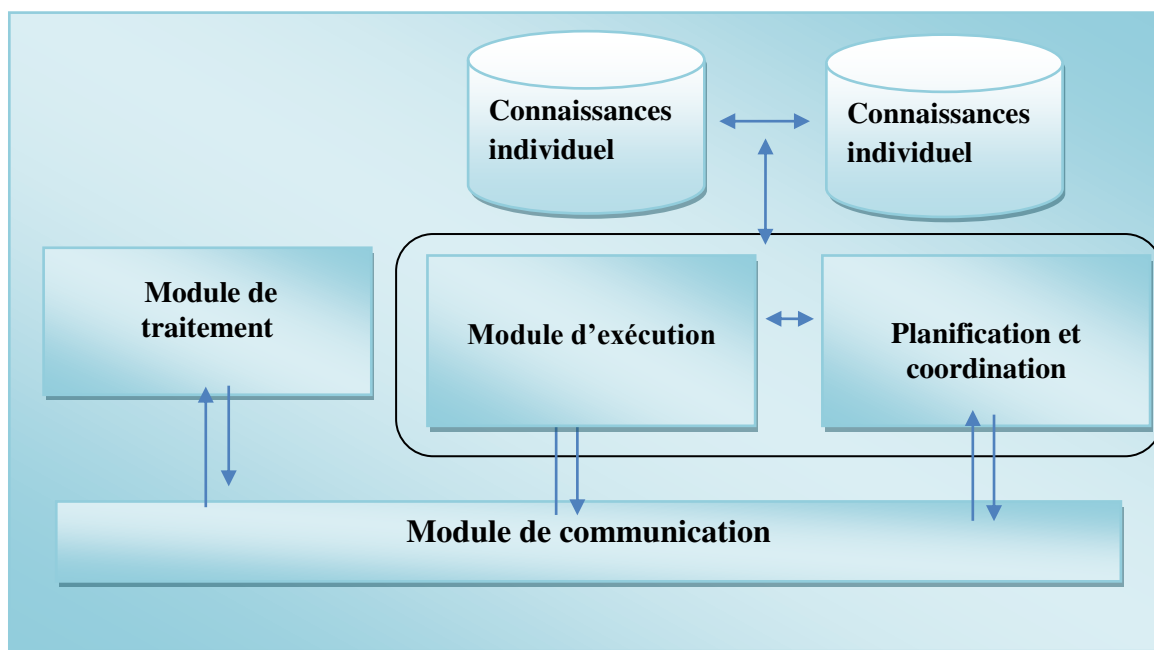
Figure 4.13 Architecture de l'agent sécurité.

Un agent analyseur est donc composé de :

- Le registre d'information  
Ce registre contient toutes les informations concernant les serveurs virtuels (nom d'enregistrement et mot de passe) pour qu'il puisse les authentifier lors de leur connexion, Ce registre contient aussi les informations sur les données échangées entre agents (signature des données) pour qu'il puisse vérifier leur authenticité et il connaît aussi l'autorisation de chaque serveur virtuel
  
- le module d'authentification et d'analyse  
Le rôle de ce module est d'authentifier les serveurs virtuels du cloud computing, et l'autre partenaire et d'analyser les données et de fournir le degré de sécurité à tout serveur virtuel connectée avec cloud computing. Pour authentifier les utilisateurs ce module se sert des informations stocké dans le registre d'information, la même chose pour l'analyse des données. La degré de sécurité est fournis par le générateur de degré de sécurité
  
- Le générateur de degré de sécurité  
Ce module génère un degré de sécurité pour chaque serveur virtuel qui veut demander un service.

#### **IV.5.7. Agent exécuteur**

C'est un agent local au niveau du cloud computing, Il est responsable de répondre aux requêtes des agents mobiles arrivant à son cloud computing. L'agent exécuteur est représenté par une structure comportant un module de communication, un module de planification et de coordination et un module d'exécution et module de traitement. L'agent exécuteur possède deux autres modules de connaissances, qui sont : les connaissances individuelles et les connaissances du cloud computing



**Figure 4.14 Architecture de l'agent exécuteur.**

- Le module de communication : contient tous les processus de prise en charge des messages, à savoir : la réception, le filtrage et la traduction des messages entrants et la formulation et l'envoi des messages sortants.
- Le module de planification et de coordination c'est le module responsable de la gestion de la coopération et la formulation des offres pour répondre au but annoncé par l'agent contrôleur.
- Le Module d'exécution  
ce module ne contient pas les ressources du cloud computing mais, il contient les informations sur les ressources internes du cloud computing (application, usagers, sources de connaissances, etc.) qui permettent la réalisation des tâches locales assignées à cette cloud.  
Ce module a le rôle de réaliser la correspondance entre la tâche assigné à l'agent et la ou les ressources internes de le cloud aptes à l'achèvement de cette tâche.
- Connaissances du cloud computing : contient les informations concernant les règles organisationnelles et opérationnelles définies par le cloud, il comprend pour chaque cloud que cet agent fait partie d'elle, la liste des tous les agents membres. Ce module contient aussi des informations sur les droits et les obligations des agents dans le cloud computing.
- Connaissances individuelles : contient des informations sur l'agent lui-même ses capacités et compétences, l'état et la charge de travail actuelle, c'est-à-dire, pour chaque tache des indicateurs sont assignés pour déterminer la disponibilité et le coût de cette tache.

#### IV.6. Exemple de scénario

Dans cette section nous allons donner un exemple de scénario de déroulement d'un processus de demande de service dans notre cloud computing.

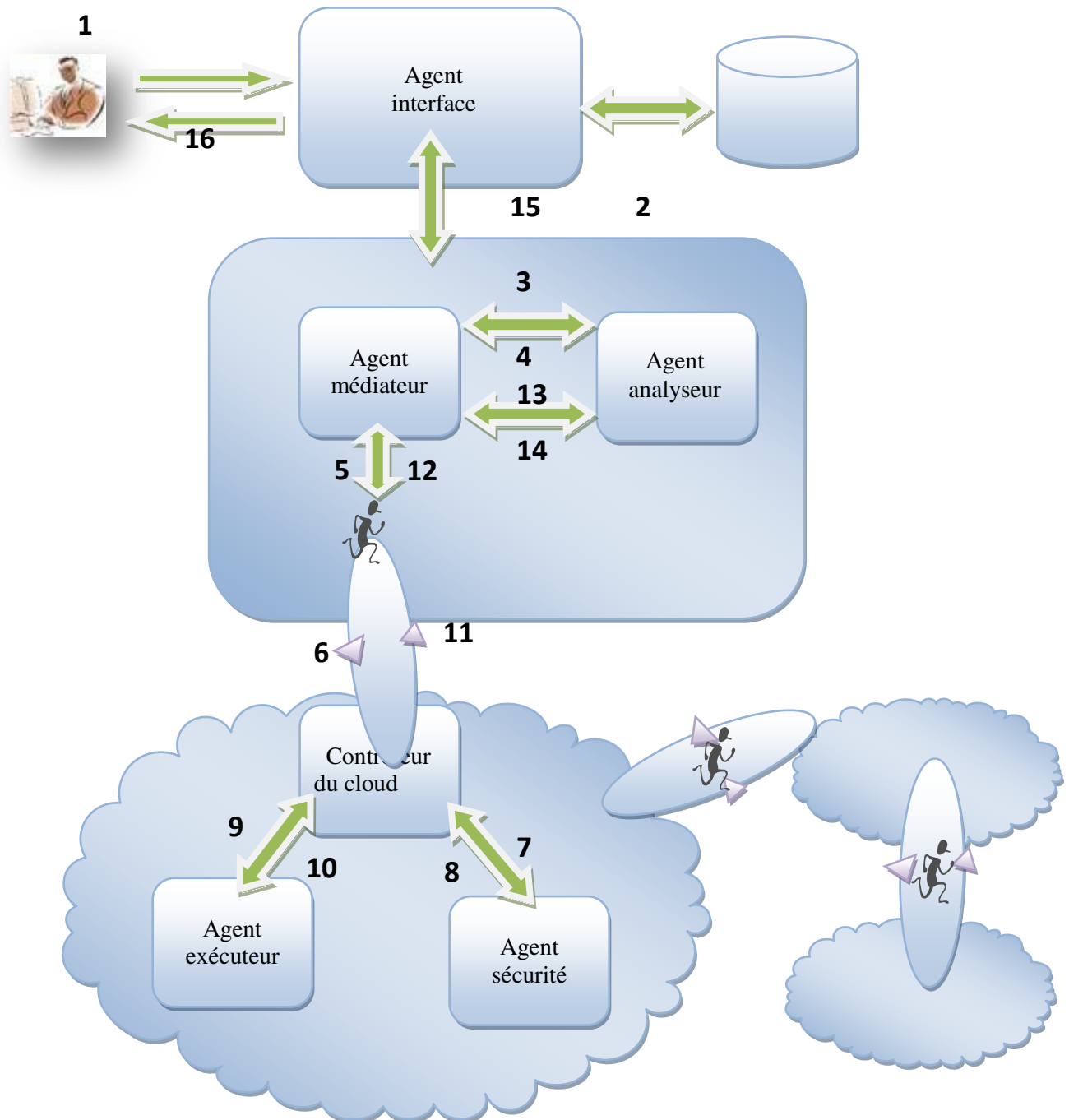


Figure 4.15 Représentation une architecture détaillée.

ETAPE 1 : Envoi de la requête client vers un agent interface.

ETAPE 2 : Lancement de l'agent interface et envoi la requête à l'agent médiateur.

ETAPE 3 : L'agent médiateur envoie la requête et l'information d'accès à l'agent analyseur.

ETAPE 4 : L'agent analyseur renvoie la réponse à l'agent médiateur, si l'information d'accès est correcte la demande sera passée sinon sera abandonnée.

ETAPE 5 : L'agent médiateur crée les agents de transfère.

ETAPE 6 : L'agent transfère migre vers le cloud computing pour transfère l'information ou le service demandé par le client.

ETAPE 7 : contrôleur des tâches envoie la requête à l'agent sécurité pour vérifier d'où la demande arrivée et vérifié l'information d'accès du lui et consultée le degré de la sécurité de notre système ou la demande arrivée.

ETAPE 8 : L'agent sécurité renvoie la réponse au contrôleur des tâches, si l'information d'accès est correcte et notre système ou l'autre cloud qui demande un service sont confiance la demande sera passée sinon sera abandonnée.

ETAPE 9 : Lancement de l'agent exécuter

ETAPE 10 : L'agent exécuter exécute le service demandé

ETAPE 11 : le contrôleur des tâches envoie le résultat à l'agent transfère et l'agent transfère migre vers notre système.

ETAPE 12 : L'agent transfère envoie résultat à l'agent médiateur.

ETAPE 13 : L'agent médiateur envoie les résultats à l'agent analyseur.

ETAPE 14 : L'agent analyse trié les résultats envoyée par l'agent médiateur et renvoie la réponse a cet agent.

ETAPE 15 : L'agent médiateur envoie le résultat à l'agent interface.

ETAPE 16 : Affichage du résultat au client.

## **IV.7. Fonctionnement du système**

On rappelle que l'approche qu'on va présenter a pour but de trouver ou d'exécuter des services dans un système cloud computing à partir des requêtes de l'utilisateur, tout en utilisant les agents mobiles comme un moyen de communication dans la réalisation de ce système.

Dans ce chapitre nous avons présenté une description de l'architecture proposée pour assurer la sécurité dans le cloud computing en utilisant la technologie d'agents mobiles. Dans cette section, nous allons aborder les exigences que l'architecture proposée devrait prendre en compte pour assurer le bon fonctionnement du cette système.

Ces exigences sont :

- Le support de migration des agents.
- Le support de communication entre les agents.
- Le support d'interaction entre les agents.

#### **IV.7.1. La migration des agents**

Comme nous avons mentionné dans le premier chapitre la technologie d'agents mobiles a deux types de migration :

- (i) la migration forte.
- (ii) la migration faible.

Selon la difficulté d'implantation de la migration forte nous avons opté d'utiliser la migration faible dans notre architecture afin d'augmenter la performance de notre système.

Aussi la migration des agents est réactive, où la destination de chaque agent n'est pas déterminée par l'agent lui-même, mais elle est dictée par l'agent médiateur ou l'agent contrôleur.

Dans notre proposition, les données et le code de l'agent mobile sont déplacés avec l'agent lorsque celui-ci visite le cloud désiré. Lorsque l'exécution de l'agent mobile est terminée, il effectue une deuxième migration et retourne éventuellement vers l'agent médiateur afin de lui fournir les résultats de son exécution.

#### **IV.7.2. La communication entre les agents**

Un système à agents fournit en général trois primitives de communication permettant aux agents de communiquer entre eux, aussi bien qu'avec les services de plate-forme ou de l'environnement

Ces primitives de communication prennent la forme d'envois de messages, d'appels de procédures (méthodes), ou bien l'utilisation d'un blackboard. Par conséquent, les langages de communication entre les agents standardisés devront être fournis.

- la communication d'un agent mobile avec un agent statique.
- la communication entre agents mobiles
- la communication entre un agent et un environnement (client, serveur virtuel ou cloud computing).

Pour la communication entre agents mobiles il est nécessaire de trouver des mécanismes de localisation et de notification pour réaliser la mise en correspondance des agents.

La communication entre l'agent mobile et un environnement où un agent, les systèmes utilisent la communication classique par échange de messages ou blackboard.

### **IV.7.3. L'interaction entre les agents**

Pour bien comprendre le fonctionnement de notre architecture nous pouvons de choisir un langage de modélisation l'interaction globale des entités de système et ensuite les interactions détaillés entre les entités.

## **IV.8. Les diagrammes en AUML**

Le langage A-UML (Agent -UML) a été proposé par Bauer, Odell et alen 1999 comme une extension de la notation UML pour la modélisation d'agents. Cette extension concerne les diagrammes de séquence pour modéliser les protocoles d'interaction entre agents.

Donc, AUML est un mécanisme de modélisation des interactions entre agents. D'une part, les agents sont actifs, ils sont capables de prendre des initiatives et peuvent contrôler la communication entre eux. D'autre part, les agents coopèrent et coordonnent leurs travaux pour atteindre un but commun. 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 et c'est pour cette raison que nous avons utilisé AUML pour modéliser notre système.

Nous commençons d'abord par l'élaboration du diagramme de cas d'utilisation afin de montrer les acteurs du système ainsi que les cas d'utilisation qui existent

### **IV.8.1. Diagramme de cas d'utilisation**

La description des cas d'utilisation se fait par l'identification des scénarios qui peuvent être représentés au moyen d'un diagramme de séquence pour chacun, ce diagramme rend compte des événements qui définissent l'interaction de les agents entre eux et l'interaction des agents et le système (utilisateur, serveur virtuel et cloud contrôleur).

