

*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 M-Business**

*Par :*

**Mr. ALOUI Ahmed**

Soutenu le : / /

### **Devant le jury :**

Dr. BILAMI Azeddine,  
Dr. KAZAR Okba,  
Dr. ZIDANI Abdelmadjid,  
Dr. BELATTAR Brahim,

Pr. Université de Batna  
Pr. Université de Biskra  
M.C. Université de Batna  
M.C. Université de Batna

Président  
Rapporteur  
Examineur  
Examineur

# *Remerciements*

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

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

*Mes remerciements vont au Mr. BILAMI Azeddine, Professeur à l'université de Batna, qui a accepté de présider le jury et à tout les membres de jury : Mr. ZIDANI Abdelmadjid, Mr. BELATTAR Brahim Maîtres de conférence à l'université de Batna, 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 chères parents, pour leur patience et leur soutien morale que spirituel pendant les moments difficiles. Sans leur conseil et sans leur amour rien du tout ceci ne s'aurait arrivé. Je ne serais oublier de remercier tout mes collègues de la promo de poste 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 tout la période de ce mémoire.*

**Résumé :** Des utilisateurs aujourd'hui veulent avoir la possibilité de faire (ou gérer) des transactions commerciales à tout moment et n'importe où via leurs appareils mobiles. Les applications e-business (business électronique) exigent que les utilisateurs doivent se connecter aux sites Web prévus à partir de leurs ordinateurs personnels ou des terminaux publics. En outre, les utilisateurs ont souvent besoin de visiter de nombreux sites et sont toujours impliqués dans un processus fastidieux. La plupart des applications d'e-business utilisent le modèle traditionnel client/serveur dans laquelle une opération commerciale exige généralement un lien de communication stable établi entre le client et le serveur, ce modèle possède l'inconvénient d'augmenter le trafic sur le réseau et exige une connexion permanente. Dans ce travail, nous proposons une nouvelle approche qui utilise les agents mobiles. Ces derniers apparaissent dans ce contexte comme une solution prometteuse facilitant la mise en œuvre d'application m-business. Ce mémoire propose une architecture basée agents mobiles pour les business mobiles (m-business). Le m-business est apparu comme l'approche prometteuse pour conduire la vague suivante d'e-business, le M-business représente les affaires commerciales conduites à l'aide d'appareils mobiles (portables, assistants numériques, etc...). Les agents mobiles seront utilisés sur un seul niveau : agent de recherche. Chaque agent mobile de recherche sera utilisé pour visiter le site serveur cible de l'application afin de collecter des informations pour son client, ce qui permet à ce dernier d'interagir localement avec un serveur, et donc de réduire le trafic sur le réseau en ne transmettant que les données utiles.

**Mots-clés :** Système multi-agents, agent mobile, M-Business

**Abstract:** Users today want the opportunity to make (or manage) a businesses in anytime and anywhere via their mobile devices. The e-business (electronic business) applications require that the users must connect to Web sites under their personal computers or public terminals. Besides, the users often need to visit numerous sites and are always involved in a tedious process. The most of e-business applications using the traditional client / server model in which a commercial operation generally requires a stable communication link being established between the client (customer) and server, this model possesses the disadvantage of traffic's increasing on the network and requires a permanent connection. In this work, we propose a new approach that uses the mobile agents. The mobiles agent appear in this context as a promising solution facilitating the implementation of an m-business application. This work proposes mobile agents based architecture for mobile business (m-business), m-business appeared as the promising approach to drive the next wave of e-business, the m-business represents the commercial business conducted using mobile devices (portable, digital assistants, etc.). The Mobile agents will be used on a single level: research agent. Every research mobile agent will be used to visit the target server site of the application to collect the information's for his client, which allows it to interact locally with a server, and so to reduce the traffic on the network by transmitting only the useful data.

**Key words:** System multi-agent, mobile agent, M-Business.

**ملخص :** في الوقت الحالي أصبح المستخدمين يريدون إجراء أو إدارة المعاملات التجارية في أي وقت وفي أي مكان عبر أجهزتهم النقالة. تطبيقات الأعمال الإلكترونية تتطلب على المستخدمين الاتصال إلى المواقع المقدمة من أجهزة الكمبيوتر الشخصية الخاصة بهم أو المحطات العامة. بالإضافة إلى ذلك، المستخدمين كثيرا ما يحتاجون إلى زيارة العديد من المواقع و تطبيق العديد من العمليات الشاقة. معظم تطبيقات الأعمال الإلكترونية تستخدم النموذج التقليدي "العميل / الخادم" في صفقة تجارية والتي تتطلب عادة خط اتصال مستقر بين العميل والخادم، هذا النموذج يحتوي على العديد من المساوئ و المشاكل. من أجل هذا نقترح طريقة جديدة مبنية على أساس العون المتنقل من أجل تسهيل تنفيذ الأعمال المتنقلة. يمكن دور هذا العميل في اختصار وقت البحث و تقليل كثافة المعلومات المتنقلة عبر الشبكة.

**الكلمات الأساسية :** الأنظمة متعددة الوكلاء, العون المتنقل, الأعمال المتنقلة.

# *Table des matières*

<b>Introduction Générale .....</b>	<b>1</b>	
<b>Chapitre I</b>	<b>Les Agents Mobiles.....</b>	<b>4</b>
<b>1. Introduction .....</b>		<b>4</b>
<b>2. Les systèmes multi-agents.....</b>		<b>5</b>
<b>2.1 Définition.....</b>		<b>5</b>
<b>3. Etude du concept d'agent .....</b>		<b>6</b>
<b>3.1 Définition.....</b>		<b>6</b>
<b>4. Caractéristiques des agents .....</b>		<b>7</b>
<b>5. Typologie des agents.....</b>		<b>8</b>
<b>5.1 Classification des agents selon la granularité.....</b>		<b>8</b>
<b>5.1.1 Les agents cognitifs.....</b>		<b>8</b>
<b>5.1.2 Les agents réactifs.....</b>		<b>9</b>
<b>5.1.3 les agents pro-actifs .....</b>		<b>10</b>
<b>5.2 Classification des agents selon la mobilité.....</b>		<b>10</b>
<b>6. L'environnement dans un système multi-agents .....</b>		<b>10</b>
<b>6.1 L'interaction entre les agents .....</b>		<b>11</b>
<b>6.1.1 La coopération .....</b>		<b>12</b>
<b>6.1.2 la collaboration .....</b>		<b>12</b>
<b>6.1.3 La coordination.....</b>		<b>12</b>
<b>6.1.4 La négociation.....</b>		<b>12</b>
<b>6.2 La communication entre les agents.....</b>		<b>13</b>
<b>6.2.1 Communication par partage d'information.....</b>		<b>13</b>
<b>6.2.2 Communication par envoi de messages .....</b>		<b>13</b>
<b>6.3 Les ACL (Agent Communication Language) .....</b>		<b>14</b>
<b>7. Les agents mobiles.....</b>		<b>14</b>
<b>7.1 Définition.....</b>		<b>14</b>
<b>7.2 Motivation .....</b>		<b>15</b>
<b>7.3 Structure d'un agent mobile.....</b>		<b>16</b>
<b>7.4 Migration d'un agent .....</b>		<b>17</b>
<b>7.4.1 Migration forte.....</b>		<b>17</b>
<b>7.4.2 Migration faible .....</b>		<b>17</b>
<b>7.5 Tolérance aux pannes.....</b>		<b>18</b>

7.6	Plate-forme et Standardisation .....	19
8.	Les normes .....	19
9.	Implémentations existantes.....	21
10.	Conclusion .....	22
<b>Chapitre II</b>		
	<b>M-business : état de l'art .....</b>	<b>23</b>
1.	Introduction .....	23
2.	Les technologies mobiles.....	23
3.	Business Mobile.....	25
3.1	Définition de Business Mobile .....	25
3.2	La chaîne de valeur de l'activité mobile .....	26
3.3	Services dans Business Mobile .....	26
3.3.1	Business-to-Business.....	26
3.3.2	Business-to-Employée.....	26
3.3.3	Business-to-Consumer.....	27
3.3.4	Machine-to-Machine .....	27
4.	Business électronique .....	27
4.1	Modèle client/serveur .....	28
4.2	Les inconvénients du modèle Client/Serveur .....	28
5.	Avantages et améliorations de M-business par rapport à E-business.....	29
6.	Les technologies de communication.....	31
1)	Wireless Personal Area Network .....	32
2)	Wireless Local Area Network (WLAN).....	32
3)	Wireless Local Area Network (WWAN) .....	34
7.	Les services associés aux réseaux mobiles .....	35
7.1	Le protocole WAP (Wireless Application Protocol).....	35
7.1.1	Services WAP .....	35
7.2	Le service i-Mode.....	36
8.	Les dispositifs Mobiles.....	37
8.2	Pagers (Téléavertisseurs) .....	37
9.	Déterminants de la réussite de Business Mobile.....	38
9.1	Conditions préalables nécessaires .....	38
10.	Obstacles et les défis de M-business .....	38
11.	Conclusion .....	40
<b>Chapitre III</b>		
	<b>Approches existantes à base d'agent mobile pour m-business .....</b>	<b>41</b>
1.	Introduction .....	41

2.	La technologie agent mobile et Internet.....	42
3.	Les travaux connexes.....	42
3.1	Présentation de Zhiyong Weng et Thomas Tran.....	42
3.1.1	Les types d'agents.....	44
3.1.2	Migration des agents.....	45
3.1.3	Processus de négociation.....	45
3.1.4	Implémentation du système.....	47
3.2	Présentation du système de J-Phone.....	48
3.2.1	La plateforme de l'approche.....	50
3.3	L'architecture de <i>Gilda Pour</i> .....	50
3.3.1	Architecture de Gilda.....	50
3.3.2	Les types d'Agents.....	51
3.3.3	La plateforme de l'approche.....	51
3.3.4	La sécurité.....	52
3.3.5	Un système de gestion des stocks.....	52
3.4	Le système IMAGO.....	54
4.	Synthèse.....	56
5.	Conclusion.....	59
Chapitre IV Proposition d'un environnement de m-business.....		60
1.	Introduction.....	60
2.	Objectif et Motivation du travail.....	60
2.1	Objectif.....	60
2.2	Motivation.....	61
3.	Approche proposée.....	62
3.1	L'architecture du système.....	63
3.2	Les types d'agents.....	64
4.	L'architecture détaillée du système.....	66
a.	Partie de l'utilisateur.....	67
b.	Serveur d'application.....	67
c.	Le site du fournisseur.....	68
5.	Fonctionnement détaillé de l'approche.....	68
5.1	La migration des agents.....	69
6.	L'architecture des agents.....	70
6.1	Agent client (AC).....	70
6.2	Maître aglet (MA).....	71

6.3	Les Agents Mobile de Recherche (AMR) .....	72
6.4	Agent Répertoire (AR) .....	73
6.5	Agent de base de données (répertoire) (ABDR).....	74
6.6	Agent fournisseur (AF) .....	75
6.7	Agent de base de données (fournisseur) (ABDF).....	76
7.	Les diagrammes AUML .....	77
7.1	Diagramme de cas d'utilisation.....	78
7.2	Diagramme de séquences .....	79
7.2.1	Diagramme de séquence (inscription).....	79
7.2.2	Diagramme de séquence (Authentification) .....	80
7.2.3	diagramme séquence « Lancement requête » .....	81
8.	Protocole d'interaction du système .....	81
8.1	Lancement agent.....	82
8.2	Négociation.....	83
9.	Diagramme de classes.....	84
9.1	Les classes d'agents .....	84
9.2	Diagramme des classes du système .....	90
10.	Conclusion .....	91
Chapitre V: Etude de cas et implémentation .....		92
1.	Introduction .....	92
2.	Environnement de développement.....	92
2.1	Langage de programmation .....	92
2.2	Les servlets .....	93
2.3	J2ME.....	93
2.3.1	L'architecture de J2ME .....	93
2.4	MIDlet.....	95
2.5	Serveur de base de données : MySQL .....	95
2.6	JADE LEAP .....	95
2.7	Serveur Web .....	95
2.8	NetBeans IDE 6.7.1.....	96
2.9	Choix de la plate-forme.....	96
2.9.1	Description générale de la plate-forme Aglets .....	96
2.9.1.1	Les éléments de base d'Aglets.....	97
2.9.1.2	Opérations fondamentales .....	98
2.9.2	Serveur Tahiti .....	99

2.9.3	Communication inter-aglet.....	100
2.9.3.1	Communication par messages .....	101
2.10	La norme MASIF .....	101
3.	Description générale de l'application.....	101
4.	Présentation de l'étude de cas.....	103
5.	La réalisation du système .....	104
5.1	Partie l'utilisateur.....	104
5.2	Serveur d'application.....	106
5.3	Les sites fournisseurs.....	110
5.4	Résultat de la recherche.....	115
5.5	Communication .....	116
6.	Conclusion.....	118
	Conclusion Générale.....	119
	Bibliographie .....	121

# Liste des figures

## Chapitre I Les Agents Mobiles et leurs spécificités

Figure 1.1: Structure d'un agent.....	8
Figure 1.2: Exécution asynchrone et autonome.....	15

## Chapitre II M-business : état de l'art

Figure 2.1: Assistants numériques personnels.....	37
---	----

## Chapitre III Approches existantes à base d'agent mobile pour le M-business

Figure 3.1: Structure et processus du système proposé par Zhiyong Weng et al. ....	43
Figure 3.2: Migrations des agents. ....	45
Figure 3.3: Processus de négociation .....	46
Figure 3.4: environnement d'expérimentation.....	47
Figure 3.5: Présentation du système de vente aux enchères de J-phone. ....	49
Figure 3.6: Conception de haut niveau de l'architecture à base d'agents mobile.....	51
Figure 3.7: Diagramme de séquence pour un système de gestion des stocks.....	53
Figure 3.8: Un exemple d'utilisation des agents mobiles dans le M-commerce. ....	55

## Chapitre IV Proposition d'un environnement de M-business

Figure 4.1: Un environnement de business mobile. ....	63
Figure 4.2: Architecture du système.....	64
Figure 4.3: L'architecture détaillée. ....	66
Figure 4.4: Architecture interne de l'agent client.....	71
Figure 4.5: architecture interne de Maître Aglet. ....	72
Figure 4.6: Architecture interne des agents mobiles de recherche. ....	73
Figure 4.7: architecture interne d'agent répertoire. ....	74
Figure 4.8: architecture interne d'agent de base de données (répertoire).....	75
Figure 4.9: architecture interne d'agent fournisseur.....	76
Figure 4.10: Architecture interne d'agent de base de données (fournisseur). ....	77
Figure 4.11: Diagramme de cas d'utilisation du système.....	78
Figure 4.12: Diagramme de séquence « Inscription fournisseur» .....	79
Figure 4.13: Diagramme de séquence « Inscription client» .....	80
Figure 4.14: Diagramme de séquence « Authentification client» .....	80
Figure 4.15: Diagramme de séquence « Lancement requête».....	81
Figure 4.16: Diagramme de séquence « Lancement agent».....	82
Figure 4.17: Diagramme de séquence pour informé client.....	83
Figure 4.18: Diagramme de séquence « négociation ».....	84
Figure 4.19: Diagramme de classe de l'agent mobile de recherche .....	85
Figure 4.20: Diagramme de classe de Maître Aglets.....	86
Figure 4.21: Diagramme de classe d'agent répertoire.....	87
Figure 4.22: Diagramme de classe d'agent fournisseur. ....	88
Figure 4.23: Diagramme de classe d'agent Client.....	89
Figure 4.24: Diagramme de classe du système.....	90

## Chapitre V Etude de cas et Implémentation

Figure 5.1: Relation entre un Aglet et son Proxy. ....	98
Figure 5.2: serveur Tahiti. ....	99
Figure 5.3: Environnement de test.....	104
Figure 5.4: les agents de répertoire (Agent répertoire et Agent de base de données de répertoire). ....	107
Figure 5.5: Création des agents de recherche par Maître Aglets.....	108
Figure 5.6: Maître Aglets. ....	108
Figure 5.7: Maître Aglets créer un agent mobile de recherche.....	109
Figure 5.8: Création d'un <i>clone</i> dans le même contexte. ....	109
Figure 5.9: Maître Aglets envoyé les agents mobile à des fournisseurs.....	110
Figure 5.10: Fournisseur 01 (Agent fournisseur et Agent de base de données de fournisseur). .....	111
Figure 5.11: Fournisseur 02 (Agent fournisseur et Agent de base de données de fournisseur). .....	111
Figure 5.12: Fournisseur 03 (Agent fournisseur et Agent de base de données de fournisseur). .....	112
Figure 5.13: Fournisseur 04 (Agent fournisseur et Agent de base de données de fournisseur). .....	112
Figure 5.14: Fournisseur 04 reçoive l'agent mobile.....	113
Figure 5.15: Fournisseur 02 reçoive l'agent mobile.....	113
Figure 5.16: L'agent mobile 01 revenir vers le Maître Aglets.....	114
Figure 5.17: L'agent mobile 02 revenir vers le Maître Aglets.....	115
Figure 5.18: les agents mobiles retour à Maître Aglets.....	115
Figure 5.19: Affichage des résultats. ....	116
Figure 5.20: Communication par message entre Maître Aglets et Agent mobile de recherche. .....	117
Figure 5.21: Communication par message. ....	117

## *Liste des tables*

<b>Tableau 2.1</b> : Les différences entre les applications e-business et les applications m-business.....	<b>30</b>
<b>Tableau 2.2</b> : Réseaux sans fil et leurs technologies.....	<b>32</b>
<b>Tableau 5.1</b> : Les copies d'écran d'agent Client.....	<b>106</b>

# Introduction Générale

L'évolution des réseaux à grande échelle a permis la naissance d'un grand nombre de nouvelles applications qui se développent autour de ce type de réseau comme business mobile (m-business). Aujourd'hui les affaires commerciales sont une partie importante dans la vie quotidienne depuis les années immémoriales, et avec les progrès dans les capacités de calcul et de communication des appareils portatifs tels que les PDA, Pocket PC, ordinateurs portables et d'autres, nous commençons à voir l'émergence d'une variété d'applications dans m-business. La puissance de ces appareils mobiles incitant de plus en plus des consommateurs et des entreprises à les utiliser dans leurs affaires quotidiennes, ainsi que dans des activités récréatives.

Les applications d'e-business (électronique business) reposant sur le modèle client/serveur qui constituent un obstacle au développement d'application de m-business. Ce modèle souffre de nombreux problèmes car dans ce modèle, seul le client représente une application au sens propre du terme et le rôle du serveur est de répondre aux demandes des clients. Le serveur construit ses réponses indépendamment du client. Ainsi une partie des données envoyées est inutile augmentant ainsi le trafic sur le réseau. De plus, ce modèle exige une connexion permanente entre le client et le serveur, ce qui n'est pas le cas des terminaux mobiles qui sont exposés à la perte de la connexion.

Nous avons utilisé les agents mobiles pour régler ses problèmes qui constituent un paradigme de programmation adéquat et efficace pour ce genre d'application. L'objectif principal de notre projet est de proposer une architecture d'environnement basée agents mobiles pour M-business, en s'appuyant sur la capacité de la mobilité des agents.

Un agent mobile est une entité logicielle qui se déplace d'un site à un autre en cours d'exécution pour accéder à des données ou à des ressources distantes. Il se déplace avec son code, son état d'exécution et ses données propres. La décision de migration peut se faire à l'initiative de l'agent lui-même, de manière autonome ; la mobilité est ainsi contrôlée par l'application et non par le système d'exécution. Le but du déplacement 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. Nous distinguons deux rôles essentiels pour un agent mobile : les échanges de données et l'agent mobile calculateur. Dans une application donnée, un agent peut jouer l'un des deux rôles ou les deux en même temps [Fal 06].

Le but de l'utilisation d'un agent de communication est la réduction du trafic, ainsi en envoyant l'agent là où les tâches se font, les messages échangés deviennent locaux et libèrent d'autant la charge du réseau.

D'autre part, de nombreux travaux ont été élaborés afin d'introduire la technologie d'agents mobiles et les concepts liés à cette dernière pour les affaires commerciales mobiles. Le concept d'agent mobile apparaît dans ce contexte comme une solution facilitant la mise œuvre d'applications dynamiquement adaptables, et il offre un cadre générique pour le développement des applications réparties.

Notre travail s'inscrit dans cette optique. Nous étudions l'utilisation de la technologie d'agents mobiles dans le domaine du business mobile. Le business mobile représente des affaires commerciales conduites à l'aide d'appareils mobiles. Généralement, dans la m-business, 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 permet d'améliorer la qualité de la solution proposée et de réduire le délai d'attente du client. Le rôle de chaque agent est de se déplacer entre les différentes machines de l'application, sur chaque machine il effectue des échanges et filtre les informations collectées. Ce filtrage permet de réduire la quantité d'informations transportées avec ces agents et par conséquent le trafic sur le réseau. Chaque agent transporte avec lui les données demandées par son client.

L'objectif de ce mémoire consiste donc à proposer une architecture d'environnement basée agents mobiles pour M-business, en s'appuyant sur la capacité de la mobilité des agents.

Ce mémoire est constitué de cinq chapitres répartis comme suit :

Dans le premier chapitre, nous présentons un état de l'art sur les systèmes multi-agents et les agents mobiles. Nous détaillons les principaux services qu'un environnement d'exécution devrait fournir à ces agents. Nous évoquons également la migration d'un agent et l'interaction entre les agents. Finalement nous discutons les différentes plates-formes d'agents mobiles existantes de nos jours.

Dans le deuxième chapitre, nous présentons la définition de business mobile, services dans business mobile et les avantages et améliorations de m-business par rapport à e-business. Puis nous présentons les technologies de communication et les dispositifs mobiles et les obstacles et les défis de m-business.

Le troisième chapitre est consacré à la présentation de quelques architectures basées agent mobile, qui sont conçus pour les affaires commerciales mobiles.

Dans le quatrième chapitre, nous allons décrire notre contribution qui est la proposition d'une approche basée agents mobiles pour les affaires commerciales mobiles.

Dans le cinquième chapitre, nous poursuivons notre recherche avec une étude de cas afin de valider notre proposition. On a choisi comme cas à étudier l'organisation des achats à l'aide d'appareils mobiles (portables, assistants numériques, etc...) car elle est considérée comme un exemple typique pour les affaires commerciales mobiles. Nous précisons ensuite la plate-forme « Aglets » et comment les différents composants de notre architecture (interaction d'agents, spécification d'agents, etc ...) peuvent être implémentés en utilisant cette plate-forme.

Enfin, nous terminons ce mémoire par une conclusion générale, qui récapitule les travaux réalisés et fait le point sur un ensemble de perspectives envisagées.

# Chapitre I

---

## Les Agents Mobiles

---

### 1. Introduction

Les systèmes multi-agents (SMA) sont devenus un paradigme dominant dans le domaine de développement des systèmes distribués complexe. Leurs avantages consistent notamment en leur capacité d'aborder les problèmes complexe d'une manière distribuée et de proposer, en outre, des solutions réactives robustes. Les concepts qui y sont liés font les SMA une approche prometteuse pour la conception des systèmes complexe. La mobilité est une caractéristique essentielle des stations mobiles qui peuvent utiliser une connexion sans-fil pour communiquer pendant un déplacement et un point d'accès fixe au réseau traditionnel pour se reconnecter après un déplacement.

D'autre part le paradigme agent mobile est devenu aussi un formalisme très puissant pour le développement des applications réparties. Les agents mobiles sont des programmes autonomes qui peuvent se déplacer à travers un réseau hétérogène sous leur propre contrôle. Il se déplace avec son code son état d'exécution et ses données propres. Le but du déplacement est 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. Le business mobile pourrait aussi inclure toutes les activités, les processus et les applications qui sont réalisées ou soutenues par les technologies mobiles.

L'objectif de ce chapitre est d'initier le lecteur novice au domaine de système multi-agent qui constitue un des piliers de notre travail. Nous nous intéressons d'abord aux entités qui composent cette catégorie de système : les agents. Pour une telle raison, nous allons présenter, en premier lieu, des considérations générales sur les notions d'agent et de systèmes multi-agents. Par la suite, nous procédons par un état de l'art sur la technologie d'agents mobiles.

## 2. Les systèmes multi-agents

### 2.1 Définition

La plupart des auteurs s'accordent généralement pour définir un système multi-agents (SMA) comme un système composé d'agents qui communiquent et collaborent pour achever des objectifs spécifiques personnels ou collectifs.

Pour [Fer 95], « un Système Multi-Agents est un système composé des éléments suivants :

- Un environnement **E**, c'est-à-dire un espace disposant généralement d'une métrique.
- Un ensemble d'objets **O**. Ces objets sont situés, c'est-à-dire que pour tout objet, il est possible, à un moment donné, d'associer une position dans **E**. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
- Un ensemble **A** d'agents qui sont des objets particuliers ( $A \subseteq O$ ), lesquels représentent les entités actives du système.
- Un ensemble de relations **R** qui unissent des objets (et donc des agents) entre eux.
- Un ensemble d'opérations **Op** permettant aux agents de **A** de percevoir, produire, consommer, transformer, et manipuler des objets de **O**.
- Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

Toujours d'après Ferber, dans un SMA :

- chaque agent possède des informations ou des capacités de résolution de problèmes limitées, chaque agent a un point de vue partiel ;
- il n'y a pas de contrôle global du système ;
- les données sont décentralisées ;
- le calcul est asynchrone.

### **3. Etude du concept d'agent**

#### **3.1 Définition**

Il n'existe pas de définition adoptée par la communauté sur le terme agent, en revanche tout le monde s'accord à dire que la notion d'autonomie est au centre de la problématique des agents. Dans la littérature, on trouve une multitude de définitions d'agents. Elles se ressemblent toutes, mais diffèrent selon le type d'application pour laquelle est conçu l'agent [Jar 02].

Nous proposons une première définition suivante : « un agent est une entité logicielle ou physique à qui est attribuée une certaine mission qu'elle est capable d'accomplir de manière autonome et en coopération avec d'autres agents » [Bri 01]

Une deuxième définition est assez proche de la première proposée par Wooldridge and Jennings: « Un agent est un système informatique capable d'agir de manière autonome et flexible dans un environnement changeant ». [Woo 95]

Jacques Ferber [Fer 95] propose une définition plus précise et plus contraignante des agents puisqu'il fixe neuf caractéristiques pour ces derniers :

« Un agent est une entité physique ou virtuelle :

- a. qui est capable d'agir dans un environnement,
- b. qui peut communiquer directement avec d'autres agents.
- c. qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser).
- d. qui possède des ressources propres.
- e. qui est capable de percevoir (mais de manière limitée) son environnement.
- f. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune).
- g. qui possède des compétences et offre des services,
- h. qui peut éventuellement se reproduire,

i. dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit ».

#### **4. Caractéristiques des agents**

En pratique, un agent nécessite quelques propriétés additionnelles, la liste ci-dessous présente quelques propriétés que les agents puissent posséder, certaines ces propriétés sont importantes (tel que d'être autonomes, interactifs et réactifs) car sans elles, un agent n'est pas efficace. Les autres sont optionnelles (tel que la mobilité, l'intelligence, etc...) et peuvent être employées dans différentes combinaisons [Fra 99].

Les caractéristiques principales des agents sont:

- **La nature:** agents physiques ou virtuels.
- **Autonomie :** capable d'agir sans intervention externe directe.
- **Intelligence:** On appelle agent intelligent un agent cognitif, rationnel, proactif et adaptatif.
- **Raisonnement:** l'agent peut être lié à un système expert ou à d'autres mécanismes de raisonnements plus ou moins complexes.
- **Mobilité :** cette aptitude caractérise uniquement les agents mobiles. Ceux-ci peuvent se déplacer vers d'autres environnements, ils peuvent transporter avec eux des données avec des instructions qui seront exécutées sur des sites distants.
- **L'anticipation:** l'agent peut plus ou moins avoir les capacités d'anticiper les événements futurs.
- **Efficacité :** C'est une autre caractéristique très importante, C'est-à-dire la capacité d'agent à résoudre le problème, à atteindre ses buts avec une rapidité d'exécution et d'intervention.
- **La contribution:** l'agent participe plus ou moins à la résolution du problème ou à l'activité globale du système.
- **Adaptabilité:** un agent adaptable est un agent capable de contrôler ses aptitudes (communicationnelles, comportementales, etc...) selon l'environnement.

- **Social** : l'agent doit être capable d'interagir avec des autres agents (logiciels ou humains) afin d'accomplir des tâches ou aider ces agents à accomplir les leurs.
- **proactif** : l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au bon moment.

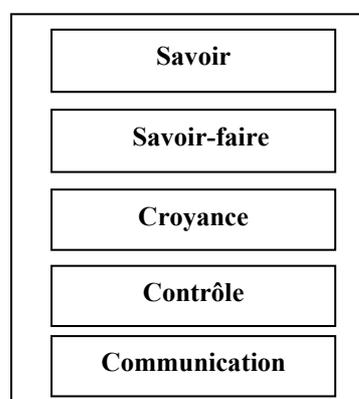
## 5. Typologie des agents

Les agents peuvent être classés selon différentes propriétés [Zga 07], par exemple : la granularité, le rôle, la mobilité, etc...

### 5.1 Classification des agents selon la granularité

#### 5.1.1 Les agents cognitifs

Les agents cognitifs sont dotés de capacités de raisonnement importantes. Leur comportement ne se limite pas à répondre immédiatement aux stimuli venant de l'extérieur. Ils disposent d'une représentation de leur environnement et accumulent de l'expérience. Pour atteindre leurs buts, ces agents sont capables d'échafauder des plans et de coopérer. Ils utilisent l'expérience qu'ils ont acquise pour prendre des décisions. Dans de nombreux systèmes, ces agents sont insérés dans des réseaux sociaux et cherchent à coopérer avec d'autres agents pour arriver à atteindre leurs fins. Dans ce cadre, un système multi-agent est composé d'un petit nombre d'agents "intelligent". Ces agents possèdent des buts et des plans explicites, leur permettant d'accomplir leurs buts. On dit aussi que ces agents sont «intentionnels» c'est-à-dire qu'ils possèdent des buts et des plans explicites leur permettant d'accomplir leurs buts [Fer 95].



**Figure 1.1:** Structure d'un agent.

- **Savoir-faire**

C'est une interface permettant la déclaration des connaissances et les compétences de l'agent. Il permet la sélection des agents à solliciter pour une tâche de donnée.

Il n'est pas nécessaire mais il est très utile pour améliorer les performances du système quel que soit le mode de coopération.

- **Croyance**

Dans un univers multi-agents, chaque agent possède des connaissances sur lui-même et sur les autres agents de son univers. Cette est à la base de la conception de système multi-agents puisque c'est elle qui détermine en grande partie le comportement «intelligent» de chacun des agents. Les logiques des connaissances contrôlent et des croyances se préoccupent d'une telle représentation.

- **Contrôle**

Le contrôle dans un agent regroupe la connaissance sur les buts, les plans et les tâches.

- **Savoir**

C'est la connaissance sur la résolution de problème, pour un système expert utilisant le formalisme de règles, par exemple, cette connaissance correspond à sa base de règle.

- **Communication**

La communication est l'ensemble des processus physiques et psychologiques par lesquels s'effectue l'opération de mise en relation d'un ou plusieurs acteurs ou agents émetteurs avec un ou plusieurs acteurs ou agents récepteurs en vue d'atteindre certains objectifs [Kaz 05].

### **5.1.2 Les agents réactifs**

Ce sont des agents passifs qui réagissent seulement à un stimulus. Ce type d'agent ne dispose pas de module de raisonnement interne.

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. Des mécanismes de réaction aux événements, ne prend en compte

ni une explication des buts, ni des mécanismes de planifications, peuvent alors résoudre des problèmes qualifiés de complexes [Fer 95]. Les agents réactifs leurs actions sont provoquées par des réactions réflexes aux changements de leurs environnements. Généralement un système réactif comprend un grand nombre d'agent de faible granularité.

### 5.1.3 les agents pro-actifs

Ce sont des agents dynamiques qui entreprennent car ils possèdent, en plus de leurs attributs et méthodes, des processus internes qui leur permettent de prendre des initiatives pour réaliser leurs buts. Un agent pro-actif est donc un agent dirigé buts.

## 5.2 Classification des agents selon la mobilité

- **les agents stationnaires** : cet agent agit localement, pendant tout son cycle de vie. dans la machine là où il a été implanté initialement.
- **les agents mobiles** : Contrairement à un agent stationnaire, un agent mobile est capable de se déplacer à travers un réseau, d'un nœud à un autre pour agir à son propre compte ou à la demande d'un autre agent. C'est un paradigme de plus en plus utilisé dans le contexte des réseaux largement distribués. La notion de la mobilité est évoquée plus en détail dans section 7.

## 6. L'environnement dans un système multi-agents

Pour former un système multi-agents les agents doivent être immergés dans un environnement. En effet, l'ensemble des perceptions et des actions qu'un agent est susceptible de réaliser est entièrement défini par rapport à l'environnement où celui-ci va opérer. Un environnement peut représenter :

- Un lieu où des actions individuelles ou collectives sont réalisées et où des réactions sont perçues ;
- Un espace de déplacement : grille, position des agents, etc. ;
- Un moyen de structuration des agents : relations de proximité, définition des topologies spatiales ou temporelles, etc. ;
- Une source de données pour le système ;
- Un lieu où des ressources sont disponibles.

Pour Russell [Rus 03] les propriétés permettant de caractériser les environnements sont les suivantes :

- **Accessible / inaccessible** : un environnement est dit accessible lorsqu'un agent peut obtenir la totalité des informations sur l'environnement où plus simplement l'ensemble de celles qui lui sont nécessaires pour décider et agir. Généralement les agents n'ont qu'une perception limitée de leur environnement et la portée de leur action reste locale.
- **Statique / dynamique** : un environnement est dit statique lorsqu'il ne possède pas de dynamique propre, dans ce cas les agents sont les seuls à pouvoir le modifier.
- **Discret / continu** : un environnement est discret lorsqu'il existe seulement un nombre fini et fixe de perceptions et d'actions possibles sur lui.

Ferber [Fer 95] ajoute une propriété supplémentaire permettant de caractériser un environnement :

- **Centralisé** : un environnement est centralisé lorsque tous les agents ont accès à la même structure de données contenant toutes les informations environnementales.
- **Distribué** : un environnement est distribué lorsqu'il est composé d'un ensemble de cellules disposées en réseau. Chaque cellule se comporte comme un mini environnement centralisé. Dans un environnement distribué les cellules interagissent avec leurs voisines, propagent des signaux, hébergent des agents et permettent de modéliser des phénomènes liés au voisinage.

## 6.1 L'interaction entre les agents

L'interaction représente l'une des plus importantes facettes d'un SMA. Elle est définie par Doniec [Don 06] comme étant une relation dynamique instaurée entre plusieurs agents du fait de leurs actions combinées et réciproques.

Nous présentons ci-dessous, quelques formes et situation importants connues et exploitées pour l'interaction :

### **6.1.1 La coopération**

La coopération est nécessaire quand un agent ne peut pas atteindre ses buts sans l'aide des autres agents. Souvent les buts nécessitant la coopération sont des buts sociaux, ils assurent la survie du groupe ou de l'espèce. Quelquefois, ce sont des buts individuels, un agent qui en aide un autre peut attendre une aide en retour ou se faire payer son travail. Un agent peut avoir besoin d'un autre agent parce que cet agent a des compétences qu'ils n'a pas, ou parce qu'il faut être plusieurs pour réaliser la tâche.

### **6.1.2 la collaboration**

La collaboration, qui s'exprime par le travail collectif d'une équipe d'agents pour satisfaire un objectif global, nécessite la décomposition de la tâche globale en des sous tâche complémentaires, suivie par leur répartition et leur allocation aux agents, tout en considérant les compétences des différents agents et les ressources disponibles dans le système.

### **6.1.3 La coordination**

La coordination des agents dans un environnement où les ressources sont limitées, et où les objectifs locaux des différents agents peuvent se contredire provoquant l'apparition de situation de conflits qui influencent sur le rendement globale de tout le système, et ainsi diminuent les avantages de coopération. Le concept de coordination regroupe l'ensemble d'outils et méthodes qui peuvent être employés pour résoudre des conflits (dus aux ressources partagées limitées ou aux objectifs incompatibles) ou pour optimiser des comportements (éliminer des actions redondantes et inutiles) et plus généralement pour assurer un tout cohérent.

### **6.1.4 La négociation**

La négociation joue un rôle fondamental dans les activités de coopération. En général, une négociation intervient lorsque des agents interagissent pour prendre des décisions communes, alors qu'ils poursuivent des buts différents. Plus précisément, l'objectif de la négociation est de résoudre des conflits qui pourraient mettre en péril des comportements coopératifs. Verrons [Ver 05] définit la négociation comme étant un processus grâce auquel plusieurs parties aboutissent successivement à une décision

commune, après avoir diffusé initialement, leurs demandes respectives. Ces demandes initiales convergent donc petit à petit, vers un accord final, par une suite de concessions ou par la recherche de nouvelles alternatives [Fra 00].

## **6.2 La communication entre les agents**

La communication est l'un des concepts pertinents dans les systèmes multi-agents. On distingue essentiellement deux modèles de communication :

- Communication par partage d'information.
- Communication par envoi de messages.