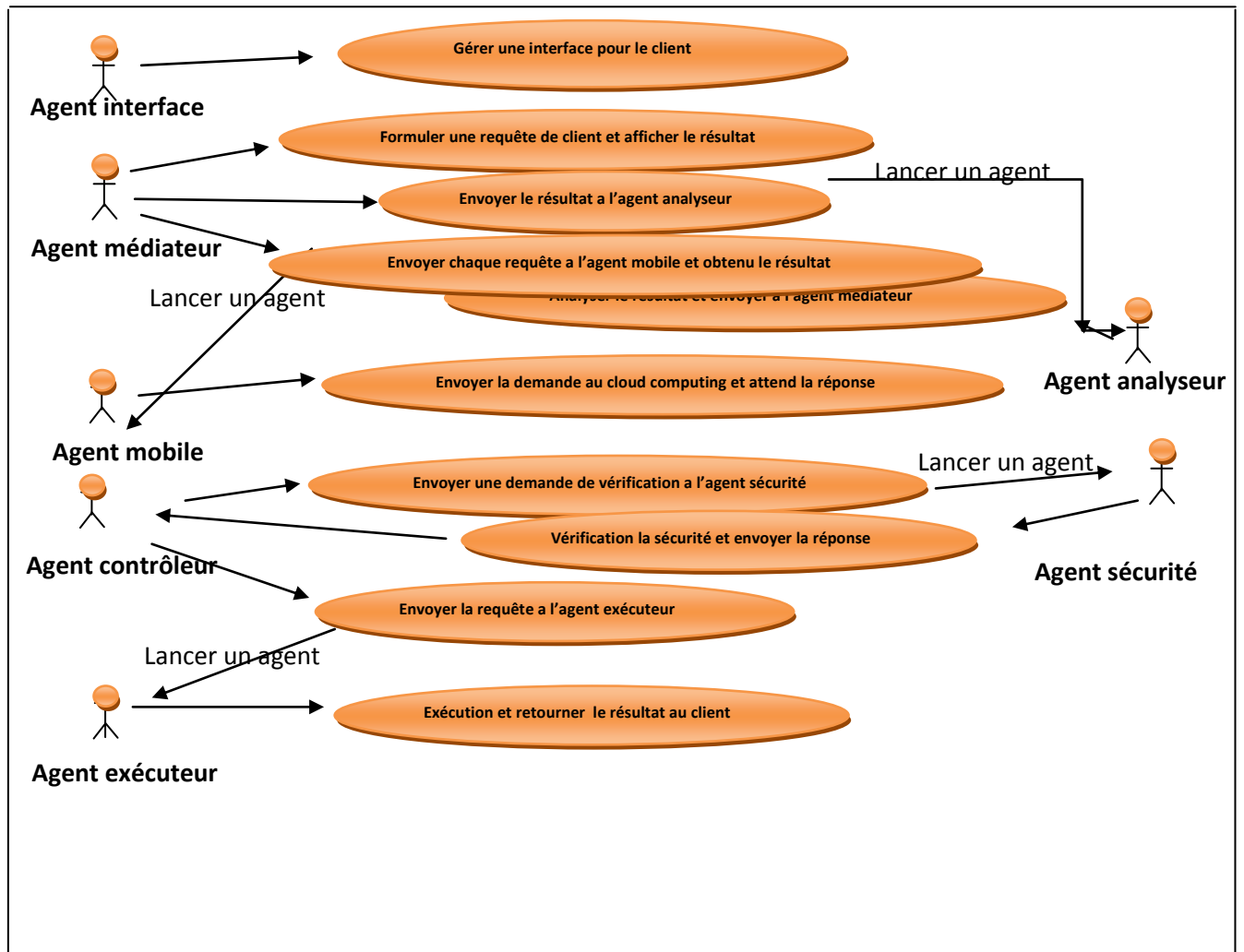
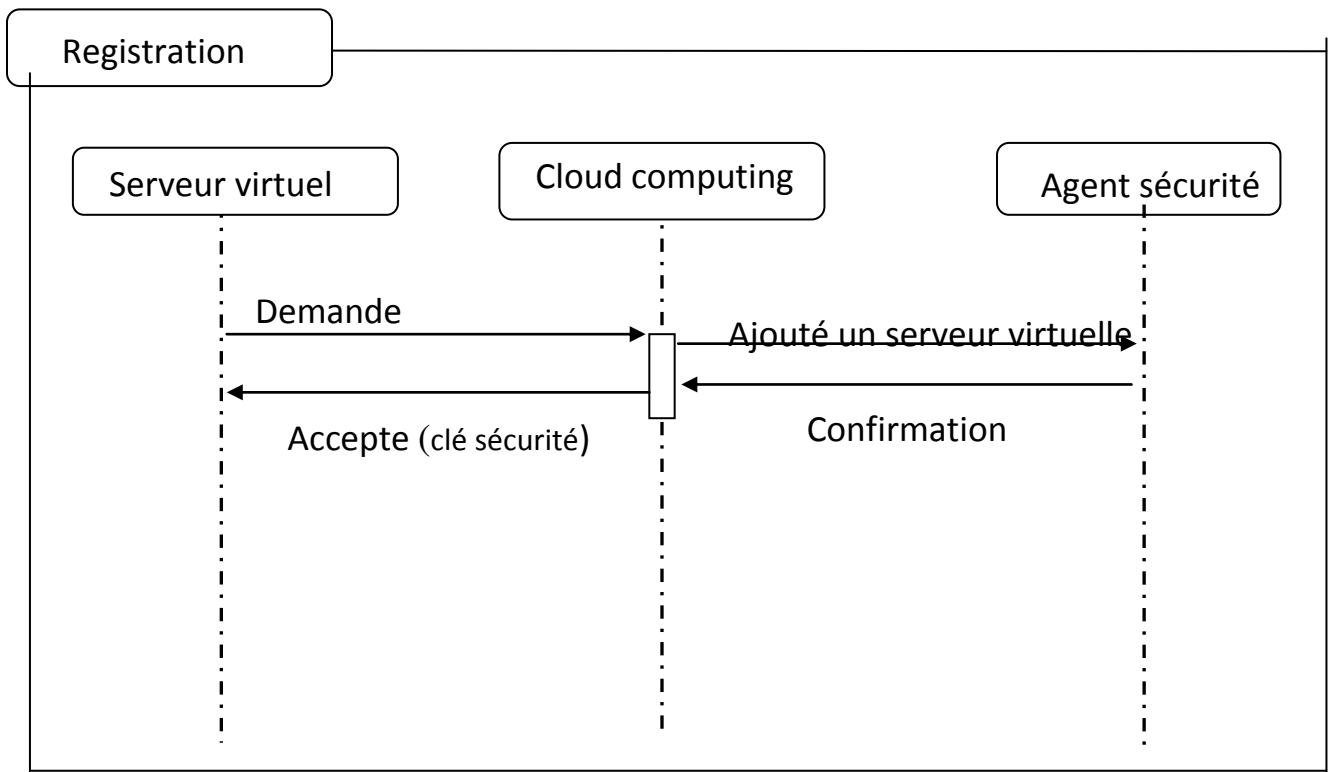


Figure 4.16 Diagramme de cas d'utilisation de système

#### IV.8.2. Les diagrammes des séquences

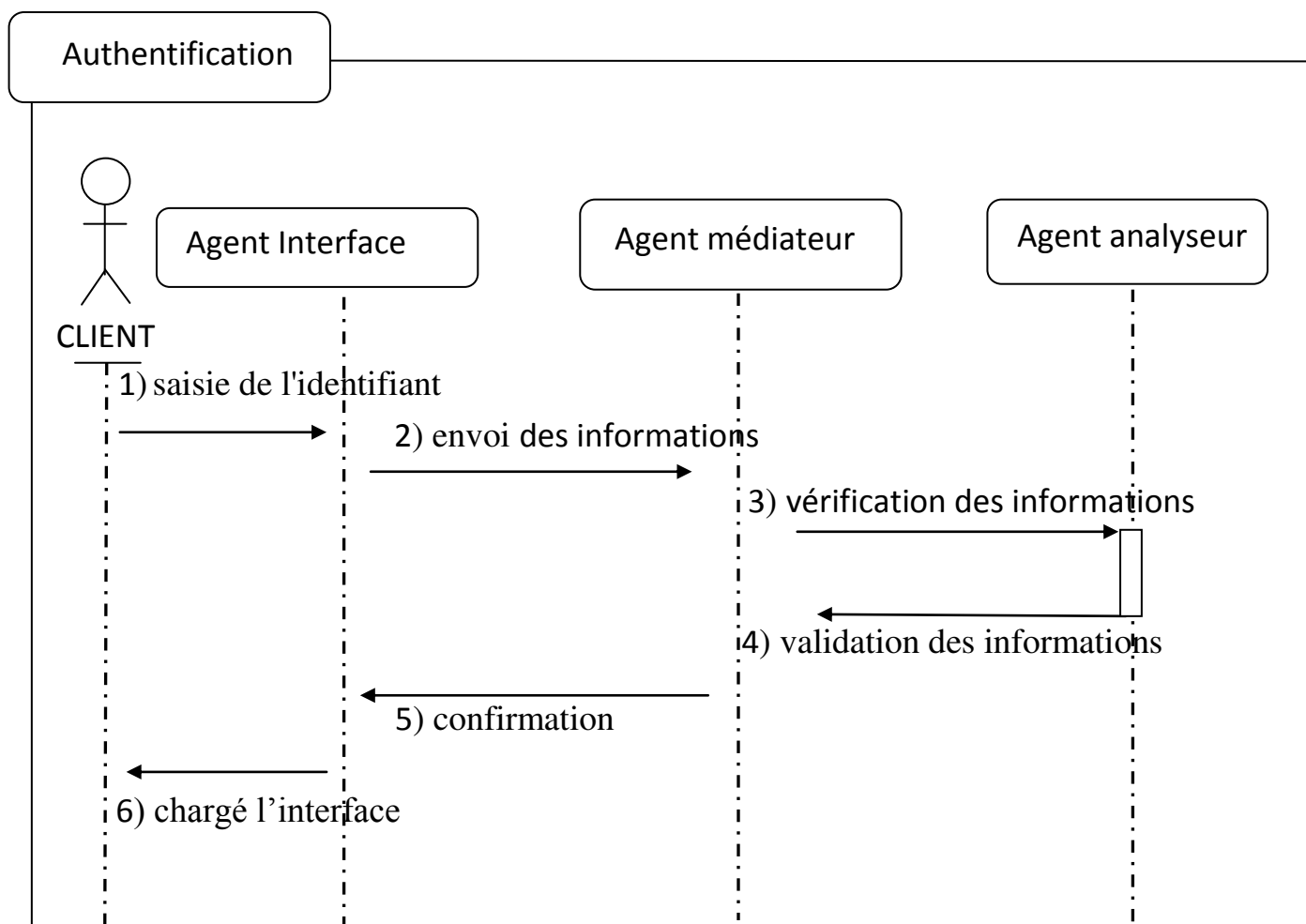
Dans la partie de diagrammes de séquences, nous avons un ensemble des diagrammes simples et explicites Schématisent les relations et les interactions dans notre Système

- Diagramme de séquence de registre un serveur virtuelle  
Ce diagramme présente l'interaction entre le serveur virtuel, et le contrôleur du cloud computing pour registre cette serveur dans le cloud computing.



**Figure 4.17 Diagramme de séquence « Inscription serveur»**

- Diagramme de séquence d'agent interface  
Le client qui désire utiliser le système doit s'authentifier en saisissant son identifiant et son mot de passe



**Figure 4.18 Diagramme de séquence « authentication »**

- Diagramme de rôle du serveur virtuel  
Ce diagramme présente l'interaction entre l'agent médiateur, et l'utilisateur et le système a base d'agents mobiles et il joue comme le responsable de système.

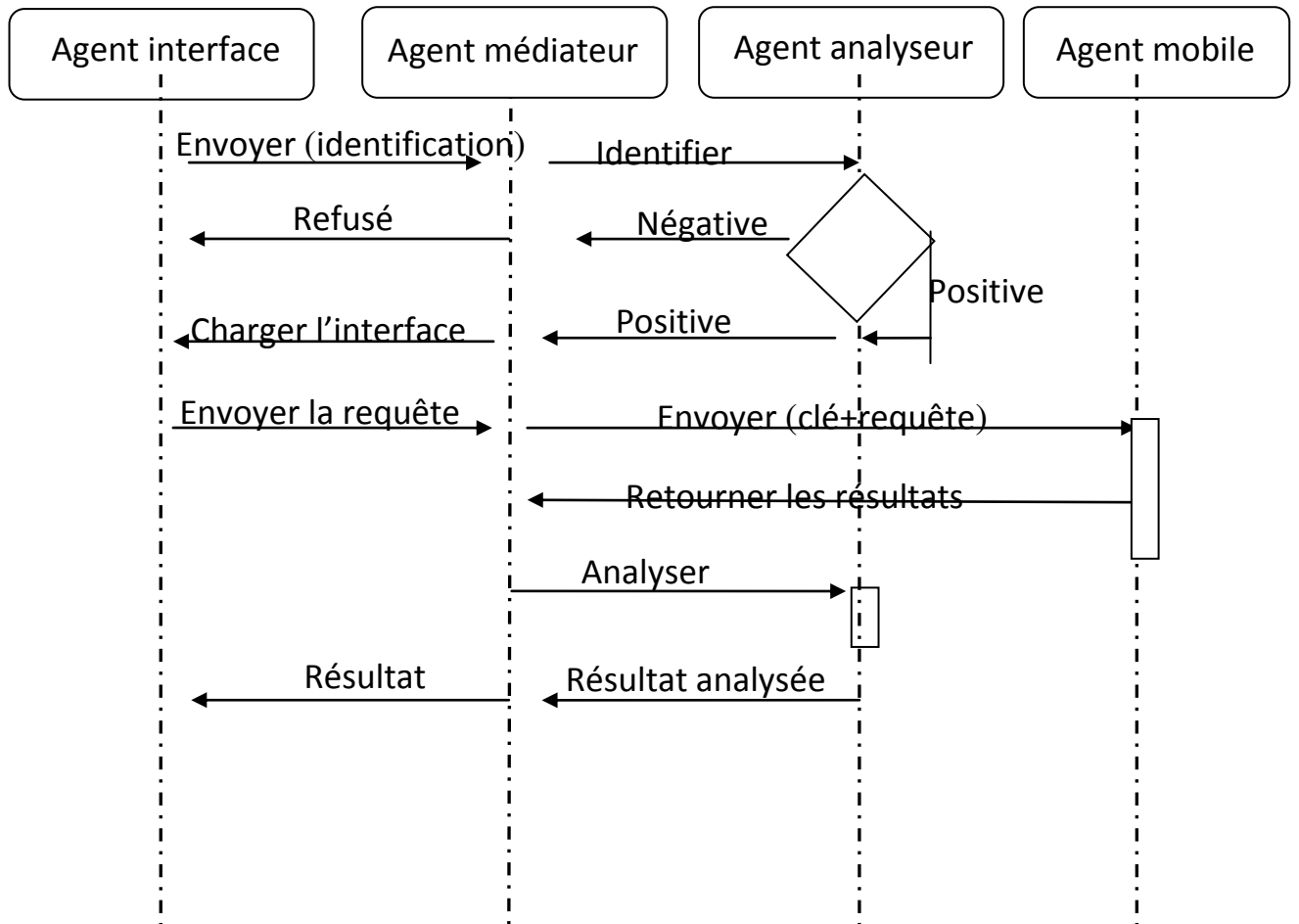


Figure 4.19 Diagramme de séquence « lancer requêtes»

- Diagramme de rôle de l'agent contrôleur dans le cloud computing  
Ce diagramme présente l'interaction entre l'agent contrôleur, et les autres agents dans le cloud computing et le système à base d'agents mobiles et il joue comme le responsable de système.