### **6.2.1 Communication par partage d'information**

La communication entre les différents agents du système est réalisée par partage d'information lorsque ceux-ci disposent d'une zone de données commune dans laquelle ils rangent les conclusions qu'ils ont pu tirer. Outre ces résultats partiels, cette zone renferme les données du problème initial. Les agents peuvent ainsi y puiser les informations dont ils ont besoin pour résoudre une partie du problème globale. Ce type de communication correspond à ce que la littérature désigne communément sous le nom de modèle du blackboard (tableau noir).

En intelligence artificielle la technique du tableau noir est très utilisée pour spécifier une mémoire partagée par divers systèmes. Dans un SMA utilisant un tableau noir, les agents peuvent écrire des messages, insérer des résultats partiels de leurs calculs et obtenir de l'information.

### **6.2.2 Communication par envoi de messages**

Les systèmes multi-agents fondés sur la communication par envoi de messages se caractérisent par le fait que chaque agent possède une représentation propre et locale de l'environnement qui l'entoure. Chaque agent va alors interroger les autres agents sur cet environnement ou leur envoyer des informations sur sa propre perception des choses. La communication se fait soit en mode point à point, soit en mode par diffusion.

- **Mode point à point** : l'agent émetteur du message connaît et précise l'adresse de ou des agent(s) destinataire(s). ce type de communication est généralement le plus employé par les agents cognitif.
- **Mode par diffusion** : le message est envoyé à tous les agents du système. Ce type de transmission est très utilisé dans les systèmes dynamiques ainsi que les systèmes d'agent réactif. En fait, ceci suppose en général une messagerie : un agent spécialisé gère autant de files d'attente que de destinataire, chaque agent peut traite le premier message de sa file.

### 6.3 Les ACL (Agent Communication Language)

À base de la théorie des actes de langage, des langages pour la communication entre agents cognitifs artificiels ont vu le jour. L'idée était de créer un langage explicite pour une communication intentionnelle, comme le cas des langages naturels exploite dans les sociétés humaines. Pour cela, il fallait :

- **Définir un dictionnaire de vocabulaire commun** : ce vocabulaire sera utilisé pour exprimer le contenu propositionnel. Il doit assurer une interprétation commune de ce contenu.
- **Définir un ensemble d'acte de langage** : les actes manipulent le vocabulaire commun, et ils servent de brique de base pour la communication. Ces actes correspondent aux attitudes propositionnelles transmises entre agents.
- **Définir des protocoles de conversation (les règles de conversation)** : ces règles servent à structurer le dialogue entre agents.

L'intérêt des langages de communication est de faciliter l'échange et l'interprétation des messages et l'interopérabilité entre les agents. [Tar 06].

## 7. Les agents mobiles

### 7.1 Définition

Un agent statique est un agent qui s'exécute seulement dans les systèmes où il commence son exécution. Il utilise un mécanisme de communication tel que RPC. Par contre, un agent mobile n'est pas lié au système dans lequel il débute son exécution. L'agent mobile est capable de se déplacer d'un hôte à un autre hôte dans le réseau. Il

peut transporter son état et son code d'un environnement vers un autre dans le réseau où il poursuit son exécution [Far 03].

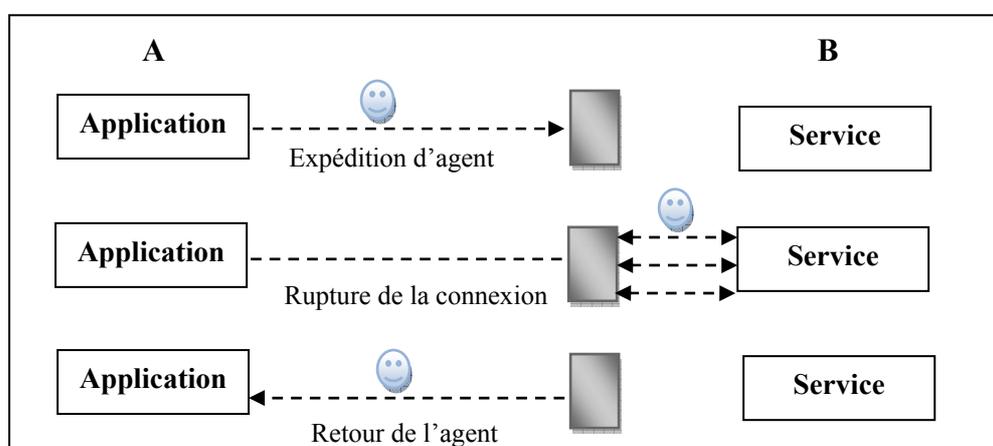
Selon [Che 99], un agent mobile est un programme informatique capable de s'exécuter sur un ordinateur distant, il peut se déplacer d'un site à un autre en cours ses données propres, mais aussi avec son état d'exécution.

Donc un agent est un processus, incluant du code et des données, pouvant se déplacer entre des machines pour réaliser une tâche et communiquer potentiellement avec d'autres agent.

## 7.2 Motivation

L'idée de la mobilité du code et des agents mobiles trouve plusieurs motivations dans les applications modernes de l'informatique. Nous décrivons ci-dessous certains de ces motivations :

- **Plus d'efficace:** au lieu d'avoir des interactions à distance avec un serveur (en générale coûteuse) c'est-à-dire appel de procédure à distance (RPC : Remote procedure call), le mieux est de déplacer le code vers le serveur et avoir des interactions locales moins coûteuses.
- **Robustesse et tolérance aux fautes:** lorsqu'un système/machine hôte est en difficulté (rupture de communication), les agents mobiles visiteurs prévenus ont la possibilité de se dispatcher ailleurs dans le réseau. En plus, l'exécution asynchrone et autonome des agents mobiles rend l'application tolérante en vers les ruptures possible entre hôte source et hôte destination.



**Figure 1.2:** Exécution asynchrone et autonome.

- **Réduction de la charge dans le réseau** : les agents mobiles réduisent le trafic du réseau, en déplaçant le traitement vers les données brutes plutôt que de déplacer les données vers le traitement.
- **Adaptation dynamique à l'environnement** : les agents mobiles sont autonomes, ils peuvent « découvrir » le réseau, c'est-à-dire qu'ils pourront visiter des hôtes qui leurs étaient inconnus. à chaque nouvel hôte parcouru, ils pourront prendre des décisions basées sur leurs propres connaissances ainsi que sur les connaissances acquises des hôtes visités. Des agents mobiles ont la propriété de se répartir uniformément entre les machines d'un réseau pour résoudre un problème complexe de manière optimale.
- **Une solution stratégique** : au lieu d'avoir trop de services sur un certain serveur dès le début, on peut adapter ces services selon les besoins spécifiques des clients.
- **Plates-formes hétérogènes** : l'informatique dans les réseaux à grand échelle, est fondamentalement hétérogènes (logiciel, matériel). Les agents procurent une configuration unique, dans laquelle les applications réparties peuvent être implémentées facilement et de manière efficiente. Les agents mobiles étant totalement indépendants de leur environnement, ils fournissent d'excellentes conditions d'intégration de systèmes à partir de leur contexte d'exécution.
- **Equilibrage de charge** : la mobilité d'agent lui permet de déplacer la charge de calcul d'un site à un autre, afin d'essayer d'équilibrer l'utilisation des ressources dédiées à l'application sur les différents sites participant.

### 7.3 Structure d'un agent mobile

Un agent mobile est une entité qui possède cinq attributs : son état, son implémentation, son interface, son identifiant et son autorité. Quand un agent se déplace à travers le réseau, il transporte ses attributs [Far 03] :

- **L'état** : l'état d'un agent peut être considéré comme une photo instantanée de son exécution. Quand un agent voyage, il transporte avec lui son état, ceci lui permet de reprendre son exécution quand il à arrive à destination.
- **L'implémentation** : comme n'importe quel autre programme, l'agent mobile a besoin d'un code pour pouvoir s'exécuter. Quand il se déplace à travers le réseau, l'agent peut soit emporter son code soit aller à destination, voir quel

code est disponible sur la machine distante et récupérer le code manquant à partir du réseau (c'est la technique du « code on demande »).

- **L'interface** : un agent fournit une interface qui permet aux autres agents et autres systèmes d'interagir avec lui. Cette interface peut être un ensemble de méthode qui permet aux autres agents et applications d'accéder aux méthodes de l'agent par un système de messagerie.
- **L'identifiant** : chaque agent possède un identifiant unique durant son cycle de vie, qui lui permet d'être identifié et localisé. Puisque l'identifiant est unique, il peut être utilisé comme clé dans les opérations qui exigent un moyen pour référencer une instance particulière d'agents.
- **L'autorité** : une autorité est une entité dont l'identité peut être authentifiée par n'importe quel système auquel elle essaye d'accéder. Une autorité peut être soit une personne privée, soit une organisation. L'identité est constituée d'un nom et d'autres attributs.

## 7.4 Migration d'un agent

Les migrations d'agents mobiles peuvent s'effectuer selon deux modes :

### 7.4.1 Migration forte

La migration forte, où la totalité de l'agent (c'est-à-dire code, données et unité d'exécution) migre vers le nouveau site. Pour cette migration réelle, l'agent est suspendu ou capturé avant d'être transféré. Une fois arrivé sur le site distant, il redémarre son exécution au point de contrôle précédent, en conservant l'état du processus. Une autre possibilité proposée consiste à stopper l'exécution de l'agent avant la migration puis d'en créer une copie distante identique sur le site distant (migration par réplication).

### 7.4.2 Migration faible

La migration faible ne fait que transférer avec l'agent son code et ses données. Sur le site de destination, l'agent redémarre son exécution depuis le début en appelant la méthode qui représente le point d'entrée de l'exécution de l'agent, et le contexte d'exécution de l'agent est réinitialisé.

## 7.5 Tolérance aux pannes

Le modèle d'exécution de l'agent mobile implique son interaction avec plusieurs sites ce qui expose l'agent à une éventualité de disparation à cause de la défaillance ou de la déconnexion soudaine et imprévue d'un site sur lequel il s'exécute. La disparition d'un agent entraîne un dysfonctionnement de l'application basée sur ce dernier. Une application distribuée sûre doit pouvoir continuer de fonctionner en cas de défaillance d'une partie du système. Pour certains types d'applications, il est essentiel que les environnements d'exécution d'agents mobiles offrent des mécanismes de tolérance aux fautes.

Dans une architecture de services, les défaillances de sites peuvent conduire à un comportement défaillant d'un service et donc le rendre inutilisable par le client. Plusieurs types de défaillances sont à considérer [Fal 06] :

- Une défaillance par arrêt (crash, panne) quand un serveur ne rend pas de résultats suite à des invocations répétées.
- Une défaillance par omission quand un serveur omet de répondre à son client.
- Une défaillance temporelle quand la réponse du serveur est fonctionnellement correcte mais n'est pas arrivée dans un intervalle de temps donné.
- Une défaillance de valeur quand le serveur rend des résultats incorrects.

Pour une application distribuée basée sur le concept d'agents mobiles, la migration des agents engendre d'autres types de défaillance :

- Un agent peut disparaître après avoir visité plusieurs sites. Si aucune précaution n'est prise, les résultats de son exécution sur ces sites peuvent être perdus.
- Il est important de détecter la disparition d'un agent pour en informer la place qui l'a lancé.

Pour faire face à la défaillance, les environnements d'agents mobiles offrent un mécanisme de point de reprise [Bre 98]. Les points de reprise sont conservés sur disque, support supposé fiable. Ils peuvent ainsi être utilisés ultérieurement pour restaurer l'agent en cas de défaillance. Cependant, lors de la restauration de l'agent il faut veiller à ne pas avoir deux agents actifs en même temps.

## 7.6 Plate-forme et Standardisation

La plupart des plates-formes pour agents mobiles ont introduit un ensemble de concepts similaires qui ont servi d'apport dans la standardisation des intergiciels agents.

Elles sont constituées de plusieurs entités. Nous pouvons distinguer [Cub 05] :

- **Les agents** ils sont définis par un identifiant unique au sein de la plate-forme et d'un état contenir plus d'un agent.
- **Les places** C'est l'élément qui sera en charge d'exécuter le code de l'agent. Une place peut contenir plus d'un agent. Une place définit un environnement d'exécution vers lequel un agent sera orienté pour gérer son exécution.
- **Les agences** Encore appelées «ports d'accueil » des agents, représentent les environnements où évoluent les agents. Une agence constitue le cœur de gestion de la plate-forme en assurant la délégation de l'exécution aux places qui la composent, l'administration, le contrôle, le transport effectif et la communication des agents grâce au système d'exploitation sous-jacent.
- **Les régions** Il s'agit du regroupement de plusieurs agences, pas nécessairement de même type, appartenant à un même domaine d'expertise, d'activité ou autre. Le but est de faciliter les activités d'administration. On peut associer par exemple une région à des politiques d'accès aux services.
- **Les services** On retrouvera un ensemble de services accessibles aux agents. Ils pourront être construits sur différents modèles, Client/serveur ou agents, et fourniront aux agents en visite les fonctionnalités recherchées.

## 8. Les normes

L'émergence des nombreuses plates-formes expérimentales a rendu nécessaire 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.

La technologie agent faisant l'objet de nombreux travaux, une intégration des concepts clés est apparue essentielle avec comme finalité la production de spécifications destinée à l'industrie. C'est l'objectif que s'est fixé la (Foundation for

Intelligent Physical Agents, FIPA) que de promouvoir le développement et la spécification de la technologie agent [Web 04]. De même (Object Management Group, OMG) a étendu le champ de ses travaux initialement fondé sur le paradigme objet pour intégrer la technologie agent mobile en vue de produire des spécifications, Depuis peu, les deux organismes font des efforts conjugués pour fusionner les deux approches.

On peut trouver à l'heure actuelle deux normes principales [Cub 05] :

- **FIPA (Fondation for intelligent physical Agents)**

En revanche, la communauté d'origine de FIPA étant celle des systèmes multi-agents, plus proche de l'intelligence artificielle, FIPA s'intéresse plus particulièrement à l'interopérabilité des agents intelligents (les efforts sont placés au niveau du langage, des protocoles et des infrastructures de communication), 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) et des protocoles de négociation permettant ainsi de définir parfaitement les interactions entre les agents.

- **MASIF (Mobile Agent System Interoperability Facility)**

La norme MASIF a été spécifiée par l'Object Management Group (OMG) qui se préoccupe généralement de l'hétérogénéité entre les systèmes, l'OMG s'intéresse à l'interopérabilité des agents mobiles à travers sa spécification appelée MAF. 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 [Far 03].

## **8.1 But de la liaison**

La raison de créer l'accord de liaison OMG-FIPA est d'encourager des normes de technologie d'agent à se développer compatible avec des normes de technologie d'objet et à la nouvelle coordination entre OMG et le travail lié du FIPA vers des normes de technologie d'agent [Fra 93].

## 9. Implémentations existantes

A l'heure actuelle, il existe beaucoup d'implémentations, incompatibles entre elles. Certaines de ces plates-formes existantes à nos jours sont présentées ci-dessous :

- **LIME** : le premier système que nous étudions est l'intergiciel basé sur Java nommé LIME (Linda In Mobile Environnement) qui propose une couche de coordination pour les agents en réutilisant le modèle Linda [Mur 01]. Ce n'est pas à proprement parler une plate-forme car elle ne prend pas en charge les différents éléments décrits par les normes. Elle se focalise plus sur la coordination qui peut être exploitée pour construire des applications réparties.
- **Aglets** : utilise la notion de référence distante (AgletProxy), le passage de messages (multicast, broadcast), développé par IBM [Dan 97]. Il offre en outre une certaine notion de sécurité en limitant les ressources allouées aux aglets.
- **JADE** : (java Agent Development Framework), est un intergiciel construit sur Java afin de permettre une programmation multi-agents simplifiée en prenant en compte la norme FIPA.
- **TACOMA** : le projet TACOMA (Troms And Cornell Moving Agents), de Norway & Cornell University, a été développé dans le but d'offrir différentes abstractions de haut niveau pour les agents. L'accent est mis sur un découplage, entre le langage et le niveau agent, semblable à celui introduit dans les normes à travers la notion de place [Joh 02].
- **PLANGENT** est une plate-forme basée sur le langage Java s'intéressant à l'adaptation des agents durant leurs déplacements au sein des environnements dynamiques [Ohs 97]. Son but est de proposer aux agents des mécanismes pour modifier les objectifs intermédiaires nécessaires à la réalisation de leur tâche finale.

## **10. 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 m-business 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 m-business.

# Chapitre II

---

## M-business : état de l'art

---

### 1. Introduction

Les technologies mobiles et **m-business (business mobile)** sont considérés comme la prochaine innovation potentielle pour les organismes. Ils fournissent une communication supplémentaire et le canal de la transaction, qui peut être appliquée afin d'améliorer les applications existantes d'e-business. La technologie mobile et internet peuvent se compléter et s'enrichir mutuellement. L'expressivité limitée de la technologie mobile peut être réduite par des applications internet complémentaires qui pourraient définir un contexte plus large pour les applications mobiles. En même temps un canal mobile supplémentaire pour des applications Internet existantes pourrait augmenter leur portée à une communication réelle (n'importe quand, n'importe où, n'importe qui et n'importe quoi). **Leung** et **Antypas** définissent business mobile comme «la diffusion de contenu (notification et de rapport) et les transactions (achat et la saisie des données) sur des appareils mobiles» [Leu 01].

Nous présentons dans ce chapitre la définition de business mobile, services dans business mobile et les avantages et améliorations de m-business par rapport à e-business. Puis nous présentons les technologies de communication et les dispositifs mobiles et les obstacles et les défis de m-business.

### 2. Les technologies mobiles

Sous le terme de technologies mobiles nous permettons de comprendre intuitivement que c'est la communication sans fil et des terminaux mobiles. Dans la littérature, dans de nombreux cas, sous le terme générique des technologies mobiles, différents types de technologies et d'applications sont discutées tel que par exemple les communications mobiles, les services de données mobiles, l'informatique ubiquitaire, etc. En général, tous ces termes renvoient à une caractéristique en commun « communication sans fil » [Kat 04].

Les composants de base des technologies mobiles sont les suivants: les technologies de transmission sans fil et les protocoles, tels que Bluetooth, UMTS, WLAN, et différents types des dispositifs spécifiquement conçus pour la communication sans fil. Ces appareils finaux sont par exemple des téléphones mobiles ou assistants numériques personnels (PDA) et dans le cas de communication de machine à machine, cela pourrait être n'importe quel objet équipé de puces spéciales permettant la communication sans fil. Des exemples bien connus de ces puces sont Remote Frequency Identification (RFID). Les étiquettes RFID sont de petites puces capables d'enregistrer et de stocker des données et de communiquer avec d'autres objets en réseau. Les choses sont équipées de RFID appelée choses intelligentes.

Les technologies mobiles permettent trois types de communication sans fil:

- **La communication person-to-person** : Qui pourrait être la voix mobile ou la transmission de données,
- **La communication person-to-machine** : qui se réfère à la communication sans fil de personnes, soit à des serveurs afin de récupérer et de stocker des données ou à d'autres choses intelligentes,
- **La communication machine-to-machine**, c'est-à-dire objet-à-objet qui est également connu sous le terme informatique ubiquitaire.

Par rapport à l'Internet, les technologies mobiles à l'étape actuelle de développement ont des inconvénients et des avantages. Les inconvénients résultent d'une part des petits écrans et des claviers de dispositifs mobiles ; d'autre part de leurs mémoires et la puissance de calcul limitée. Les caractéristiques des terminaux mobiles limitent la taille des informations qui peuvent être affichées sur l'écran du mobile, assurent la saisie des données difficiles et inconfortables, et ne permet pas de traitement compliqué du côté client.

Les inconvénients de la technologie mobile décrits ci-dessus sont compensés par les avantages suivants uniques par rapport à Internet câblé:

- **Ubiquité** : Dans un monde de mobilité personnelle croissante et la portabilité de la taille moyenne de l'information tient compte d'un échange permanent d'informations et des transactions à tout moment.

- **Identification** : Chaque personne ou chose capable de communication sans fil peuvent être identifiés basé sur l'enregistrement avec l'opérateur mobile.
- **Localisation** : Chaque appareil mobile ou un objet intelligent peut être localisé par les opérateurs mobiles ou en utilisant d'autres technologies de positionnement. Cette fonctionnalité offre des possibilités sans précédent pour des services innovants basés sur la localisation [Kat 04].

Un ordinateur personnel filaire est nécessaire pour accéder aux catalogues en ligne, alors que pour accéder aux catalogues condensés mobiles, un téléphone mobile est suffisant. Comparé à un ordinateur personnel, le téléphone mobile est moins cher et permet d'accéder de partout. La technologie mobile et Internet peuvent se compléter et s'enrichir mutuellement.

### **3. Business Mobile**

#### **3.1 Définition de Business Mobile**

Le business mobile est une activité mobile qui résulte de la capacité de l'individu à avoir un accès permanent à l'information et des services indépendants du lieu et de temps en utilisant un téléphone portable, assistants numériques personnels (PDA) ou PC de poche via une radio sans fil ou une connexion infrarouge. Il prend en charge l'échange de biens, services, informations et connaissances. Toutes les transactions sont effectuées via un périphérique de poche. Elles peuvent survenir *business-to-client (B2C)*, *business-to-business (B2B)*, *business-to-employé (B2E)*, *business-to-machine (B2M)* ainsi que de machine à machine (*M2M*) [Sas 03].

En plus de la transmission de la voix, l'amélioration des télécommunications mobiles et les technologies sans fil permettent le transfert de données et de contenu à des vitesses raisonnables sur les réseaux radio. Ainsi, il suggère la convergence des technologies Internet à la communication mobile.

De nombreuses entreprises ont déjà établi des applications e-business et le canal de communication mobile est ainsi considéré comme un canal supplémentaire qui doit être positionné par rapport aux canaux existants [Kem 02].

Les applications existantes dans l'e-business peuvent fournir un point de départ pour l'identification des composantes du m-business. Nous pouvons dire que m-business peut être considéré comme un canal supplémentaire pour l'e-business.

Toutefois, cela ne signifie pas que les approches pour le développement d'applications e-business peuvent être appliquées au développement d'applications m-business sans adaptation [Shi 02].

### **3.2 La chaîne de valeur de l'activité mobile**

La chaîne de valeur de l'activité mobile est composée des domaines d'activité suivants, *L'infrastructure* comprend tout ce qui est nécessaire pour faire des applications sur le travail de dispositifs mobiles y compris le réseau, les matériels, les logiciels, les plates-formes et les dispositifs etc. Les acteurs dans ce segment sont, par exemple, des producteurs de dispositifs mobiles comme Nokia, Siemens et Ericsson. Et pour *Contenu*, les fournisseurs se spécialisent dans la collecte et l'agrégation des informations qu'ils vendent directement ou via le portail au client mobile. Le contenu est rendu disponible grâce à des *applications* différentes qui sont fournis par **wireless application service providers (WASP)**. La dernière liaison dans la chaîne de valeur est les *portails*. Ils agrègent des applications comme le courrier électronique, des calendriers, la messagerie instantanée aussi bien que le contenu pour leurs abonnés. Dans le business mobile, ces services sont supportés par accrue des offres à base d'emplacement [Sas 03].

### **3.3 Services dans Business Mobile**

Selon le type de relation que l'entreprise entretient avec leurs partenaires, on distingue trois types de M-business :

#### **3.3.1 Business-to-Business.**

Noté B2B, des solutions d'entreprise dans business mobile favoriseront l'efficacité de processus commerciaux dans beaucoup d'aspects, B2B considèrerait la communication entre les entreprises avec d'autres entreprises.

#### **3.3.2 Business-to-Employée**

Noté B2E, D'autres solutions d'entreprise considèrent la communication entre les entreprises et leurs employés. Les entreprises peuvent faire leur intranet accessible

à tous leurs agents de terrain. Cela comprend tous les outils d'administration tels que la gestion voyage, carnets d'adresses, ... etc. En outre, toutes les applications d'entreprise sont disponibles sur les appareils mobiles afin que les calendriers peuvent être synchronisés, contrôler les courriels ... etc.

Avec l'utilisation d'un appareil mobile, de toutes les informations seront disponibles à tout moment et peuvent communiquer avec d'autres appareils via Bluetooth ou WLAN.

### **3.3.3 Business-to-Consumer**

Noté B2C, Outre l'intégration de services de plus en plus hors ligne de typiques industries comme banques, prestataires de services financiers, agences d'assurance, tourisme, divertissement, automobile, commerce et fabricants de biens de consommation. Après avoir identifié l'emplacement d'un dispositif mobile, il est possible de fournir à l'utilisateur des informations appropriées selon l'endroit et le temps.

### **3.3.4 Machine-to-Machine**

Utiliser les réseaux filaires, la télétransmission des données de surveillance, prise de décision et le contrôle des machines, dénommée télémessure, a déjà produit une large gamme d'application. Par le déploiement de réseaux 3G, même les dispositifs qui ont été trop onéreux câbler peut être fournis avec la technologie mobile. Distributeurs automatiques peuvent appeler ayant des besoins spécifiques de rechange ou de sociétés de services publics peuvent lire leurs compteurs automatiquement.

## **4. Business électronique**

E-business est un modèle d'affaires de l'achat et la vente de marchandises et de services via internet. Ce modèle comporte généralement des activités économiques, les transactions commerciales traversant des entreprises et les interactions entre consommateurs et producteurs. La plupart des applications d'e-business utilisent le modèle traditionnel client/serveur dans lequel une opération commerciale exige généralement un lien de communication stable étant établi entre le client et le serveur.

#### 4.1 Modèle client/serveur

Dans le modèle client/serveur, un serveur représente un objet géré sur un site et offrant des services aux clients. Les fonctionnalités d'une application sont alors encapsulées dans des services et exécutées au sein de serveurs. Les serveurs sont ainsi vus comme des fournisseurs de services. On trouve soit des serveurs de données, dans lesquels des clients accèdent à des données localisées sur chaque serveur, soit des serveurs de calculs, dans lesquels des clients utilisent les ressources de chaque serveur afin d'exécuter une tâche précise.

Le modèle client/serveur est une des modalités des architectures informatiques distribuées. Au sein de cette architecture, les processus sont classés entre offreurs de services (serveurs) et consommateurs de services (clients). Le terme **serveur** s'applique à tout programme qui offre un service que l'on peut atteindre à travers un réseau. Le serveur accepte des demandes issues du réseau, les traite et renvoie le résultat au demandeur. Quant au terme client, il s'applique à tout programme qui émet une demande à un serveur et qui attend une réponse.

#### 4.2 Les inconvénients du modèle Client/Serveur

Le modèle client/serveur présente des inconvénients que nous avons résumés dans des points suivants :

- Une surcharge au niveau du serveur : tout est concentré à son niveau ; si trop de clients veulent communiquer avec le serveur en même temps, ce dernier risque de ne pas supporter la charge.
- Sensibilité aux pannes réseau : le risque d'indispensabilité de tout le système si le serveur tombe en panne ; s'il n'est plus disponible, plus aucun des clients ne marche.
- Les détails et tarifs de livraison.
- Les coûts de téléphone.
- Les difficultés de recours en cas d'ennuis.
- Exiger une connexion permanente entre le client et le serveur
- Perte de temps dans la recherche.
- Le consommateur faisant des affaires avec restriction de temps et d'espace.
- Coût d'exploitation et de maintenance des applications élevé.
- Augmentation du trafic réseau

- Mode connecté (synchrone) : le client émet une requête et se bloque jusqu'à l'arrivée de la réponse.

Grâce à la flexibilité et la mobilité des agents mobiles, la technologie d'agents mobiles peut pallier les problèmes du modèle client /serveur pour permettre aux consommateurs de faire des affaires sans restrictions de temps et d'espace.

## **5. Avantages et améliorations de M-business par rapport à E-business**

Les applications de M-business sont différentes des applications e-business dans de nombreux aspects. Développement d'applications de m-business nécessite donc un ensemble différent des stratégies, outils et techniques. La plupart des applications e-business ne peuvent être facilement transférées aux plates-formes mobiles. Quand une entreprise cherche à reproduire leur succès de l'e-business, simplement en déplaçant ses applications Web dans un environnement sans fil. Pour réussir à déplacer une application Web pour un environnement sans fil, le champ d'application doit être redéfini, l'interface utilisateur doit être repensée et son réseau et les exigences de traitement doivent être réévalués. Pour ce faire, nous devons commencer par comprendre les différences entre les applications e-business et m-business [Gor 05].

**Singhal et al** ont remarqué que les applications m-business diffèrent à partir d'applications e-business dans trois domaines: dispositif, le réseau et l'utilisateur. Le tableau 2.1 résume certaines de ces différences [Sin 01]. Les différences exigent que des développeurs pensent avec créativité sur la façon d'offrir les avantages des applications m-business pour les utilisateurs tout en répondant aux contraintes des dispositifs de client et des réseaux sans fil. Dans le même temps, nous devons également reconnaître les possibilités uniques des applications m-business actuel. Par exemple, la nature personnelle de dispositifs de client (par exemple, les PDA) permet aux applications d'être plus personnalisées et ciblées; les technologies d'identification d'emplacement ont créé beaucoup d'occasions nouvelles et passionnantes de fournir des produits à base du contexte et des services; les réseaux sans fil permettent de diffuser de l'information en temps opportun aux utilisateurs n'importe quand et n'importe où.

	<b>Applications E-Business</b>	<b>Applications M-Business</b>
<b>Dispositif</b>	Résolutions d'écran allant de 640x480 à 1600x1200.	Les tailles d'écran allant de 4 lignes de texte et 12 caractères par ligne à 120x160 et affiche 11 lignes de texte.
	Moniteur affiche des centaines de millions de couleurs.	Très grande majorité des écrans de portable sont en niveaux de gris ou soutenir un nombre limité de couleurs.
	Clavier et souris sont les périphériques d'entrée principale.	Les méthodes d'entrée (clavier et stylet) sont laborieuses à utiliser.
	Machines clientes ont processeur capable et la mémoire considérable.	Machines clientes ont peu de puissance de traitement et de mémoire.
<b>Réseau</b>	Gammes de vitesses du réseau de 28,8 KB par seconde à plus de 1 Mo par seconde.	Des vitesses de données allant de moins de 100 bits par seconde à 28,8 Ko par seconde.
	Les réseaux sont stables.	Plus de latence, moins de stabilité de connexion.
<b>Utilisateur</b>	L'utilisateur est devrait être un expert en informatique.	L'utilisateur ne doit pas nécessairement être un expert en informatique.
	L'utilisateur peut passer de longues périodes de temps « naviguer sur Internet ».	L'utilisateur s'attend à terminer la tâche en quelques minutes.
		L'utilisateur a tendance à être engagé dans d'autres activités en même temps.

**Tableau 2.1** : Les différences entre les applications e-business et les applications m-business.

Dans les paragraphes suivants, nous offrons de courtes descriptions de chacun des expériences variables qui sont susceptibles de définir m-business indépendamment d'e-business [Web 05]:

- **Services multi transaction :** services de M-business pourraient être prévus et livrés de multiples façons. Les utilisateurs peuvent choisir d'avoir une variété de services au moment et les lieux qu'ils spécifient. Dans certains cas, le réseau et le dispositif peuvent faire des évaluations intelligentes de ce que les services sont nécessaires, puis offrir de tels services.
- **Intégration avec les applications d'entreprise :** Avec m-business, une entreprise pourrait passer la plupart de ses capacités sur le terrain. Services et les applications nécessitant des réunions et des consultations de bureau pouvaient maintenant être livrées en déplacement, avec un accès complet à toutes les applications d'entreprise résidant sur l'entreprise.
- **Positionnement géographique :** réseaux m-business serait en mesure de localiser l'utilisateur et d'adapter la gamme des services à la localisation géographique, en tenant compte contrainte et des possibilités de la situation géographique ainsi que les préférences de l'utilisateur.
- **Intégration avec les services mobiles :** Nouveaux services m-business serait facile à intégrer avec préexistants services mobiles. Par exemple, les offres de m-business pourraient facilement intégrer une variété de services de messagerie existants, service de messages courts (SMS), et e-mail... .

## **6. Les technologies de communication**

La technologie sans fil désigne le matériel et le logiciel qui permet la transmission d'informations entre les périphériques sans utiliser les connexions physiques. La livraison des applications de m-business s'appuie sur la fiabilité et la disponibilité des technologies de réseau sans fil. Solutions de réseau sans fil d'aujourd'hui comprennent les réseaux personnels sans fil (**WPAN**), les réseaux locaux sans fil (**WLAN**) et les larges réseaux sans fil (**WWAN**). Tableau 2.2 résumés les technologies attendant pour différents types de réseaux sans fil [Gor 05].

Réseaux sans fil	Technologies Attendant
WPANs	Line-of-sight infrared, radio frequency (RF), et Bluetooth.
WLANs	Spread spectrum technology (802.11b), Orthogonal Frequency Division Multiplexing (802.11a and 802.11g), Infrared, et narrowband technology.
WWANs	Analog cellular networks, digital cellular systems et Personal Communications System (PCS), Cellular Digital Packet Data (CDPD), Code Division Multiple Access (CDMA), Time Division Multiple Access (TDMA), Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS), (EDGE), CDMA2000, W-CDMA, et satellites.

**Tableau 2.2 :** Réseaux sans fil et leurs technologies.

Dans des zones personnelles (bureau et automobile), une des nouvelles technologies les plus prometteuses de cette région est Bluetooth.

### 1) Wireless Personal Area Network

#### ➤ Bluetooth

C'est un standard mondial pour la connectivité sans fil rapidement adopté par de nombreux fabricants d'appareils. Il utilise la technologie de radio courte portée pour connecter des petits périphériques, tels que les ordinateurs portables, PDA, souris et autres périphériques. La technologie a le potentiel pour remplacer les câbles qui relient les appareils numériques et offre aux utilisateurs un niveau plus élevé de la liberté dans leurs domaines personnels. La technologie Bluetooth a été particulièrement conçue pour faire des interactions ad hoc entre des dispositifs différents dans un environnement informatique hétérogène. Sa vitesse de transfert de données relativement élevé et extrêmement faible consommation d'énergie en ont fait la technologie idéale pour bon nombre de situations commerciales.

### 2) Wireless Local Area Network (WLAN)

De nombreuses organisations ont adopté la technologie **Wi-Fi** pour fournir l'accès sans fil aux utilisateurs dans une zone géographique locale (par exemple un bâtiment, campus, aéroport, café et hôtel). Technologies de réseau local sans fil

permettent des deux communications de pair-à-pair (peer-to-peer) entre les périphériques et point-multipoints (point-to-multipoint) communications par le biais de points d'accès qui couvrent un rayon de 50 à 100 mètres. La norme Wi-Fi la plus largement adoptée est aujourd'hui IEEE802.11b ; elle transmet des données à spectre 2,4 GHz à une vitesse de 11 Mbit/S à l'aide de l'étalement de spectre à séquence directe (DSSS). Récemment, IEEE802.11a et IEEE802.11g sont devenus disponibles sur le marché. Contrairement à IEEE802.11b, les IEEE802.11a et IEEE802.11g utilisant *Orthogonal Frequency Division Multiplexing (OFDM)* et transmettant des données à la vitesse de 54 Mbit/S.

La technologie de réseau sans fil d'aujourd'hui a encore ses faiblesses. Une enquête récente par **Businessweek** a révélé que les défis rencontrés par la technologie Wi-Fi ont inclus des normes peu claires, la sécurité irrégulière, la gamme limitée, des coûts cachés et le manque d'interopérabilité. [Gre 03].

Il existe plusieurs technologies concurrentes :

- **Le Wifi** (ou **IEEE 802.11**), soutenu par l'alliance WECA (Wireless Ethernet Compatibility Alliance) offre des débits allant jusqu'à 54 Mbp/s sur une distance de plusieurs centaines de mètres.

Grâce au Wi-Fi, il est possible de créer des réseaux locaux sans fils à haut débit pour peu que l'ordinateur à connecter ne soit pas trop distant par rapport au point d'accès. Dans la pratique, le WiFi permet de relier des ordinateurs portables, des ordinateurs de bureau, des assistants personnels (PDA) ou tout type de périphérique à une liaison haut débit (11 Mbp/s ou supérieur) sur un rayon de plusieurs dizaines de mètres en intérieur (généralement entre une vingtaine et une cinquantaine de mètres) à plusieurs centaines de mètres en environnement ouvert.

- **hiperLAN2** (*High Performance Radio LAN 2.0*), norme européenne élaborée par l'ETSI (*European Telecommunications Standards Institute*). HiperLAN 2 permet d'obtenir un débit théorique de 54 Mbp/s sur une zone d'une centaine de mètres dans la gamme de fréquence comprise entre 5 150 et 5 300 MHz.

### 3) Wireless Local Area Network (WWAN)

Réseaux étendus sans fil (WWAN) permettent aux utilisateurs de communiquer et accéder à des ressources dans une zone géographique large comme une ville, une région, un pays entier ou même dans le monde entier. Cela est obtenu par l'utilisation de satellites et de réseaux cellulaires. Tout au long de l'histoire des communications sans fil, plusieurs générations de réseau cellulaire et de technologies ont été développées, et différentes régions ont adopté des normes de technologies différentes et parfois incompatibles. Alors que WWAN une offre liberté sans précédent aux utilisateurs et offre de nombreuses possibilités de m-business, dans de nombreuses régions, l'absence de réseaux cellulaires de nouvelle génération (2 ou 3G) avec des hauts taux de transmission de données ont empêché la croissance de m-business. Les technologies WWAN peuvent être classées par génération. Premières générations de réseaux cellulaires ont été conçues principalement pour les communications vocales à l'aide de commutation de circuit. Créées en 1978, les premières générations réseaux sans fil cellulaire (**1G**) étaient des réseaux analogiques appelés des systèmes avancés de téléphone Mobile (**AMPS**). Par rapport aux nouvelles générations de technologies cellulaires, réseaux (**AMPS**) étaient inefficaces dans l'utilisation du spectre sans fil limité et pris en charge le cryptage limité et les services avancés. Ces limitations ont incité le développement de la deuxième génération (**2G**) des réseaux sans fil cellulaire [Agr 03].

Les technologies de base 2G comprennent Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), et Global System for Mobile Communications (GSM). Ces technologies sont aujourd'hui largement utilisées dans le monde. GSM a été adopté par les pays notamment asiatiques et européens, alors que CDMA, le bloc de construction pour les 2,5G et de la troisième génération (3G), a été adopté aux États-Unis. Même si une amélioration énorme sur les technologies de 1G, 2G réseaux cellulaires sont centrés sur la voix et commutation de circuits. Le taux de transmission de données des réseaux 2G était limité à 14,4 Kbits/s.

L'augmentation de la nécessité d'une communication de données sans fil, nouvelles générations de technologies cellulaires ont été conçues pour rendre la communication de données plus efficace. Technologies 2.5G, tels que General Packet Radio Service (GPRS), CDMA 2000 1x, et Enhanced Data Rates pour l'évolution du

GSM (EDGE) offrent des taux de transmission de données de jusqu'au 384 kbps [Sas 03].

En plus des réseaux cellulaires terrestres, les satellites fournissent également un soutien pour WWAN. Ils sont particulièrement utiles dans les régions éloignées où les services cellulaires sont rares. Les satellites sont également des éléments essentiels du système de positionnement global (GPS). Largement utilisé dans les appareils commerciaux aujourd'hui, GPS utilise des satellites pour suivre la latitude, longitude et altitude d'une personne ou un objet à l'aide d'une technique appelée 'triangulation'. Il est extrêmement précis, mais cher à exploiter et ne fonctionne pas bien à l'intérieur. GPS et *Enhancer Observes Time Difference* (E-ODD) ces nouvelles technologies permettent aux opérateurs de déterminer la position exacte d'un appareil mobile [Gor 05].

## **7. Les services associés aux réseaux mobiles**

### **7.1 Le protocole WAP (Wireless Application Protocol)**

C'est un protocole de communication qui permet d'accéder à internet à partir d'un appareil de transmission sans fil, comme par exemple un téléphone portable ou un assistant personnel. Le WAP redéfinit le protocole HTTP et aussi la manière dont les documents doivent être structurés, grâce à un langage dérivant du HTML et nommé pour l'occasion WML (*Wireless Markup Language*) et un langage de script baptisé *WMLScript* pour le monde des périphériques ayant un écran de taille réduite, un processeur de faible puissance et une autonomie limitée (téléphones mobiles). Le Wireless Mark-up Language (WML) est un langage à balises conçu spécifiquement pour le WAP, de manière à pouvoir s'afficher sur un écran de téléphone portable. Il est basé sur XML. Sa syntaxe est proche de HTML. WML est doté de son propre format d'image, appelé *Wireless Bitmap* (WBMP), dérivé du format BMP, mais en noir et blanc [Oma 02].

#### **7.1.1 Services WAP**

Il existe trois types de services WAP : services de communication, services d'information et services m-commerce.

- **Les services de communication** regroupent tous les services que l'on pourrait qualifier de base tels que service e-mail, service d'annuaire, service d'agenda personnalisé et service de forums et d'achat.
- **Les services d'information** peuvent être déclenchés par l'utilisateur mobile qui demande une information particulière ou à l'initiative du fournisseur de services qui envoie périodiquement ou selon des critères prédéfinis des informations à l'abonné. Dans ce dernier cas, l'utilisateur devrait au préalable être abonné à ce service. Les services d'information sont nombreux : météo, actualités, sports, horaires de train, résultats de jeux, etc.
- **Les services de m-commerce** correspondent aux services de réservation en ligne. Parmi ces services figurent les commandes par correspondance (livres, CDs, etc.) et les services bancaires [Web 01].

## 7.2 Le service i-Mode

C'est l'appellation commerciale d'un ensemble de services et de protocoles permettant de connecter des téléphones portables à internet. L'i-mode est un concurrent du WAP, bien que les téléphones i-mode permettent également d'afficher du contenu WAP. Le nom i-mode est une marque déposée par NTT DoCoMo et a été inventé par Mari Matsunaga. On peut citer Takeshi Natsuno comme acteur principal de la création et du développement d'i-mode au Japon.

Les deux fonctions principales de l'i-mode sont la consultation de services et la messagerie électronique. La consultation de services se fait sur des sites Web adaptés à la taille d'un écran de téléphone mobile. Ces sites peuvent être scindés en deux groupes : les sites payants et les sites gratuits. L'i-mode permet d'envoyer des courriers électroniques à d'autres abonnés à l'i-mode, mais aussi sur les boîtes électroniques d'autres utilisateurs. Il permet aussi de recevoir des courriers électroniques tant des autres utilisateurs qu'à partir d'internet. Néanmoins, la taille des messages est limitée, et les pièces jointes ne sont pas toujours gérées par les premières générations de terminaux. On peut aussi envoyer des images, des sons, et des vidéos grâce à ce service de messagerie [Web 03].

## 8. Les dispositifs Mobiles

Les dispositifs mobiles deviennent plus intelligents et intègrent plusieurs fonctions dans un seul dispositif. Ces dispositifs mobiles, qui permettent une connexion sans fil à l'internet, sont de plus en plus utilisés pour retrouver et envoyer des informations. Cette évolution renforcera le développement de l'internet participatif et favorisera l'apparition de services fondés sur la localisation. La plupart des téléphones de nos jours peuvent être utilisés pour accéder au Web et consulter des e-mails. Il est probable que les téléphones mobiles vont devenir le lien de l'utilisateur pour toutes sortes d'information mobile et les transactions.

### 8.1 Les assistants numériques personnels (PDA)

C'est un ordinateur de poche, servant de complément à l'ordinateur de bureau ou à l'ordinateur portable. Il intègre de multiples fonctions de gestion qui lui permettent d'être utilisé comme un outil de travail accompagnant une personne dans ses déplacements. Aujourd'hui de nombreuses marques proposent des assistants personnels très sophistiqués, pouvant se connecter à internet sans fil et utilisant un navigateur WAP : *Compaq, Hewlett-Packard, Toshiba, Sagem, etc.*



**Figure 2.1:** Assistants numériques personnels.

### 8.2 Pagers (Téléavertisseurs)

Ils ont été l'un des premiers dispositifs de communications sans fil. Ils sont conçus pour avertir les utilisateurs lorsque des messages textes ou vocaux arrivent. Similaires à des téléphones mobiles, ils varient en taille, forme, couleur, et en fonctionnalité. L'évolution la plus récente dans les deux sens de communication

interactive permettant aux utilisateurs d'envoyer et de recevoir des messages. Les **Smartphones** par exemple, combinent déjà un téléphone portable et un ordinateur.

## **9. Déterminants de la réussite de Business Mobile**

### **9.1 Conditions préalables nécessaires**

- **Fiabilité** : Pour la transmission des données, les connexions doivent être stables.
- **Facturation** : Systèmes de tarification doivent être adaptés aux nouvelles habitudes de communication causées par la transmission de données.
- **Praticabilité des applications** : Les applications professionnelles mobiles doivent être faciles à utiliser et intuitifs pour garder leur avantage de gagner du temps. La taille d'affichage limitée et l'activité réduite de surf doivent être prises en compte.
- **Matériel fonctionnel** : Le succès des services de business mobiles dépend de la disponibilité des appareils compatibles offrant à chaque utilisateur une solution de communication adéquate à ses besoins, la bonne qualité d'affichage, la portabilité et un nombre adéquat de fonctions.
- **Sécurité** : Un problème majeur est la sécurité des données transmises. Normes de sécurité doivent être développées sur plusieurs niveaux, (dispositif, réseau mobile, passerelle, et sécurité du serveur). Les fabricants des dispositifs doivent construire dans des normes de sécurité et les opérateurs doivent entretenir les caractéristiques de sécurité correctement. Pour tenir compte de la fiabilité de fonctionnement d'une meilleure sécurisation des systèmes de paiement mobile.
- **Partenariats** L'interdépendance des acteurs de l'entreprise mobile afin d'assurer un système d'information mobile complet permettant les services prévus, sera les obliger à élaborer des modèles d'affaires où la coopération et le partage des revenus sont impératifs.

## **10. Obstacles et les défis de M-business**

Les entreprises qui ne considèrent pas la mobilisation sera un sérieux désavantage dans l'environnement actuel (les affaires en évolution rapide). Néanmoins, les nombreux obstacles qui nuisent à l'adoption de m-business existent

toujours. Ces obstacles comprennent principalement des frais élevés d'accès sans fil à internet, les préoccupations au sujet de la protection des renseignements personnels et de sécurité, limitations de l'appareil et le manque de normes mondiales pour la communication sans fil [Gor 05].

M-business doit résoudre un grand nombre des défis qui nous attendent. Afin de mieux illustrer ces défis, Tarasewich, Nickerson, et War Kentin identifié trois dimensions de problèmes de m-business: techniques, d'application et les enjeux mondiaux. Dans l'ordre pour m-business continuait à augmenter, questions *techniques* telles que les limitations de l'appareil, convivialité, normalisation et l'intégration des différentes technologies sans fil doivent être abordées. Le champ de m-business est régi par un ensemble chaotique des plates-formes, des dispositifs et des normes. Chaque classe de périphérique diffère considérablement des autres classes en termes de fonctions de bouton et de résolution d'écran [Tar 02].

Développement d'applications d'affaires pour tel un groupe diversifié de machines clientes s'avère extrêmement difficile. En outre, les petits écrans des appareils mobiles et leurs capacités d'entrée limitées ont créé de nombreux défis de conception. La miniaturisation des dispositifs mobiles a entraîné de nombreux périphériques avec écrans qui sont trop petits pour afficher toutes les données utiles. Les capacités d'entrée de données limitées de dispositifs mobiles font aussi les applications m-business maladroit et difficile à utiliser. Parmi les questions liées à *l'application* sont : maintenir l'intégrité des données, assurer la sécurité des données. Alors que la plupart des menaces de sécurité internet et de la vie privée envahissent les applications m-business, les applications m-business introduisent de nouveaux risques en raison de son support de mobilité et de la communication.

La construction d'une infrastructure viable répondant à toutes prévisions se révèle difficile en raison de sa grande complexité. En outre, les opérateurs sont confrontés aux effets externes tels que des investissements élevés, une réglementation stricte, et les questions administratives qui entravent le processus de mise en place de l'infrastructure nécessaire [Sas 03].

## **11. Conclusion**

Dans ce chapitre nous avons vu que le m-business s'intéresse à l'amélioration de la performance des processus clés de l'entreprise. Une compréhension profonde des implications d'affaires, les applications et les technologies de m-business contribueront à une organisation restée compétitive dans l'avenir.

Dans ce chapitre, nous avons présenté la définition de business mobile, services dans business mobile et les avantages et améliorations de m-business par rapport à e-business. Puis nous avons présentons les technologies de communication et les dispositifs mobiles, les obstacles et les défis de m-business.

La communication par agents mobiles permet un bon fonctionnement dans le cas où la communication n'est pas établie en permanence ; ainsi, le client se connecte pour créer un agent mobile, l'agent créé joue alors le rôle du client et lui transmettra en retour les résultats lors d'une prochaine connexion. La mobilité permet d'augmenter la fiabilité et le bon fonctionnement des applications m-business. Si une machine doit s'arrêter, l'agent qui s'y trouve peut changer de machine pour terminer la tâche en cours. En conclusion, le paradigme agent mobile est innovant et concerne principalement les applications répartie et mobile afin de remplacer efficacement les paradigmes classiques. Donc la technologie d'agent mobile joue un rôle important dans la conception des applications m-business.

Dans le prochain chapitre, nous présentons quelques architectures, qui sont conçues pour les affaires mobiles en utilisant l'approche agent mobile.

# Chapitre III

---

## Approches Existantes à Base D'agent Mobile

### Pour M-business

---

#### 1. Introduction

Avec les progrès dans les capacités de calcul et de communication des appareils portatifs tels que les PDA, Pocket PC, ordinateurs portables et d'autres, nous commençons à voir l'émergence d'une variété d'applications dans m-business. Ces appareils mobiles deviennent aussi puissants que les ordinateurs portables et PC de bureau, incitant de plus en plus des consommateurs et des entreprises à les utiliser dans leurs affaires quotidiennes, ainsi que dans des activités récréatives. Nous considérons qu'e business mobile est une occasion d'affaires majeures dans les prochaines années et ce en raison de la flexibilité et de coût de performance des compromis.

Les entreprises prennent conscience du potentiel d'une telle connectivité et des capacités pour permettre à un éventail d'activités pour leurs employés mobiles. De plus, beaucoup d'entreprises (cafés, des aéroports,...) commencent à capitaliser et à déployer des réseaux sans fil au voyageur d'affaires pour le voyageur des affaires nécessaires.

Dans ce qui suit, nous allons présenter quelques architectures, qui sont conçues pour les affaires mobiles en utilisant l'approche agent mobile.

## **2. La technologie agent mobile et Internet**

Des transactions d'affaires exigent l'activité humaine, comme la collection de l'information, l'analyse et l'interaction humaine, comme la négociation. Avec la croissance rapide de la technologie Internet tout au long de ces dernières années, M-business a commencé à transformer la façon dont nous menons nos affaires. Les concepts comme des chariots d'e-achats et des e-magasins nous aident à mettre en œuvre des affaires aux entreprises de consommateurs. Ces modèles de travail exigent souvent un acheteur à visiter les sites Web de fournisseurs. Les données recueillies sont analysées, et une transaction terminée, souvent sans négociation bien interactive. La technologie agent mobile existe maintenant pour fournir une autre manière de mettre en œuvre d'e-business. La nouvelle technologie permet l'automatisation et la négociation, et reçoit beaucoup d'attention des chercheurs et des industriels.

## **3. Les travaux connexes**

Dans ce contexte, de nombreux travaux ont été élaborés afin d'introduire la technologie multi-agent et en particulier la technologie d'agent mobile et les concepts liés à cette dernière, dans le cadre des affaires mobile, nous allons présenter quelques architectures :

### **3.1 Présentation de Zhiyong Weng et Thomas Tran**

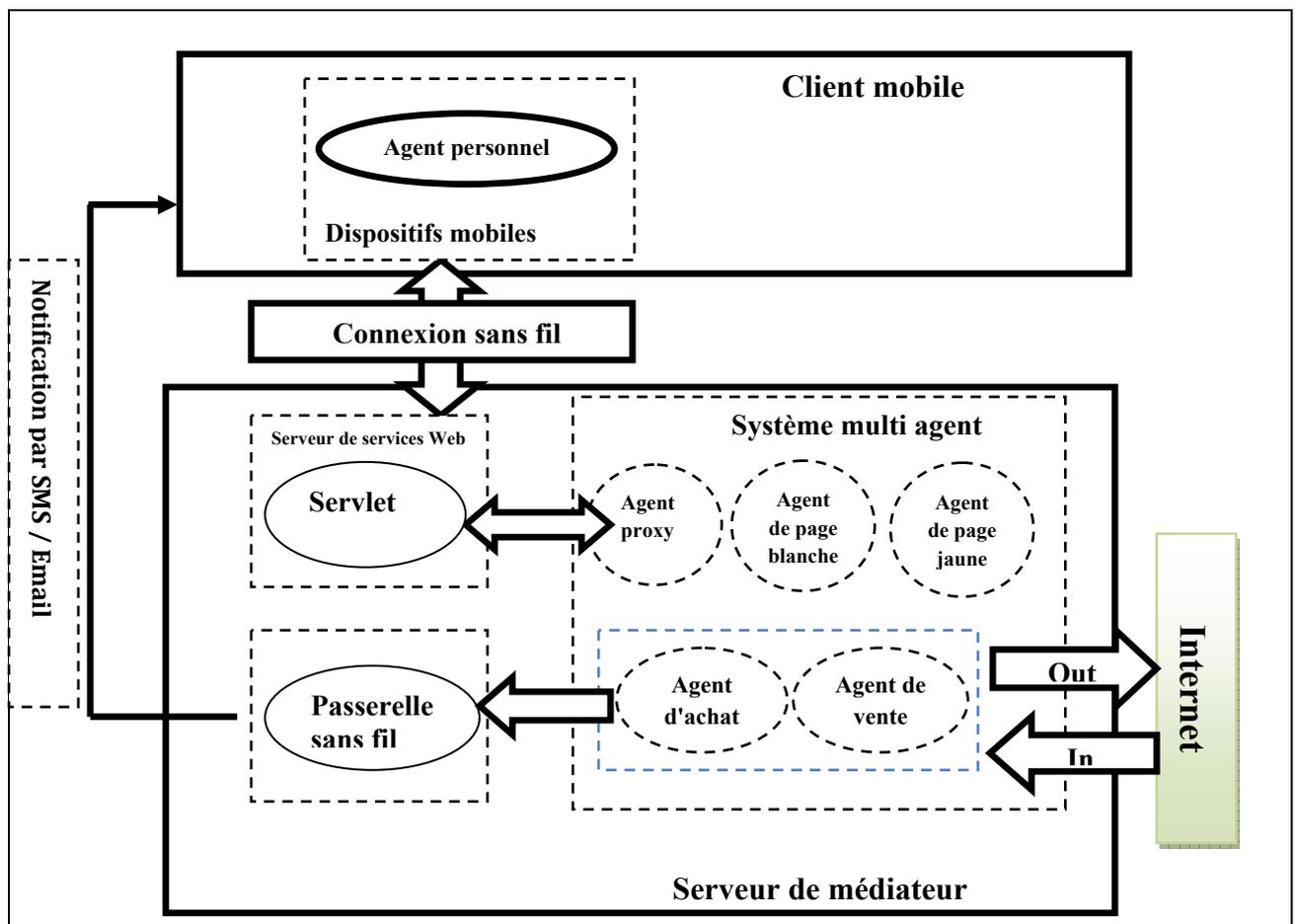
Cette approche a été proposée par **Zhiyong Weng et Thomas Tran** [Zhi 07] à la sixième Conférence internationale sur la gestion des affaires mobiles (ICMB 2007).

Dans cette structure, **Zhiyong Weng et Thomas Tran** présentent un environnement d'agent mobile pour le service de négociation des produits de seconde main, considèrent une situation classique dans laquelle un agent des ventes offre un article unique pour le meilleur enchérisseur.

Dans cette approche, un consommateur peut connecter son appareil mobile au serveur de médiateur via une connexion sans fil et ensuite envoi une demande de création d'agent mobile de vente ou d'achat pour l'offre de vente aux enchères en son nom. Le serveur de médiateur contient deux composants principaux : le serveur de services Web, qui facilite l'utilisation des agents mobiles a interagi avec d'autres

agents et le système de multi-agent, qui gère les agents et joue le rôle d'un marché semblable à un serveur à base d'agent.

La figure 3.1 illustre la structure proposée et le processus d'opération. Comme illustrée dans la figure 3.1, un Servlet répond à n'importe quelles demandes (requêtes) de l'agent personnel et est lié avec le comportement d'un agent proxy étant responsable du traitement des demandes. L'agent proxy interagit avec le servlet et construit un pont entre le serveur du service web et le système multi-agents.



**Figure 3.1:** Structure et processus du système proposé par Zhiyong Weng et al.

Zhillong Weng et Thomas Tran présentent le processus comment d'un agent mobile de vente ou d'achat sont lancés selon les préférences de l'utilisateur comme suit :

Dans la première étape, l'utilisateur configure ses préférences via un agent personnel. L'agent personnel envoie ensuite une demande au serveur de médiateur, en deuxième étape, le serveur web accepte la demande et communique avec l'agent proxy, ensuite l'agent proxy coopère avec l'agent de page blanche qui se trouve dans

le système multi-agent pour créer un agent mobile de vente ou d'achat. Dans la troisième étape, si l'agent mobile est créé avec succès, il peut se déplacer à d'autres serveurs pour entreprendre la tâche de l'utilisateur.

### 3.1.1 Les types d'agents

Zhillong Weng et Thomas Tran utilisent les types des agents suivants :

- **Un agent personnel** est un agent stationnaire qui s'exécute sur un périphérique mobile de l'utilisateur et fournit une interface graphique pour l'utilisateur.
- **L'agent mobile** sera envoyé à un marché électronique où les agents des achats et les agents de vente interagissent entre eux pour atteindre un accord.
- **Un agent proxy** est aussi un agent stationnaire qui relie le système multi-agent pour le serveur de services web. Il est l'un des agents qui est toujours en place et en cours d'exécution dans le système multi-agent. L'agent proxy coopère avec l'agent de page blanche pour créer un agent mobile de vente ou l'achat pour chaque utilisateur.
- **L'agent de page jaune** fournit le service de pages jaunes, au moyen d'un agent qui peut recevoir des informations sur des produits disponibles. L'agent de page blanche représente l'autorité et fournit des services de l'authentification des agents résidents et du contrôle d'enregistrement.
- **Les agents mobiles** (acheteurs ou vendeurs) ont d'abord négocié avec d'autres agents mobiles dans le serveur de médiateur du même hôte avant de migrer entre plusieurs sites Web pour communiquer avec d'autres agents.

Il y a trois événements de temps qui indiquent les comportements d'un agent mobile:

- l'agent commence son processus de négociation à intervalles réguliers (par exemple, chaque minute).
- l'agent commence sa migration lorsque le temps d'activité par serveur est atteint.
- l'agent termine son cycle de vie lorsque sa durée de vie est épuisée.

Notez qu'aucun agent ne provoque des agents (acheteurs ou vendeurs) à migrer, mais ils se mobilisent de manière autonome.

### 3.1.2 Migration des agents

L'agent commence sa migration de son serveur médiateur avec une liste d'itinéraire acquis auprès du médiateur. Si on a  $N$  serveurs, l'agent visitera ces  $N$  serveurs en séquence (Comme illustré à la Figure 3.2).

Dans chaque serveur, deux événements se produisent résultant de deux actions respectivement:

- Te1 : si l'agent atteint sa durée de vie, il retourne à son hôte où il a été créé et finit ensuite le processus de migration.
- Te2 : si l'agent épuise son temps d'activité de serveur, il migrera au serveur suivant.

De plus, avant que l'agent migre au serveur suivant, il devrait aussi prendre la décision s'il a accompli la tâche au serveur actuel (la tâche est finie quand l'agent reçoit une offre acceptable d'un autre agent). Ainsi, le processus de migration décrit en fait un scénario de comparaison de prix.

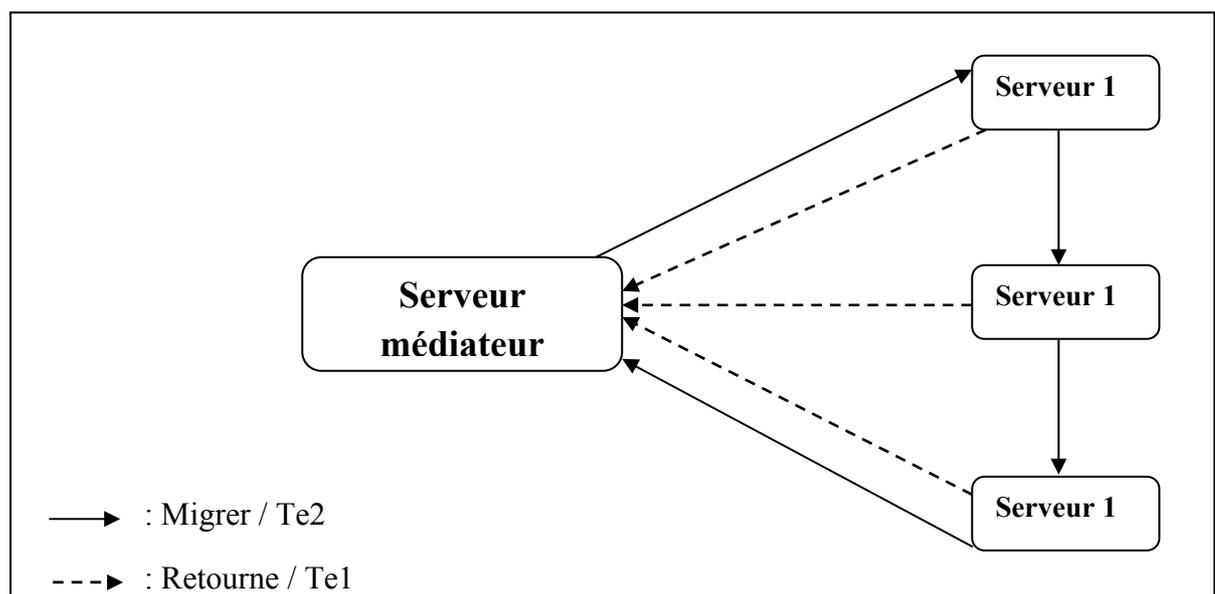
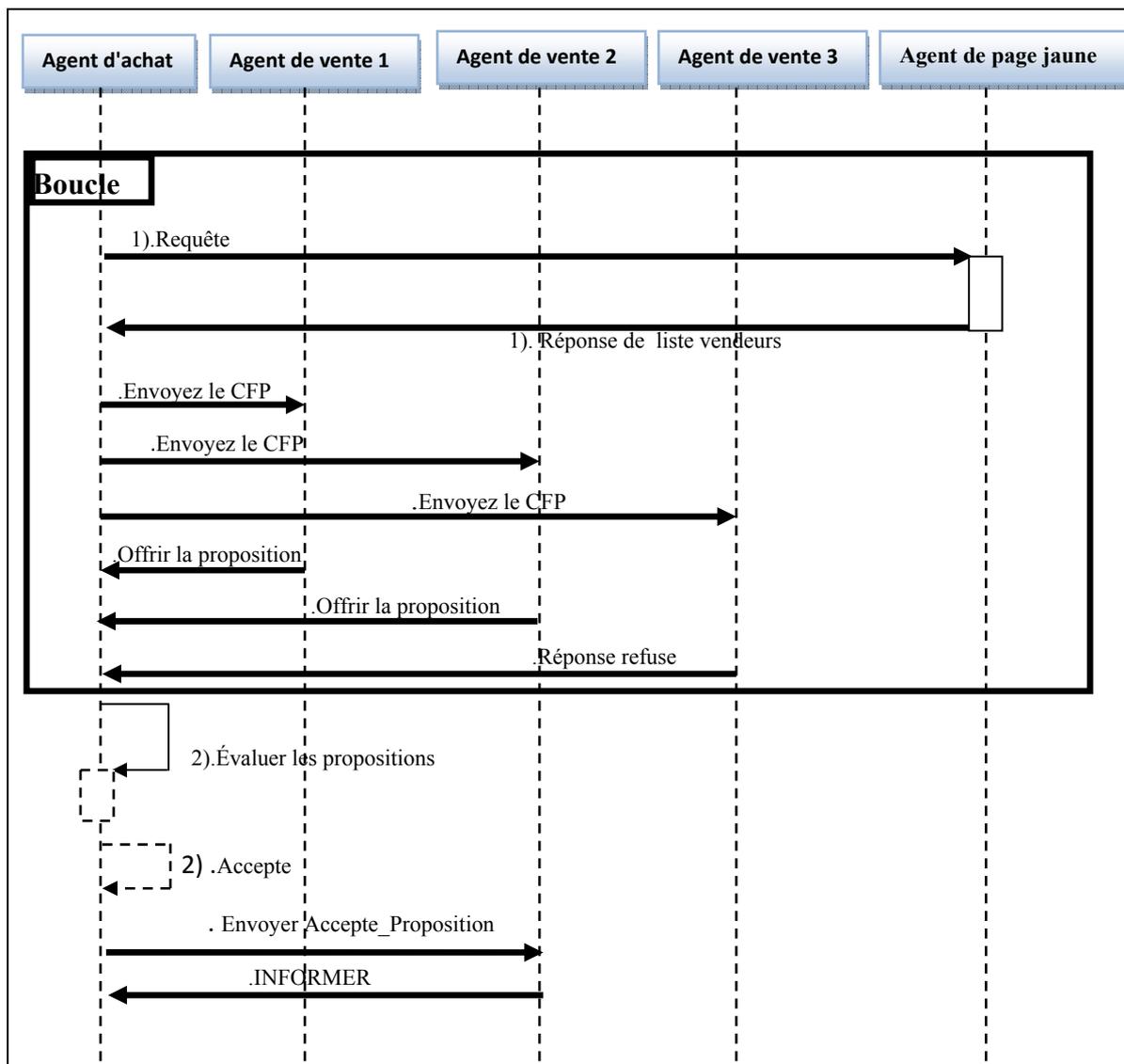


Figure 3.2: Migrations des agents.

### 3.1.3 Processus de négociation

L'interaction proposée entre des agents est conforme au protocole FIPA-Contract-Net. Le protocole réseau contractuel ("Contract Net" en anglais) a été une

des premières approches utilisées dans les systèmes multi-agents pour résoudre le problème d'allocation des tâches [Leh 02]. Les agents coordonnent leurs activités grâce à l'établissement de contrats afin d'atteindre des buts spécifiques. Ce protocole permet à un agent d'achat (l'initiateur) d'envoyer un appel à propositions (CFP) à un ensemble d'agents des ventes (des répondeurs), évaluer les propositions des agents des ventes et accepter ensuite le plus préféré (ou refuser même tous).



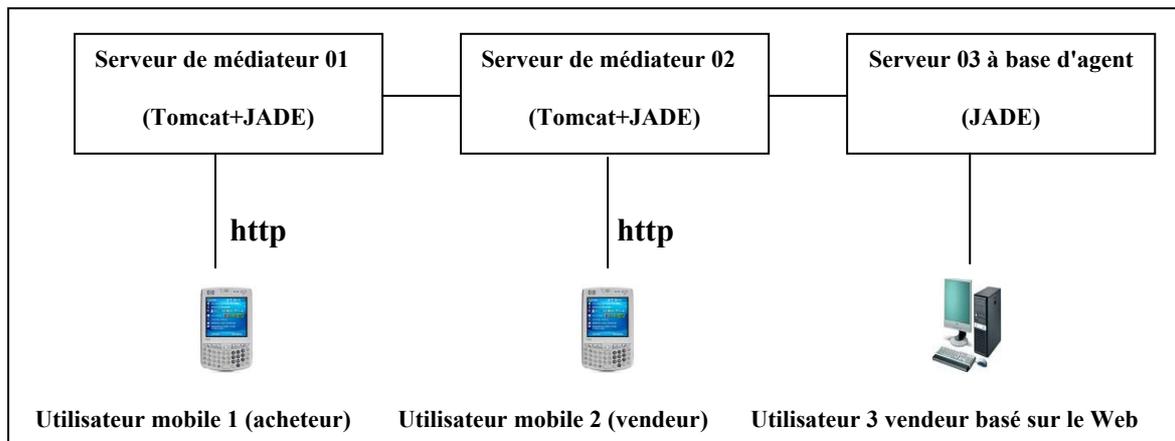
**Figure 3.3:** Processus de négociation

Comme le montre la figure 3.3, l'agent d'achat envoie CFP (un appel de propositions) à tous les agents de vente disponibles (obtenu auprès du service de pages jaunes).

Après la réception du CFP, un agent de vente peut envoyer à l'agent d'achat une proposition avec le prix du produit. Si le produit n'est pas disponible ou vendu, il ne doit pas envoyer n'importe quelle proposition. L'agent d'achat placera un bon de commande si le prix d'offre est dans le prix maximal que le client a spécifié. Les résultats de négociations des prix sont renvoyés à l'agent personnel et ont montré dans une interface graphique à l'utilisateur. Puisque le système est entièrement asynchrone, l'intention de faire un achat ne conduit pas nécessairement à un succès.

### 3.1.4 Implémentation du système

Shillong Weng et Thomas Tran ont déployé 3 serveurs dans le réseau Local, installé J2ME MIDlet dans 2 simulateurs de téléphone mobile, fournis GUI pour le vendeur sur le web simulé un scénario de simples objets utilisés dans négociation électronique. Des deux serveurs de médiateur utilisent le serveur Tomcat et le conteneur principal de plate-forme de JADE. Le troisième ordinateur joue un rôle d'un marché à base d'agents sur l'Internet (Comme illustré à la Figure 3.4).



**Figure 3.4:** environnement d'expérimentation.

Shillong Weng et Thomas Tran ont observé les résultats suivants:

- Des utilisateurs mobiles peuvent se connecter aux serveurs de médiateur via http et initie un agent mobile de vente ou d'achat dans le serveur de médiateur.
- Des utilisateurs mobiles ne doivent pas instruire leurs agents mobiles de ce qu'il doit faire après la configuration de leurs préférences.

- Des agents mobiles sont actifs dans leurs serveurs dans le temps d'activité de serveur indiqué et migrent ensuite parmi les serveurs.
- Agents d'achat peuvent faire accord avec agents de vente lorsque l'article nécessaire et le prix sont mis en correspondance. Les utilisateurs mobiles ensuite recevant des messages textes de leurs agents affichés sur l'écran des simulateurs.
- Des agents mobiles finissent leurs cycles de vie en finissant leurs tâches.

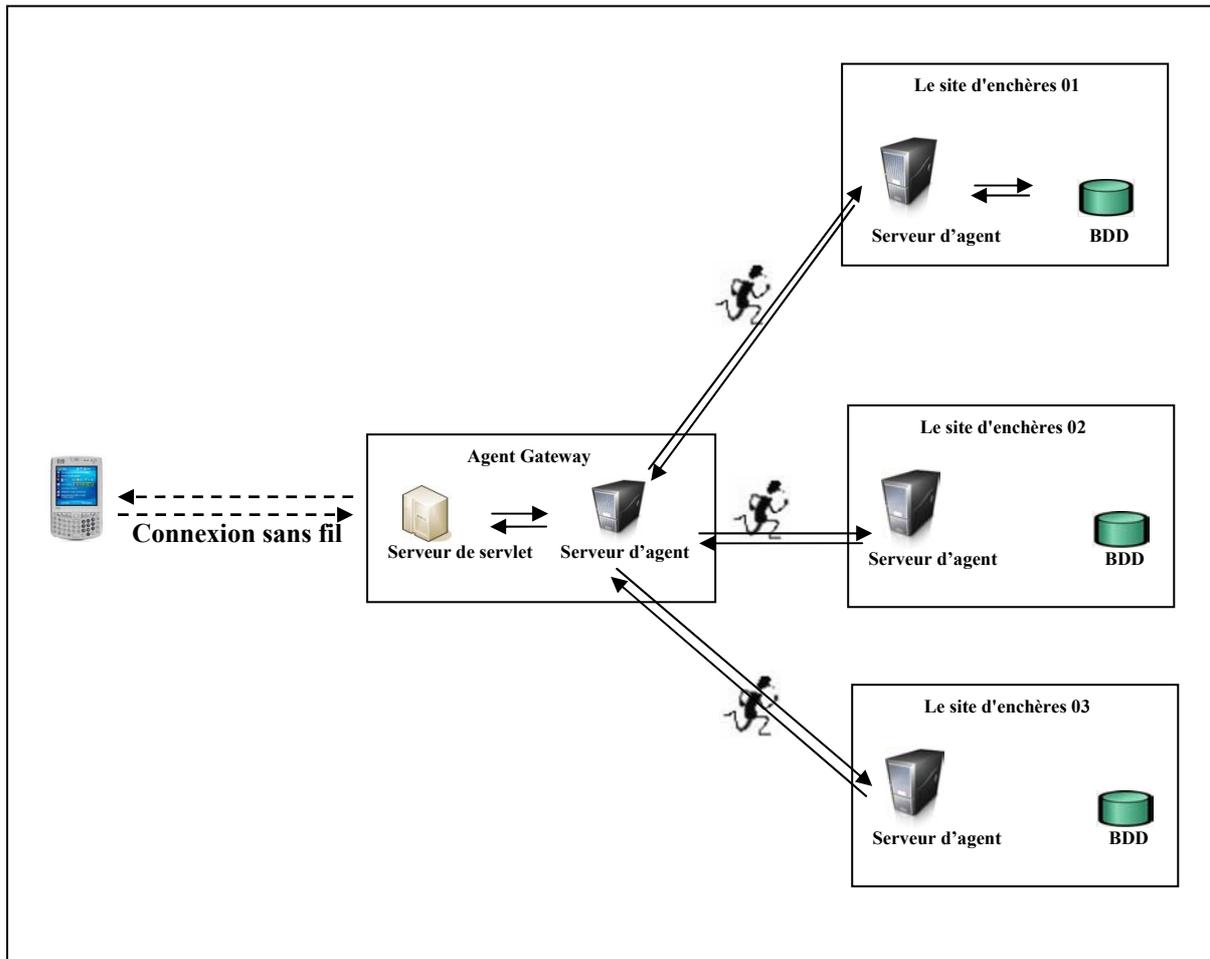
### 3.2 Présentation du système de J-Phone

Calvin Wan et Ronnie Cheung [Cal 10] ont présenté une architecture d'agent des enchères de J-phone. Le système d'agent de vente aux enchères de J-Phone consiste en trois principales parties:

L'application J2ME, AgentGateway et sites d'enchères (Comme illustré à la Figure 3.5).