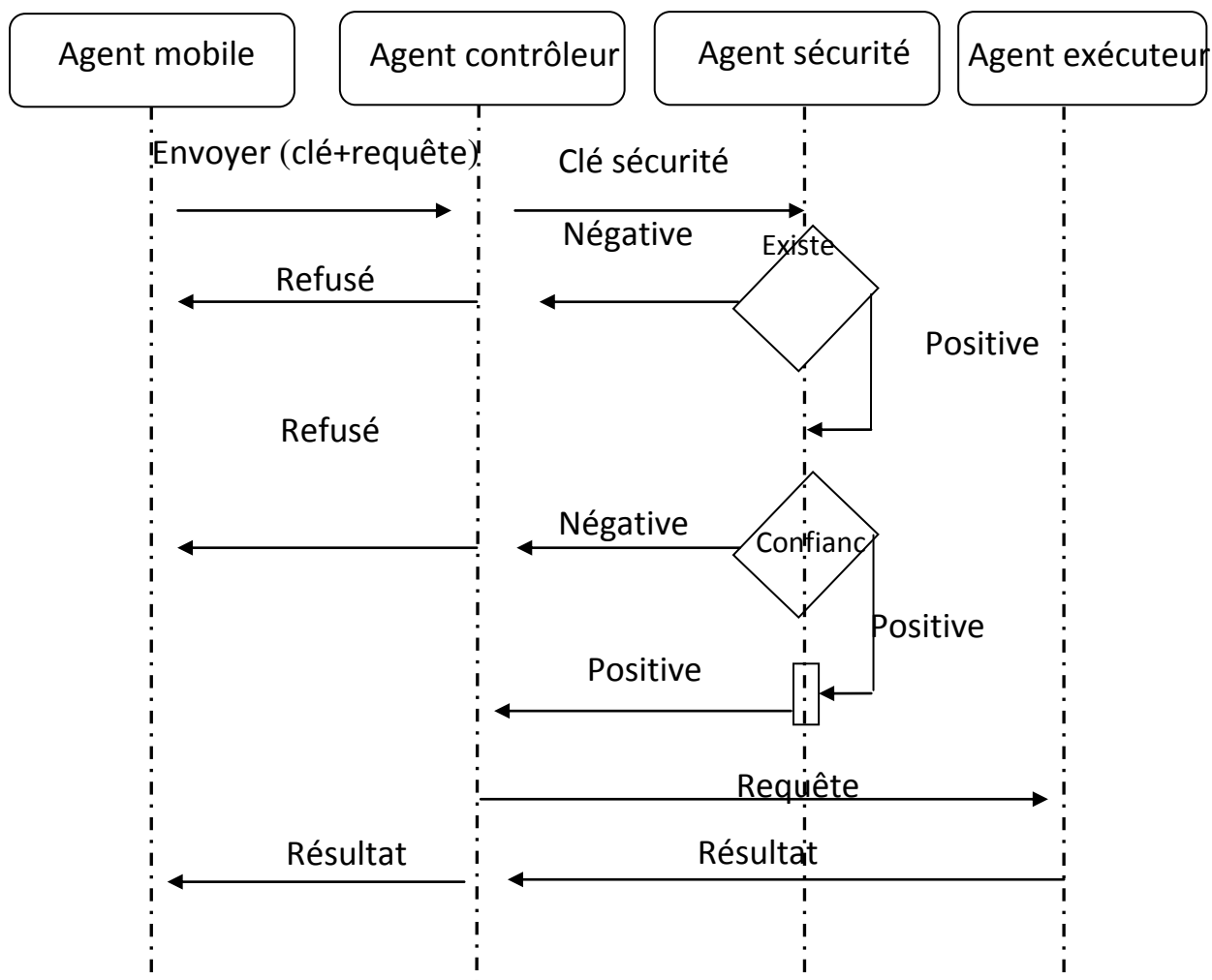


Figure 4.20 Diagramme de séquence « lancer services »

## IV.9. La Conclusion

Dans ce chapitre, nous avons donné la description de notre architecture par l'objectif de ce système et la motivation de choix de cette technologie. Ensuite, on a présenté la conception globale de l'architecture du cloud computing sécurisée qui est basée sur la technologie d'agents mobiles, ensuite nous avons détaillée la conception de notre système qui réalise la création de serveur virtuel pour accéder à un service de cloud computing par les agents mobiles.

Donc, on a présenté la description de l'architecture proposée et les structures internes des différents agents et leurs rôles en notre système. Dans la prochaine chapitre nous donnerons la réalisation du système.

# Chapitre V

---

## Etude De Cas Et Implémentation

---

### V.1. Introduction

Dans le chapitre précédent de ce mémoire, nous avons proposé une architecture basée agents mobiles pour assurer la sécurité dans un cloud computing.

Afin d'illustrer les différentes idées et concepts inclus dans l'architecture proposée, nous allons utiliser cette architecture comme base pour une étude de cas dans un environnement réel. Le but est de dérouler les principaux aspects de notre architecture sur un exemple concret afin de montrer la faisabilité et la mise en évidence de nos idées.

Pour cela, dans ce chapitre nous allons présenter, dans un premier temps, l'environnement du travail dont on va donner les différents langages de programmations ainsi l'environnement matériel et logiciel dont nous aurons besoin dans notre architecture pour passer par la suite aux étapes de l'implémentation notre architecture.

Les résultats obtenus à partir de l'implémentation de notre étude de cas sont présentés dans la dernière section.

### V.2. Environnement de développement

Dans la réalisation de ce projet, nous avons employé beaucoup des techniques (langages, outils, environnements, APIs<sup>1</sup>, etc.) à savoir : java, Aglet, MYSQL, etc...

Dans les sections suivantes, nous allons présenter ces différentes techniques, en les organisant selon leurs catégories.

#### V.2.1. Choix du langage de programmation

- **JAVA**

Java est un langage de programmation orienté objet développé par *Sun Microsystems*. Les premières versions datent de 1995, il a réussi à intéresser et intriguer beaucoup de développeurs à travers le monde . [Web 08]

La syntaxe générale du langage java est très proche de celle du langage C, et parmi ces avantages :

- Java assure une totale indépendance des applications vis-à-vis de l'environnement d'exécution : c'est à dire que toute machine supportant Java est en mesure d'exécuter un programme sans aucune adaptation (ni recompilation, ni paramétrage de variables d'environnement);
- Java est un langage orienté objet, c'est à dire que nous n'allons pas manipuler des fonctions et des procédures mais des objets qui vont s'échanger des messages. Le principal avantage est que l'on peut réaliser une programmation modulaire : tous les objets peuvent être mis au point séparément ;
- Java nous permet un accès simplifié aux bases de données, soit à travers la passerelle JDBC-ODBC ou à travers un pilote JDBC spécifique au SGBD ;
- Il est particulièrement adapté au développement d'application communiquant à l'intermédiaire d'un réseau. L'API java est très riche : différents packages permettant d'accéder aux réseaux, aux entrée/sortie et aux différents composants graphiques.

➤ Quelques notions de base JAVA

▪ **JDK**

Le Java Development Kit (JDK) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en byte code destiné à la machine virtuelle Java [Web 09].

Il existe plusieurs éditions de JDK, selon la plate-forme Java considérée (et bien évidemment la version de Java ciblée) :

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

À chacune de ces plateformes correspond une base commune de Development Kits, plus des bibliothèques additionnelles spécifiques selon la plate-forme Java que le JDK cible, mais le terme de JDK est appliqué indistinctement à n'importe laquelle de ces plates-formes.

Le Kit de développement comprend plusieurs outils, parmi lesquels :

- *javac*: le compilateur Java
- *java*: un interpréteur d'applications (machine virtuelle)
- *applet viewer*: un interpréteur d'applets
- *jdb*: un débogueur
- *javap*: un décompilateur, pour revenir du *bytecode* au code source
- *javadoc*: un générateur de documentation
- *jar* : un compresseur de classes Java

#### ▪ API

Cela signifie « Application Programming Interface ». Une API contient un ensemble de fonctions qui facilitent la programmation.

L'API de java contient paquets standard : `java.awt`, `java.io`, `java.lang`, `java.net` et `java.util`.

#### ▪ JVM

La machine virtuelle Java ou JVM (Java Virtual Machine) est un environnement d'exécution pour applications Java.

C'est un des éléments les plus importants de la plate-forme Java.

Elle assure l'indépendance du matériel et du système d'exploitation lors de l'exécution des applications Java. Une application Java ne s'exécute pas directement dans le système d'exploitation mais dans une machine virtuelle qui s'exécute dans le système d'exploitation et propose une couche d'abstraction entre l'application Java et ce système [Web 10]. La machine virtuelle permet notamment :

- L'interprétation du byte code
- l'interaction avec le système d'exploitation
- la gestion de sa mémoire grâce au ramasse miettes

Son mode de fonctionnement est relativement similaire à celui d'un ordinateur : elle exécute des instructions qui manipulent différentes zones de mémoire dédiées de la JVM. Une application Java ne fait pas d'appel direct au système d'exploitation (sauf en cas d'utilisation de JNI) : elle n'utilise que les API qui sont pour une large part écrites en Java sauf quelques unes qui sont natives. Ceci permet à Java de rendre les applications indépendantes de l'environnement d'exécution.

La machine virtuelle ne connaît pas le langage Java : elle ne connaît que le byte code qui est issu de la compilation de codes sources écrits en Java.

Les spécifications de la machine virtuelle Java définissent :

- Les concepts du langage Java
- Le format des fichiers .class
- Les fonctionnalités de la JVM
- Le chargement des fichiers .class
- Le byte code
- La gestion des threads et des accès concurrents
- ..

Les fonctionnalités de la JVM décrites dans les spécifications sont abstraites : elles décrivent les fonctionnalités requises mais ne fournissent aucune implémentation ou algorithme d'implémentation. L'implémentation est à la charge du fournisseur de la JVM.

Il existe de nombreuses implémentations de JVM dont les plus connues sont celles de Sun Microsystems, IBM, BEA, ...

Le respect strict de ces spécifications par une implémentation de la JVM garantit la portabilité et la bonne exécution du bytecode.

## **V.2.2. La plateforme J2ME**

Java 2 Micro Edition est une architecture technique dont le but est de fournir un socle de développement aux applications embarquées. Elle offre un environnement de développement allégé afin de s'adapter aux différentes contraintes d'exécution[java em].

J2ME regroupe par catégories certaines familles de produits tout en proposant la possibilité d'implémenter des routines spécifiques à un terminal donné.

### **V.2.2.1. L'architecture J2ME se découpe donc en plusieurs couches :**

- Les machines virtuelles sont allégées afin de consommer plus ou moins de ressources (KVM, CVM, ...).
- Les configurations définissent une plate-forme minimale en terme de services concernant un ou plusieurs profils donnés.
- Les profils permettent à une certaine catégorie de terminaux d'utiliser des caractéristiques communes telles que la gestion de l'affichage, des événements d'entrées/sorties (pointage, clavier, ...) ou des mécanismes de persistance (Base de données légère intégrée).

Cette architecture en couche a pour but de factoriser pour des familles de produits données un ensemble d'API permettant à une application de s'exécuter sur plusieurs terminaux sans modification de code. Dans cette optique, la plate-forme propose deux configurations:

- La configuration CDC (Connected Device Configuration) spécifie un environnement pour des terminaux connectés de forte capacité tels que le « set top boxes », les téléphones à écran, la télévision numérique, etc.
- La configuration CLDC (Connected Limited Device Configuration) est adaptée aux périphériques à ressources limitées ou faibles tels que les téléphones mobiles, les assistants personnels, ou les périphériques légers sans fil (wireless).

### **V.2.2.2. Les configurations**

Une configuration représente la base de la plateforme J2ME. Elle est constituée de la machine virtuelle et des bibliothèques bas niveaux. J2ME est constitué de la configuration CDC et de la configuration CLDC. Leur utilisation est fonction du type de matériel où s'exécutera l'environnement Java. Les configurations sont empilables, de sorte qu'une application s'exécute sur une configuration CLDC peut s'exécuter sur une configuration CDC, pas contre la réciproque n'est pas vraie.

#### **V.2.2.2.1. La configuration CDC (Connected Device Configuration)**

La configuration CDC est plus adaptée aux terminaux relativement puissants comme les PDA. En effet, elle nécessite une machine virtuelle java optimisée appelée CVM qui offre les mêmes fonctionnalités que la JVM classique.

Les caractéristiques de l'environnement matériel proposé par la configuration CDC sont :

- ✓ Minimum de 512Ko de ROM et 256Ko de RAM, processeur 32 bits
- ✓ Une connexion réseau obligatoire (sans fil ou pas)
- ✓ Support des spécifications complètes de la machine virtuelle Java (CVM)
- ✓ Cette configuration s'inscrit donc dans le cadre d'une architecture Java presque complète.

#### **V.2.2.2.2. CLDC (Connected Limited Device Configuration)**

Cette configuration s'adresse aux terminaux légers tels que les téléphones mobiles ou les assistants personnels. Ces périphériques étant limités en termes de ressources, l'environnement classique ne permet pas de respecter les contraintes d'occupation mémoire liée à ces appareils. J2ME définit donc un ensemble d'API spécifiques à CLDC et destinées à utiliser les particularités de chaque terminal d'une même famille (profile). La liste suivante résume l'ensemble de ces caractéristiques :

- ✓ Minimum de 160Ko à 512Ko de RAM, processeur 16 ou 32 bits, vitesse 16Mhz ou plus.
- ✓ Alimentation limitée, prise en charge d'une batterie
- ✓ Connexion réseau non permanente, sans fil.
- ✓ Interface graphique limitée ou inexistante

### **V.2.2.3. Les profiles**

Les profiles se composent d'un ensemble d'API particulières à un type de machines ou à une fonctionnalité spécifique. Ils permettent l'utilisation de fonctionnalités précises et doivent être associés à une configuration. Ils permettent donc d'assurer une certaine modularité à la plate-forme J2ME.

Il existe plusieurs profiles : MIDP ; Foundation Profile ; Personal Profile ; Personal Basis Profile et ETC..

MIDP (Mobile Information Device Profile) est un profile standard qui n'est pas défini pour une machine particulière mais pour un ensemble de machines embarquées possédant des ressources et une interface graphique limitée.

Il existe aujourd'hui deux implémentations majeures de profiles MIDP.

L'une, plus spécifique, destinée aux Assistants de type Palm Pilot (PalmOs) et l'autre, totalement générique, proposée par Sun comme implémentation de référence (RI) ; Le Foundation Profile est un profil de base qui s'utilise avec CDC.

Ce profil ne permet pas de développer des IHM. Il faut lui associer un des deux profils suivants :

- ✓ Personal Basic Profile permet le développement d'application connectée avec le réseau
- ✓ Personal Profile est un profil qui permet le développement complet d'une IHM et d'applet grâce à AWT.

#### **V.2.2.3.1. Foundation profile**

Ce profile sert de base pour le développement d'applications sur des outils mobiles utilisant la configuration CDC tels que des Pockets PC ou des Tablets PC. Le but du Foundation Profile est de servir de support pour le développement d'autres profiles. Il ajoute à CDC des classes de Java SE notamment relatives à la sécurité, aux locales et des utilitaires. Ce profile ne propose aucune classe pour les interfaces graphiques. Une partie importante de ce profile concerne les différentes formes de connexions au réseau.

#### **V.2.2.3.2. Le Personal Basis Profile (PBP)**

Le Personal Basis Profile repose sur le Foundation Profile. Il contient les éléments de bases pour développer une interface graphique, Son but principal est de proposer un support minimum pour les interfaces graphiques sous la forme d'un sous ensemble de l'API AWT.Ce profile propose un support pour les applications de type Xlet.