- **L'application J2ME** est l'interface pour l'utilisateur pour envoyer des demandes.
- **AgentGateway** comprend un Serveur de servlet et Serveur d'agent.
- **Le site d'enchères**, d'autre part, accueille de l'agent et interagit avec l'agent pour alerter le statut de l'appel d'offres ainsi que de placer des enchères (l'offre).

Le système fournit les fonctions aux utilisateurs incluant : la recherche des articles de vente aux enchères, lancement de l'agent d'enchère, gestion de l'agent, vent des articles, et la notification.



**Figure 3.5:** Présentation du système de vente aux enchères de J-phone.

L'architecture de ce système se compose de trois principaux éléments reliés entre eux. Les composants dans ce système sont : un dispositif mobile, une passerelle d'agent et les sites de vente aux enchères comme indiqué dans la figure 3.5.

Dans cette approche l'utilisateur est capable d'envoyer un agent pour faire quelques travaux pour le compte de l'utilisateur par le biais du serveur de Servlet. L'agent effectue son travail et en ce moment, l'utilisateur peut se déconnecter du réseau. Lorsque l'agent termine sa tâche, il revient à son environnement d'accueil et pousse les résultats aux dispositifs.

L'utilisateur peut choisir les services de l'interface et envoyer une demande à AgentGateway. Tous les codes de l'agent mobile sont stockés dans AgentGateway, et l'utilisateur peut initier et gérer les agents mobiles à partir AgentGateway.

Lorsque le Servlet reçoit la demande de client via une connexion http, il effectue des actions correspondant à créer des agents ou envoi des messages aux agents. Quand l'agent mobile recueille tout les résultats, il renvoie les résultats au serveur agent mobile. Serveur d'agent mobile, passe ensuite le résultat à Servlet. Toutes les activités d'enchère sont effectuées dans le site d'enchères. Chaque site d'enchères a des agents pour gérer les enchères.

### 3.2.1 La plateforme de l'approche

L'auteur utilise dans cette l'approche la plateforme suivante :

- **Aglets** est la plate-forme d'agent utilisée sur ce système et il fonctionne sur la passerelle d'agent. Un aglet est un agent logiciel autonome basé sur Java qui est autonome. Aglets peut décider d'exécuter des actions, comme le voyage à travers un réseau à un nouvel hôte,

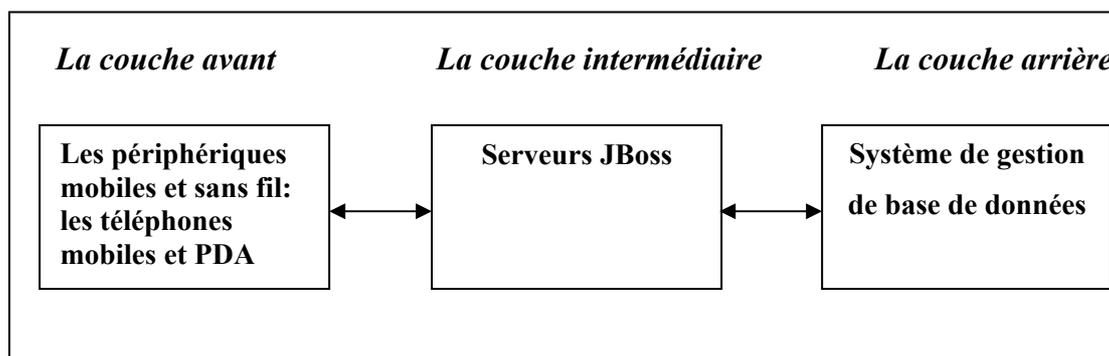
### 3.3 L'architecture de *Gilda Pour*

Cette approche a été proposée par **Gilda Pour** (Université d'État de San José) [Gil 04] lors de la Conférence internationale sur le commerce électronique 2004 (IADIS 2004).

**Gilda Pour** a développé une architecture basée sur agent mobile pour les applications de commerce mobile. L'objectif principal de ce projet de recherche est d'explorer et d'étendre les avantages des composants de l'agent (par exemple la mobilité, l'autonomie et la collaboration) pour les applications de m-commerce.

#### 3.3.1 Architecture de **Gilda**

Cette l'architecture est conçue avec trois principales couches. La figure 3.6 montre la conception de haut niveau de cette l'architecture à base d'agents mobile : (1) la couche avant, (2) la couche intermédiaire, et (3) la couche arrière. La couche avant est principalement pour la présentation, la couche intermédiaire pour la logique de l'application et la couche arrière pour la gestion des données.



**Figure 3.6:** Conception de haut niveau de l'architecture à base d'agents mobiles.

### 3.3.2 Les types d'Agents

Cette architecture contient les types d'agents suivants:

- Agents Interface de l'utilisateur
  - Agents de processus (transformation)
  - les agents wrapper : encapsulent les applications préexistantes.
- ❖ Les **Agents Interface de l'utilisateur** reçoivent des demandes des utilisateurs (les humains et les autres systèmes), emballer les demandes et les envoyer à des agents de processus approprié.
  - ❖ Les **agents de processus (transformation)** traduisent les demandes des utilisateurs qu'ils reçoivent par des agents d'interface d'utilisateur dans une série de tâches, ils déterminent les meilleures sources pour obtenir les informations exigées pour traiter les demandes et prendre des décisions parmi des alternatives. Ces agents agissent comme des substituts pour des utilisateurs dans la gestion et l'exécution du processus, **Agents de suivi (Tracking)** sont des exemples d'agents de processus (transformation).
  - ❖ Un **agent wrapper** représente les interfaces d'un système hérité, annonce les capacités du système hérité de manière compatible agent; agit comme le proxy du système hérité, manipule toutes les demandes de service destinées au système et gère les interactions du système.

### 3.3.3 La plateforme de l'approche

L'auteur utilise, dans cette l'approche, la plateforme suivante :

✚ **JADE-LEAP** : c'est la combinaison de LEAP (Lightweight Extensible Agent Platform) et Java Agent DEvelopment Framework (JADE). JADE-LEAP soutient les agents exécutés sur les périphériques légers tels que les PDA et les téléphones cellulaires. Il peut être déployé sur différents types de dispositifs, réseaux et JVMs (machines virtuelles Java). JADE-LEAP offre un ensemble homogène d'APIs (Application Programming Interface).

La couche intermédiaire se compose de deux principales classes de composants : sockets Java et Agent du client. Classes de socket Java représentent le lien entre un programme client et un serveur. Agent du client sont des agents mobiles, une fois qu'un nouvel utilisateur se connecte au système, un Agent du client sera automatiquement créé pour agir au nom de l'utilisateur. Le nom d'Agent du client est unique et dépend de l'identificateur d'utilisateur. Un Agent du client contient les informations personnelles de ses utilisateurs, et réagit aux différents messages entrants et sortants et des demandes destinées à l'utilisateur. Une fois que l'utilisateur se déconnecte, la vie d'Agent du client arrive à son terme.

Une fois le serveur reçoit une commande, il communique avec une base de données. Après que les résultats sont rendus par la base de données, le serveur organise les données et les renvoie à l'ordinateur de poche. Une fois le système est lancé, le conteneur crée automatiquement un conteneur de couche arrière pour préparer les autres activités de gestion. Si plus d'un utilisateur se connecte au système en même temps, un ensemble d'agents clients seront créés.

### **3.3.4 La sécurité**

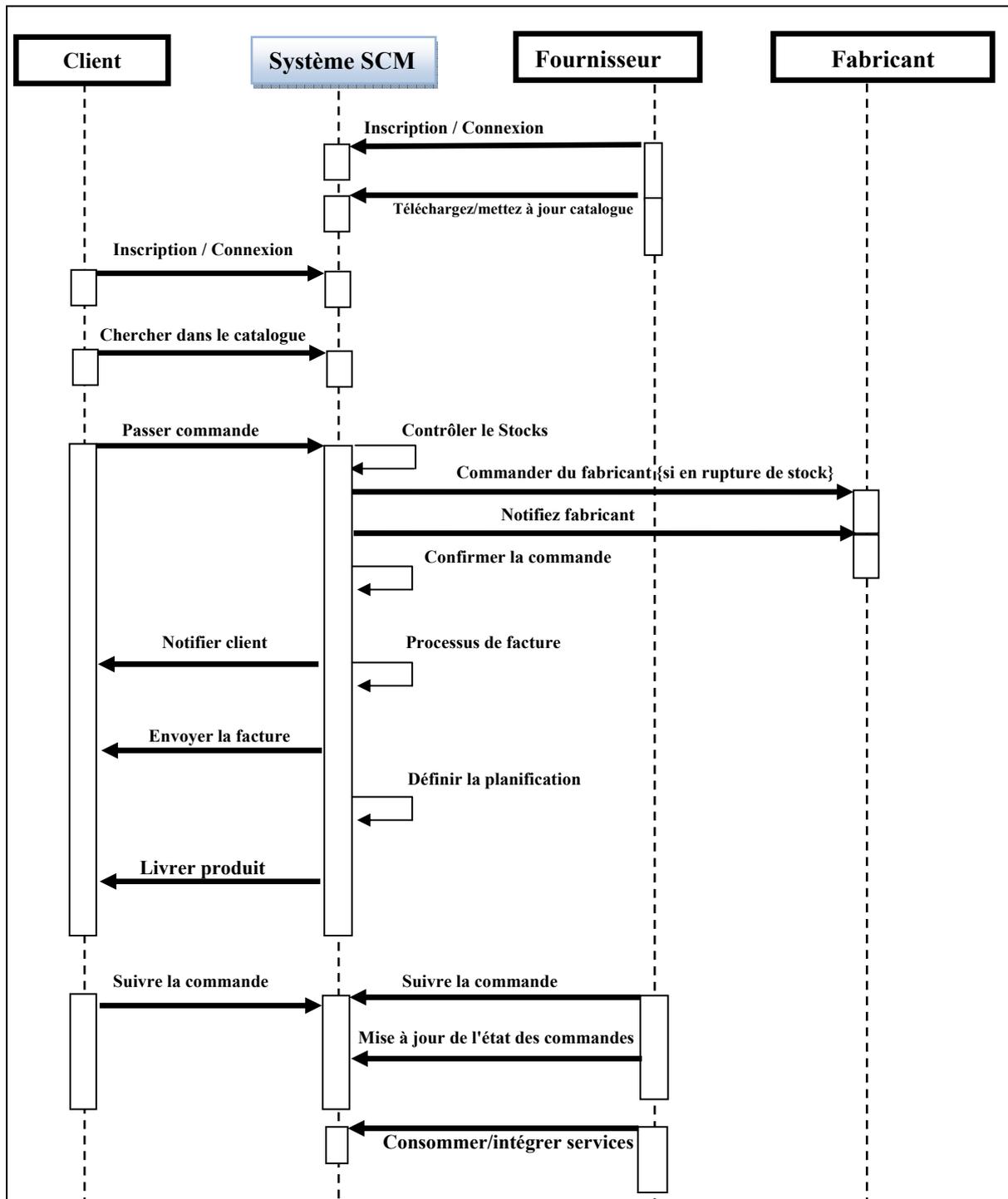
Pour résoudre le problème de sécurité, Gilda a pris les mesures suivantes:

- un utilisateur authentifié a son propre agent.
- Un agent agissant a certaines autorisations au compte l'utilisateur.
- La plate-forme vérifie le certificat de l'agent pour empêcher des agents non autorisés de prendre des actions qu'ils ne sont pas autorisés à prendre.

### **3.3.5 Un système de gestion des stocks**

Gilda a développé un prototype de démonstration de gestion de stock et des systèmes de suivi (**Tracking System**), l'auteur a utilisé les architectures à base

d'agents mobiles pour le m-commerce. Les agents utilisent la FIPA/JADE SMA (page blanche) et d'agent facilitateur (page jaune) pour l'enregistrement et la recherche. La figure 3.7 montre un diagramme de séquence pour un système de gestion des stocks.



**Figure 3.7:** Diagramme de séquence pour un système de gestion des stocks.

Comme nous venons de le voir dans le diagramme, premièrement le fournisseur fait une inscription s'il ne possède pas de compte, sinon il est connecté par

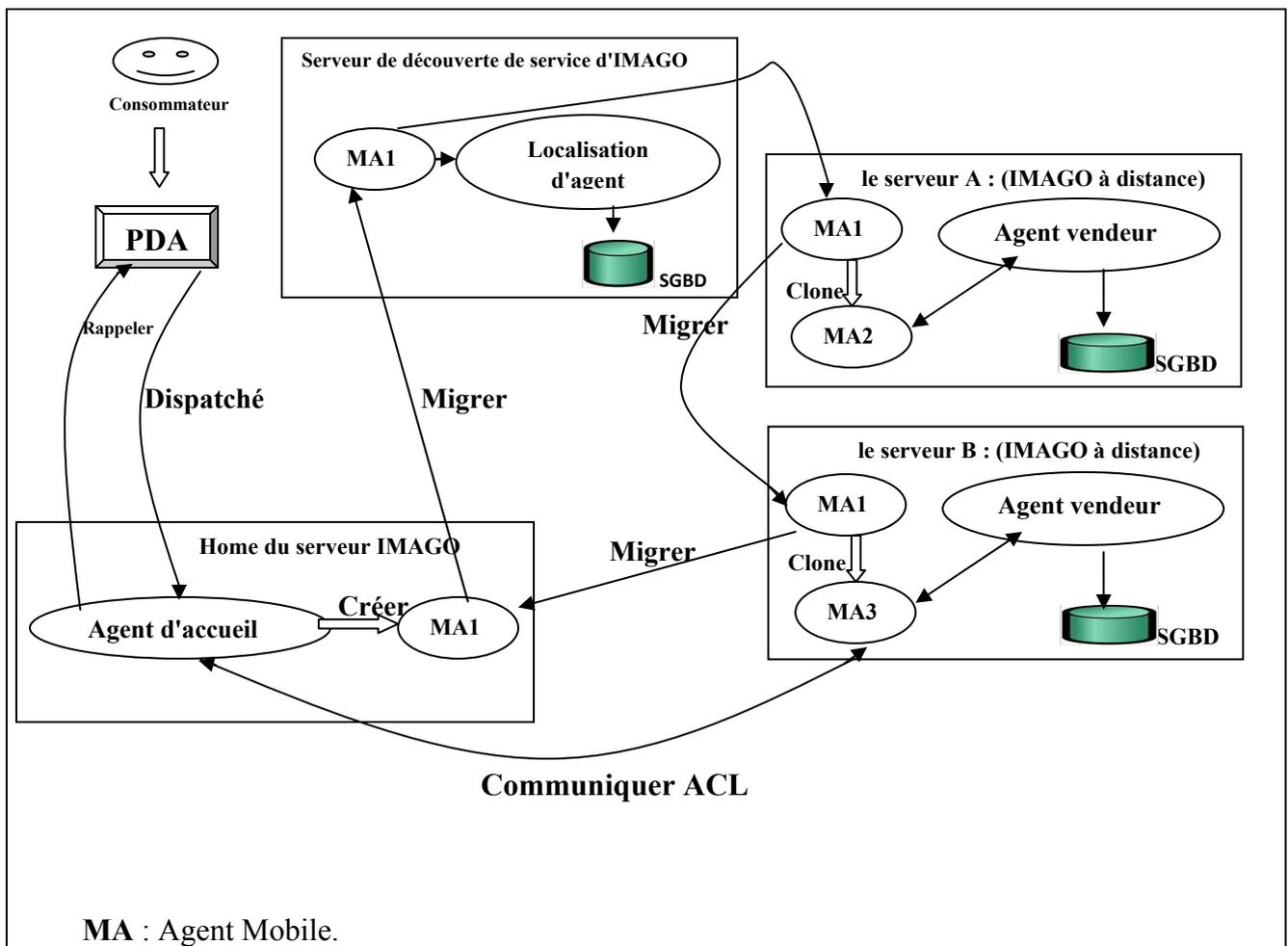
son nom et son mot de passe, et après il envoie un produit et fait une mise à jour de catalogue.

Pour la partie de l'utilisateur, celui-ci fait une inscription s'il ne possède pas de compte, sinon il est connecté par son nom et son mot de passe. Pour demander un produit il faut chercher dans le catalogue, et ensuite il passe une commande de produit au système. Ce dernier contrôle le stock, si le produit n'existe pas, le système envoie une demande à un fabricant pour fabriquer le produit demandé par l'utilisateur, sinon le système notifie le fabricant et confirme la commande. Par la suite le système informe et envoie la facture à l'utilisateur. Enfin le fournisseur suit la commande et fait MAJ de l'état des commandes pour consommer ou intégrer les services.

### **3.4 Le système IMAGO**

Le système IMAGO (Intelligent Mobile Agent Gliding On-line) est une infrastructure pour développer des applications agents mobiles, tels que M-commerce ou distribué d'exploration de données [Xin 06].

Il y a beaucoup de scénarios possibles d'affaires pour développer des applications de commerce mobile. Cette approche a été proposée par **Xining Li** (Département de l'informatique et des sciences de l'Information, Université de Guelph, Canada.) à la sixième Conférence internationale du Wuhan sur E-Business. Cette proposition se concentre sur le commerce mobile déterminé par le consommateur, c'est-à-dire, un modèle de commerce électronique dans lequel le consommateur initie une transaction commerciale. Plus précisément, cette structure représente un environnement distribué qui permet aux consommateurs de dépêcher des agents mobiles de leurs appareils portatifs pour visiter les magasins en ligne pour la recherche, en comparant, et en évaluant l'achat et le paiement des marchandises. La figure 3.8 donne un exemple d'agents qui se produisent dans le système de commerce mobile d'IMAGO et indiquent leurs comportements de base.



**Figure 3.8:** Un exemple d'utilisation des agents mobiles dans le M-commerce.

De façon générale, la structure d'IMAGO M-commerce distingue trois grands types d'agents [Xin 06]. :

- **Agent de dispositif** : il est lié avec l'utilisateur et installé sur l'ordinateur de poche. La principale responsabilité de cette agent est de permettre à un utilisateur mobile de localiser son serveur hôte (Home du serveur IMAGO) et de communiquer avec l'agent d'accueil d'invoquer une application M-commerce. il appelle l'agent dispositif comme portail IMAGO mobile.
- **Agent stationnaire**: c'est un agent stationnaire qui réside toujours dans son hôte. Pour des raisons différentes, il identifie trois types d'agents stationnaires : **agent d'accueil**, **agent de la localisation** et **l'agent vendeur**. **Agent d'accueil** fournit un pont entre l'utilisateur mobile et l'application M-commerce. Il est responsable de communiquer avec le portail mobile, accepter

les demandes de l'utilisateur et l'envoi correspondant agents mobiles pour démarrer une transaction commerciale. **Agent de la localisation** fournit des services de découverte et réside au niveau du serveur de découverte IMAGO.

**Un agent vendeur** agit comme le représentant du vendeur pour garder une trace de toutes les transactions et demandes.

- **Agent mobile** : un agent mobile représente l'utilisateur errant dans Internet pour effectuer la tâche assignée par son agent d'accueil. Il visitera l'agent de la localisation pour obtenir les adresses sur les serveurs à distance qui peuvent porter le produit désiré par le client. Quand l'agent mobile arrive à chaque serveur à distance il clone un autre agent puis se déplace vers le prochain serveur.

Dans une demande de commerce mobile, les agents ne travaillent pas seuls et ils doivent communiquer avec eux pour coopérer et générer une agrégation des données globales pour une analyse plus approfondie. Les systèmes d'agents mobiles les plus existants adoptent quelques modèles/protocoles de communication de systèmes distribués traditionnels. Cependant, le système d'IMAGO adopte une stratégie différente de faire face à cette question. L'idée est de déployer des messagers mobiles intelligents pour la communication inter-agent. Les messagers sont des agents minces consacrés pour livrer des messages. Le système IMAGO fournit un ensemble de messagers intégrés dans le cadre de son interface de programmation. Un agent mobile sur des sites éloignés et à tout moment peut dépêcher des messagers de fournir des données aux récepteurs désignés. Par exemple, supposons qu'un agent mobile a terminé son travail de M-commerce à un serveur distant de vendeur, il peut migrer sur son serveur d'origine (Accueil), ou envoyer un messager pour livrer le résultat à l'agent d'accueil. La communication entre les agents s'effectue au moyen Agent Communication Language (ACL).

#### **4. Synthèse**

Dans ce chapitre nous avons présentés quatre architectures pour le business mobile utilisant les agents mobiles. Premièrement, on a présenté une architecture de Zhiyong Weng et Thomas Tran qui ont proposé une structure à base d'agent mobile qui permet aux acheteurs et des vendeurs d'exécuter les affaires à l'aide d'appareils mobiles. Cette architecture est basée sur des systèmes multi-agents qui, de par leurs

caractéristiques, facilitent la prise en compte de la très grande dynamique de l'environnement dans lequel s'exécutent les affaires à l'aide d'appareils mobiles. Zhiyong Weng et Thomas Tran ont proposé le processus de migration où le serveur de médiateur crée un seul agent mobile, cet agent commence sa migration de son serveur médiateur avec une liste d'itinéraire acquis auprès du médiateur. Si on a  $N$  serveurs, l'agent visitera ces  $N$  serveurs en séquence dans un temps limité par le serveur de médiateur. Si l'agent atteint sa durée de vie, il retourne à son hôte où il a été créé et finit ensuite le processus de migration ; dans ce cas, l'agent peut ne pas visiter tous les serveurs. Cela peut conduire à une perte des chances de négociation avec les autres serveurs. Comme nous l'avons dit, le serveur de médiateur crée un seul agent pour tous les serveurs à distance. L'inconvénient de cette architecture est : augmenter le temps de migration et de négociation ; et si l'agent de migration échoue, ou le nœud sur lequel il est exécuté échoue, les processus de migration et de négociation seront détruits. Deuxièmement on a présenté une l'architecture de J-Phone, Calvin Wan et Ronnie Cheung qui ont présenté une architecture de J-phone à base d'agent mobile pour les ventes aux enchères. Dans cette architecture, le serveur de médiateur (AgentGateway) ne contient pas la liste d'itinéraires pour les sites d'enchères. Dans cette approche le serveur de médiateur crée les agents mobiles selon le nombre des sites d'enchères et envoie à chaque site d'enchères un agent mobile. Cette liste d'itinéraires contient le chemin de migration, les adresses des sites d'enchères et les services offerts par ces sites. Dans la première approche (l'approche de Zhiyong Weng et Thomas Tran), le serveur de médiateur possède une liste d'itinéraires. Pour cela l'agent mobile sait où et comment migrer, comme indiqué dans l'itinéraire de migration. Le principal avantage de cette approche (J-Phone), c'est que le serveur de médiateur envoie un agent mobile pour chaque site d'enchère afin de minimiser le temps de migration et éviter l'échec de la première approche, toutefois l'inconvénient de cette approche, c'est que le serveur de médiateur envoie les agents mobiles à tous les sites d'enchères.

Troisièmement, on a présenté l'architecture de Gilda, qui a développé une architecture basée sur agent mobile pour les applications de commerce mobile. L'objectif principal de ce projet de recherche est d'explorer et d'étendre les avantages des composants de l'agent pour les applications de m-commerce. Il a construit un prototype de démonstration de gestion de stock et des systèmes de traçabilité

(Tracking System) avec la nouvelle architecture pour des environnements mobiles. Le but de cette approche est d'acquérir une expérience dans les protocoles d'interaction des agents et à réaliser l'importance des normes architecturales. Gilda a utilisé JADE-LEAP (Java Agent Development Environment Platform léger Extensible Agent) pour mettre en œuvre un agent personnel sur un PDA. LEAP peut être utilisé pour déployer des systèmes multi-agents répartis sur les appareils mobiles et des serveurs, mais il exige une permanente connexion bidirectionnelle entre les terminaux mobiles et les serveurs.

La dernière architecture est IMAGO, cette architecture représente un environnement distribué qui permet aux consommateurs de dépêcher des agents mobiles de leurs appareils portatifs pour visiter les magasins en ligne pour la recherche, en comparant et en évaluant l'achat et le paiement des marchandises. L'inconvénient de cette approche, c'est que l'agent mobile fait le clonage dans les serveurs à distance. Cette méthode n'est pas recommandée car elle viole l'intégrité des serveurs. Un autre inconvénient, c'est que les consommateurs envoient l'agent mobile à partir de leurs appareils portatifs, entraînant potentiellement de longue durée de connexion avec l'Internet. Nous croyons qu'il est important de tenir compte des limites des périphériques mobiles, comme bande passante, batterie et la capacité de calcul limitée et le temps de connexion coûteux.

L'approche proposée dans ce travail est une approche basée agent mobiles conçue pour le M-business, un consommateur peut connecter son appareil mobile, comme un PDA ou un téléphone mobile, au serveur d'application via une connexion sans fil et envoyé ensuite une demande de création d'un agent mobile pour entreprendre une tâche d'affaires spécifique en son nom.

## **5. Conclusion**

Ce chapitre a présenté l'état de l'art utile sur des travaux développés dans le cadre de ce mémoire. Pour cela nous avons présenté quelques architectures basées agents mobiles pour les affaires commerciales conduites à l'aide d'appareils mobiles. Nous avons terminé cet état de l'art par une synthèse récapitulative en montrant quelques avantages de ces architectures, ainsi que quelques problèmes rencontrés.

Dans le chapitre suivant, nous présenterons notre architecture pour les affaires commerciales à base du paradigme agent mobile.

# Chapitre IV

---

## **Proposition d'un environnement de M-Business**

---

### **1. Introduction**

Dans le précédent chapitre nous avons montré l'utilité de l'utilisation des agents mobiles et leurs applications dans le domaine du business mobile.

De nombreux travaux ont été élaborés afin d'introduire la technologie d'agent mobile et les concepts liés à cette dernière pour les affaires commerciales conduites à l'aide d'appareils mobiles (portables, assistants numériques, etc...). Le concept d'agent mobile apparaît dans ce contexte comme une solution facilitant la mise en œuvre d'application mobile. Dans ce chapitre nous allons décrire notre contribution qui est la proposition d'une architecture d'environnement basée agent mobile pour M-business, en s'appuyant sur la capacité de la mobilité des agents.

Dans notre étude, nous avons utilisé les Aglets [Dan 98] dans les parties serveur et les sites de fournisseurs, car les Aglets sont spécifiques dans la mobilité que les autres plateformes, avec cela pour l'utilisation des avantages des agents, où les agents mobiles offrent plusieurs avantages améliorant la performance de plusieurs applications distribuées.

Nous commençons la description de notre architecture par la motivation de choix de cette approche. Ensuite, nous présentons la description de l'architecture proposée et les structures internes des différents agents, ainsi que leurs rôles. Enfin, la conclusion est présentée dans la dernière section.

### **2. Objectif et Motivation du travail**

#### **2.1 Objectif**

Les affaires commerciales sont une partie importante dans la vie quotidienne depuis des années, et avec les progrès dans les capacités de calcul et de communication des appareils portatifs tels que les PDA, Pocket PC, ordinateurs

portables et d'autres, nous commençons à voir l'émergence d'une variété d'applications dans m-business. La puissance de ces appareils mobiles incitant de plus en plus des consommateurs et des entreprises pour les utiliser dans leurs affaires quotidiennes, ainsi que dans des activités récréatives.

Nous avons vu que les applications **d'e-business** reposant sur le modèle **client/serveur** constitue un obstacle au développement d'application de **M-Business** et ce modèle souffre de nombreux problèmes comme :

- ✚ Les affaires commerciales exigent une connexion permanente étant établie entre le client et le serveur.
- ✚ L'augmentation du trafic réseau
- ✚ Le consommateur gérer des affaires avec restriction de temps et d'espace.
- ✚ Le mode connecté est synchrone : le client émet une requête et se bloque jusqu'à l'arrivée de la réponse.
- ✚ L'exigence aux clients de vérifier fréquemment les possibilités d'échange et de prendre des décisions plus manuellement.

Nous sommes arrivés à ce que les agents mobiles peuvent régler ces problèmes et constituent un paradigme de programmation adéquat et efficace pour ce genre d'application.

L'objectif principal de notre étude est de proposer une architecture d'environnement basée agents mobiles pour M-business, en s'appuyant sur la capacité de la mobilité des agents.

Dans cette architecture, un utilisateur, après la création et l'initialisation d'un agent mobile pour agir en son nom, peut déconnecter son appareil mobile du serveur. Il n'a besoin que de se reconnecter plus tard et de rappelle l'agent pour les résultats, réduisant aussi au minimum l'utilisation des ressources.

## 2.2 Motivation

Dans ce modèle d'agent mobile, un agent est un processus qui a un contexte d'exécution, incluant du code et des données, pouvant se déplacer de machine en machine afin de réaliser la tâche qui lui est assignée.

L'orientation vers l'approche agent mobile est motivée par les points suivants :

- ✚ la mobilité d'agent permet à un client d'interagir localement avec un serveur, et donc de réduire le trafic sur le réseau en ne transmettant que les données utiles.
- ✚ les agents sont capables de chercher l'information d'une façon plus intelligente, par exemple en cherchant selon des concepts.
- ✚ Les agents s'adaptent aux préférences et aux souhaits de chaque usager. Ils peuvent ainsi apprendre de leurs recherches précédentes et par la suite comprendre mieux les besoins des utilisateurs.
- ✚ 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 de communiquer et coopérer entre eux, ce qui accélère et facilite la recherche.
- ✚ L'utilisation des agents mobiles dans le m-business peut réduire le trafic réseau inutile, fournir des services plus avancés et réduire les frais de participation et améliorer l'efficacité commerciale.
- ✚ Grâce à la flexibilité et la mobilité des agents mobiles, l'agent mobile peuvent compléter le modèle client/serveur pour permettre aux consommateurs de faire des affaires sans restrictions de temps et d'espace.

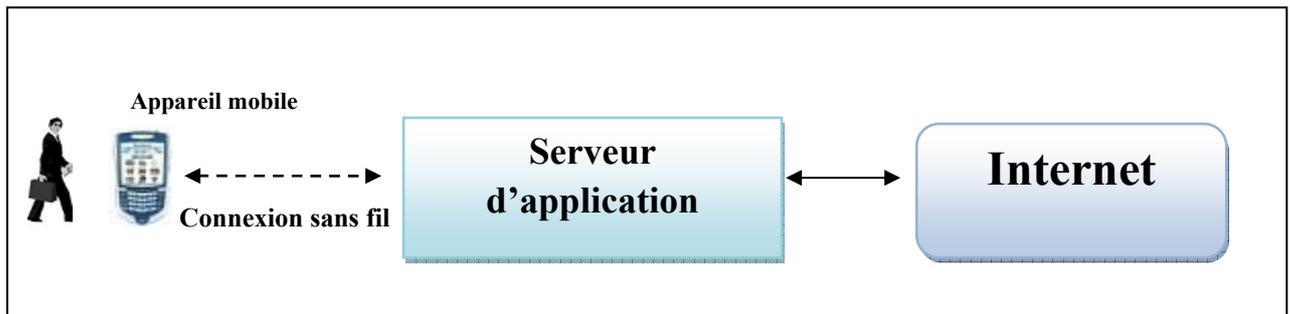
Dans la suite de ce chapitre, nous proposons notre architecture basée sur la technologie d'agents mobiles qui permette aux consommateurs de gérer les affaires commerciales conduites à l'aide d'appareils mobiles (portables, assistants numériques, ... etc.) à tout moment et n'importe où.

### **3. Approche proposée**

L'approche proposée dans ce travail est une approche basée agents mobiles conçue pour le M-business. La figure 4.1 montre un environnement Mobile business et un consommateur qui peut connecter son appareil mobile, comme un PDA ou un téléphone mobile au serveur d'application via une connexion sans fil. Il envoie ensuite une demande de création d'un agent mobile pour entreprendre une tâche d'affaires spécifique en son nom.

Le serveur d'application fournit des services comme la création des agents mobiles en fonction des demandes des consommateurs. Après avoir été créé, les

agents mobiles de façon autonome voyagent à plusieurs serveurs à base d'agents sur internet lorsque le consommateur désire procéder à une comparaison sur plusieurs marchés mondiaux.



**Figure 4.1:** Un environnement de business mobile.

Les consommateurs doivent uniquement utiliser deux fois la connexion à faible bande passante, une fois pour initier un agent mobile et une seconde fois pour collecter les résultats lorsque la tâche est terminée.

### 3.1 L'architecture du système

L'architecture générale de notre système, illustrée à la figure 4.2 s'articule autour de trois principales parties en interaction :

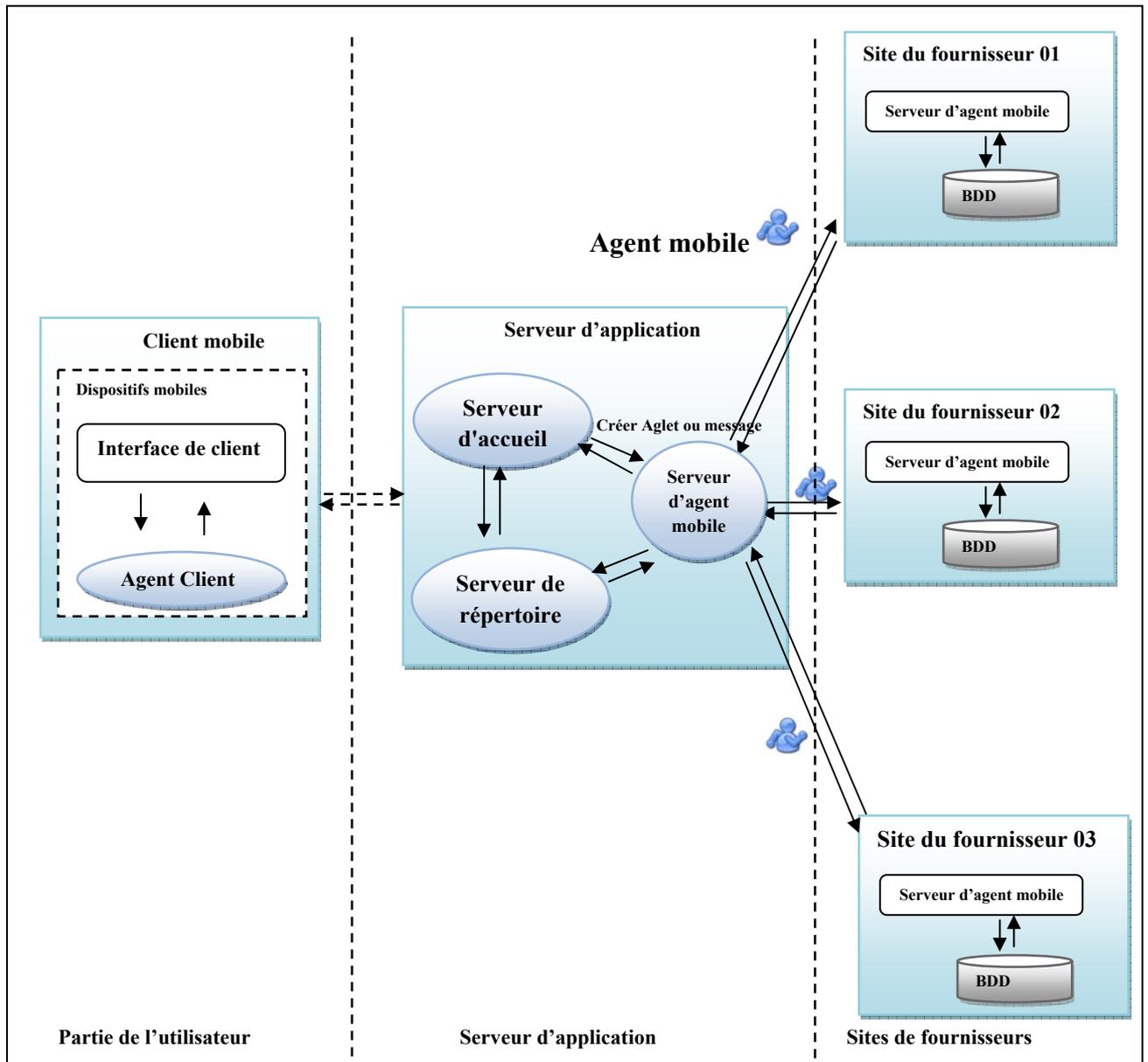
- 1) Partie de l'utilisateur.
- 2) Serveur d'application.
- 3) Sites de fournisseurs.

La figure 4.2 présente notre architecture qui est basée sur le concept « agent mobile » pour M-business.

Comme indiqués dans la figure 4.2, des dispositifs mobiles sont pris en charge par des agents clients et connectés au serveur d'application via la connexion sans fil. Un agent client est un agent statique fonctionnant dans un dispositif mobile et offre une interface utilisateur graphique (GUI) pour son utilisateur pour communiquer avec le système.

La clé de cette architecture du système est le **Serveur d'application** qui agit comme un médiateur entre l'appareil mobile et les sites de fournisseurs.

Dans le serveur d'application, un serveur d'accueil répond à n'importe quelles demandes de l'agent client.



**Figure 4.2:** Architecture du système.

### 3.2 Les types d'agents

Notre système est constitué de plusieurs agents qui coopèrent pour satisfaire les demandes d'achats ou les propositions de vente des clients, on distingue : l'agent client, les agents de répertoire, Maître Aglets, l'agent mobile de recherche et les agents de fournisseurs, plusieurs types d'agents sont utilisés :

- **Un agent client** est un agent stationnaire qui s'exécute sur un périphérique mobile de l'utilisateur et fournit une interface graphique pour permettre à l'utilisateur de configurer un agent mobile (à partir de son appareil mobile).

- **L'agent mobile de recherche** sera envoyé à des sites de fournisseurs où les agents d'achat et les agents de vente interagissent et négocient entre eux pour conclure un accord.
- Un **Aglet** est un agent logiciel autonome basé sur Java, utilisé sur ce système et il fonctionne sur Serveur d'application.
- **Agent répertoire** c'est un agent stationnaire qui dispose d'un répertoire de base de données qui stocke les adresses de tous les fournisseurs.
- **Agent fournisseur** : est un agent stationnaire, il reçoit les agents mobiles de recherche (esclaves) qui sont envoyés par Maître Aglets.

Nous proposons le processus fonctionnel suivant :

- a) en première étape, l'utilisateur initie et gère **l'agent mobile de recherche** et configure ses préférences via **l'agent client**. **L'agent client** ensuite envoie la demande de l'utilisateur au **serveur d'application**.
- b) Le **serveur d'accueil** accepte la demande et communique avec le **serveur d'agent mobile**.
- c) Le **serveur d'agent mobile** crée les agents mobiles et à l'aide du **serveur de répertoire**, il envoie les agents mobiles à d'autres serveurs pour entreprendre la tâche de l'utilisateur.
- d) Quand **l'agent mobile de recherche** recueille tous les résultats, il renvoie les résultats au **serveur agent mobile**. Le **serveur d'agent mobile** passe ensuite le résultat au **serveur d'accueil** ou il envoie des SMS à l'utilisateur.

Ce processus est asynchrone au moyen duquel l'utilisateur peut se déconnecter du réseau à volonté. Cependant, il recevra toujours une notification à base de SMS ou à base de l'e-mail du serveur d'application via le serveur d'agent mobile ou le serveur d'accueil. Le serveur d'application fournit les supports exigés pour la création d'agents mobiles, la messagerie parmi des agents, la migration d'agent, la collaboration, la protection, la destruction et le contrôle d'agents mobiles.

#### 4. L'architecture détaillée du système

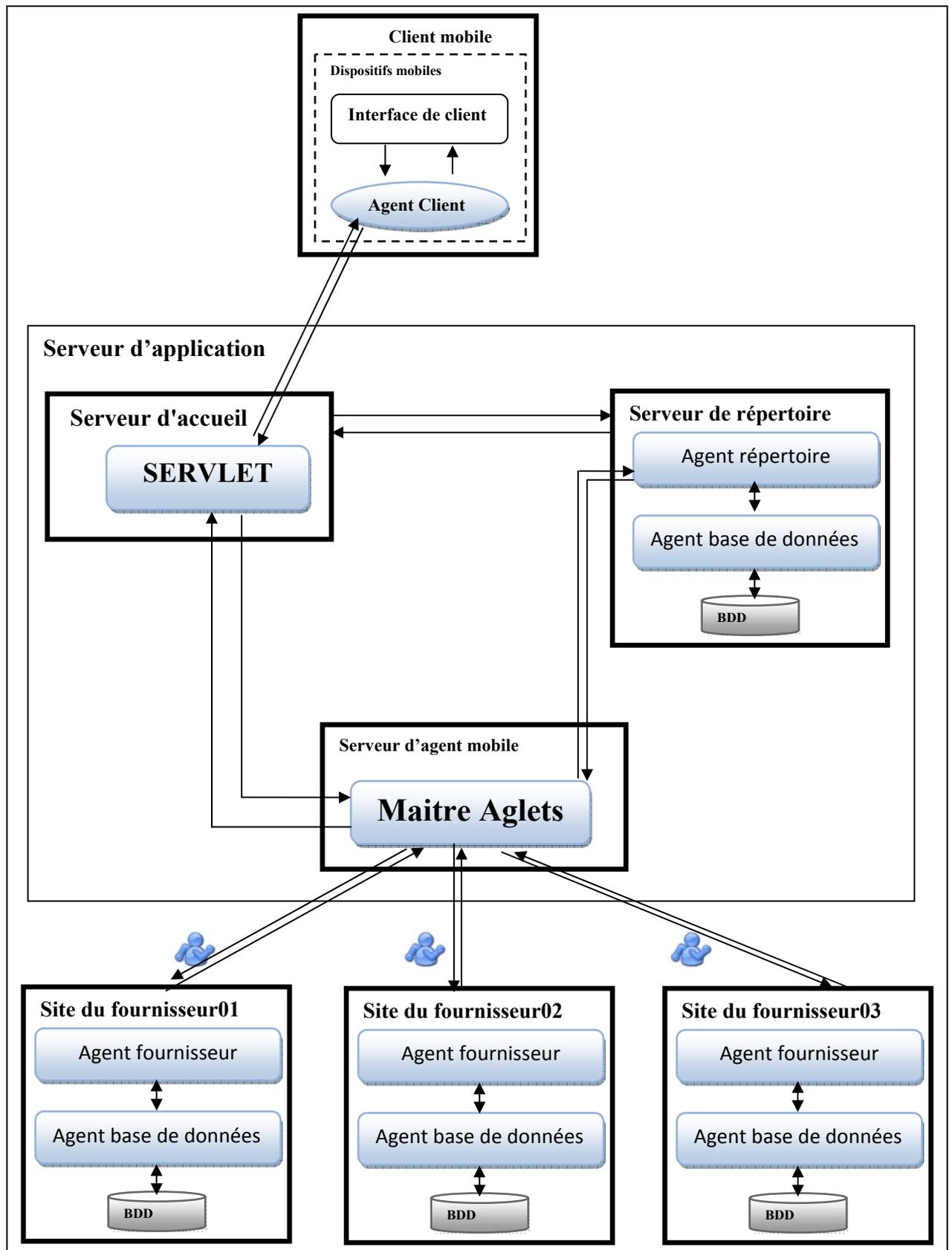


Figure 4.3: L'architecture détaillée.

L'architecture détaillée de notre système, illustrée à la figure 4.3, s'articule autour de trois principales parties en interaction :

#### **a. Partie de l'utilisateur**

Généralement l'interface est l'élément de référence qui permet à l'utilisateur de juger de la qualité d'un système. Elle présente le seul moyen qui permet l'interaction directe entre le système et l'utilisateur.

Cette interface permet à l'utilisateur d'envoyer des demandes, et comprend:

- **Agent Client** est un agent stationnaire qui s'exécute sur un périphérique mobile de l'utilisateur et fournit une interface graphique (interface client) pour interagir directement avec les consommateurs et d'examiner ses préférences personnelles (à partir de son appareil mobile). L'agent client représente les intérêts des consommateurs et permet au consommateur d'avoir un choix de produire et de distribuer un agent de vente ou d'achat mobile.

Les attributs peuvent être configurés par un utilisateur via son dispositif mobile, selon les caractéristiques suivantes : le type d'agent qui spécifie agent d'achat ou un agent de vente, serveur d'agent (l'adresse de serveur d'application), identification de l'utilisateur qui peut être un numéro de téléphone portable ou une adresse e-mail, informations sur le produit prédéterminées, quantité, le prix que l'agent peut offrir, l'offre au meilleur prix actuel; durée de vie (fonctionnement) de l'agent.

#### **b. Serveur d'application**

Il fournit des services comme la création des agents mobiles en fonction des demandes des consommateurs. Après avoir été créé, les agents mobiles de façon autonome voyage à plusieurs serveurs à base d'agents sur l'internet lorsque le consommateur désire procéder à une comparaison sur plusieurs marchés mondiaux. Il comprend :

- 1) **un serveur d'accueil** qui est utilisé pour traiter la demande de partie de l'utilisateur qui inclut la création et/ou gère le **Maître Aglets**. Dans le serveur d'accueil, les **Servlet** répond à n'importe quelles demandes (requêtes) de l'agent client.

- 2) **Le serveur d'agent mobile** comprend **Maître Aglets** pour créer ou envoyer des agents mobiles (esclaves) à des sites de fournisseurs. Maître Aglets coopère avec le **serveur de répertoire** pour recevoir une liste de fournisseurs qui possède des informations sur le produit demandé par client.
- 3) **le serveur de répertoire** dispose d'un répertoire de base de données qui stocke les adresses de tous les fournisseurs, et peut recevoir des informations sur des produits disponibles, ou trouver d'autres agents fournissant les services nécessaires pour réaliser son but. Dans ce serveur on a deux agents :
  - ✚ **Agent répertoire** c'est un agent stationnaire qui dispose d'un répertoire de base de données qui stocke les adresses de tous les fournisseurs. En fonction des demandes, l'agent répertoire sélectionne une liste de fournisseurs qui possède des informations sur le produit demandé, avec l'aide d'un agent base de données.
  - ✚ **Agent base de données** est responsable de l'accès et la récupération de données à partir de la base de données.

### **c. Le site du fournisseur**

Joue le rôle d'un marché à base d'agents, accueille et interagit avec l'agent recherche, il comprend :

- ✚ **Agent fournisseur** : est un agent stationnaire, il reçoit les agents mobiles de recherche (esclaves) qui sont envoyés par Maître Aglets et transmet les requêtes des consommateurs à l'agent base de données.
- ✚ **Agent base de données** : va générer une requête, les résultats correspondant à la demande du consommateur sont extraits de la base de données et retournés comme résultat à l'agent fournisseur.

## **5. Fonctionnement détaillé de l'approche**

Nous allons utiliser les agents mobiles parce qu'ils sont capables de chercher les informations d'une façon plus intelligente. En plus, ils sont capables de communiquer et coopérer entre eux, ce qui accélère et facilite la recherche.

Nous proposons le processus fonctionnel suivant :

- (1) En première étape, l'utilisateur initie et gère l'agent mobile de recherche et configure ses préférences via **l'agent client**. L'agent client envoie ensuite la demande de l'utilisateur au serveur d'application, l'utilisateur peut choisir les services de l'interface et envoie une demande au serveur d'application via une connexion HTTP.
- (2) En deuxième étape **Servlet** accepte la demande et communique avec **Maître Aglets**, Maître Aglets crée ensuite les agents mobiles (les aglets esclaves) selon la demande de **Servlet** et à l'aide l'agent répertoire envoie les agents mobiles à des sites de fournisseurs qui possèdent des informations sur le produit demandé, pour entreprendre la tâche de l'utilisateur.
- (3) Quand les agents mobiles arrivent aux sites fournisseurs (chaque agent pour chaque site fournisseur), chaque agent mobile de recherche envoie un appel à propositions (CFP) à l'agent fournisseur, ensuite l'agent fournisseur offre le résultat à l'agent mobile de recherche à l'aide l'agent de base de données de fournisseur.
- (4) Quand l'agent mobile de recherche recueille tout les résultats, il renvoie les résultats au serveur agent mobile. Serveur d'agent mobile, puis passe le résultat à Servlet ou envoie des SMS à l'utilisateur.
- (5) Toutes les affaires demandées par le client sont effectuées dans les sites fournisseur, chaque site de fournisseurs a des agents pour gérer les affaires de client.

Dans une demande de business mobile, les agents ne travaillent pas seuls et ils doivent communiquer avec eux pour coopérer et à générer une agrégation des données globales pour une analyse plus approfondie. Les systèmes d'agents mobiles les plus existants adoptent quelques modèles/protocoles de communication de systèmes distribués traditionnels.

Cependant, le système M-business adopte une stratégie différente de faire face à cette question. L'idée est de déployer des messagers mobiles intelligents pour la communication inter-agent.

## 5.1 La migration des agents

Lorsque l'on parle de la technologie d'agents mobiles, deux types de migration sont à considérer : la **migration forte** qui permet à un agent de se déplacer quelque

soit l'état d'exécution et de communication avec l'extérieur dans lequel il se trouve et de reprendre son exécution après la migration exactement là où elle en était avant son déplacement ; la *migration faible* ne fait que transférer avec l'agent son code et ses données. Les **Aglets** sont des agents mobiles dont la mobilité est réalisée par migration faible.

La migration forte est beaucoup plus exigeante à implanter que la migration faible, pour cela, nous avons opté pour l'utilisation de notre architecture à la migration faible afin d'augmenter la performance de notre système.

La migration des agents est aussi réactive, parce que la destination de notre agent de recherche n'est pas déterminée par l'agent lui-même, mais elle est dictée par Maître Aglets.

Un agent mobile de recherche sur des sites éloignés et à tout moment peut dépêcher des messages de fournir des données aux récepteurs désignés. Par exemple, supposons qu'un agent mobile de recherche a terminé son travail à un serveur distant, il peut migrer vers un serveur d'origine (Accueil), ou envoyer un message pour livrer le résultat à Servlet.

## **6. L'architecture des agents**

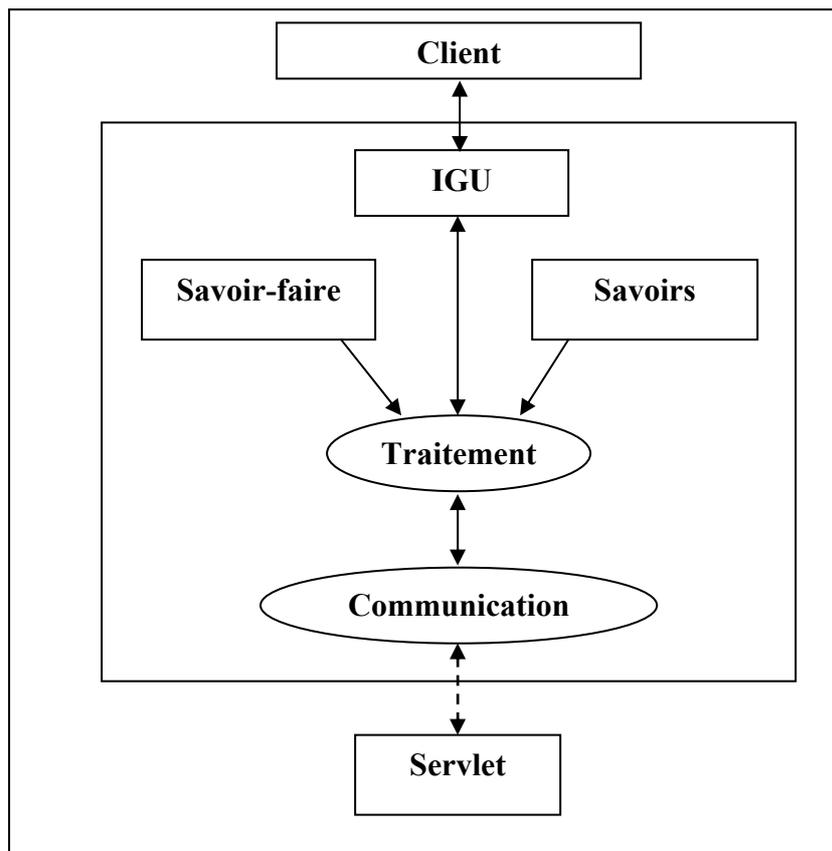
Nous passons maintenant à raffiner les architectures internes des classes d'agents proposés. Dans ce travail, on considère qu'un agent est caractérisé par des savoirs (connaissance sur son état, sur son environnement, ...), des savoir-faire (des actions ou des interactions qu'il peut mener) :

### **6.1 Agent client (AC)**

Cet agent peut être vu comme un simplificateur permettant aux utilisateurs d'interagir avec le système. C'est un agent stationnaire qui est responsable principalement à acquérir toutes les requêtes des utilisateurs, envoyer ces requêtes aux Servlet et présenter les résultats aux utilisateurs.

- **Les savoirs** : il connaît :
  - Le client
  - Servlet

- **Les savoir-faire :**
  - Il interagit avec le client.
  - Il interagit avec Servlet.
  - Il collecte les informations à partir l'interface graphique.
  - Il crée la requête et l'envoi à Servlet.
  - Il renvoie la facture au client.
- **Rôle :** satisfaire la demande du client.

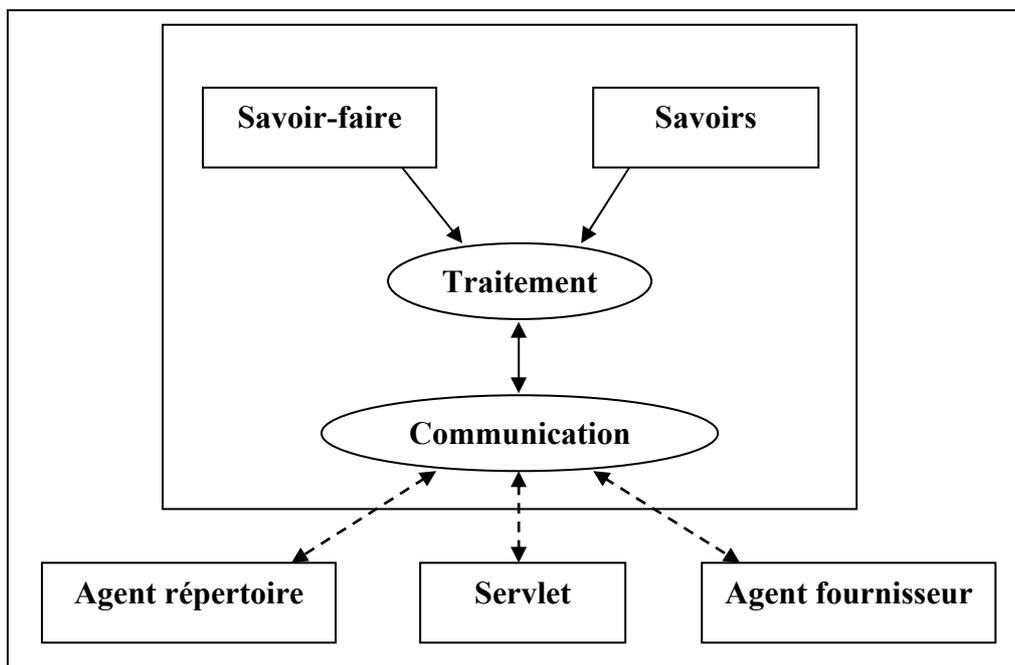


**Figure 4.4:** Architecture interne de l'agent client.

## 6.2 Maître aglet (MA)

- **Les savoirs :** il connaît :
  - Agent répertoire
  - Servlet
  - Les Agents mobiles de recherches.
- **Les savoir-faire :**
  - Il reçoit et analyse les messages de Servlet.
  - Il établit un Message de demande des adresses fournisseurs et l'envoi à l'agent répertoire.

- Il reçoit et analyse le Message de réponse de l'agent répertoire.
  - Crée les agents esclaves mobiles.
  - Cloner l'agent esclave mobile selon le nombre de fournisseurs dans le Message de réponse, et les envoie aux fournisseurs.
  - Il reçoit les résultats à partir des agents mobiles de recherche (esclaves).
- **Rôle** : satisfaire la demande reçue, par l'interaction avec l'agent répertoire et les agents fournisseurs.



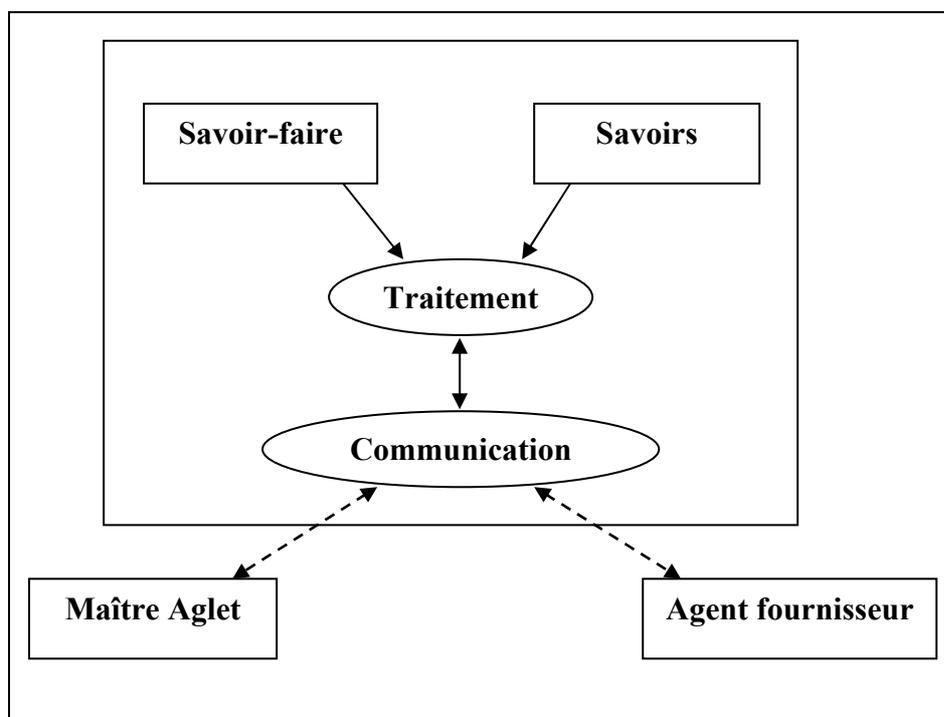
**Figure 4.5:** architecture interne de Maître Aglet.

### 6.3 Les Agents Mobile de Recherche (AMR)

Ces agents de recherche sont des entités qui se déplacent d'une machine à l'autre sur le réseau afin de satisfaire les besoins de leurs clients. Sur chaque machine, ils effectuent des échanges et traitent les informations collectées. Ce traitement permet de réduire la quantité d'informations transportées avec l'agent et par conséquent le trafic sur le réseau. L'agent transporte avec lui les données demandées par Maître Aglets.

- **Les savoirs** : ils connaissent :
- Maître Aglet

- Agent fournisseur
- **Les savoir-faire :**
  - Ils se déplacent vers les sites fournisseurs.
  - Ils soumission la demande à des agents fournisseurs.
  - Ils interagissent avec les agents fournisseurs.
  - Ils reçoivent les résultats à partir des agents fournisseurs.
  - Ils envoient les résultats à Maître Aglet.
- **Rôle :** livraison les demandes à fournisseurs, et recueil les résultats.



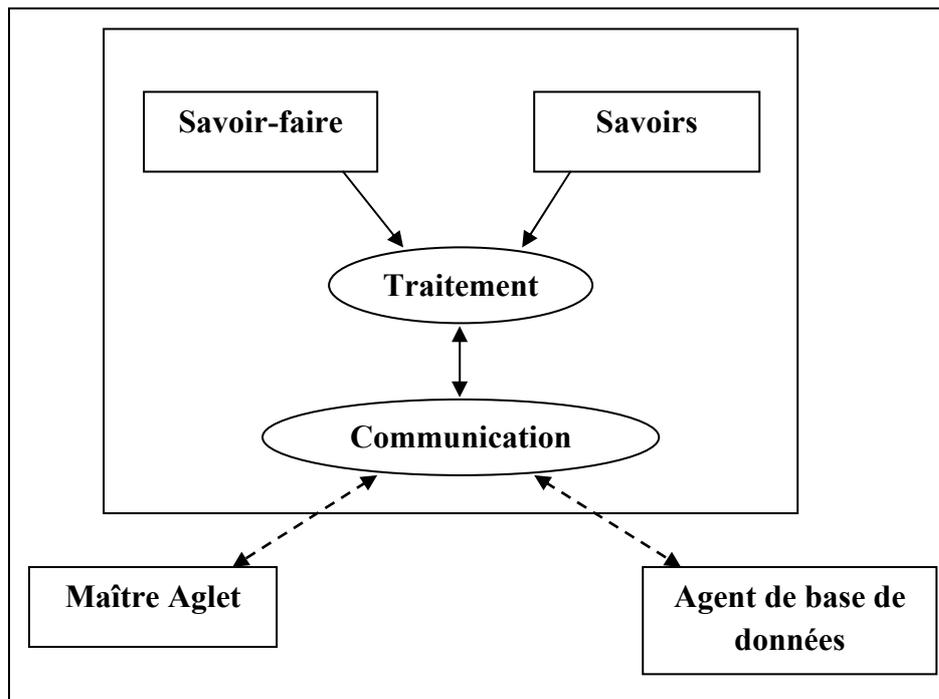
**Figure 4.6:** Architecture interne des agents mobiles de recherche.

#### 6.4 Agent Répertoire (AR)

C'est un agent stationnaire, il agit localement pendant tout son cycle de vie, dans la machine là où il a été implanté initialement.

- **Les savoirs :** il connaît:
  - Maître Aglet
  - Agent de base de données (répertoire).

- **Les savoir-faire :**
  - Il reçoit le Message de demande de Maître Aglet.
  - Il interagit avec l'agent de base de données.
  - Il reçoit la liste de fournisseurs qui possèdent des informations sur le produit demandé à partir l'agent de base de données.
  - Il envoie la liste de fournisseurs qui possèdent des informations sur le produit demandé à Maître Aglet.
- **Rôle :** en fonction des demandes, l'agent répertoire sélectionne une liste de fournisseurs qui possèdent des informations sur le produit demandé, avec l'aide d'un agent base de données.

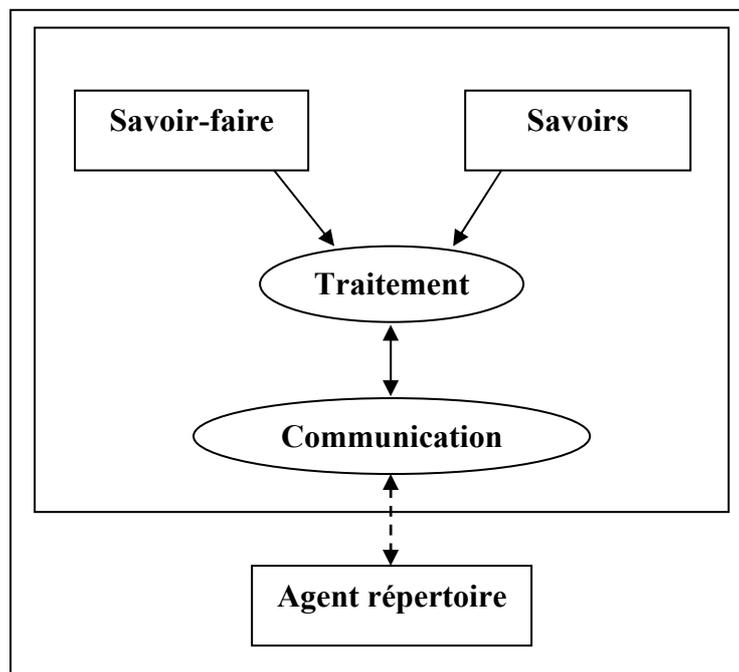


**Figure 4.7:** architecture interne d'agent répertoire.

## 6.5 Agent de base de données (répertoire) (ABDR)

- **Les savoirs :** il connaît:
  - Agent répertoire.
- **Les savoir-faire :**
  - Il consulte la base de données.

- Il interagit avec l'agent répertoire.
  - Il envoie la liste de fournisseurs (les adresses) qui possèdent des informations sur le produit demandé à l'agent répertoire.
  - Il reçoit la demande de la liste de fournisseurs qui possèdent des informations sur le produit demandé à partir d'agent répertoire.
- **Rôle** : l'accès et la récupération des données à partir de la base de données.



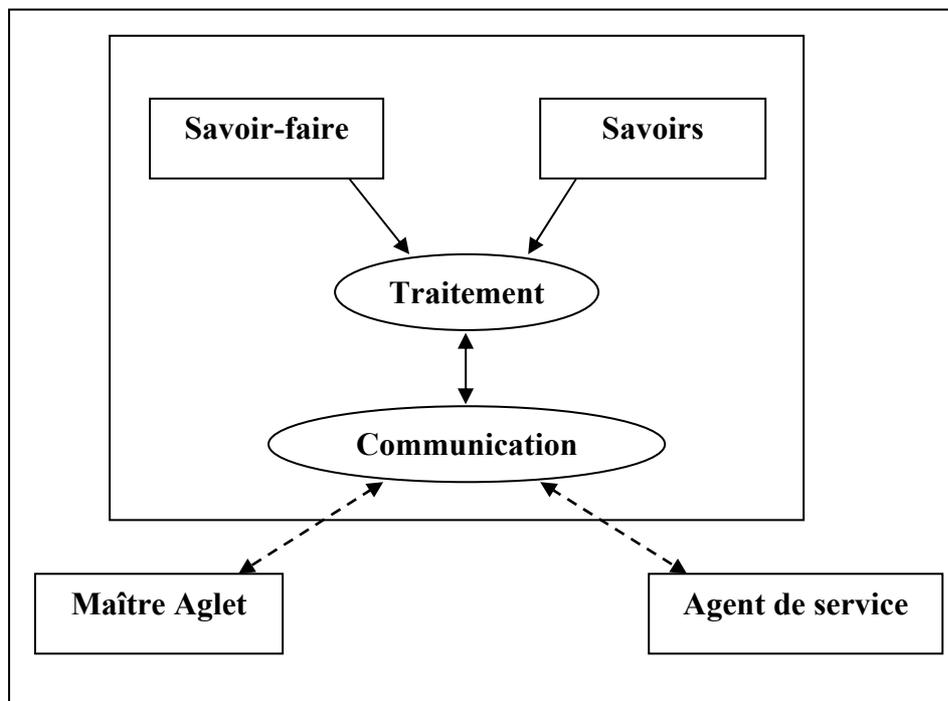
**Figure 4.8:** architecture interne d'agent de base de données (répertoire).

## 6.6 Agent fournisseur (AF)

- **Les savoirs** : il connaît:
  - Agents mobiles de recherche.
  - Agent de base de données (fournisseur).
- **Les savoir-faire** :
  - Il reçoit les informations sur le produit demandé à partir d'agent mobile de recherche.
  - Il reçoit les agents mobiles de recherche (esclaves) qui sont envoyés par Maître Aglet.

- Il interagit avec l'agent mobile de recherche.
- Il envoie la demande à l'agent de base de données (fournisseur).
- Il interagit avec l'agent de base de données (fournisseur).
- Il reçoit le résultat à partir d'agent de base de données (fournisseur).
- Il envoie le résultat sur le produit demandé à agents mobiles de recherche.

➤ **Rôle** : Offrir le résultat sur le produit demandé.

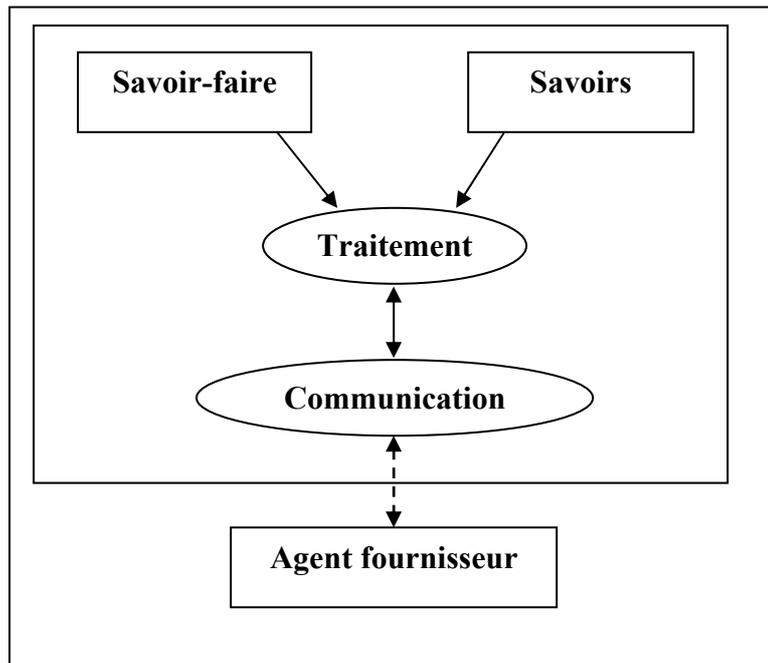


**Figure 4.9:** architecture interne d'agent fournisseur.

### 6.7 Agent de base de données (fournisseur) (ABDF)

- **Les savoirs** : il connaît:
  - L'agent fournisseur.
- **Les savoir-faire** :
  - Il consulte la base de données.
  - Il interagit avec l'agent fournisseur.
  - Il envoie le résultat sur le produit demandé à l'agent fournisseur.

- Il reçoit la demande sur le produit demandé à partir d'agent fournisseur.
- **Rôle** : l'accès et la récupération des données à partir de la base de données.



**Figure 4.10:** Architecture interne d'agent de base de données (fournisseur).

## **7. Les diagrammes AUML**

Nous utilisons AUML (Agent-base Unified Modeling Language) qui est une variante d'UML pour modéliser les interactions entre les agents. AUML est un langage de modélisation des systèmes multi-agent. D'une part, les agents sont actifs, ils sont capables de prendre des initiatives et peuvent contrôler la communication entre eux. D'une 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,

ensuite pour chaque cas d'utilisation un diagramme de séquence sera associé. Nous enrichissons notre conception par le diagramme de classes du système.

### 7.1 Diagramme de cas d'utilisation

Les cas d'utilisation (use cases) permettent de modéliser les besoins des clients d'un système et doivent aussi posséder ces caractéristiques. Ils ne doivent pas chercher l'exhaustivité, mais clarifier, filtrer et organiser les besoins.

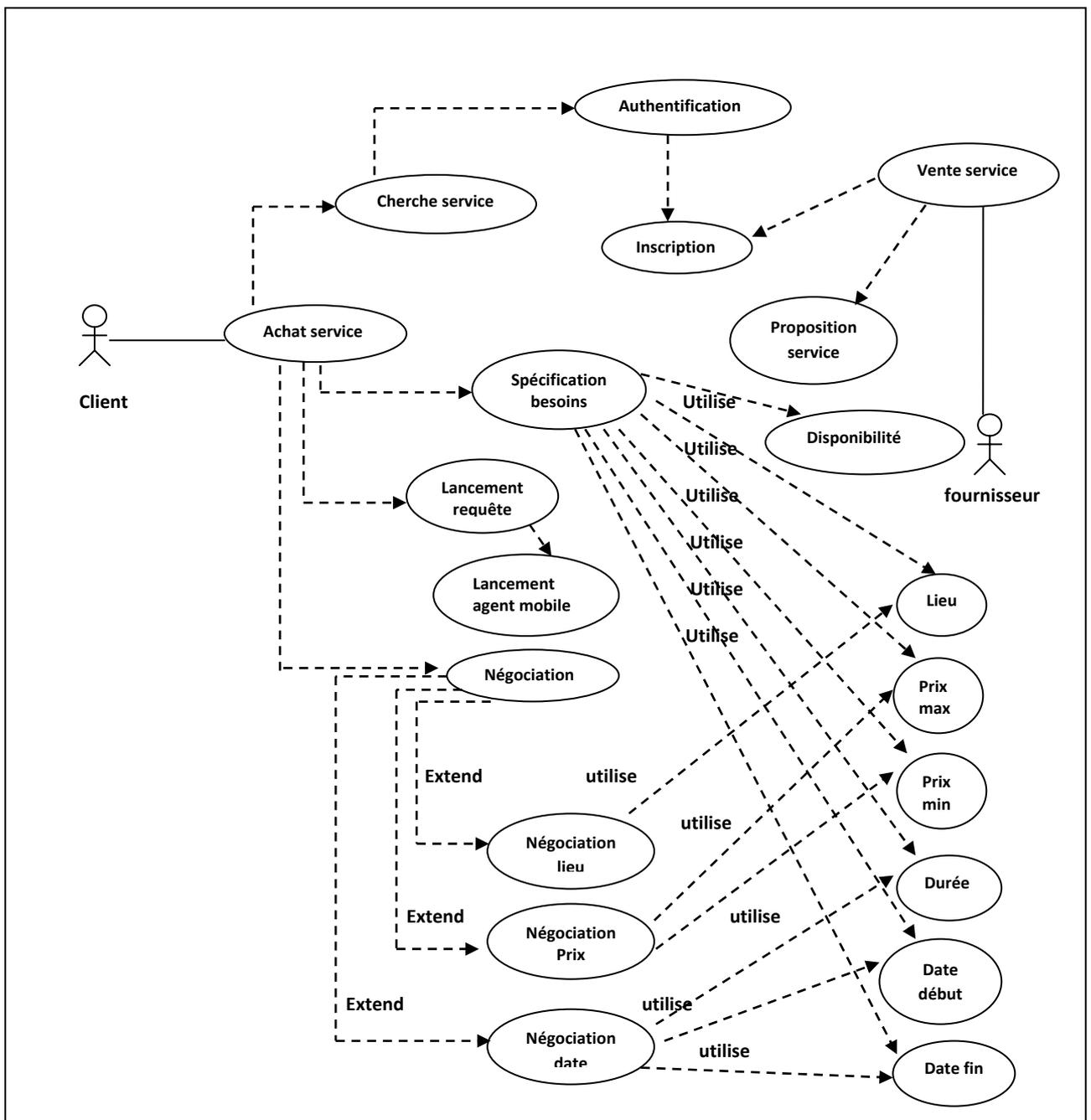


Figure 4.11: Diagramme de cas d'utilisation du système.