#### **V.2.2.3.3. Le Personal Profile (PP)**

Le Personal Profile repose sur le Personal Basis Profile. Il est destiné au développement d'applications sur des PDA disposant de ressources importantes telles que les Pockets PC. Ce profile permet notamment le développement d'IHM évoluées. Son but est de proposer un support pour les interfaces graphiques sous la forme d'un sous ensemble assez complet de l'API AWT et des applets.

#### **V.2.3. Description générale de la plate form JADE**

JADE (Java Agent Development framework) est une plate-forme multi-agent créé par le laboratoire TILAB. JADE permet le développement de systèmes multi-agents et des applications conformes aux normes FIPA.

Elle est implémentée en JAVA et fourni des classes qui implémentent « JESS » pour la définition du comportement des agents. Toute la communication entre agents est exécutée par messages FIPA ACL.

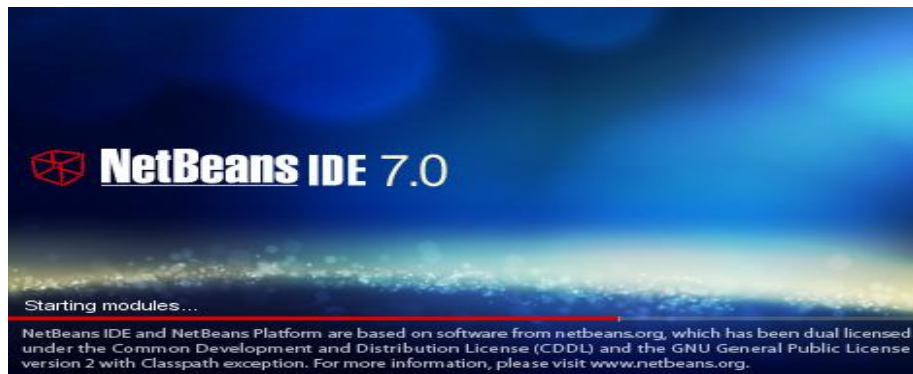
JADE fournit les facilités suivantes : Une plate-forme agent distribuée : la plateforme peut être distribuée (partagée) entre plusieurs hôtes connectées via RMI de Java, de telle façon qu'une seule application Java, par conséquent une seule "Machine Virtuelle Java" est exécutée sur chaque hôte.

- ✓ Une interface utilisateur graphique (GUI): l'interface GUI assure un traitement plus commode de la plate-forme, elle permet à l'utilisateur d'exécuter plusieurs ordres tel que créer un nouvel agent dans la même plate-forme, cloner l'agent, le déplacer, le suspendre, le tuer, etc....
- ✓ Un support d'exécution : pour les activités multiples, parallèles et concurrentes des agents via le modèle du comportement (behaviour).
- ✓ Un transport efficace des messages ACL : à l'intérieur de la même plate-forme.
- ✓ Une bibliothèque de protocoles : compatibles aux standards FIPA et prêts à être employés pour gérer l'interaction inter-agent.

#### **V.2.4. NetBeans IDE 7.0**

Pour notre choix de l'environnement de développement Java, nous avons opté sur l'utilisation de NetBeans, NetBeans est un environnement de développement intégré (IDE) open source. Il est développé par Sun et se trouve sous licence CDDL (Common Development and Distribution License). En plus de Java, il propose tous les outils nécessaires à la création d'applications professionnelles pour les particuliers, les entreprises, le web et les applications mobiles avec le langage C / C + +, et même les langages dynamiques tels que PHP, JavaScript, Groovy, Ruby XML et HTML. NetBeans IDE est facile à installer et à utiliser.

Il comprend toutes les caractéristiques d'un IDE moderne (coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web, etc.).



**Figure 5.1 : l'IDE netbeanse7.0**

### **V.2.5. La plate-forme d'agent mobile**

Il existe plusieurs plate-forme de développement des systèmes multi-agent, certaines écrites en Java pur (ce qui est aussi noté 100% Java), citons donc d'abord quelques systèmes écrits en Java : Odyssey, Aglets, Concordia, Voyager, le choix d'une plate-forme dépend de plusieurs critères :

- ✓ Facilité d'apprentissage.
- ✓ Disponibilité et souplesse des outils de développement et débogage.
- ✓ Disponibilité de documents.
- ✓ Clarté dans la structure, produit fameux.
- ✓ Très bons mécanismes de sécurité.
- ✓ ...etc.

Nous avons choisi la plate-forme Aglets qui couvre la majorité des critères précités.

#### **V.2.5.1. Description générale de la plate-forme Aglets**

L'API Aglet est une norme basée sur java proposée pour développer des agents mobiles. Elle a été développée par une équipe de chercheurs du laboratoire de recherche d'IBM à Tokyo au début 1995; son but est de fournir une plate-forme uniforme pour les agents mobiles dans un environnement hétérogène tel que celui de l'Internet

### V.2.5.1.1. Définition

Les Aglets (Agents Applets) sont des objets Java mobiles qui peuvent se déplacer d'une machine à une autre. Ainsi, un Aglet qui s'exécute sur un hôte peut stopper son exécution, se déporter vers un hôte distant et continuer cette exécution dans son nouvel environnement. [Bett 03] [WEB 11]

### V.2.5.1.2. Les éléments de base d'Aglets

Voici les éléments de base d'Aglets [WEB 11]:

✓ Aglet

Objet mobile de Java qui visite les serveurs où les agents sont autorisés, dans un réseau informatique. Un aglet est autonome puisqu'il peut reprendre son exécution dès son arrivée à destination et réactif car il peut répondre (réagir) à des événements de son environnement.

✓ Proxy

Un proxy est un représentant d'un aglet. Il sert de bouclier à l'aglet contre l'accès direct à ses méthodes publiques. Le proxy fournit également la transparence à l'emplacement pour l'aglet. C'est-à-dire qu'il peut cacher le vrai emplacement de l'aglet.

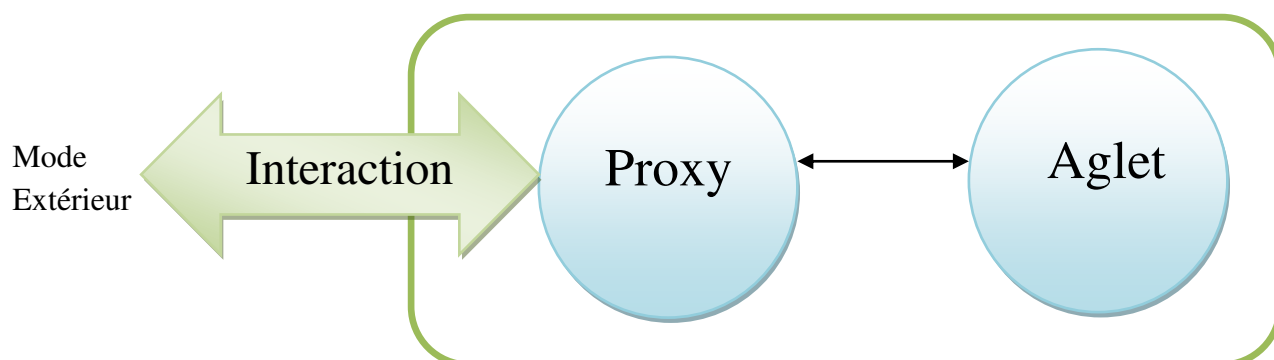


Figure 5.2: Relation entre un Aglet et son Proxy.

✓ Contexte

Le contexte est l'environnement d'exécution de l'aglet. Il fournit des moyens pour mettre à jour et contrôler des aglets dans un environnement uniforme d'exécution.

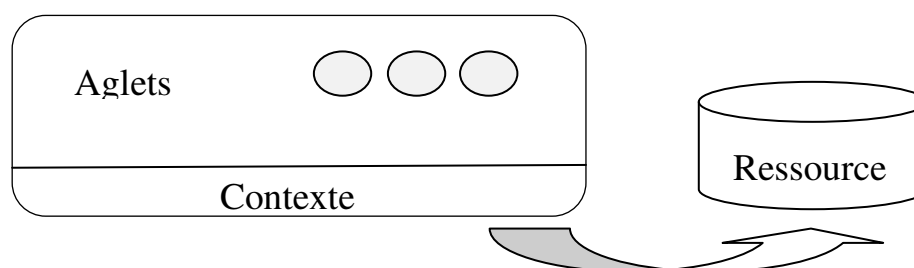


Figure 5. 3: contexte de l'aglet.

- ✓ Hôte  
Un hôte est une machine capable d'héberger plusieurs contextes.  
L'hôte est généralement un nœud dans un réseau.

### V.2.5.1.3. Cycle de vie d'un Aglet

Les types de comportement des Aglets ont été implémentés de manière à répondre aux principaux besoins des agents mobiles. Les principales opérations affectant la vie d'un aglet sont : [Bett 03] [WEB 11]

- ✓ Création `createAglet()`  
Se fait dans un Contexte. Un Identifiant unique est assigné.  
L'initialisation et l'exécution de l'aglet commence immédiatement.
- ✓ Clonage `oncloning()`  
Création d'un clone dans le même contexte que l'original. Un Identifiant différent est alors attribué. A noter que les processus (thread) ne peuvent pas être clonés.
- ✓ Déportation (Dispatching) `ondispatching()`  
Transfert d'un aglet d'un contexte à un autre. On dit que l'aglet à été poussé vers son nouveau contexte.
- ✓ Récupération (Retraction) `onreverting()`  
L'aglet déporté est récupéré (*tiré*) dans son contexte d'origine.
- ✓ Activation et Désactivation  
La désactivation d'un aglet une interruption temporaire de son exécution et stockage de son état dans un support secondaire de stockage.
- ✓ `sendMessage()`, `handleMessage()` et `sendReply()`  
pour la communication entre les Aglets.
- ✓ `onarrival()`  
pour continuer son exécution dans le nouvelle environnement.
- ✓ Libération ou destruction  
Fin de vie de l'aglet et son retrait du context.

#### **V.2.5.1.4. Les Serveurs d'Aglets**

- Tahiti [Bett 03]: un gestionnaire d'agents visuel. Tahiti utilise une interface graphique unique pour suivre et contrôler l'exécution des aglets. Il est possible en utilisant le glisser déposer de faire communiquer deux agents ou de les faire migrer vers un site particulier. Tahiti dispose d'un gestionnaire de sécurité paramétrable qui détecte toute opération non autorisée et empêche l'agent de la réaliser.
- Fiji - un lanceur d'agent sur le Web. Fiji est un applet Java capable de créer ou de détruire un aglet sur un navigateur Web. Fiji utilise comme unique paramètre l'URL de l'agent, qui est intégré directement dans le code HTML d'une page Web. Comme pour un applet, l'exécution d'un aglet commencera par le téléchargement du code, puis par son lancement grâce à Fiji. Si le serveur Web est complété par un démon ATP, il devient très facile de répartir des aglets sur les sites Web.

### **V.3. Présentation de l'étude de cas**

Afin de montrer la validité, la fiabilité et l'extensibilité de notre architecture, on a intérêt à faire une étude de cas. D'où nous allons appliquer notre approche sur un exemple d'organisation de voyage, cet exemple considéré comme un exemple simple pour simuler le travail sur un cloud computing.

On suppose le scénario suivant : Walid habite à Batna, il veut Passer la vacance à une autre ville. Il prépare les arrangements de voyage. Il peut utiliser un aéroport près de Batna et un aéroport près de la ville désirée. Il doit consulter une compagnie aérienne pour le vol, il doit déterminer le prix et il doit déterminer le lieu désiré et la durée désirée Et ensuite il lance se requête à partir leur appareil.

A partir du scénario précédant, on peut dire que le processus d'organisation du voyage est la consultation de site d'une compagnie aérienne pour faire une réservation d'un vol, notre exemple représente une application come un service dont le but est de chercher une réservation d'un vol, de bon prix, dans la durée désirée et vers la ville désirée.

Nous remarquons que ce processus est complexe et nécessite plus d'attention et de temps. Notre but est d'automatiser ce processus par notre système en utilisant la technologie d'agents mobiles.

#### **V.4. Description général de l'application**

Comme nous avons dit auparavant, notre exemple représente une application de service sur Internet dont le but est de chercher un vol dans une ville donnée dans la durée désirée.

Notre système est constitué de plusieurs agents qui coopèrent pour satisfaire Les demandes des services des clients, on en distingue :

L'agent client, l'agent médiateur, agent analyseur, l'agent mobile de transfère et l'agent contrôleur l'agent sécurité et l'agent ressources ou exécuter

Les agents de notre système sont implémentés utilisant le langage JAVA et la plateforme de développement des agents Aglets, Aglets supporte le développement des agents avec la possibilité de les transporter d'un système à l'autre. L'agent client, l'agent médiateur, agent analyseur, 'agent contrôleur, l'agent sécurité et l'agent ressources ou exécuter sont tous des agents stationnaires, tandis que les agents de transfère sont des agents mobiles parcourant le réseau pour collecter des informations pour leurs clients.

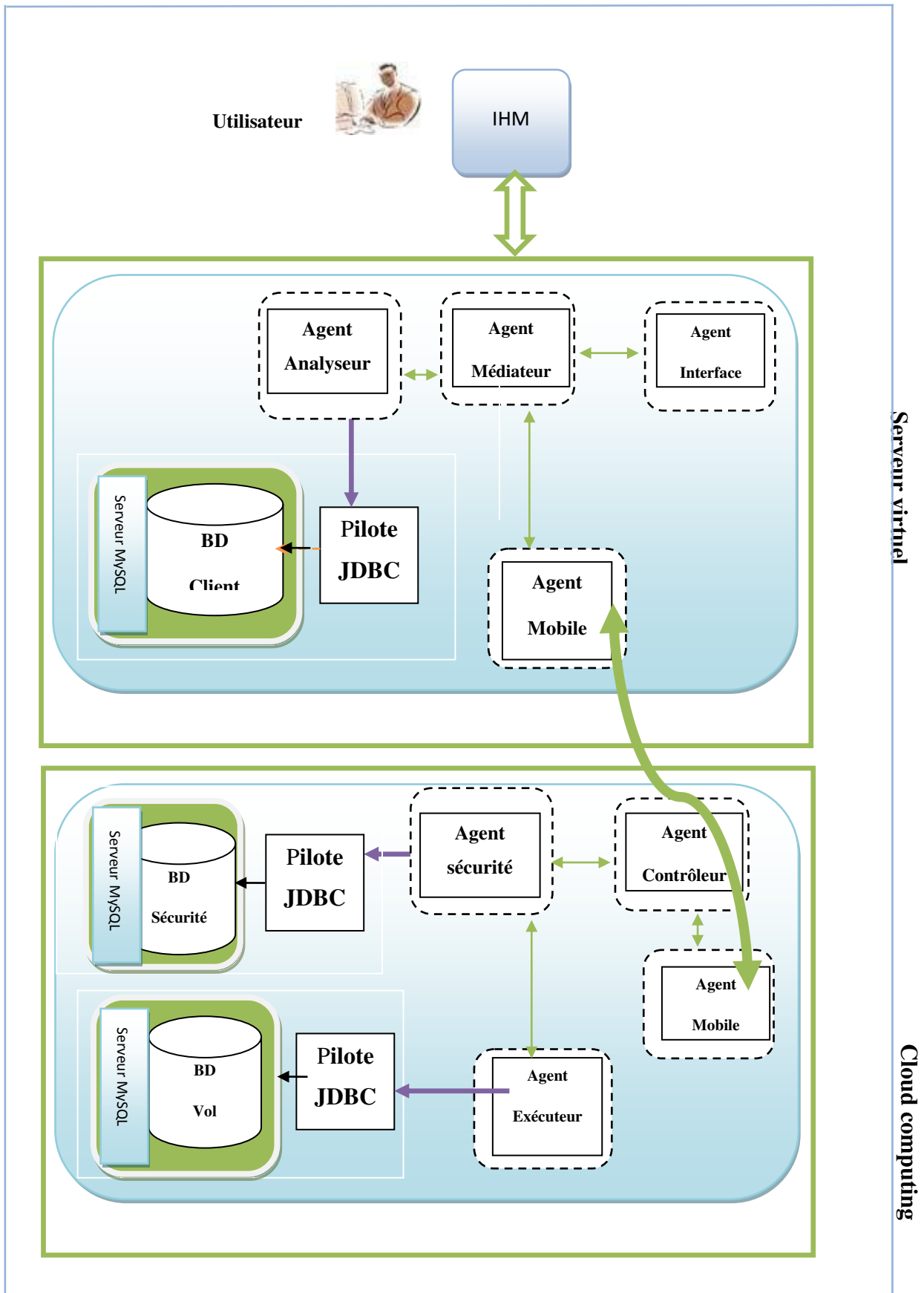


Figure 5.4 : L'architecture logicielle du prototype

## V.5. L'implémentation des agents

Pour décrire l'implémentation de SMA, nous allons présenter les différentes classes agents du système sous forme d'un pseudo-code:

### V.5.1. Agent interface

C'est un agent de type Gui, agent avec interface c.-à-d. qu'il est indispensable de programmer deux classes pour que cet agent puisse fonctionner Cette agent est la partie qui communique avec l'utilisateur où ce dernier peut ouvrir son compte par l'interface login et procéder à une inscription par l'interface d'inscription ou lancer une recherche d'un produit par l'interface de recherche.