Les acteurs principaux sont :

- **Client** : demandeur du service, c'est la personne qui visite le site pour faire la recherche d'un service.
- **Fournisseurs** : fournisseur du service, c'est la personne qui visite le site pour faire une proposition de service.

Les cas d'utilisation : On distingue deux grands cas d'utilisation :

- **Achat service** : l'expression du besoin du client.
- **Vente service**: l'expression des propositions du fournisseur.

## 7.2 Diagramme de séquences

Les diagrammes de séquences permettent de présenter les interactions entre agents avec une dimension temporelle et présenter les messages échangés.

### 7.2.1 Diagramme de séquence (inscription)

Pour fournisseur :

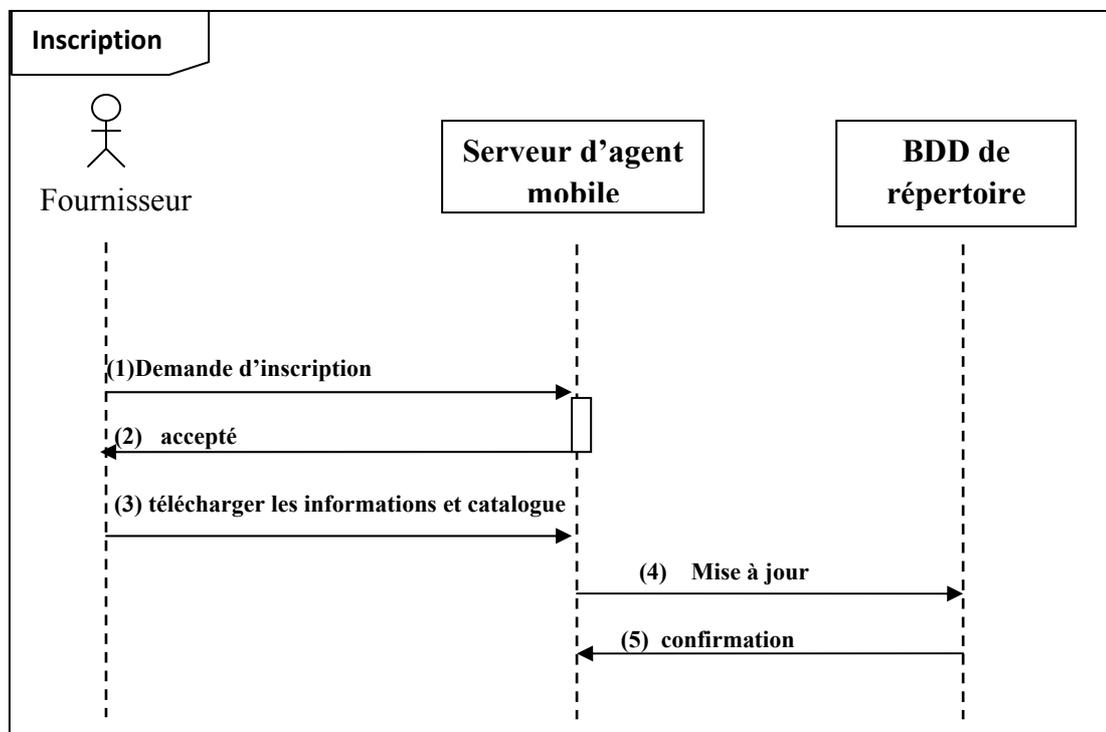


Figure 4.12: Diagramme de séquence « Inscription fournisseur »

Pour client :

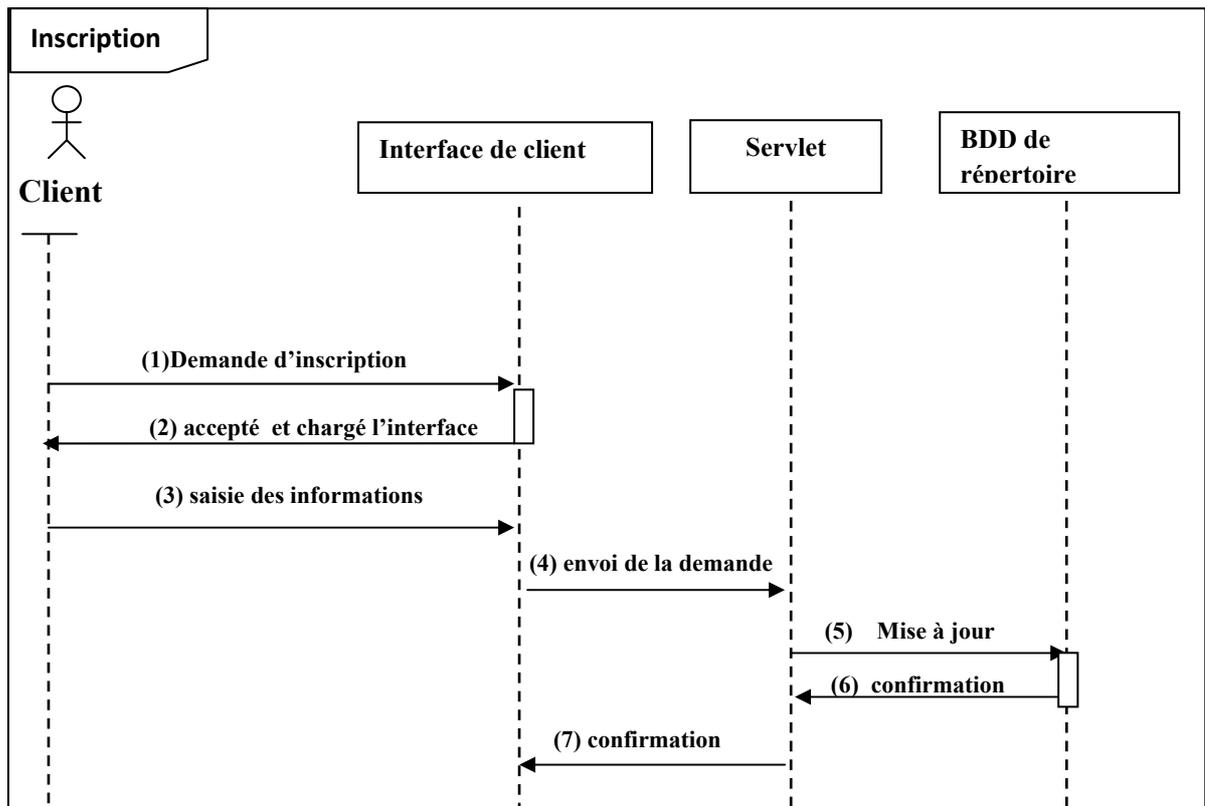


Figure 4.13: Diagramme de séquence « Inscription client »

### 7.2.2 Diagramme de séquence (Authentification)

Le client qui désire utiliser le système doit s'authentifier en saisissant son identifiant et son mot de passe :

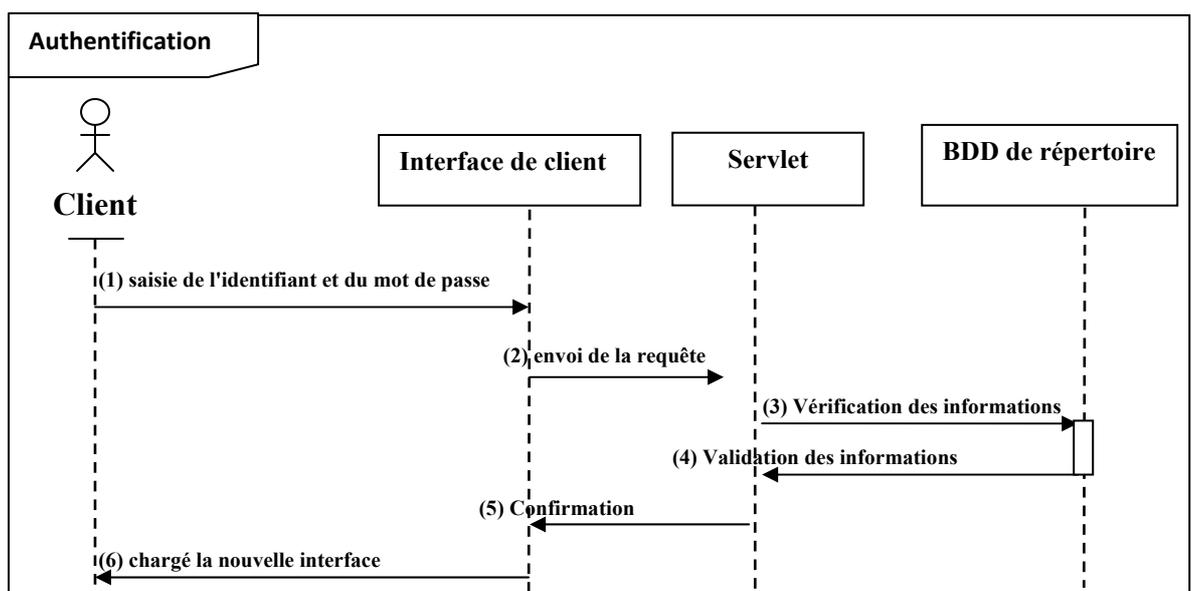


Figure 4.14: Diagramme de séquence « Authentification client »

### 7.2.3 diagramme séquence « Lancement requête »

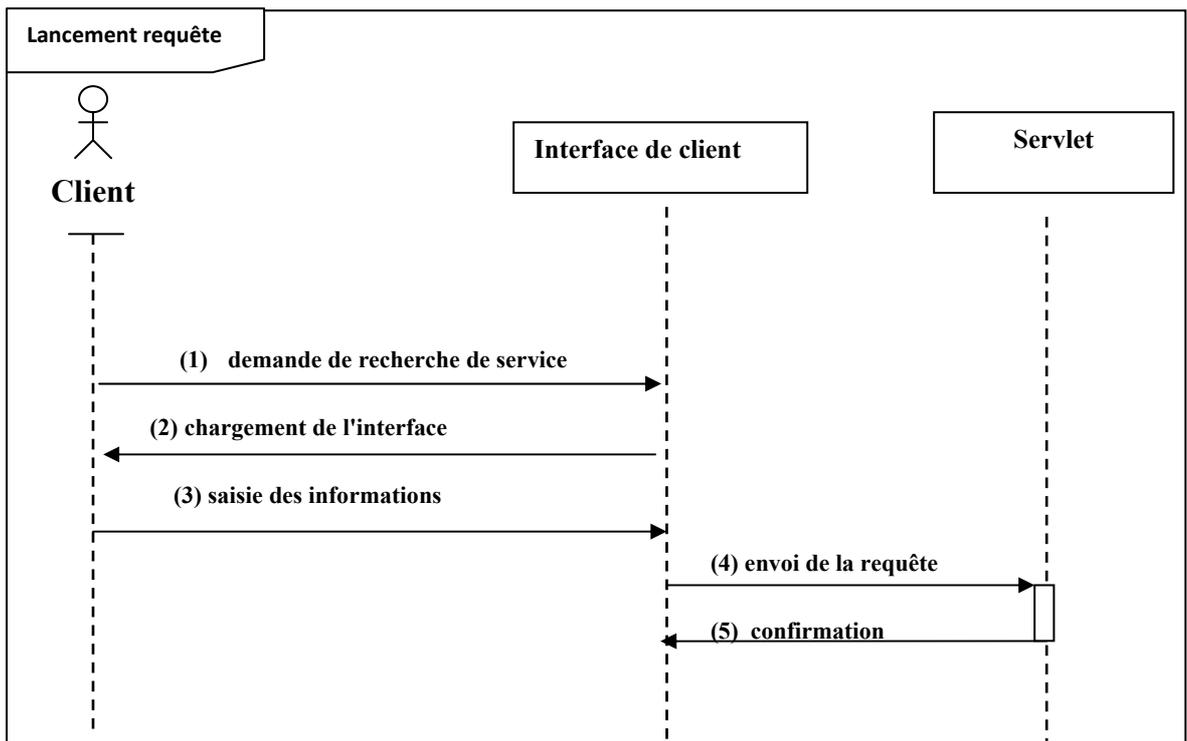


Figure 4.15: Diagramme de séquence « Lancement requête»

## 8. Protocole d'interaction du système

Nous présentons ci-dessous les protocoles d'interaction entre les agents, nous modélisons les différentes interactions par des diagrammes de séquence AUML.

Notre système est basé sur l'interaction de 07 agents :

- Agent client : AC
- Agent répertoire : AR
- Agent base de données de répertoire : ABDR
- Maître Aglet : MA
- Les agents mobiles de recherche (esclaves) : AMR
- Agent fournisseur : AF
- Agent base de données de fournisseur : ABDF

## 8.1 Lancement agent

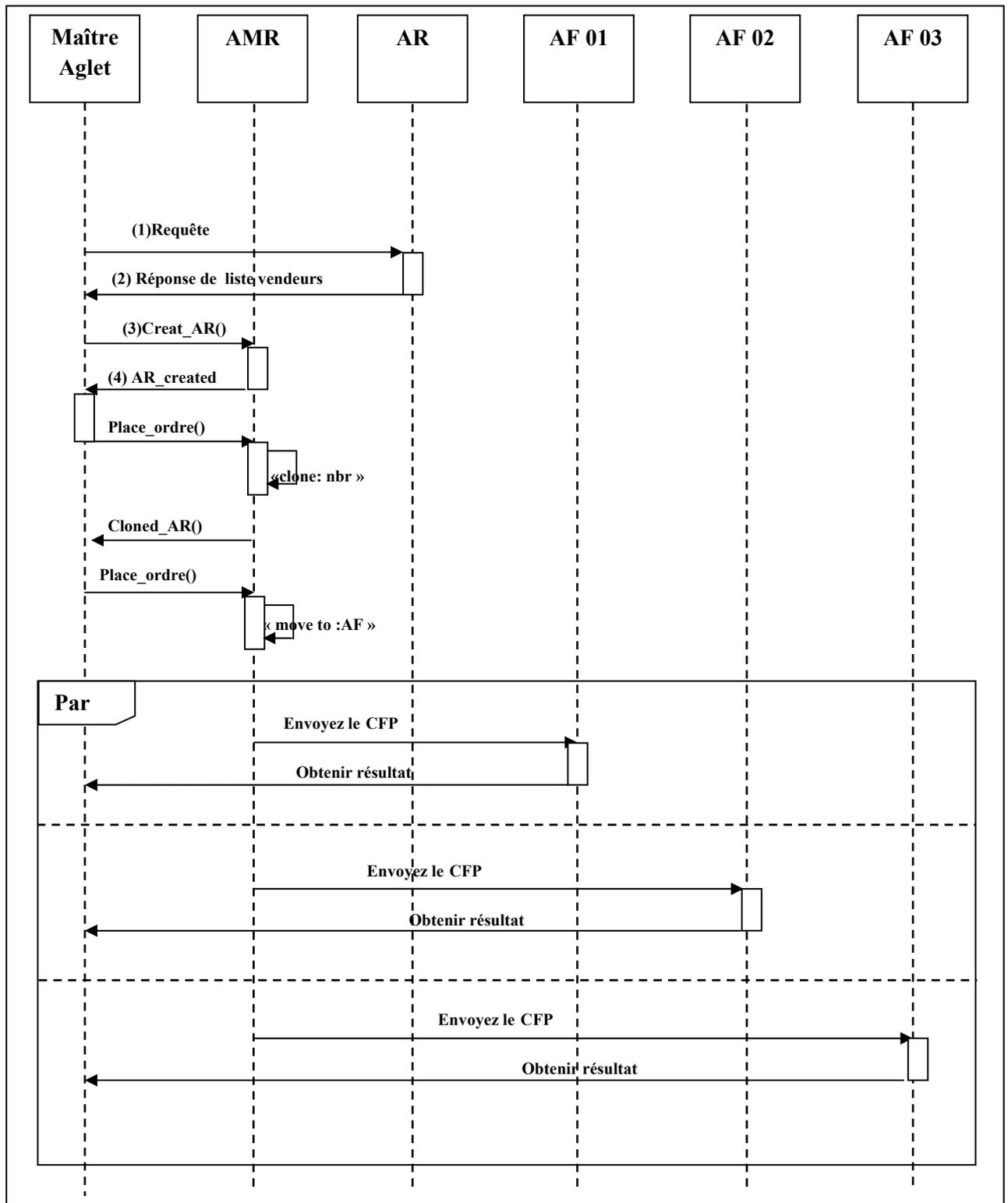
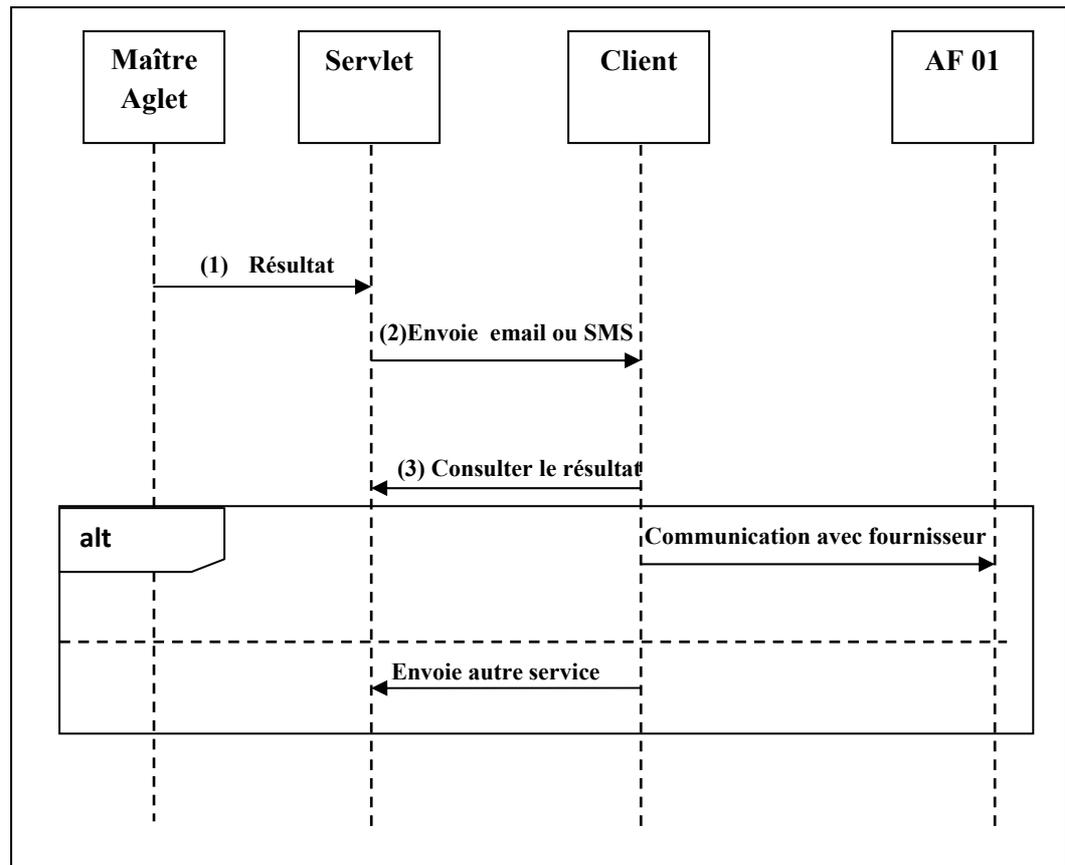


Figure 4.16: Diagramme de séquence « Lancement agent »

**Pour informer le client :**

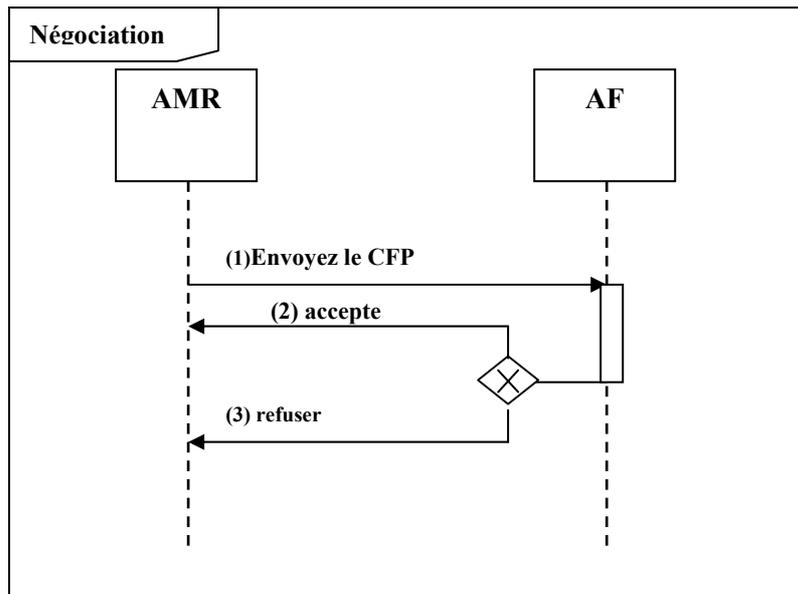


**Figure 4.17:** Diagramme de séquence pour informer client.

## 8.2 Négociation

Le protocole réseau contractuel (Contract Net) a été une des premières approches utilisées dans les systèmes multi-agents pour résoudre le problème d'allocation des tâches, les agents coordonnent leurs activités grâce à l'établissement de contrats afin d'atteindre des buts spécifiques.

Ce protocole permet à un agent d'achat (l'initiateur) d'envoyer un appel à propositions (CFP) à un ensemble d'agents des ventes (des répondants), évaluer les propositions des agents des ventes et accepter ensuite le plus préféré (ou les refuser tous).



**Figure 4.18:** Diagramme de séquence « négociation »

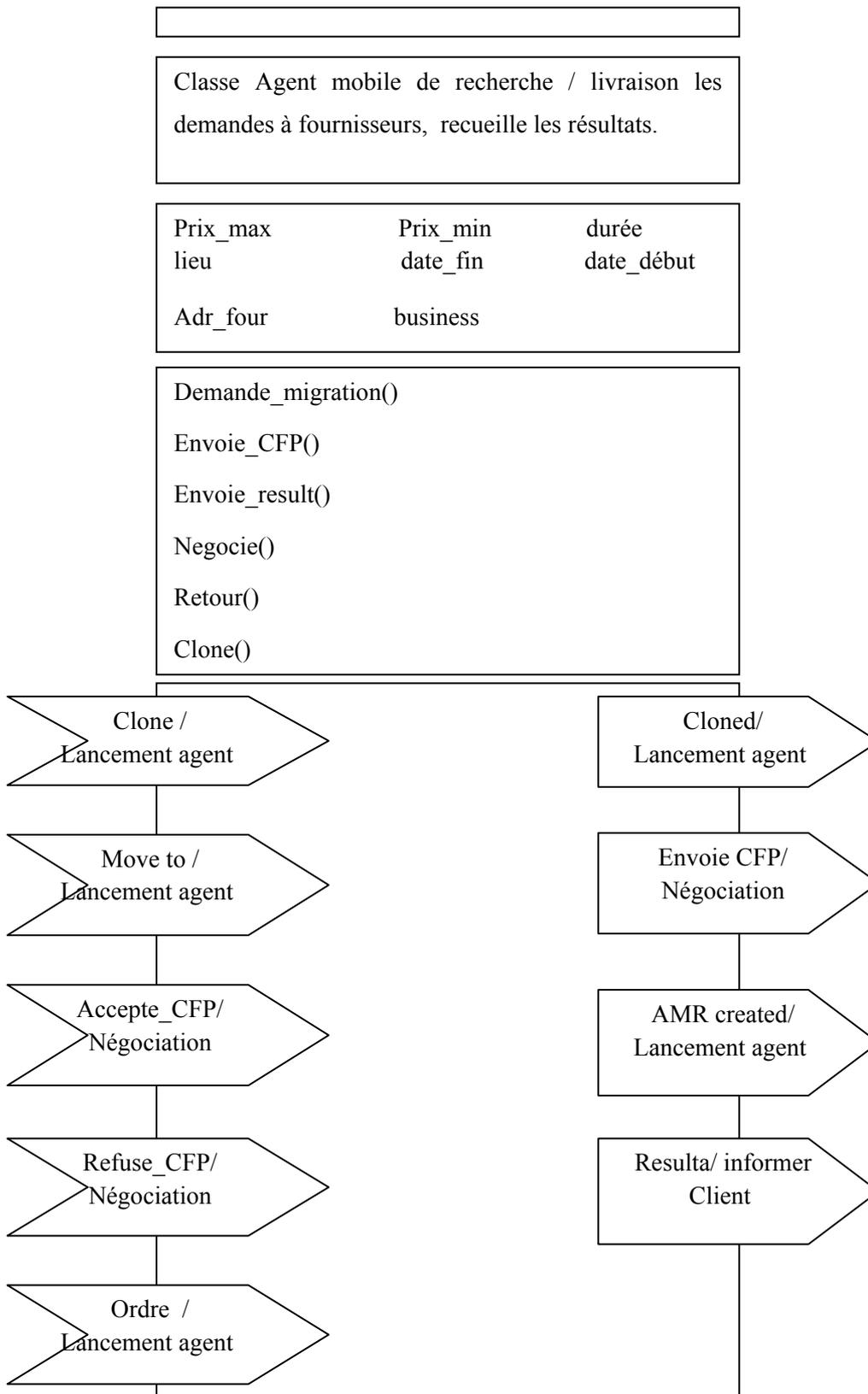
## 9. Diagramme de classes

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des agents du système qui vont interagir ensemble pour réaliser les cas d'utilisation.

### 9.1 Les classes d'agents

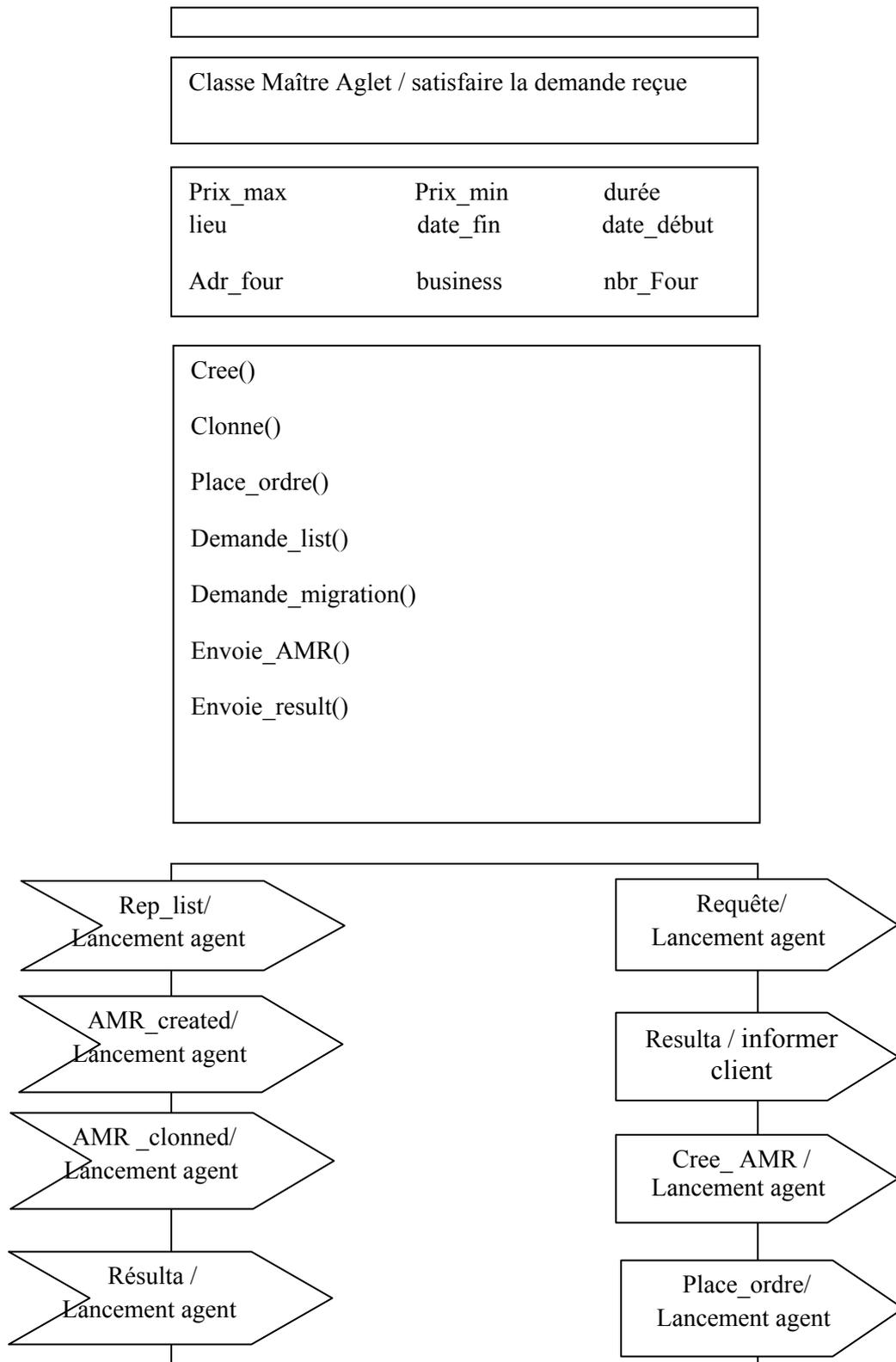
Dans cette section, nous allons présenter chaque classe d'agent avec un diagramme de classe en AUML. Ceci pour pouvoir montrer les états internes, les actions, les messages entrants et sortants de chaque agent qui sont des avantages d'AUML. Notre système est composé des agents suivants :

### 1- Agent mobile de recherche AMR :



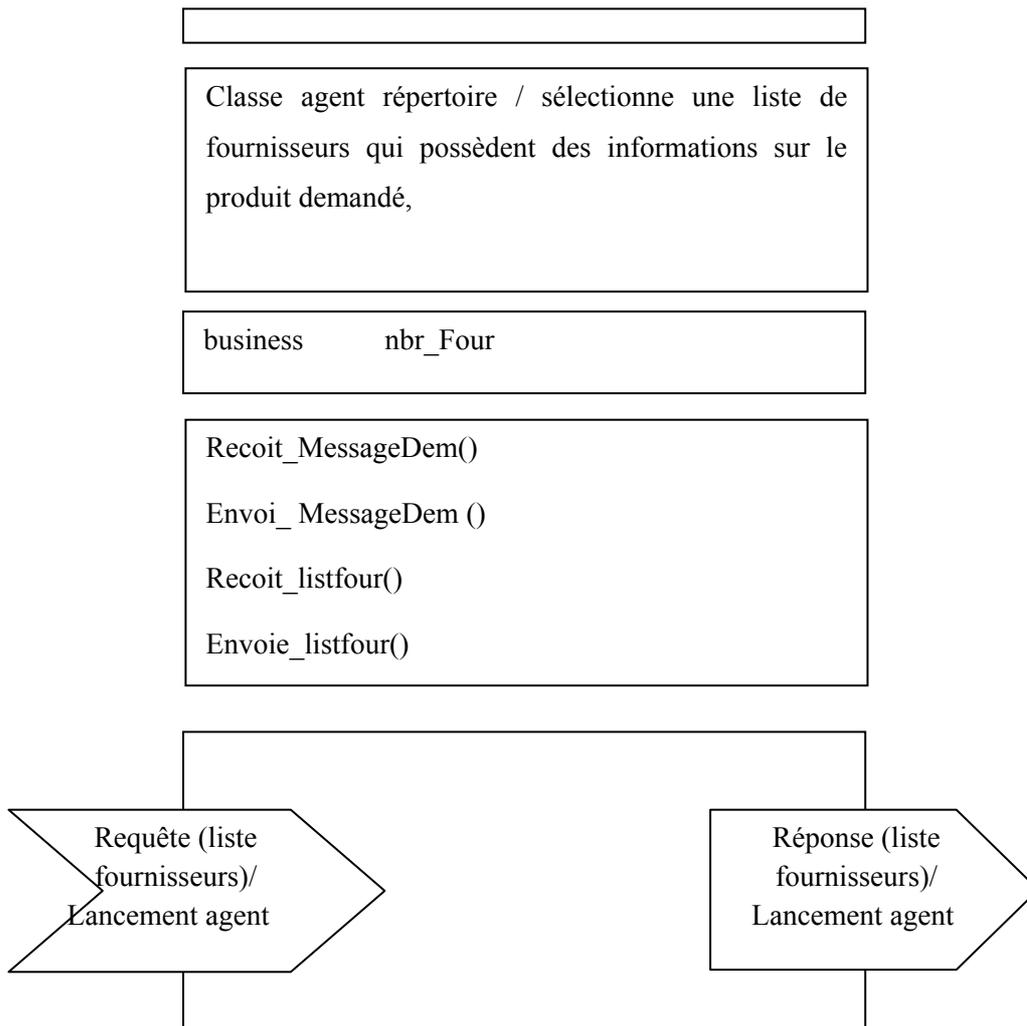
**Figure 4.19:** Diagramme de classe de l'agent mobile de recherche

## 2- Maître Aglet (MA) :



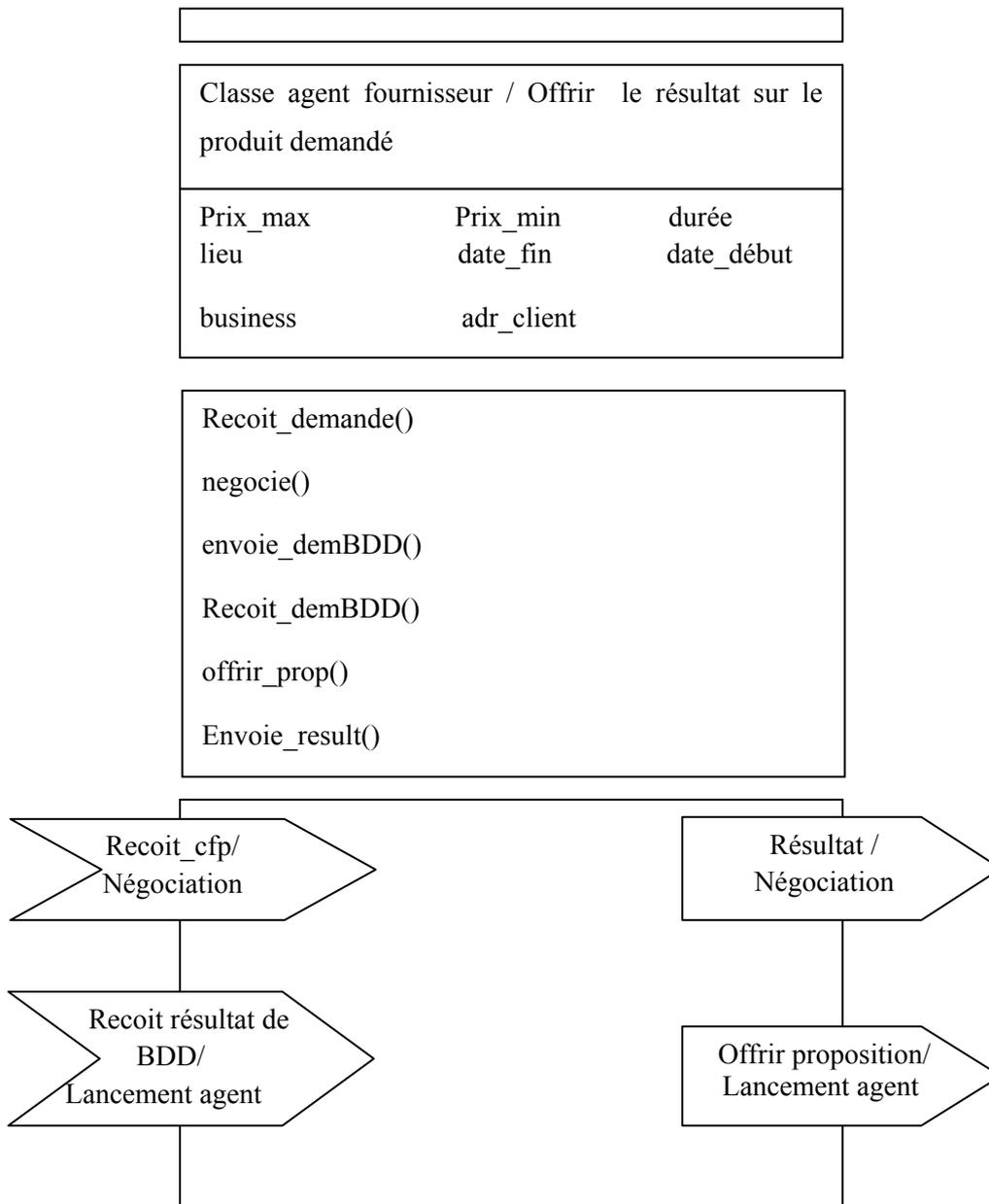
**Figure 4.20:** Diagramme de classe de Maître Aglets.

### 3- Agent répertoire (AR) :



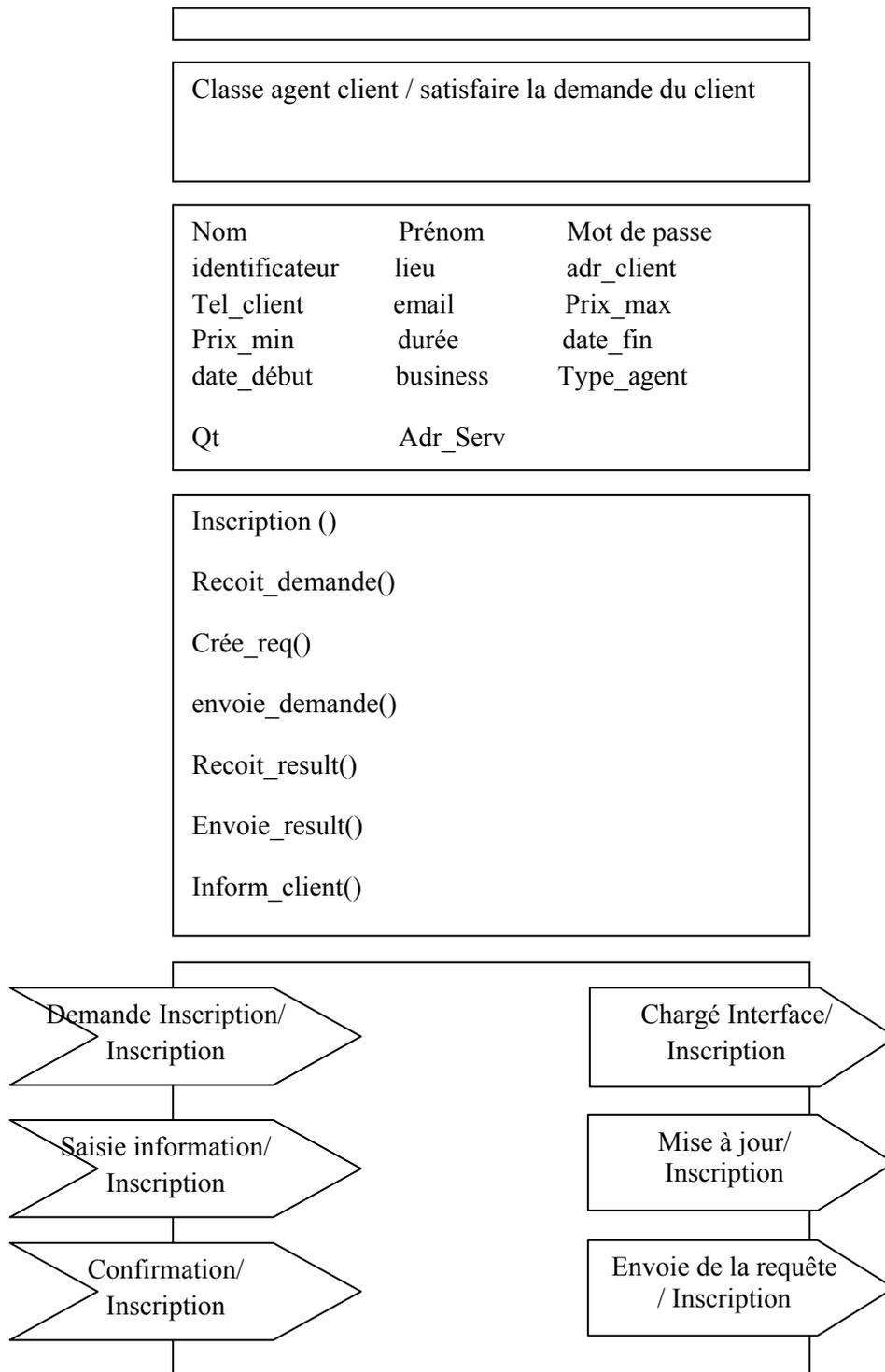
**Figure 4.21:** Diagramme de classe d'agent répertoire.

#### 4- Agent fournisseur (FA) :



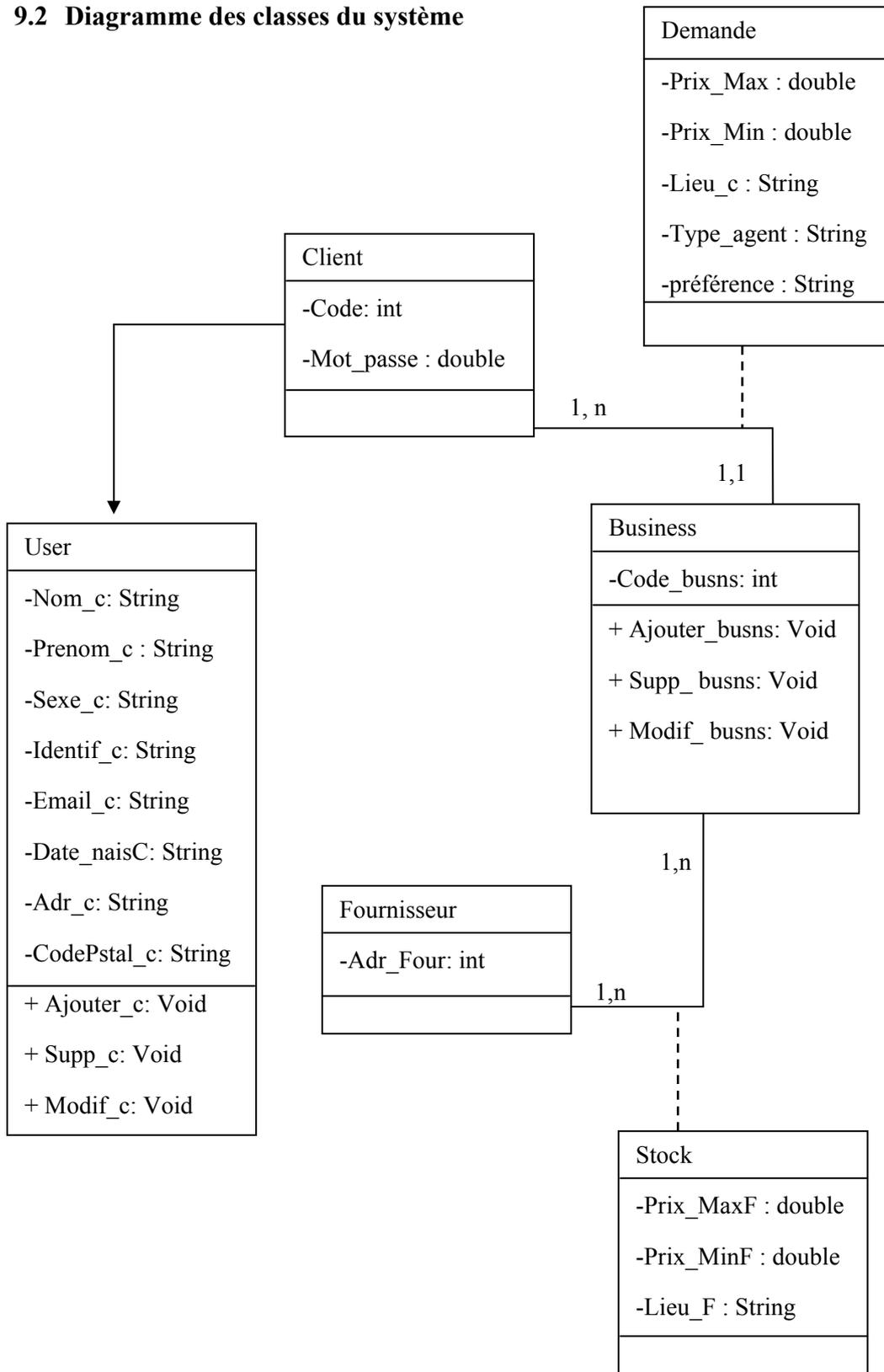
**Figure 4.22:** Diagramme de classe d'agent fournisseur.

## 5- Agent Client (CA) :



**Figure 4.23:** Diagramme de classe d'agent Client.

## 9.2 Diagramme des classes du système



**Figure 4.24:** Diagramme de classe du système.

## **10. Conclusion**

Dans ce chapitre, nous avons présenté une architecture basée agent mobile pour M-business. Cette architecture comporte les concepts nécessaires pour assurer les exigences que nous avons prises en compte afin d'assurer le bon fonctionnement du système proposé. Alors les structures internes des différents agents sont illustrées, les mécanismes d'interaction, de migration et de communication sont discutés.

Néanmoins, une étude de cas est nécessaire pour l'évaluation des différentes idées dans un environnement réel. Cette étude de cas va permettre d'aborder la phase d'implémentation. Cette dernière phase fera l'objet du chapitre suivant pour la validation de notre approche.

# Chapitre V

---

## Etude De Cas Et Implémentation

---

### 1. Introduction

Dans le précédent chapitre de ce mémoire, nous avons proposé une architecture basée agents mobile pour le business mobile, 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 idée, nous arrivons dans ce chapitre à la mise en place de notre application.

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.

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

### 2. Environnement de développement

#### 2.1 Langage de programmation

Nous avons implémenté notre système avec le langage JAVA qui est un langage de programmation développé par Sun Microsystems. Les premières versions datent de 1995, il a réussi à intéresser beaucoup de développeurs à travers le monde. C'est aussi un langage multi plate formes disposant de JVM (Java Virtual Machine) lui permettant de s'exécuter dans des environnements hétérogènes en permettant une indépendance envers les réseaux et les systèmes d'exploitation.

Le langage Java a la particularité principale d'être portable sur plusieurs systèmes d'exploitation tels que Windows, MacOS ou Linux. C'est la plateforme qui garantit la portabilité des applications développées en Java.

## 2.2 Les servlets

Un servlet est une classe du langage de programmation de Java utilisé pour étendre les fonctionnalités de serveurs qui hébergent des applications accessibles via un modèle de programmation de requête-réponse. Ce programme Java s'exécute dynamiquement sur le serveur Web et permet l'extension des fonctions de ce dernier, typiquement : accès à des bases de données, transactions commerce, etc.

Servlets ne sont pas conçus pour des protocoles spécifiques. Ils sont les plus couramment utilisés avec les protocoles HTTP, Servlets utilisent les classes dans les packages Java du javax.servlet et javax.servlet.http [Jas 98].

HTTP Servlet est généralement utilisé pour :

- Fournir du contenu dynamique comme obtenir les résultats d'une requête de base et le retour au client.
- Traiter et stocker les données soumises par le html.
- gérer les demandes pour de multiples clients simultanés.

## 2.3 J2ME

Sun a proposé plusieurs plates-formes pour le développement d'application sur des machines possédant des ressources réduites. En 1999, Sun propose de mieux structurer ces différentes plates-formes sous l'application J2ME (Java 2 Micro Edition) pour périphérique mobiles et embarqués. Des plates-formes Java compatibles avec J2ME sont embarquées dans de nombreux téléphones portables et PDA [Joh 01].

### 2.3.1 L'architecture de J2ME

L'architecture de j2me est composée de trois couches:

- 01- La première couche est la couche de Configuration qui inclut la JVM, qui interagit directement avec le système d'exploitation. La couche de Configuration gère également l'interaction entre le profil et la JVM. La couche de configuration définit les caractéristiques de bases d'un environnement d'exécution pour un certain type de machine possédant un ensemble de caractéristiques et de ressources similaires. Elles se composent d'une machine virtuelle et d'un ensemble d'API (Application Programming Interface) de base. Dans un sens, une configuration définit les éléments communs des

caractéristiques plateformes Java et les bibliothèques que les développeurs peuvent assumer d'être disponibles sur tous les appareils appartenant à une catégorie particulière.

Les configurations les plus courantes sont [Joh 01]:

- **CLDC** (Connected Limited Device Configuration), que l'on retrouve par exemple dans les téléphones mobiles.
- **CDC** (Connected Device Configuration), qui est plutôt utilisé dans des décodeurs de télévision numérique.

02- La deuxième couche est la couche de profil: la couche de profil définit l'ensemble minimal d'interfaces de programmation d'applications (API) disponibles sur une famille particulière d'appareils. Les profils sont mis en œuvre sur une configuration particulière. Les applications sont écrites pour un profil particulier et sont donc portables à n'importe quel dispositif qui supporte ce profil. Un dispositif peut supporter plusieurs profils. (Profile=ensemble d'Api minimum pour des appareils similaires).

Les profils les plus courants sont [Joh 01]:

- **MIDP** (Mobile Information Device Profile) est un profile standard défini pour un ensemble de machines embarquées possédant des ressources et une interface graphique limitée. Toutes les applications MIDP doit dériver d'une classe spéciale Classe MIDlet, Le cycle de vie d'une MIDlet est semblable à celui d'une applet.
- **DoJa**, développé par NTT DoCoMo pour les téléphones i-mode J2ME

03- La troisième couche est MIDP (Mobile Information Device profile). La couche MIDP contient des API de java pour les connexions de réseau d'utilisateur, le stockage de persistance et l'interface utilisateur. Il a aussi l'accès aux bibliothèques CLDC et des bibliothèques MIDP.

- ❖ **Couche JVM** (Java Virtual Machine) : Cette couche est une implémentation d'une Java Machine virtuelle est personnalisé pour le système d'exploitation d'un périphérique particulier, qui prend en charge une configuration particulière de J2ME.

## **2.4 MIDlet**

Les applications créées avec MIDP sont des midlets : ce sont des classes qui héritent de la classe abstraite `javax.microedition.midlet.Midlet`. Cette classe permet le dialogue entre le système et l'application [Joh 01].

## **2.5 Serveur de base de données : MySQL**

C'est la base de données de nombreuses applications web comme les gestionnaires de contenu. MySQL est un système de base de données utilisé sur le Web. Essentiellement, une base de données MySQL vous permet de créer une structure de base de données relationnelle sur un serveur Web quelque part pour stocker des données ou automatiser des procédures.

MySQL fonctionne sur beaucoup de plates-formes différentes, incluant Unix, Windows 95, 98, NT, 2000 et XP [Web 03].

## **2.6 JADE LEAP**

À la fin de l'année 1999, un groupe constitué des acteurs principaux du secteur de la télécommunication mobile se crée pour développer une plateforme conforme aux normes FIPA pouvant être utilisée sur les supports mobiles. Début 2000, un projet nommé LEAP (Lightweight Extensible agent Platform) est créé. Celui-ci a comme but premier la création d'un middleware suffisamment léger pour les supports ayant des contraintes avec leurs ressources. Ces contraintes permettent de choisir la plateforme JADE comme une plateforme de base et modifient certaines parties du noyau, à la fin de ce travail ils obtenu une extension de JADE appelé JADE-LEAP.

Parce que l'Aglets est lourd et on ne peut pas utiliser dans les MIDlet on va utiliser le JADE-LEAP dans cette partie.

## **2.7 Serveur Web**

Apache Tomcat est une implémentation des technologies Java Servlet et JavaServer Pages. Le serveur Tomcat est un serveur Open Source qui agit comme un conteneur de servlet J2EE, et permettant de compiler et d'exécuter les Servlets et les JSP. Il contient une machine virtuelle java et des éléments associés pouvant fournir un environnement de développement pour java. Donc, Tomcat implémente les

spécifications des servlets et des JSP de Sun Microsystems. Comme Tomcat celui-ci inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP.

Tomcat a été écrit en langage Java, il peut donc s'exécuter via la JVM (machine virtuelle java) sur n'importe quel système d'exploitation.

## **2.8 NetBeans IDE 6.7.1**

NetBeans est un environnement de développement intégré (IDE) développé par Sun. Il est disponible pour Windows, Mac, Linux et Solaris. Le projet NetBeans est une plateforme d'application qui permet aux développeurs de créer rapidement des pages Web et des applications mobiles utilisant la plateforme Java, il fournit également une meilleure intégration avec le serveur Oracle WebLogic, ainsi que le support pour Oracle Database et GlassFish 3.1 et le support d'édition HTML5.

## **2.9 Choix de la plate-forme**

Actuellement, 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.

### **2.9.1 Description générale de la plate-forme Aglets**

Aglets Software Development Kit est un environnement pour programmer les agents Internet en Java, développé par une équipe de chercheurs du laboratoire de recherche d'IBM Tokyo au Japon. Ce nom a été créé à partir des mots Agent et Applet, ce qui exprime assez clairement ce qu'est Aglet. La distribution d'Aglet est fournie avec un serveur appelé **Tahiti**.

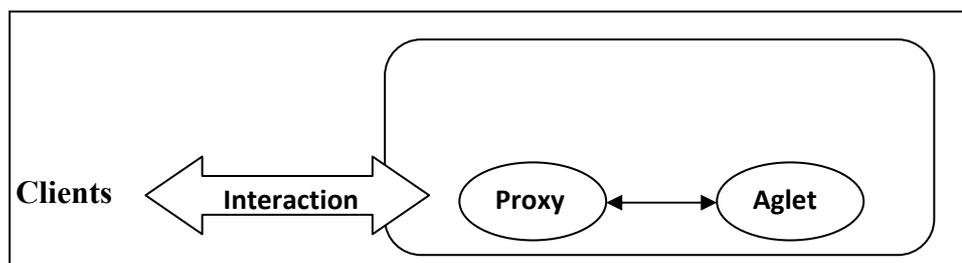
La plate-forme Aglet est une technologie d'agent mobile de Java pure de 100 %, Aglets sont des objets Java peuvent se déplacer d'un hôte à un autre. L'API Java d'Aglet (J-AAPI) est l'interface de construire des aglets et leur environnement. J-AAPI définit les méthodes pour la création d'aglet, manipulation de message, envoi, rétraction, désactivation/activation, clonage et en éliminant l'aglet. Cette API est indépendante de la plate-forme et il a besoin de JDK ou plus tard à exécuter.

Comme toutes les mises en œuvre d'agent, le système Aglets a un serveur d'agent appelé «Tahiti ». Par défaut, il commence à utiliser le port 434 pour des communications, mais il peut être modifié. Pour communiquer entre les serveurs, il utilise le Protocole de Transfert d'Agent (ATP) qui offre le protocole indépendant à plate-forme pour transférer des agents entre des ordinateurs en réseau. Le but de l'ATP est d'offrir un protocole indépendant simple et de la plate-forme pour le transfert des agents entre les ordinateurs en réseau. ATP offre également la possibilité de gérer la mobilité de l'agent d'une manière générale et uniforme, indépendamment de la langue de mise en œuvre d'agent et plate-forme de fournisseur spécifique.

### 2.9.1.1 Les éléments de base d'Aglets

Voici les éléments de base d'Aglets :

- **Proxy.** Représente l'Aglet. Protège la méthode publique de l'Aglets. Il fournit la transparence de l'emplacement pour l'Aglet. Proxy de l'Aglet ne se déplace pas.
- **Aglets** sont des agents mobiles. Ils sont mis en œuvre en Java, réactif à son environnement.
- **Identité:** chaque Aglet a un ID unique.
- **Contexte.** la place de l'Aglet ou environnement. Il fournit des services pour l'Aglet et protège l'hôte contre les Aglets malveillants. Les contextes sont nommés et peuvent ainsi être localisés par leur hôte, port et le nom (Comme illustré à la Figure5.1).



**Figure 5.1:** Relation entre un Aglet et son Proxy.

### 2.9.1.2 Opérations fondamentales

Le comportement supporté dans le modèle d'aglets inclut création, cloning, dispatching, retraction, deactivation, activation, disposal et messaging:

➤ **Création : createAglet()**

L'aglet est chargé dans un contexte. Il est initialisé et commence immédiatement à exécution.

➤ **Clonage : oncloning()**

L'API de Aglet fournit une installation clonage pour Aglets. C'est effectivement une deuxième façon pour créer un nouveau Aglets, en faisant des copies des Aglets déjà existantes. Le clonage peut être utile si le traitement parallèle est nécessaire. Seuls différence entre un Aglet et son clone est qu'ils ont AgletID différent (qui signifie que leurs proxys ne sont pas tout à fait les mêmes).

➤ **Dispatching : ondispatching()**

Déplace l'aglet vers un nouveau contexte. À son arrivée, il démarre en s'exécutant dans un nouveau thread à partir de son point de départ.

➤ **Rétraction : onreverting()**

Se rappelle un aglet de son contexte présent au contexte et l'insère dans le contexte dont l'appel de rétraction a été exécuté.

➤ **Activation/désactivation**

Un aglet peut être arrêté et transféré temporairement sur le disque. Après un certain temps l'aglet peut être activé à nouveau dans le même contexte.

➤ **Disposition**

Se débarrasse de l'agent et permet à l'éboueur de Java d'enlever ses restes.

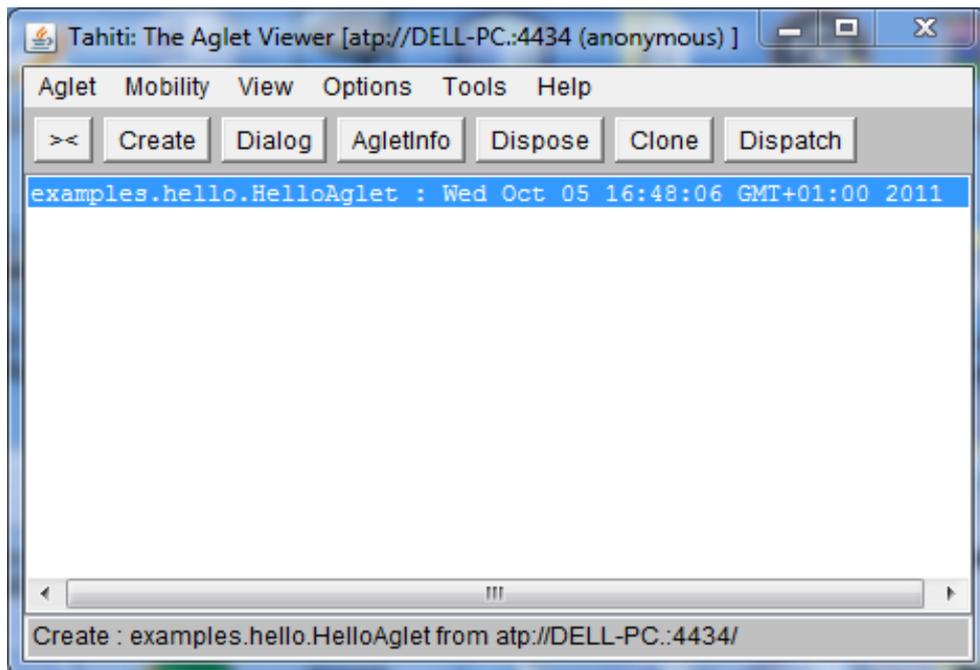
➤ **sendMessage(), handleMessage() et sendReply()** : pour la communication entre les Aglets.

➤ **onarrival()** : pour continuer son exécution dans le nouvelle environnement.

### 2.9.2 Serveur Tahiti

Tahiti est une application qui fonctionne comme un serveur d'agent (voir figure 5.2). On peut exécuter plusieurs serveurs Tahiti sur un seul ordinateur en leur attribuant des numéros de port différent. Tahiti fournit une interface utilisateur pour la surveillance, création, envoie et dispose d'agents.

Tahiti dispose d'un gestionnaire de sécurité paramétrable qui détecte toute opération non autorisée et empêche l'agent de la réaliser.



**Figure 5.2:** serveur Tahiti.

Pour l'utilisation de l'Aglet il faut installer toutes les applications suivantes :

- **JDK**

Est un outil important pour programmer avec java, où il permet de faire toutes les étapes essentielles pour exécuter un programme de la compilation jusqu'à l'exécution. Dans notre application on utilise le JDK: `jdk-6u20-windows-i586.exe`.

- **JVM (Java Virtual Machine)**

Il crée par Microsoft son rôle principale est d'exécuter les programmes Java sous Windows, dans notre application on utilise la machine virtuelle car il permet d'exécuter plusieurs agents en parallèle dans le même serveur nous avons utilisé le JVM : `msjvax86.exe`.

- **JMF**

Il est l'abréviation de Java Media Framework (JMF) est une API pour Java récent, son but est de traiter en temps réel une présentation multimédia et le traite des effets. On utilise dans notre application le `jmf 2_1_1e-windows-i586.exe`.

Dans notre étude nous avons utilisé les Aglets [Dan 98] dans les parties serveur et les sites de fournisseurs, car les Aglets sont spécifiés dans la mobilité que les autres plateformes, pour l'utilisation des avantages des agents, où les agents mobiles offrent plusieurs avantages améliorant la performance de plusieurs applications distribuées. Cette amélioration peut être ressentie dans :

- La répartition dynamique de charge,
- La réduction du trafic réseau,
- Surmonte la latence du réseau,
- Encapsulation des protocoles,
- Exécution asynchrone et autonome,
- Hétérogène,
- Robustes et tolérants aux défaillances,
- La commodité par rapport aux programmeurs ou simplement dans l'habilité de continuer l'interaction avec un utilisateur durant une déconnexion du réseau.

### **2.9.3 Communication inter-aglet**

Les aglets peuvent communiquer avec d'autres aglets résidents dans la même place ou avec des aglets résidents dans d'autres places. Un aglet peut évoquer une méthode d'un autre aglet, l'autre façon est d'un aglet de laisser un message dans l'environnement à être ramassé par un autre aglet quand il arrive. La première de ces méthodes n'est pas recommandée à tout car elle viole l'intégrité des aglets.

#### **2.9.3.1 Communication par messages**

Ceci est la méthode préférée de communication inter-aglet. Le mécanisme de message préserve l'intégrité d'Aglets et n'encombre pas l'environnement.

Les messages peuvent être envoyés parmi aglets au même endroit, c'est-à-dire localement, ou à aglets aux contextes à distance.

Il existe plusieurs types de messages d'Aglet:

- **Now-type messaging** : C'est un type synchrone. Il bloque l'exécution de l'émetteur du message jusqu'à ce que le receveur aura complètement téléchargé le message et aura envoyé sa réponse.

- **Future-type messaging** : C'est un type de messagerie asynchrone qui ne bloque pas l'exécution. L'expéditeur retient une variable (FutureReply) qui peut être utilisée pour obtenir le résultat. Ce type de messagerie est utile en particulier quand plusieurs agents communiquent ensemble.
- **One-way-type messaging** : C'est un type asynchrone qui ne bloque pas l'exécution courante. L'expéditeur ne va pas retenir une variable pour ce message et le receveur ne va jamais répondre. Ce type est utile quand deux agents engagent une conversation où l'agent expéditeur n'a pas besoin de la réponse de l'agent receveur.

## **2.10 La norme MASIF**

La norme MASIF a été spécifiée par l'Object Management Group (OMG) qui se préoccupe généralement de l'hétérogénéité entre les systèmes, l'OMG s'intéresse à l'interopérabilité des agents mobiles à travers sa spécification appelée MAF. 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 [Fra 98].

## **3. Description générale de l'application**

Beaucoup de gens utilise aujourd'hui des périphériques mobiles tels que les assistants numériques personnels (PDA) ou de téléphones mobiles d'accéder aux informations par l'internet. Aussi, ils veulent avoir la possibilité de faire des affaires à tout le temps et n'importe où via leurs appareils mobiles. Les applications e-business actuelles sont généralement développées sur le Web pour l'interaction homme-machine. Ces applications exigent que les utilisateurs doivent connecter aux sites Web prévus de leurs ordinateurs personnels ou les terminaux publics. En outre, les utilisateurs ont souvent besoin de visiter des nombreux sites et sont toujours impliqués dans un processus fastidieux.

La plupart des applications d'e-business utilisent le modèle traditionnel client/serveur dans laquelle une opération commerciale exige généralement un lien de communication stable étant établi entre le client et le serveur. L'approche

traditionnelle client/serveur constitue un obstacle au développement d'application de m-business. Tout d'abord, il deviendra cher et peu fiable lorsque beaucoup de données doivent être transférées entre le client et le serveur. Deuxièmement, il sera impossible de conserver longtemps la connectivité entre le client avec un appareil mobile et le serveur. Troisièmement, il faut généralement pour les clients la vérification fréquente des possibilités d'échange et de prendre des décisions plus manuellement.

La proposition bénéficie des consommateurs mobiles qui souhaitent accéder à un large choix de produits et services de manière n'importe où et n'importe quand. La mobilité d'agent permet à un client d'interagir localement avec un serveur, et donc de réduire le trafic sur le réseau en ne transmettant que les données utiles.

Nous proposons alors comme solution d'utiliser la technologie agents mobiles. Lorsqu'un client se connecte à notre système via leur appareil mobile, il lance sa requête, et peut ensuite déconnecter son appareil mobile du serveur. Il n'a besoin que de se reconnecter plus tard à rappel l'agent pour les résultats, réduisant ainsi au minimum l'utilisation des ressources.

Dans l'objectif :

- ❖ Permet à l'utilisateur de gérer les affaires commerciales conduites à l'aide des appareils mobiles (portables, assistants numériques, ... etc.) à tout moment et n'importe où, et
- ❖ **Minimiser** premièrement **le temps d'attente** du client, et deuxièmement **la quantité d'informations transférées**.

#### **4. 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 typique pour les affaires commerciales conduites à l'aide d'appareils mobiles à tout moment et n'importe où.

On suppose le scénario suivant : Hatem habite à Cité Bouzourane (**Batna**), il veut acheter un téléphone Portable de type: **Nokia N82** à partir de leur PDA (à tout moment et n'importe où). Il prépare les arrangements d'achat, il doit déterminer

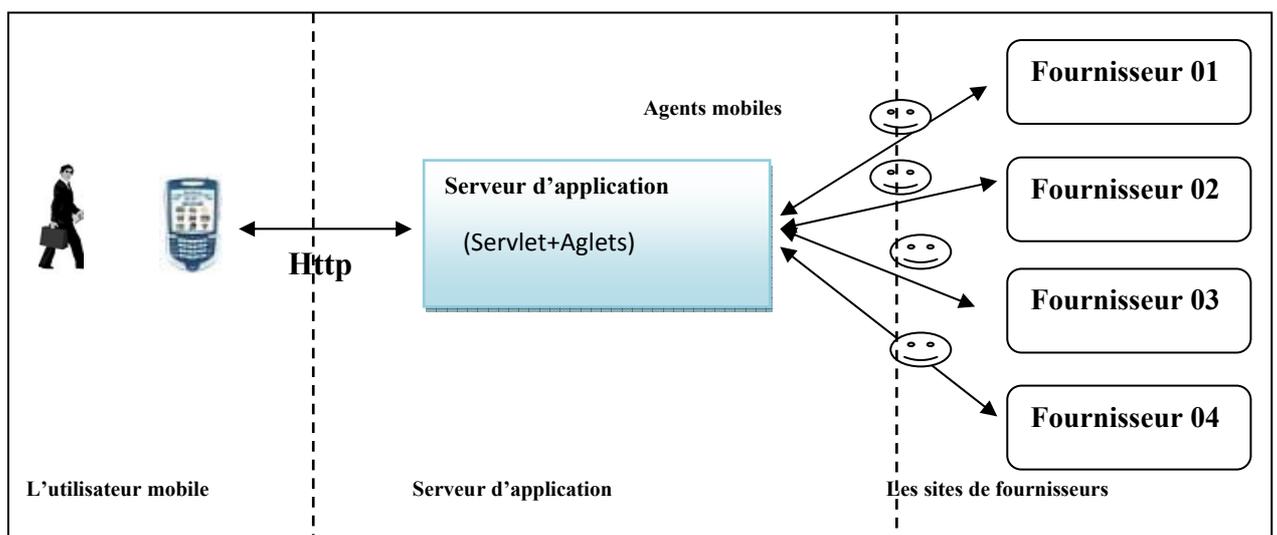
l'intervalle de *prix (maximum et minimum)*, le **prix minimum** est le prix en dessous duquel le client pense que le produit ne peut pas être de bonne qualité, et le **prix maximum** est le prix au-dessus duquel, on n'est plus disposé à payer le produit, et il doit déterminer le lieu désiré et la durée désirée. Et ensuite il lance sa requête à partir de son PDA.

Notre exemple représente une application m-business (B2C) dont le but est de chercher un téléphone portable de bon prix, dans la durée désirée, et le fournisseur de produits désirés proche d'un lieu connu et dans une ville donnée. 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. Le tableau 5.1 montre les captures d'écran d'un Agent Client.

Nous avons utilisé un serveur d'application et 04 serveurs représentés les fournisseurs des produits (02 fournisseurs pour les ordinateurs portables et 02 fournisseurs pour les téléphones portables) dans le réseau local. Le serveur d'application utilise le serveur Tomcat pour compiler et d'exécuter les Servlets et les JSP, et utilise la plate-forme Aglets pour créer et initialiser les agents.

Les autres serveurs jouent le rôle d'un marché à base d'agents sur l'internet. (Comme illustrée à la figure 5.3).

Nous avons installé J2ME MIDlet dans un simulateur de téléphone portable, fournis GUI pour le client sur le web, l'émulateur de téléphone mobile est un outil fourni par Sun J2ME wireless toolkit 2.5.2.



**Figure 5.3:** Environnement de test.

## **5. La réalisation du système**

Notre système est constitué de plusieurs agents qui coopèrent pour satisfaire les demandes d'achats ou les propositions de vente des clients, on en distingue : l'agent client, les agents de répertoire, Maître aglets, l'agent mobile de recherche et les agents de fournisseurs.

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 de Client, les agents de répertoire et les fournisseurs et Maître Aglets sont tous des agents stationnaires, tandis que les agents de recherche sont des agents mobiles parcourant le réseau pour collecter des informations pour leurs clients.

Notre système est composé de trois parties essentielles :

### **5.1 Partie l'utilisateur**

#### **❖ Agent client**

Cette partie 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. Comme la montre le tableau 5.1, l'utilisateur communique avec Agent client, alors s'il a un compte il ouvre son compte et après il lance une recherche, mais s'il est un nouvel utilisateur il va s'inscrire premièrement puis lancer la recherche.

En première étape, L'utilisateur initie et gère l'agent mobile et configure ses préférences via l'agent client (Tableau 5.1). L'agent client ensuite envoie la demande de l'utilisateur à un serveur d'application via une connexion http.



1) Interface

2) Menu

3) inscription



4) Choisir le business

5) Sélectionné le produit

6) Remplissage du formulaire et Sélectionné le produit

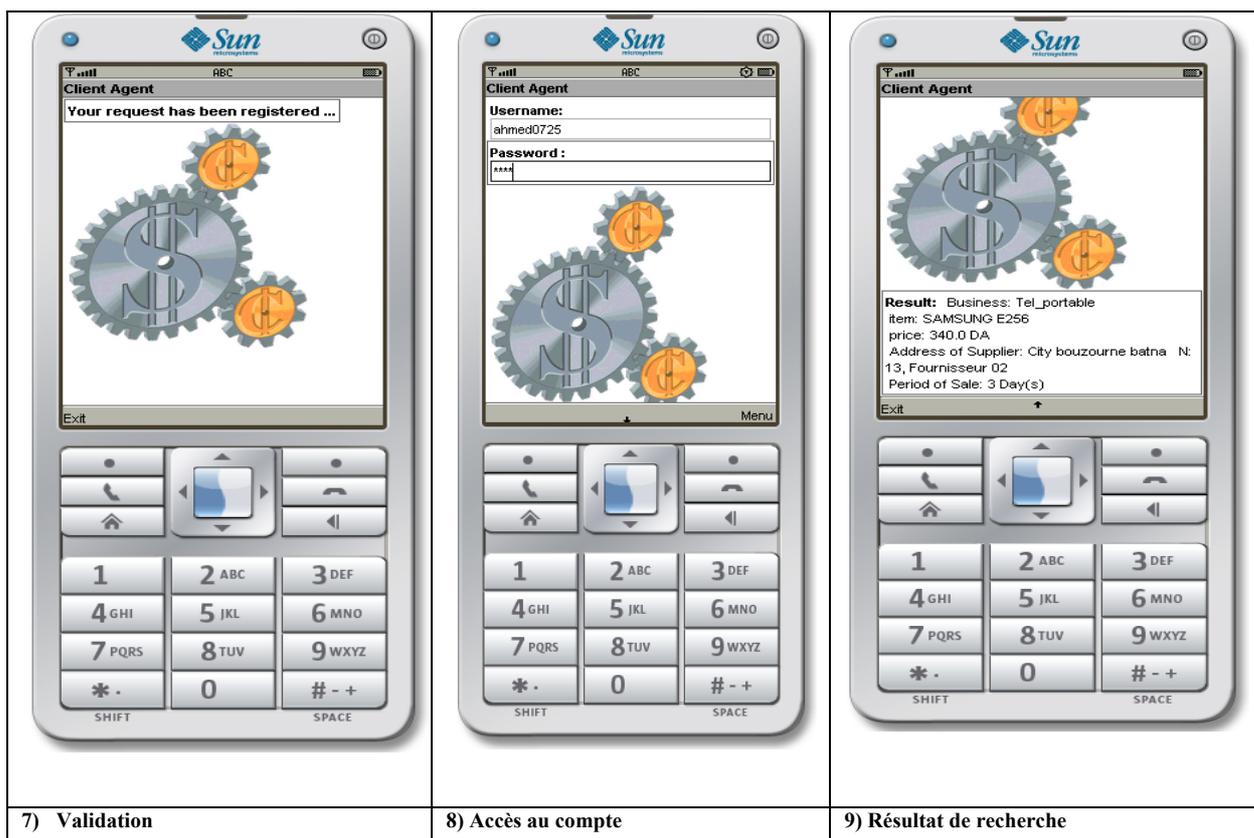


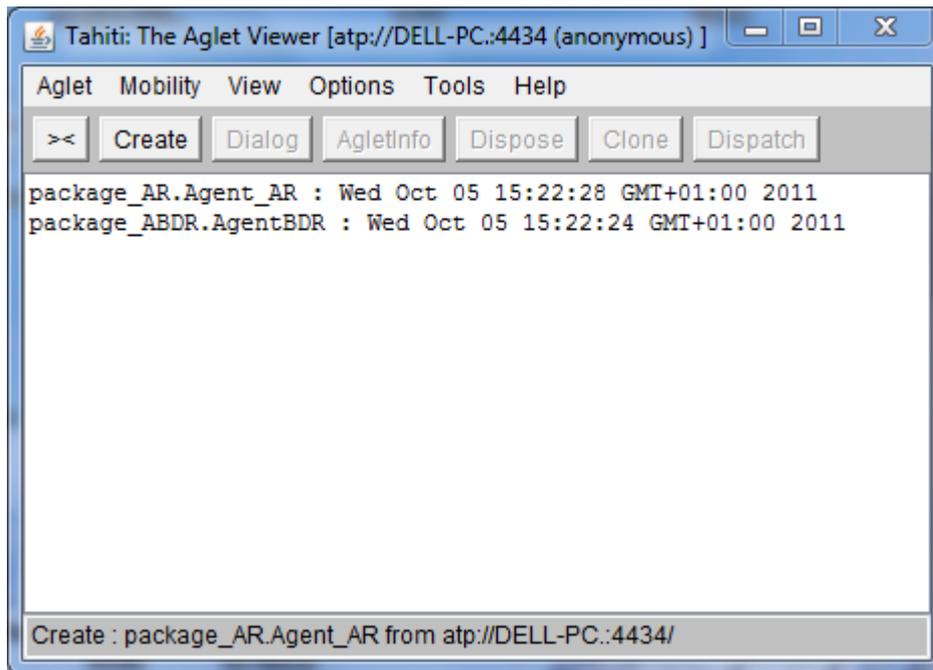
Tableau 5.1 : Les copies d'écran d'agent Client.