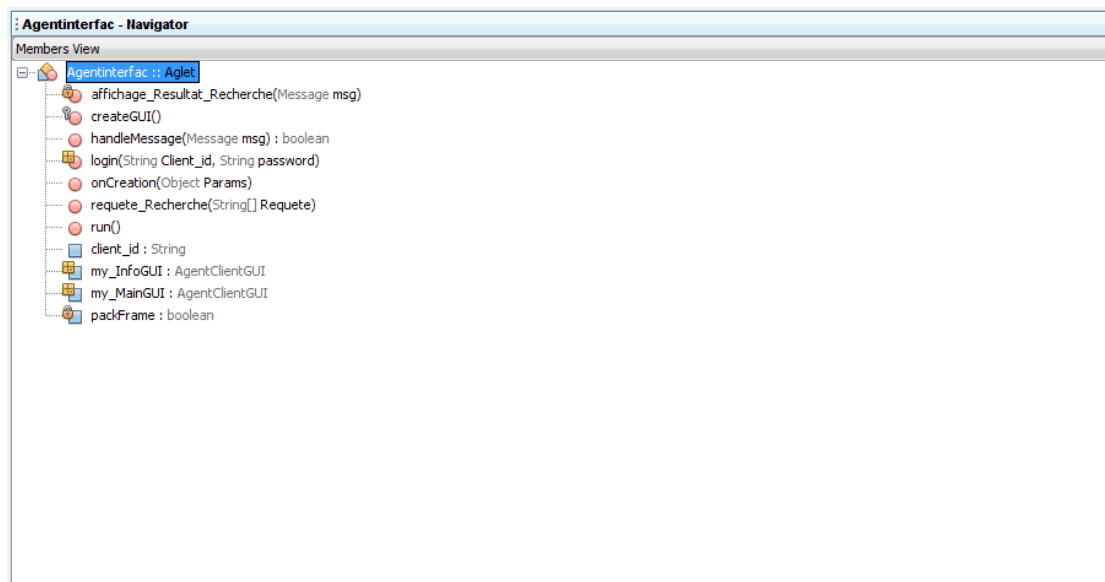


Figure 5.5: Pseudo code de l'agent interface.

### V.5.1. Agent médiateur

C'est un agent de type Gui, agent avec interface c.-à-d. qu'il est indispensable de programmer deux classes pour que cet agent puisse fonctionner.

C'est un agent Aglet stationnaire, communique seulement avec des autres Aglets (agent interface, agent mobile et Agent analyseur).

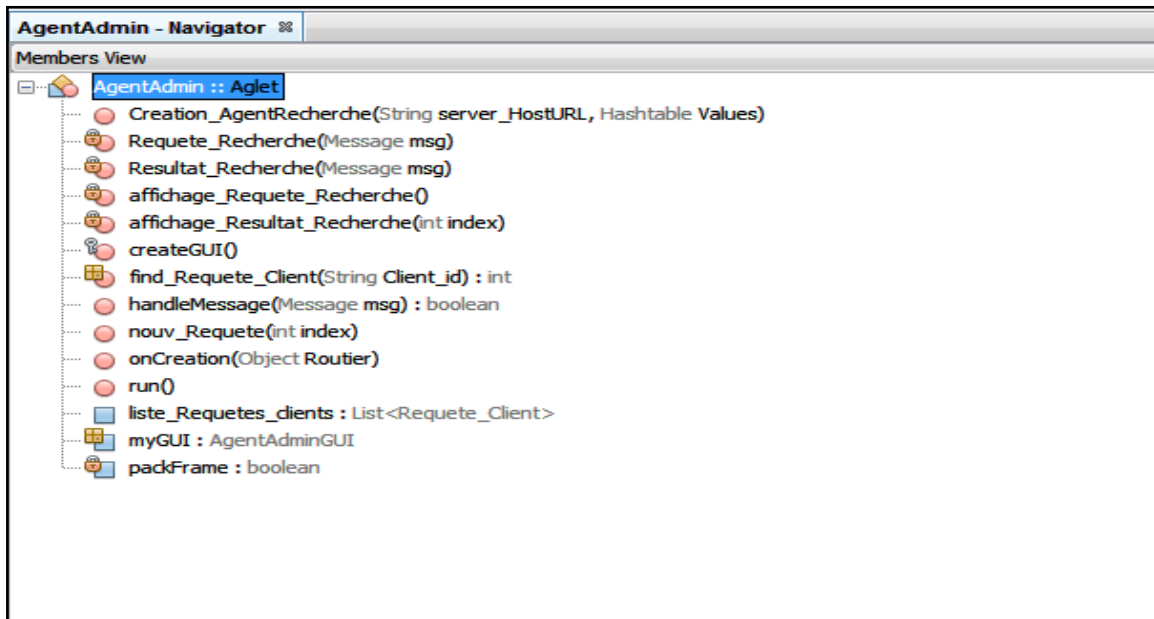


Figure 5.6: Pseudo code de l'agent médiateur.

### V.5.1. L'agent Recherche

C'est un agent Aglet mobile, migre du serveur virtuel vers le site cloud, pour communiquer avec l'agent contrôleur. Enfin il retourne vers le Serveur avec la réponse.

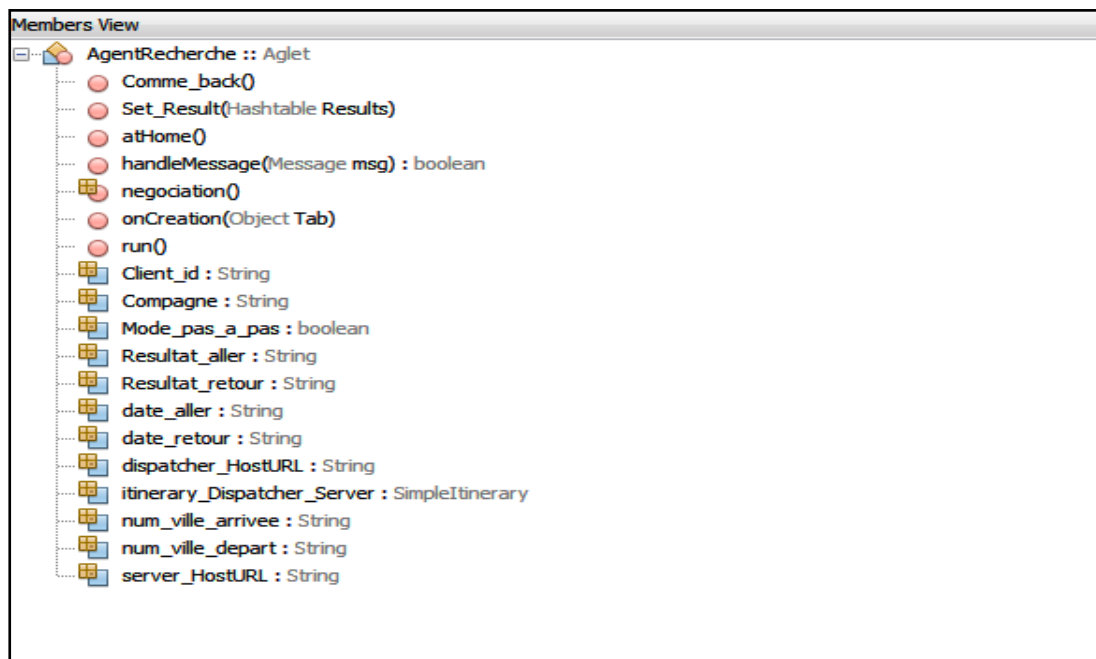


Figure 5.7: Pseudo code de l'agent mobile.

### V.5.1. Agent Analyseur

C'est un agent Aglet stationnaire, communique seulement avec l'agent Admin, il reçoit la requête et envoie la réponse traite.

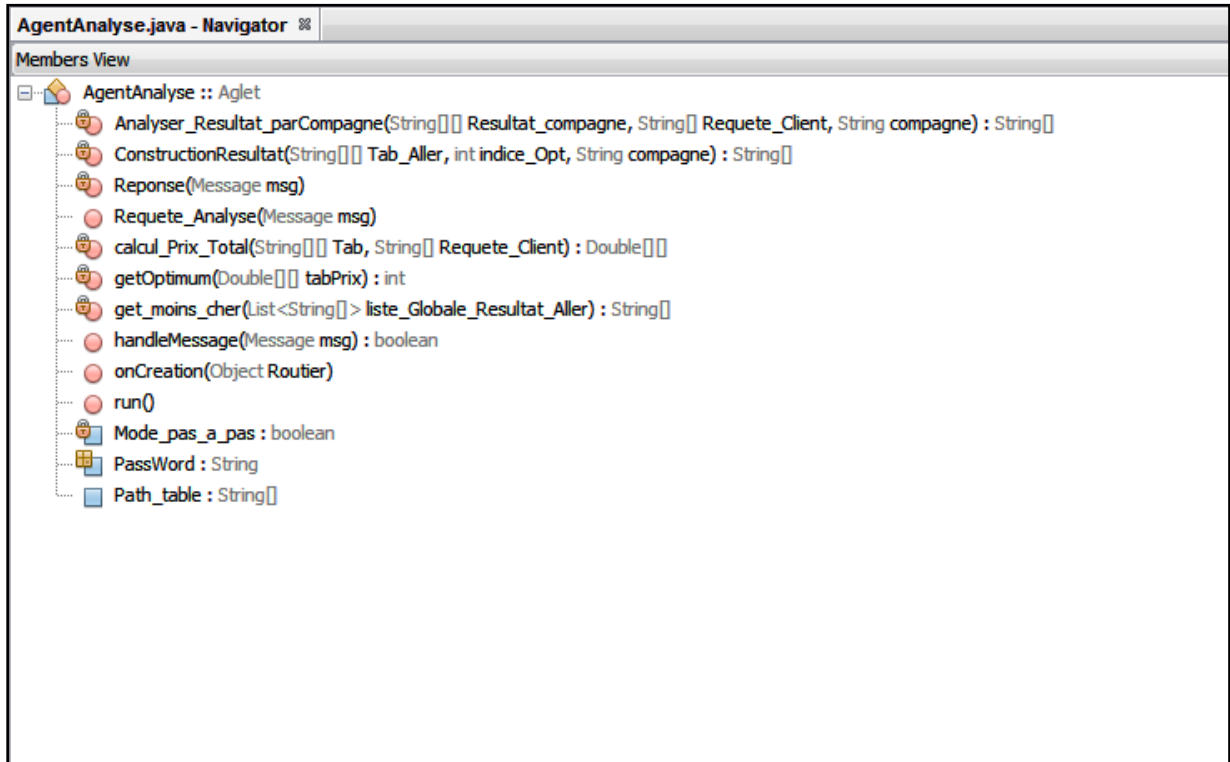


Figure 5.8: Pseudo code de l'agent analyseur.

### V.5.5. agent Ressource

C'est un agent Aglet stationnaire, communique seulement avec l'agent contrôleur, il reçoit la requête et envoie la réponse.

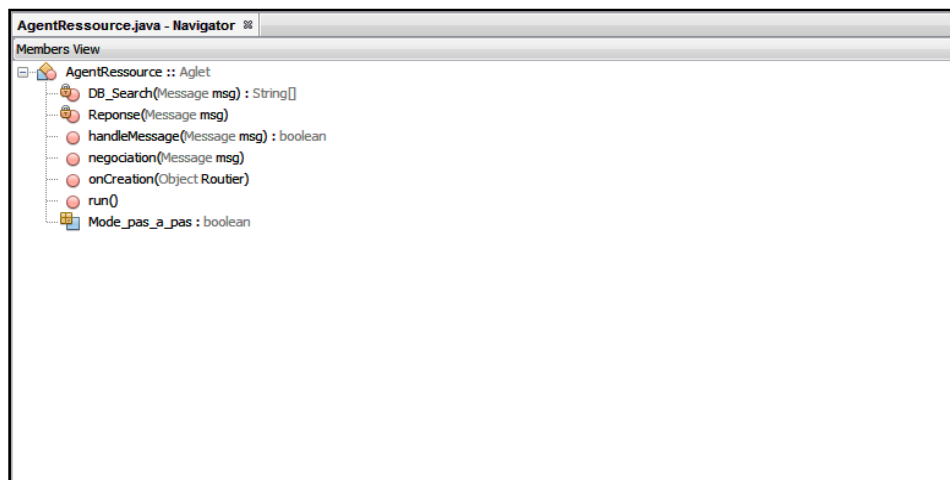


Figure 5.9: Pseudo code de l'agent ressource.

### V.5.1. Agent contrôleur

C'est un agent de type Gui, agent avec interface c.-à-d. qu'il est indispensable de programmer deux classes pour que cet agent puisse fonctionner. Il est responsable de contrôler les activités dans un cloud computing C'est un agent Aglet stationnaire, communique seulement avec des autres Aglets (agent sécurité, agent mobile et Agent exécuteur).

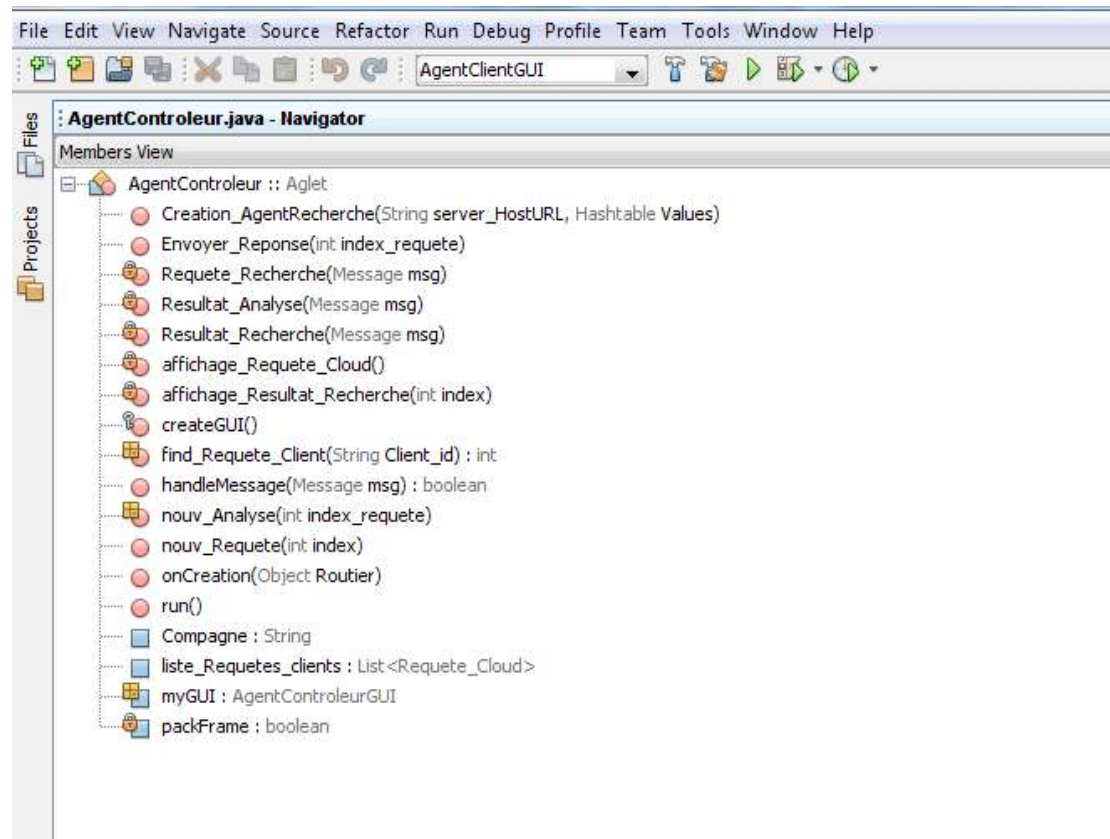


Figure 5.10: Pseudo code de l'agent contrôleur.

### V.5.1. Agent sécurité

C'est un agent Aglet stationnaire, communique seulement avec l'agent contrôleur, il reçoit la clé de sécurité pour analyser le degré de confiance de cette demande et envoie la réponse traite.

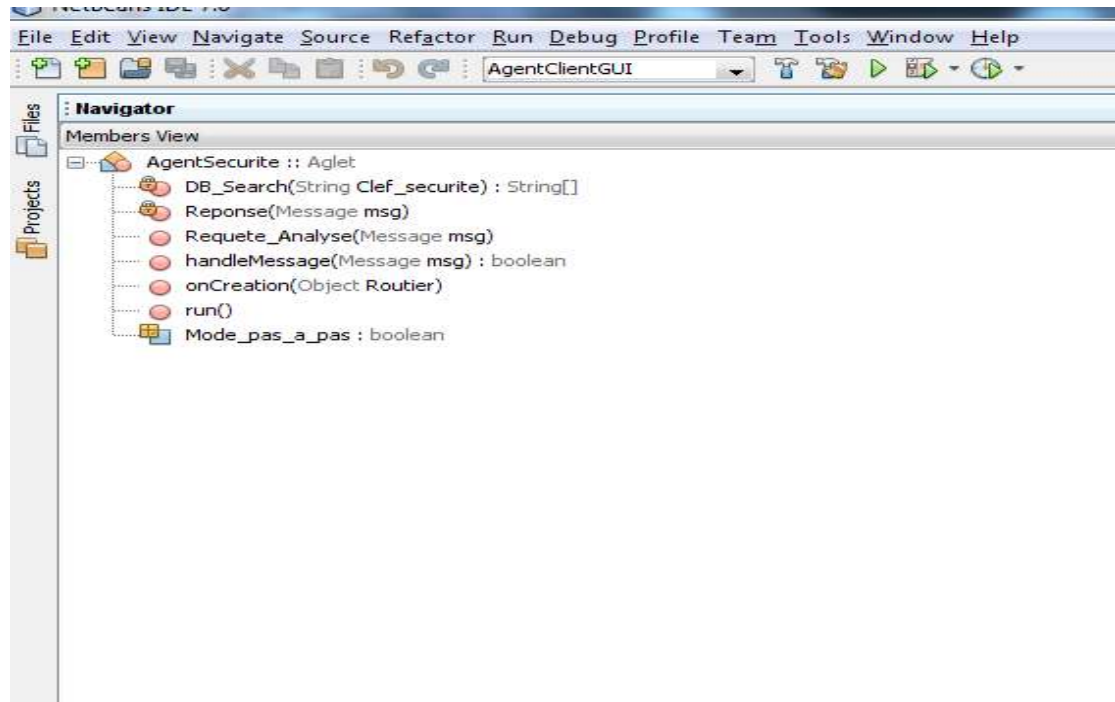


Figure 5.11: Pseudo code de l'agent sécurité.

## V.6. Présentation de Résultat de notre système Résultats

Pour exploiter le système, nous devons lancer la plateforme Aglet, on obtient

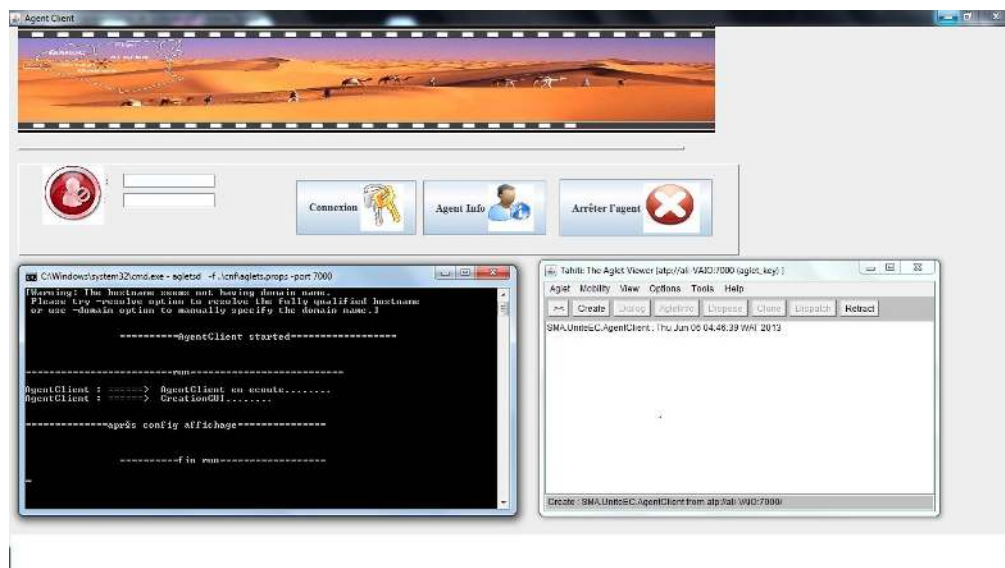


Figure 5.12: lancement de l'agent interface client.

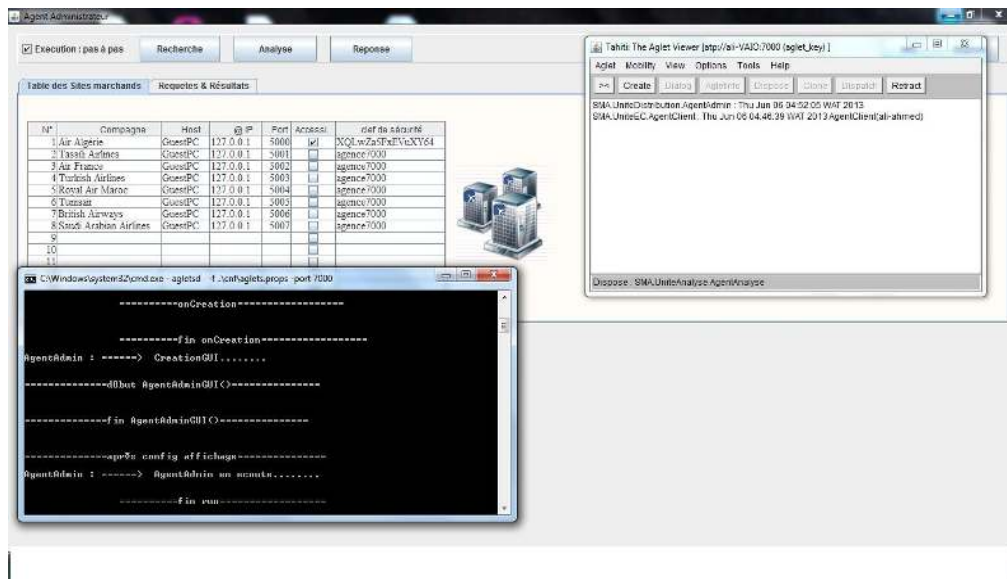


Figure 5.13: lancement de l'agent médiateur.

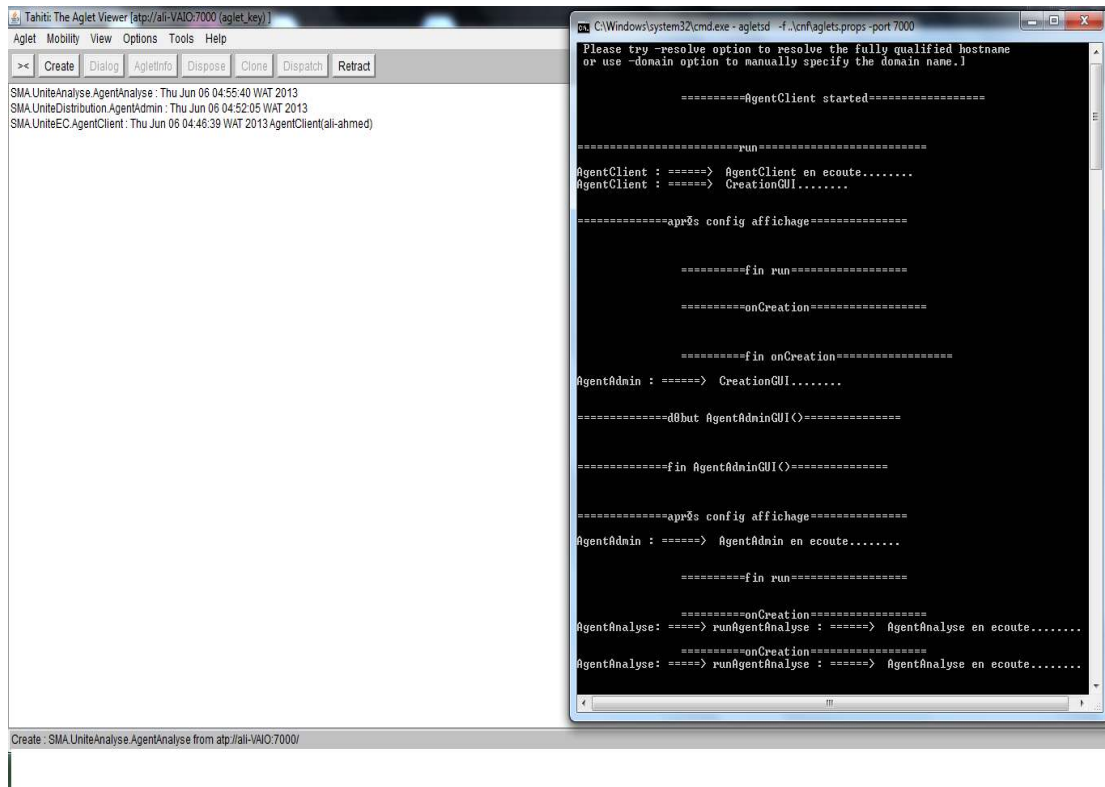


Figure 5.14: lancement de l'agent analyseur.

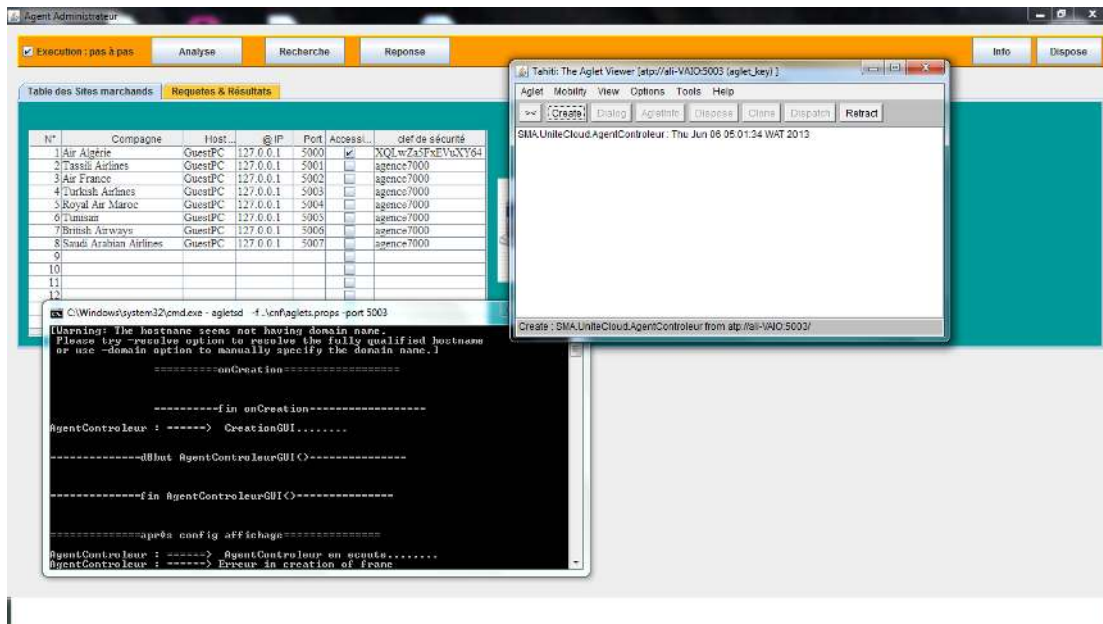


Figure 5.15: lancement de l'agent contrôleur.

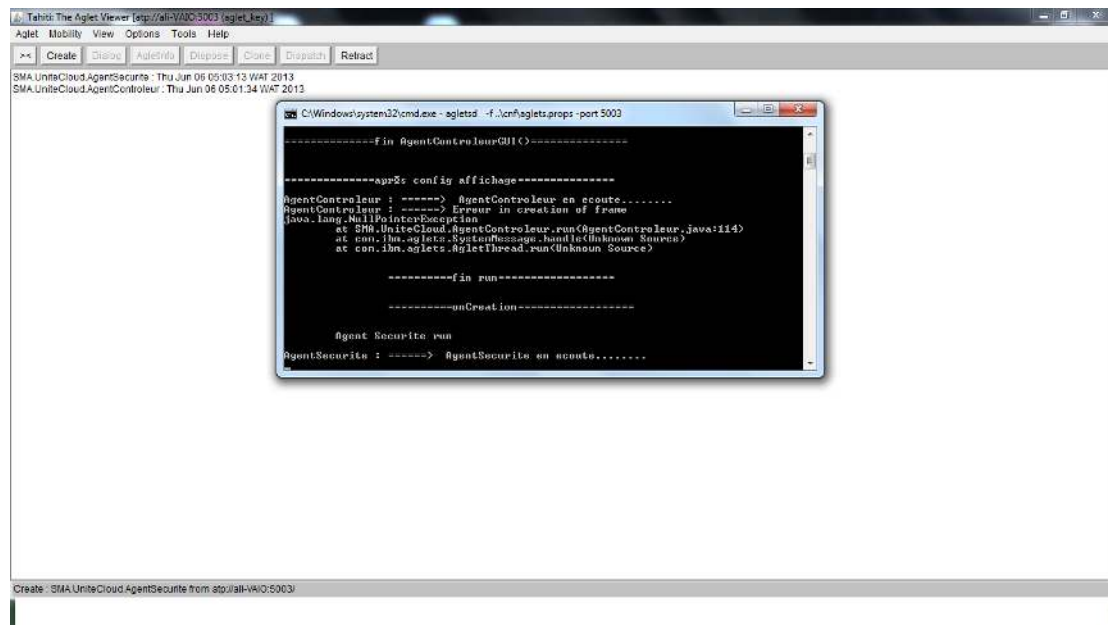


Figure 5.16: lancement de l'agent sécurité.

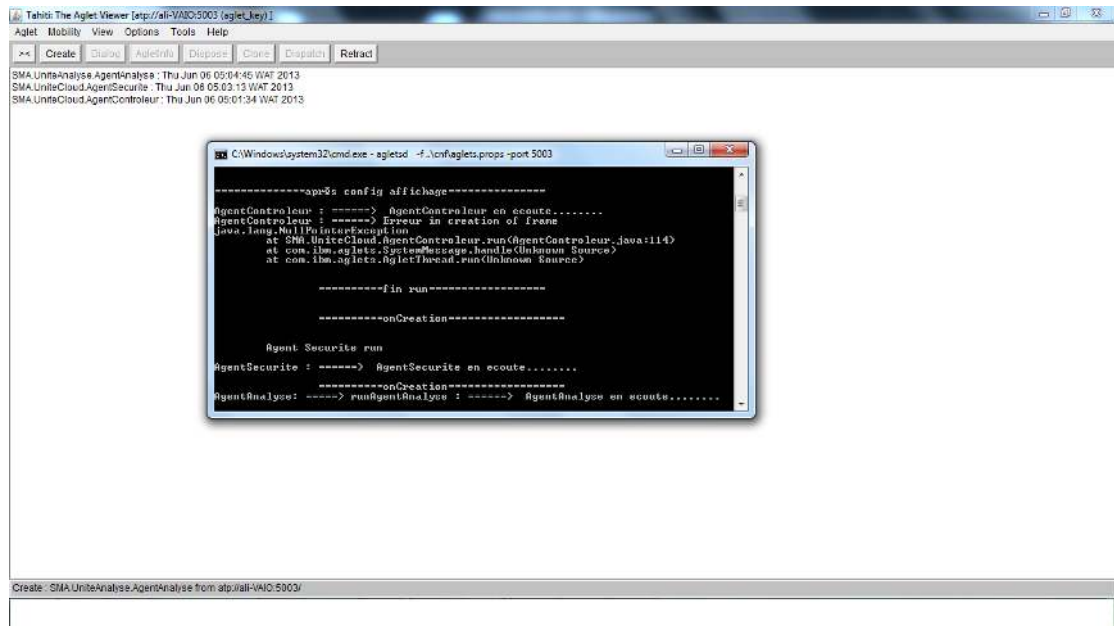


Figure 5.17: lancement de l'agent ressource.

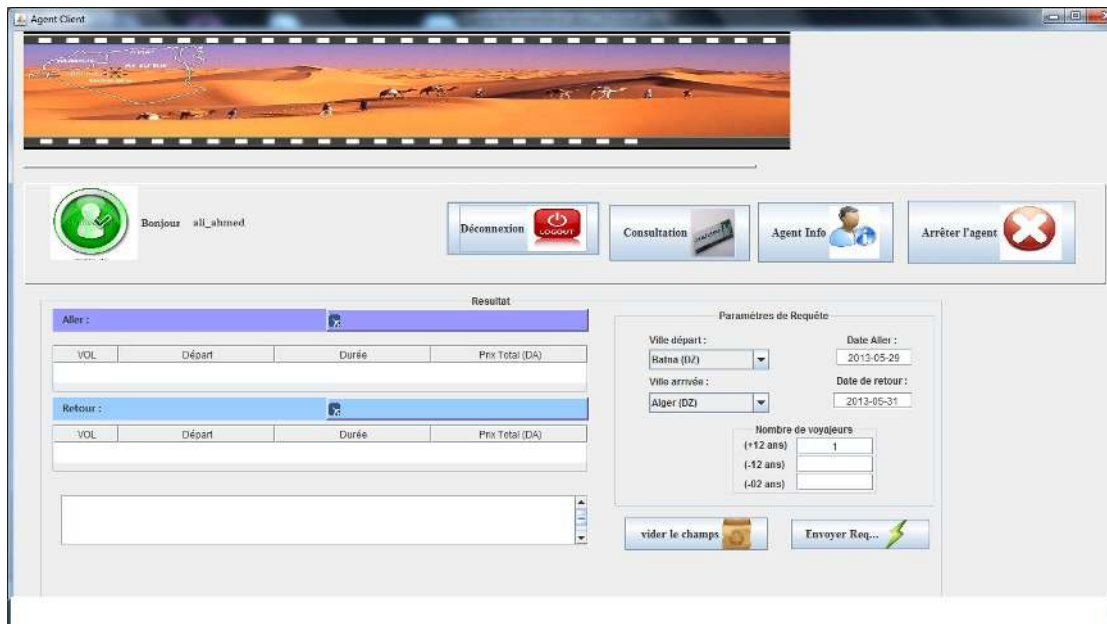
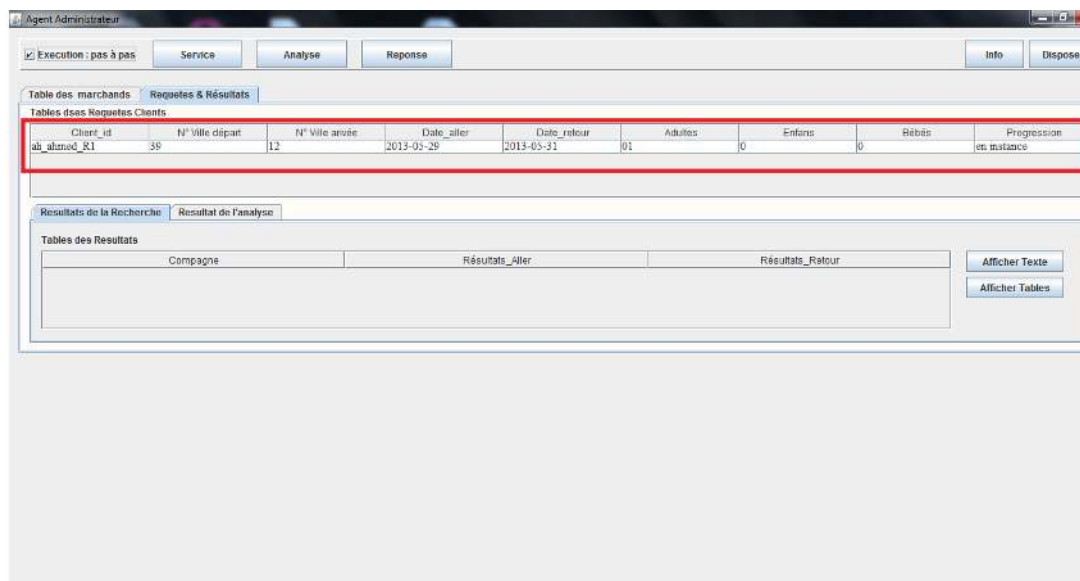


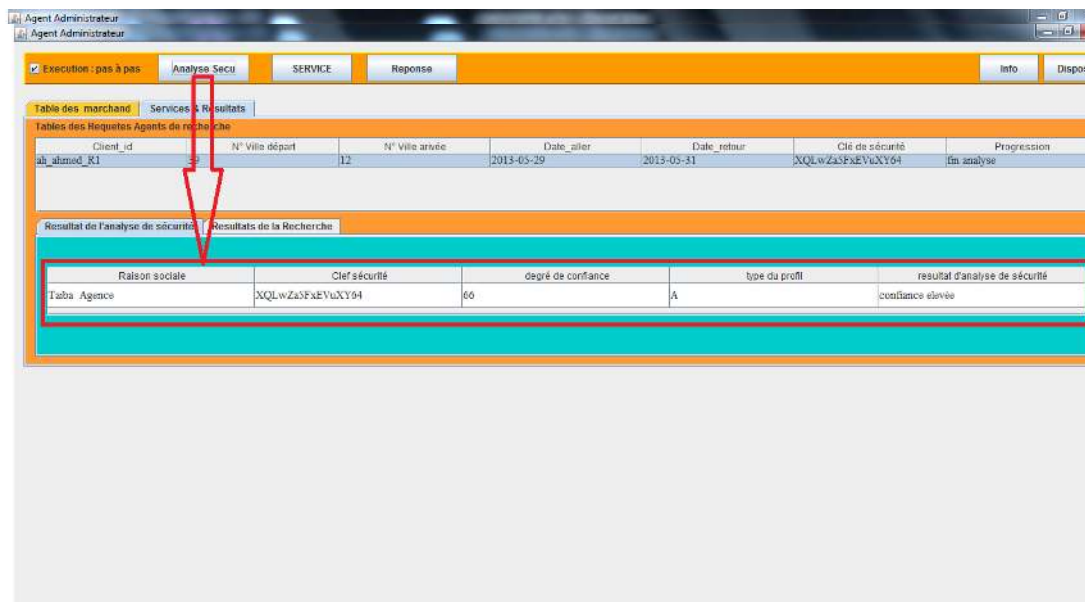
Figure 5.18 : Illustre l'interface de lancement d'une requête

Après l'exécution d'une requête Client, on peut visualiser les échanges de messages entre l'Agent Client et l'Agent Admin, et cela en cliquant sur le bouton correspondant (envoyer requête), voir Figure 5.19.



**Figure 5.19 : arriver une requête a l'agent médiateur**

Après avoirs lancer un service d'un client, on peut visualiser les échanges de messages entre l'Agent médiateur et l'Agent contrôleur, et cela en cliquant sur le bouton correspondant (service), voir Figure .20.



**Figure 5.20 : arriver une requête a l'agent contrôleur et analyser**

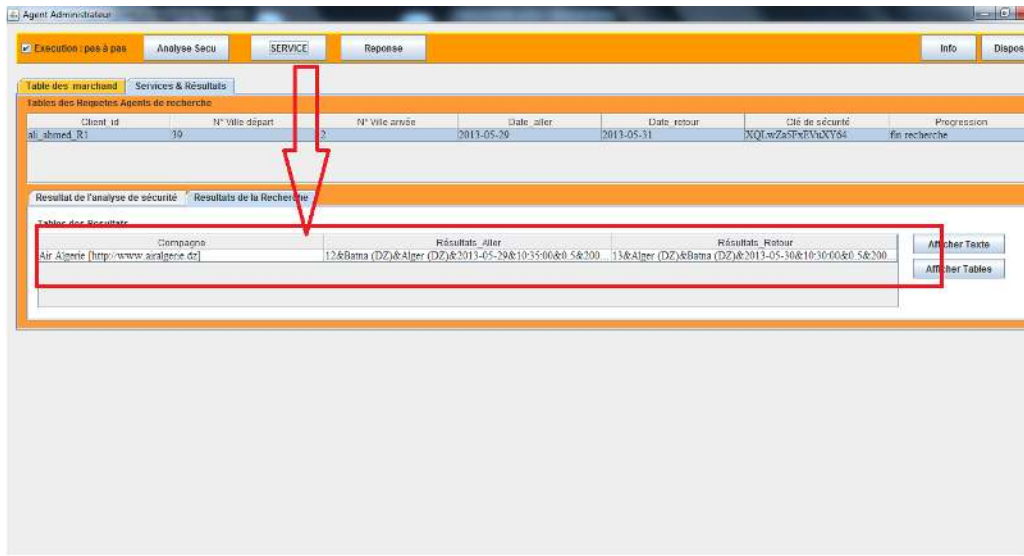


Figure V.21 : lancer un service après vérification la sécurité

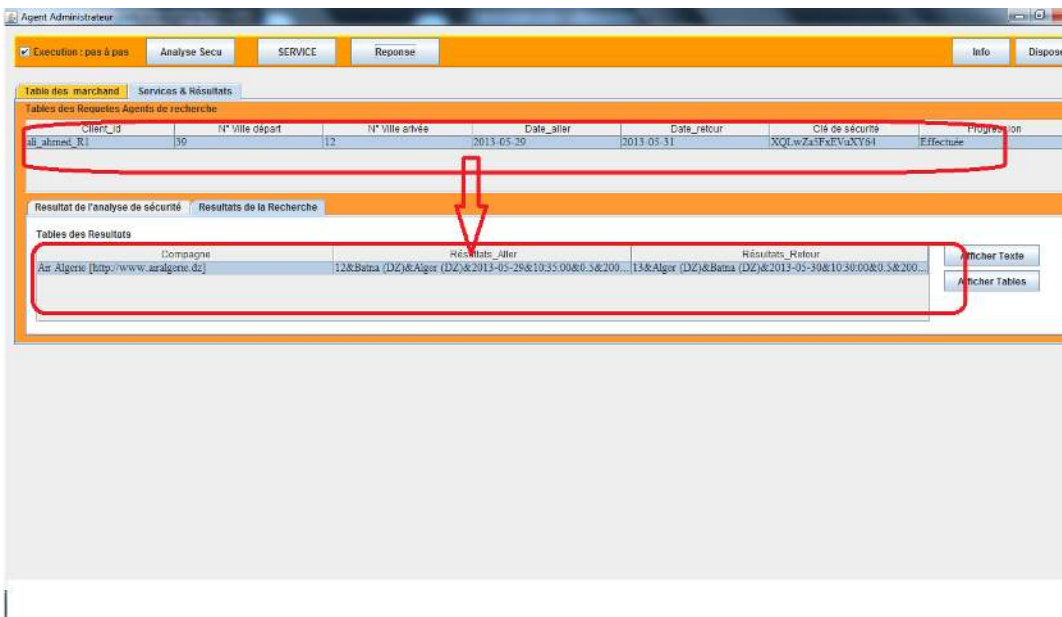


Figure V.22 : le résultat à partir de l'agent ressource

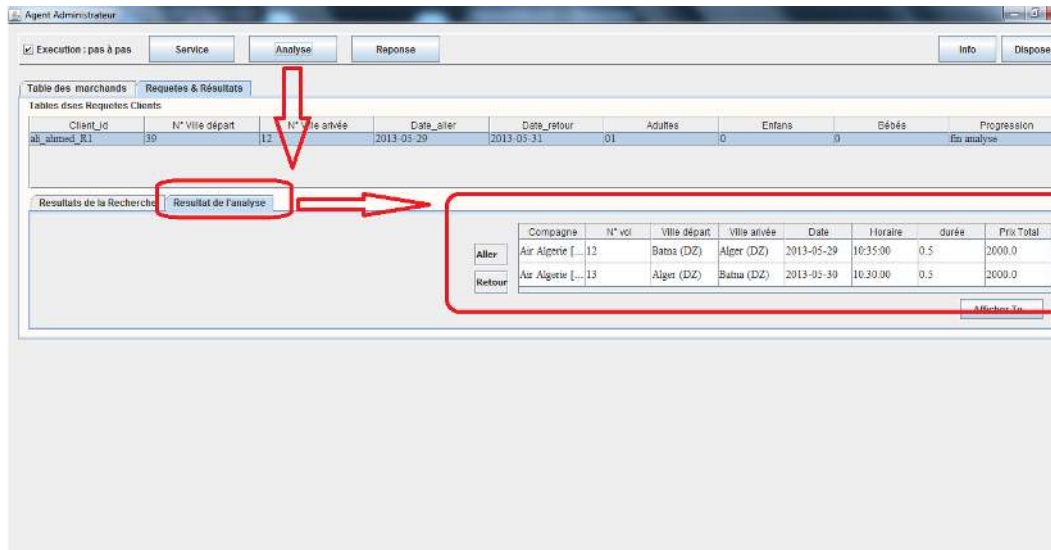


Figure V.23 : arriver le résultat vers l'agent analyseur pour analyser

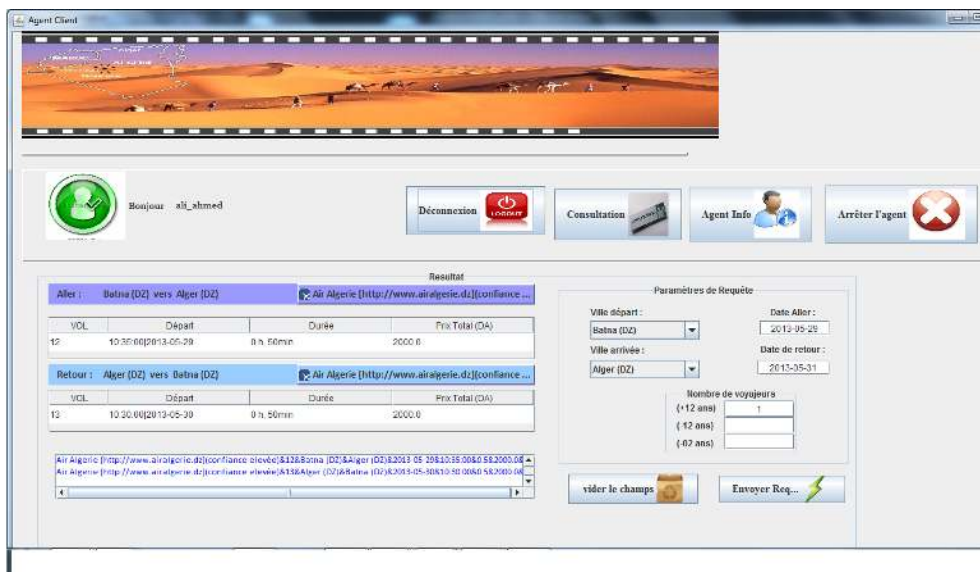


Figure V.24 : arriver le résultat vers le client

## **V.7. La conclusion**

Dans ce chapitre, nous avons proposé une étude de cas comme application de notre architecture. Nous avons essayé de mettre en œuvre l'ensemble des idées qui caractérise l'architecture proposée en se concentrant sur l'implémentation de la mobilité des agents.

Notre architecture est bien implémentée en utilisant le langage JAVA avec la plate-forme Aglet, qui permet d'un côté de développer des agents, et d'un autre côté d'être utilisée comme serveur des agents mobiles. Nous avons exploité la mobilité des agents à travers un réseau LAN.

Dans le cloud computing un seul agent mobile est crée. Il est chargé de se déplacer pour réaliser la tâche demandée par le client. Les résultats obtenus à partir de notre implémentation nous confirment que l'utilisation de plusieurs agents mobiles pour le cloud computing permet d'améliorer la securite et la qualité de la solution proposée et d'interagir localement avec un serveur, et donc de réduire le trafic sur le réseau en ne transmettant que les données utiles.

## Conclusion Générale

Tout au long de ce mémoire, nous avons présenté les différentes technologies nécessaires pour proposer une approche basée agents mobiles pour le cloud computing. Nous nous sommes intéressés à la technologie d'agents mobiles et à son utilisation dans un cloud computing. Le but du déplacement de ces agents est généralement d'accéder localement à des données ou à des ressources initialement distantes, d'effectuer le traitement en local et de ne déplacer que les données utiles. Un agent mobile est souvent exécuté dans un environnement d'exécution qui lui offre des services pour son déplacement, sa localisation et la communication avec les autres agents. La performance et la fiabilité de l'environnement d'exécution influencent beaucoup sur la qualité d'une application basée sur la technologie à "agents mobiles".

L'avantage de l'architecture proposée est qu'elle utilise les agents mobiles comme une entité de communication. Il s'agit ici de réduire le trafic sur le réseau et de diminuer la quantité d'informations échangées ; dans ce cas l'agent se déplace vers la source d'informations et effectue des échanges locaux.

Nous avons étudié l'utilisation de la technologie d'agents mobiles dans le domaine de le cloud computing. Dans ce contexte, un seul agent mobile est créé. Il est chargé de se déplacer pour réaliser la tâche demandée par le client. Nous pensons que l'utilisation de plusieurs agents mobiles pour le cloud computing permet d'améliorer la qualité de la solution proposée et de réduire le délai d'attente du client. Deux cas sont à envisager : 1) les agents se connaissent et peuvent communiquer entre eux afin de réaliser la tâches demandées ; on parlera d'agents coopératifs (c'est une perspective). 2) Les agents ne se connaissent pas et dans ce cas, chaque agent réalise sa tâche indépendamment des autres agents (c'est le cas de notre approche).

La réalisation de cette étude nous a conduites à suivre les étapes suivantes :

1. En premier lieu, nous avons fait le point sur les différents concepts clés du paradigme agent mobile.
2. Nous avons effectué une étude générale sur le cloud computing afin d'identifier leurs différentes caractéristiques et leurs besoins. Ainsi, nous illustrons l'intérêt d'utiliser les agents mobiles pour des applications réelles afin de montrer l'efficacité de la bonne utilisation de cette technologie.
3. Après, nous avons discuté un certain nombre de travaux et projets de recherche. La perspective de ces travaux est la gestion de le cloud computing en utilisant la technologie d'agents mobiles.
4. Ensuite, nous avons présenté notre architecture à base d'agent mobile pour le cloud computing, où on donne un intérêt particulier pour le rôle, les fonctions et les différentes interactions entre les agents.
5. Finalement, dans le but de montrer la faisabilité de l'architecture proposée d'un côté, et pour mieux comprendre son fonctionnement d'autre côté, nous avons

développé un prototype du système de réservation des billets aériens dans lequel nous avons montré l'apport de la technologie agent mobile dans la mise en œuvre des applications dans le cloud computing .

Ce mémoire constitue une base de travail à partir du quelle, de nouvelles activités de recherche peuvent être lancées afin d'améliorer le travail présenté. Les perspectives que nous proposons peuvent donc s'orienter vers les directions suivantes :

- L'intégration de techniques de traitement des langues naturelles notamment au niveau d'interface utilisateur, ceci afin de rendre le système plus performant.
- Utiliser des mécanismes de coopération entre les agents mobiles afin de réaliser effectivement les tâches demandées.
- Prendre en compte l'aspect d'adaptabilité des agents.
- Etendre l'architecture à l'usage de services web.

## Bibliographie

- [Amaz] Amazon voit le Cloud privé comme une arnaque  
: <http://www.zdnet.fr/blogs/cloud-news/amazon-voit-le-cloud-privé-comme-une-arnaque-39755491.htm>
- [Arsh 09] **Arshad, J. Townend, P. Jie Xu**, “Quantification of Security for compute Intensive Workloads in Clouds”, 15th International Conference on Parallel and Distributed Systems, School of Computation, pages 478-486, Dec. 2009, UK.
- [Bett 03] **Bettahar Aoued**, « Les Aglets d'IBM », Université de Montréal, BETA08036508, Cours IFT6802 – H2003..
- [Bla 05] **P. Blangi** : Etat de l'art sur les plates formes et les langages Multi-Agents Appliqués aux écosystèmes. Master recherche:modélisation et simulation des systèmes complexes (2004/2005).
- [Clou 09] **Cloud Computing & SaaS** de Guillaume Plouin, éditions Dunod, mars 2009.
- [CROS 05] **CROS, C. C. D.** Agents Mobiles Coopérants pour les Environnements Dynamiques Institut National Polytechnique de Toulouse, 2005
- [Desc 09] **Descher, M. Masser, P. Feilhauer, T. Tjoa, A.M. Huemer, D.**, “Retaining data control to the Client in Infrastructure Cloud”, International Conference on Availability, Reliability and Security, 2009, pages 9-16, Dornbirn.
- [Dian 09] **Diana Kelley**,” Cloud computing security model overview: Network infrastructure issues”, <http://searchcloudsecurity.techtarget.com/tip/>, 2009.
- [Dro 93]: **A. Drogoul** De La Simulation Multi-Agents à la résolution collective de problèmes. Une Étude De l'Émergence De Structures D'Organisation Dans Les Systèmes Multi-Agents. Thèse de Doctorat de L'université de Paris 1993.
- [Fer 95] **J. Ferber** Les systèmes Multi-Agents Vers une intelligence collective. Inter Edition, 1995, Paris.

- [Fug 98] **Fuggetta, A.; Picco, G. P. & Vigna, G.** Understanding Code Mobility  
IEEE Trans. Softw. Eng., IEEE Press, 1998, 24, 342-361
- [Jar 03] **I. Jarras, B. Chaib-Draa** Aperçu sur les systèmes Multi-Agents. Série  
Scientifique du centre inter universitaire de recherche en analyse des  
organisations -CIRANO-2002.
- [Ji Hi 09] **Ji Hu Klein**, “A Benchmark of transparent data encryption for migration  
of web application in cloud”, Eighth IEEE International Conference on  
Dependable, Autonomic and Secure Computing, 2009, pages 735 – 740,  
Chengdu.
- [John 61] **Informatique utilitaire de John McCarthy :**  
[http://computinginthecloud.wordpress.com/2008/09/25/utility-cloud  
computingflashback-to-1961-prof-john-mccarthy/](http://computinginthecloud.wordpress.com/2008/09/25/utility-cloud-computingflashback-to-1961-prof-john-mccarthy/)
- [Jens 09] **Jensen, M. Schwenk, J. Gruschka, N. Iacono**, “On technical security  
issues in Cloud” IEEE International Conference on Cloud Computing,  
2009, pages 109-16, Germany.
- [Lange 99] **Lange, D. B. & Oshima, M.** Seven good reasons for mobile agents  
Commun. ACM, ACM, 1999, 42, 88-89
- [LivC ] **Livre blanc** « Windows Azure™ Security Overview » par Charlie  
Kaufman et Ramanathan Venkatapathy
- [LivF 09] **Livre blanc de François Tonic** « Cloud Computing » Septembre 2009
- [LivIUT] **IUT Nancy Charlemagne** , « Le livre blanc du Cloud Computing »
- [LivM 09] **Livre blanc** « La protection de la vie privée à l'ère du Cloud Computing  
- Le point de vue de Microsoft », Microsoft, Novembre 2009
- [LivS ] **Syntec Informatique**, « Le livre blanc du Cloud Computing »
- [Mag 99] **Magedanz, T.** Activities Agents Technology in Europe ACTS Activities,  
1999
- [Math 09] **Mather, T., Kumaraswamy, S., & Latif, S.** *Cloud Security and  
Privacy*. New York: O'Reilly, 2009.

- [Milo 99] **Milojicic, D.** "Mobile Agent Applications," IEEE Concurrency (7:3), 1999, pp.80-90.
- [Mil 99] **Milojčić, D.; Breugst, M.; Busse, I.; Campbell, J.; Covaci, S.; Friedman, B.; Kosaka, K.; Lange, D.; Ono, K.; Oshima, M.; Tham, C.; Virdhagriswaran, S. & White, J.** MASIF, the OMG Mobile Agent System Interoperability Facility ACM Press/Addison-Wesley Publishing Co., 1999, 628-641
- [Pear 09] **Pearson, S.** "Taking account of privacy when designing cloud computing services" Software Engineering Challenges of Cloud Computing, 2009, pages, 44 – 52, Vancouver, BC.
- [Pos 01] **Poslad, S. & Charlton, P.** Standardizing agent interoperability: the FIPA approach Springer-Verlag New York, Inc., 2001, 98-117
- [Pri 11] **Priyank Singh Hada, Ranjita Singh , Mukul Manmohan Meghwal** Security Agents: A Mobile Agent based Trust Model for Cloud Computing *International Journal of Computer Applications (0975 – 8887) Volume 36– No.12, December 2011*
- [Saur 09] **Saurabh,** "Security issues in cloud Computing", <http://serl.iiit.ac.in/cs6600/saurabh.ppt>, 2009.
- [Tets 09] **Tetsuya, M. Kazuhiro, S. Hirotugu, K.** "A system for search, access restrictions and agents in the Clouds", Ninth Annual International Symposium on Applications and the Internet Cloud, 2009. Pages 201204,Japan.
- [Thor 97] **Thorn, T.** Programming languages for mobile code ACM Comput. Surv., ACM, 1997, 29, 213-239
- [Tsal 02] **Tsalgatidou, A. & Pilioura, T.** An Overview of Standards and Related Technology in Web Services Distrib. Parallel Databases, Kluwer Academic Publishers, 2002, 12, 135-162
- [Web 01] **Etude du marché du Cloud Computing :**  
<http://www.silicon.fr/le-cloud-computing-mode-ou-tendance-lourde-40238.html>  
<http://www.silicon.fr/cloud-computing-un-marche-a-149-milliards-de-dollars-en-2014-40946.html>  
<http://www.commentcamarche.net/news/5850141-la-virtualisation-et-le-cloud-bouleversent-le-monde-de-l-hebergement>

<http://www.ethantang.fr/2009/10/1378-le-cloud-computing-sera-la-tendance-technologique-en-2010/>

**[Web 02] Les couts cachés du Cloud :**

<http://davidpallmann.blogspot.com/2010/08/hidden-costs-in-cloud-part-1-driving.html>

**[Web 03] Eucalyptus.** <http://www.eucalyptus.com/>.

**[Web 04] Nimbus.** <http://www.nimbusproject.org/>.

**[Web 05] Opennebula.** <http://opennebula.org/>.

**[Web 06] Xen.** <http://wiki.xen.org/>.

**[Web 07] Abicloud.** <http://community.abiquo.com/>.

**[Web 08]** [http://fr.wikipedia.org/wiki/Java\\_%28langage%29](http://fr.wikipedia.org/wiki/Java_%28langage%29)

**[Web 09]** [http://fr.wikipedia.org/wiki/Java\\_Development\\_Kit](http://fr.wikipedia.org/wiki/Java_Development_Kit)

**[Web 10]** <http://www.jmdoudoux.fr/java/dej/chap-jvm.htm>

**[WEB 11]** <http://stromboli.it-sudparis.eu/~bernard/ipr/projets97-98/agents-rapport/aglet.htm>

**[Zha 07] Zhang, J.; Wang, Y. & Varadharajan, V. Mobile Agent and Web Service Integration Security Architecture SOCA '07: Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications, IEEE Computer Society, 2007, 172-179**