## 5.2 Serveur d'application

En deuxième étape, **Servlet** accepte la demande et communique avec **Maître Aglets**, Maître Aglets crée ensuite les agents mobiles selon la demande de **Servlet**. À l'aide l'agent répertoire le Maître Aglets envoie les agents mobiles à des sites de fournisseurs qui possèdent des informations sur le produit demandé (téléphone portable), pour entreprendre la tâche de l'utilisateur.

### ❖ Les agents de répertoires

En fonction des demandes, l'agent répertoire sélectionne une liste de fournisseurs qui possèdent des informations sur le produit demandé, avec l'aide d'un agent base de données, dans cet exemple le produit demandé est téléphone portable donc l'agent répertoire ne sélectionne que les fournisseurs des téléphones portables (fournisseur 02 et fournisseur 04).



**Figure 5.4:** les agents de répertoire (Agent répertoire et Agent de base de données de répertoire).

#### ❖ Maître Aglets

Maître Aglets crée l'agent mobile, puis ensuite à l'aide agent répertoire Maître Aglets cloné l'agent mobile selon le nombre de fournisseurs qui possèdent des informations sur le produit demandé (voir les figures 5.6, 5.7 et 5.8), et ensuite à l'aide l'agent répertoire Maître Aglet envoie les agents mobiles (figure 5.9) à ces fournisseurs (fournisseur 02 et fournisseur 04).

La mobilité des agents est la capacité pour qu'un agent puisse se déplacer d'un site (hôte) à un autre à travers le réseau. Avec la plate-forme Aglets, il est possible de construire des agents ayant ces capacités. La méthode **dispatch**(URL destination) permet de déporter un aglet vers un autre emplacement (figure 5.9) . La classe Aglet contient un ensemble de méthodes dédiées à la gestion de mobilité des agents tel que : **onDispatching()**, **onArrival()**. Donc, pour qu'un agent puisse migrer vers un autre emplacement, il est nécessaire de faire un appel à la méthode **dispatch (destination)**.

La figure ci-dessous illustre une portion du code java de Maître Aglet pour la création des agents mobiles de recherches.

```

AgletContext acc = getAgletContext();

URL homeBase = acc.getHostingURL();

Object[] args = new Object[] { homeBase, agletProxy};

try{
proxy=getAgletContext().createAglet(getCodeBase(),"Package_Maiter_Aglet.AgentMobile01", args);

} catch (Exception e) {

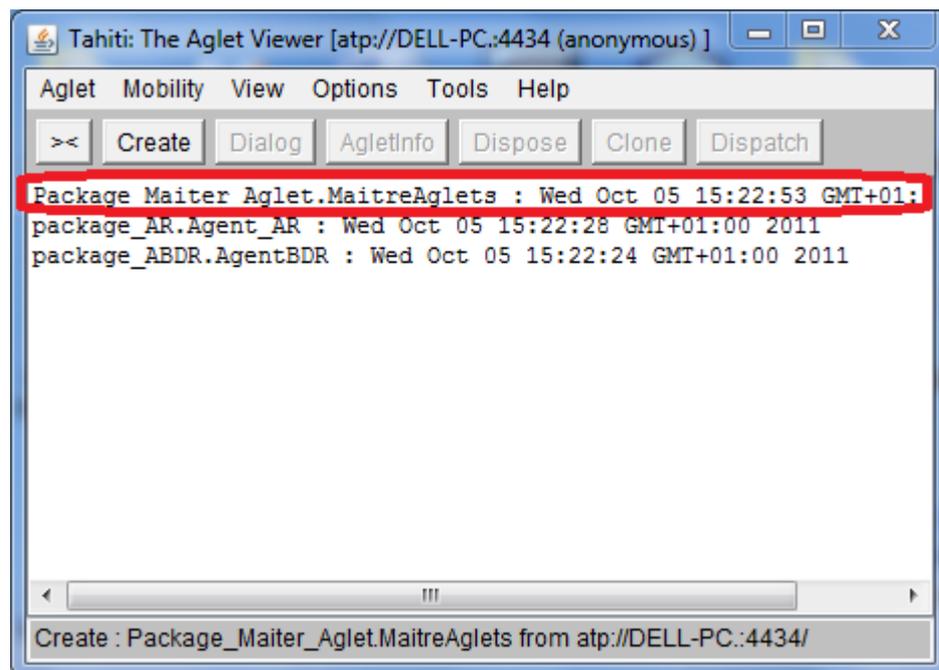
print("Failed to create the child.");

print(e.getMessage());

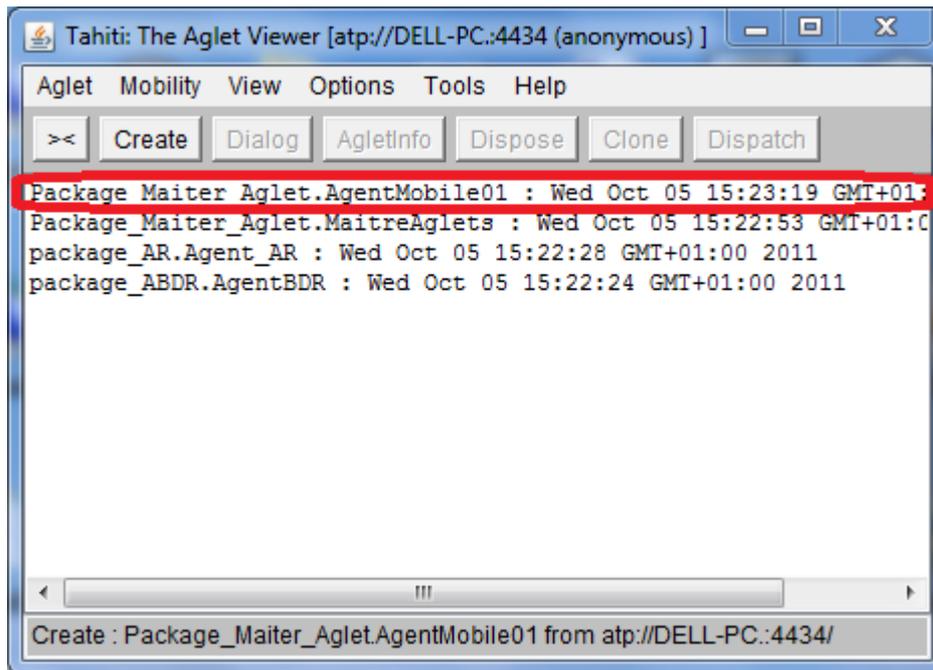
}

```

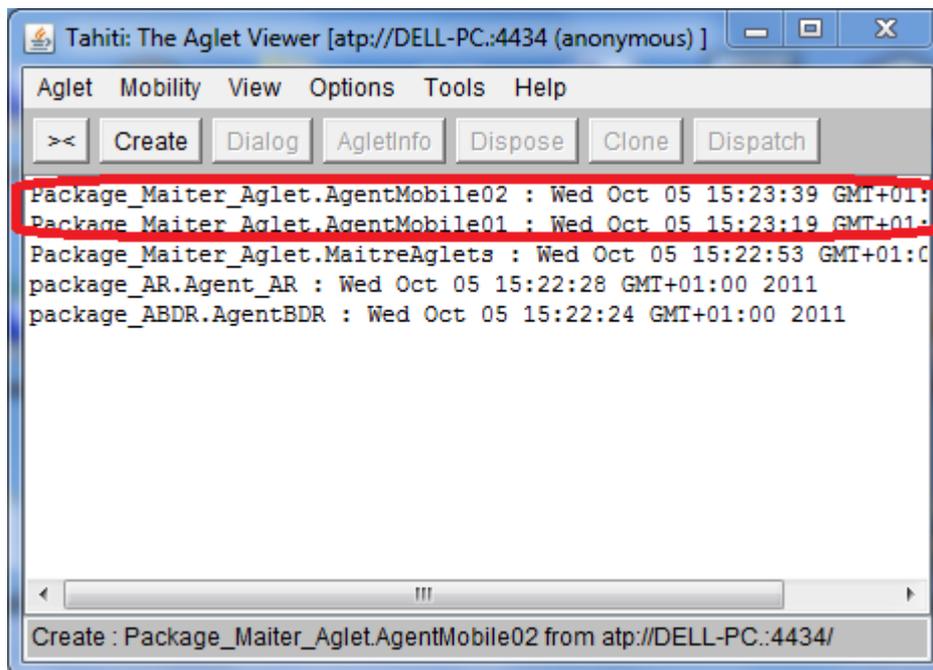
**Figure 5.5:** Création des agents de recherche par Maître Aglets.



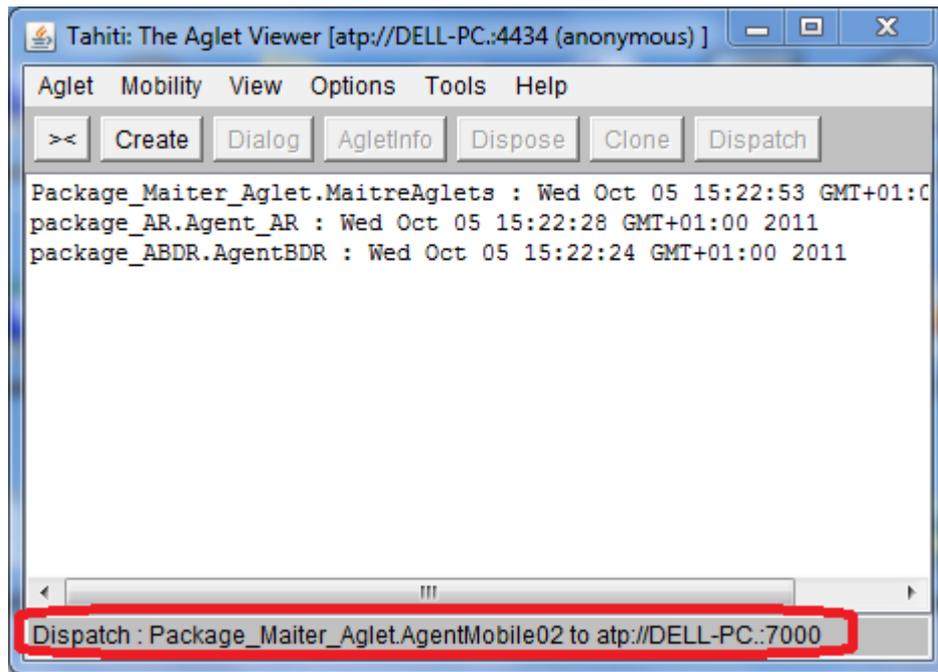
**Figure 5.6:** Maître Aglets.



**Figure 5.7:** Maître Aglets créer un agent mobile de recherche.



**Figure 5.8:** Création d'un *clone* dans le même contexte.



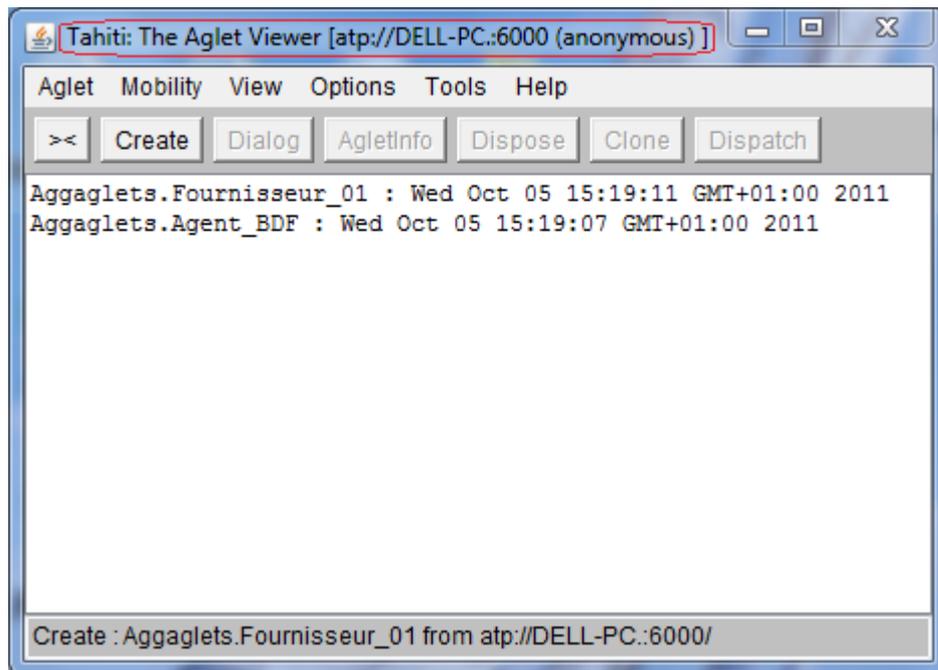
**Figure 5.9:** Maître Aglets envoyé les agents mobile à des fournisseurs.

### 5.3 Les sites fournisseurs

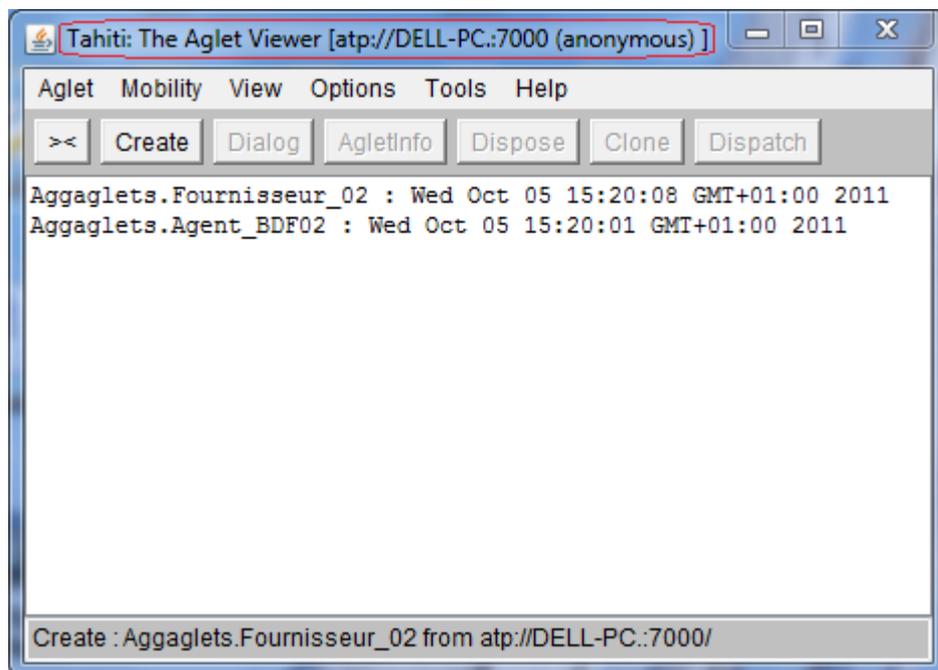
Pour offrir le résultat sur le produit demandé, on a, dans cet exemple, quatre fournisseurs, deux pour les téléphones portables et deux pour les ordinateurs portables, voici les adresses fournisseurs:

- atp://DELL-PC. :6000 fournisseur 01 ; pour les ordinateurs portables (figure 5.10).
- atp://DELL-PC. :7000 fournisseur 02 ; pour les téléphones portables (figure 5.11).
- atp://DELL-PC. :8000 fournisseur 03 ; pour les ordinateurs portables (figure 5.12).
- atp://DELL-PC. :9000 fournisseur 04 ; pour les téléphones portables (figure 5.13).

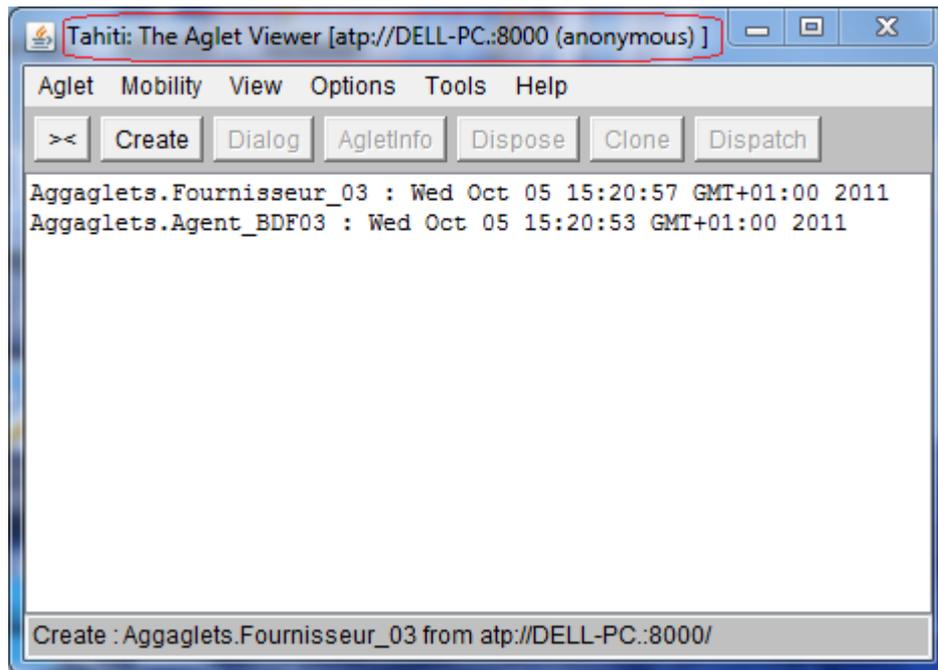
❖ Les agents de fournisseurs



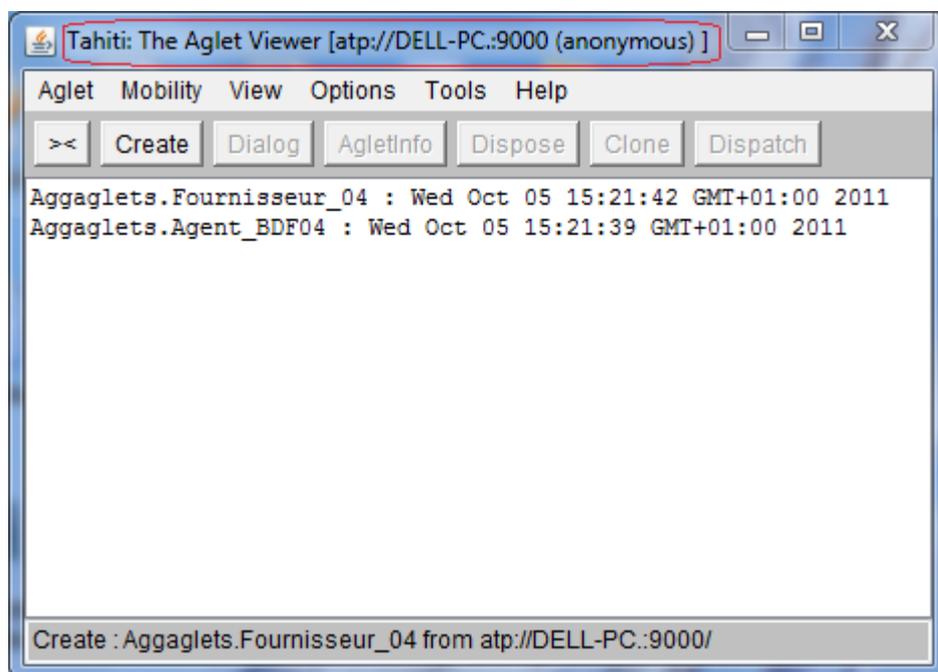
**Figure 5.10:** Fournisseur 01 (Agent fournisseur et Agent de base de données de fournisseur).



**Figure 5.11:** Fournisseur 02 (Agent fournisseur et Agent de base de données de fournisseur).



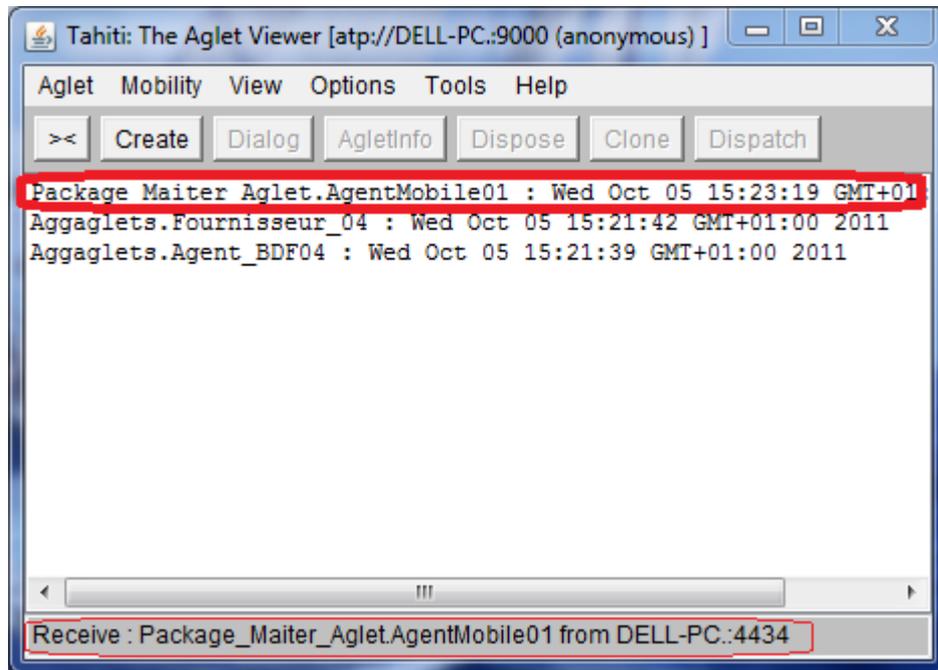
**Figure 5.12:** Fournisseur 03 (Agent fournisseur et Agent de base de données de fournisseur).



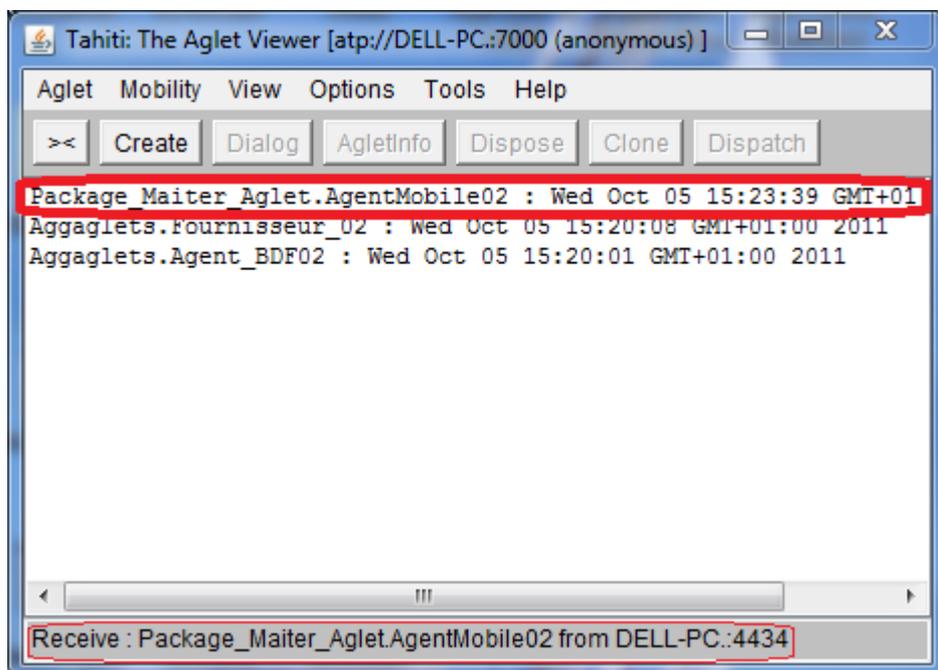
**Figure 5.13:** Fournisseur 04 (Agent fournisseur et Agent de base de données de fournisseur).

Quand les agents mobiles arrivent aux les sites fournisseurs (fournisseur 02 et fournisseur 04), comme illustrée dans les figures 5.14 et 5.15, chaque agent mobile envoie un appel à propositions (CFP) à l'agent fournisseur, ensuite l'agent

fournisseurs offre le résultat à l'agent mobile à l'aide l'agent de base de données de fournisseur.



**Figure 5.14:** Fournisseur 04 reçoit l'agent mobile.



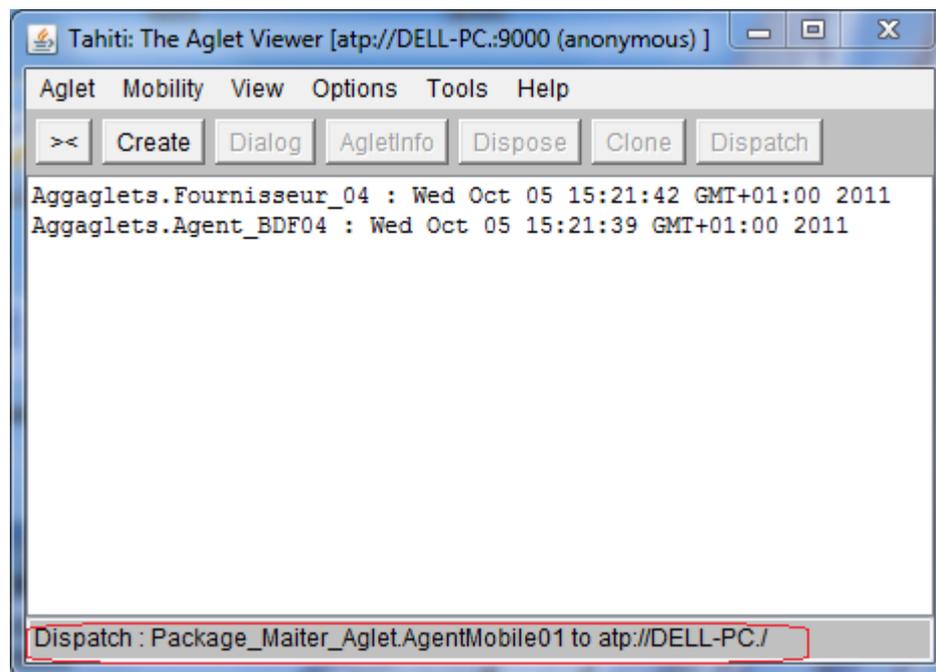
**Figure 5.15:** Fournisseur 02 reçoit l'agent mobile.

Après l'extraction des informations par les agents de fournisseurs depuis les bases de données locales et le traitement par les agents mobiles de recherche, ces

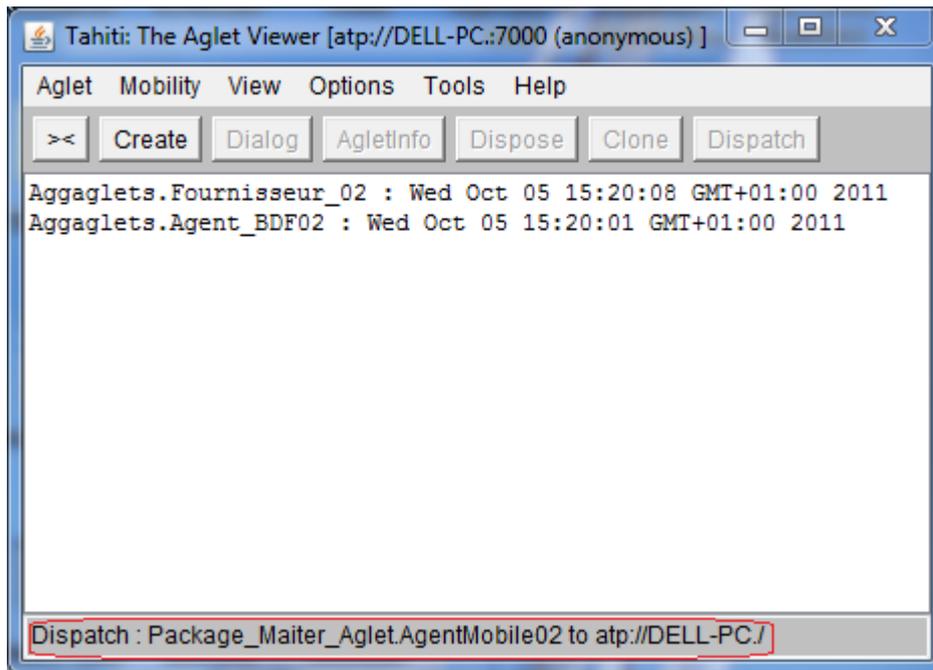
derniers retournent les résultats trouvés au Maître Aglets, puis Maître Aglets passe le résultat (figure 5.18) à Servlet ou envoie des SMS ou e-mail à l'utilisateur.

### ❖ Négociation

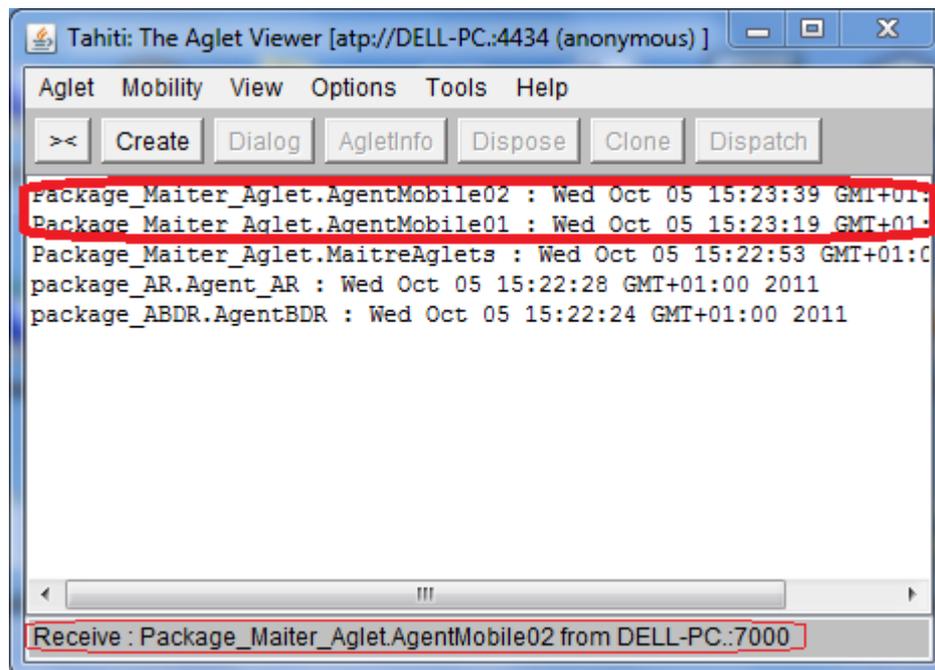
Une fois chez un fournisseur, l'agent mobile de recherche commence la négociation avec l'agent fournisseur selon le protocole ContractNet selon 3 critères d'achat la durée, le lieu et le prix.



**Figure 5.16:** L'agent mobile 01 revenir vers le Maître Aglets.



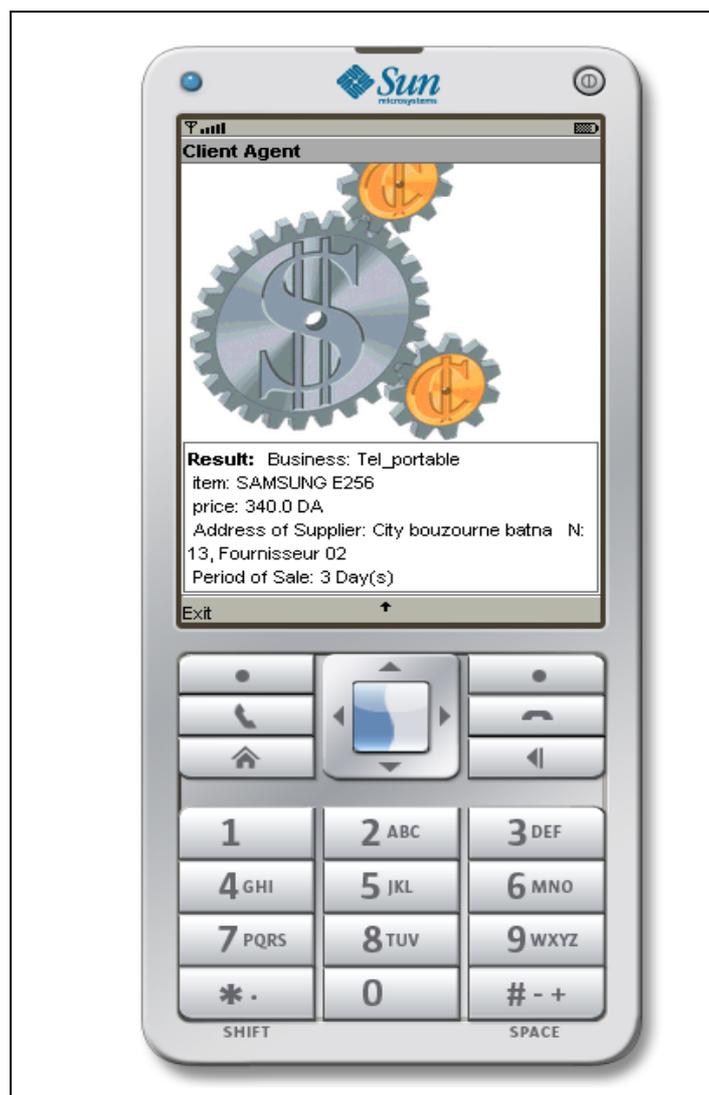
**Figure 5.17:** L'agent mobile 02 revenir vers le Maître Aglets.



**Figure 5.18:** les agents mobiles retour à Maître Aglets.

#### 5.4 Résultat de la recherche

Quand le Servlet reçoit les résultats, il renvoie un email ou un SMS à l'utilisateur pour consulter les résultats.



**Figure 5.19:** Affichage des résultats.

Comme nous l'avons vu, les consommateurs mobiles ont besoin seulement d'utiliser une connexion à faible bande passante deux fois ; une fois pour initier un agent mobile et seconde fois pour la collecte des résultats lorsque la tâche est terminée.

### **5.5 Communication**

Dans le cadre de notre travail, la communication est envisageable selon plusieurs dimensions telles que la communication entre Maître Aglets et agent répertoire, communication entre Maître Aglets et l'agent mobile, communication entre l'agent mobile et l'agent fournisseur.

La communication est l'une des plus importantes opportunités offertes par la plateforme Aglets, le paradigme adopté est le passage des messages asynchrones

(**Future-type messaging**). Maître Aglets envoie d'un message à l'agent mobile pour lancer la recherche afin de satisfaire la demande de l'utilisateur par exemple (voir Figure 5.20).

```
Message msg = new Message("DEMANDE");
msg.setArg("Service", new String(service));
msg.setArg("Type", new String(type));
msg.setArg("Prix Max", new Float(prixMax));
msg.setArg("Prix Min", new Float(prixMin));
msg.setArg("Lieu", new String(lieu));
msg.setArg("durée", new Integer(2));
proxy.sendMessage(msg);
```

**Figure 5.20:** Communication par message entre Maître Aglets et Agent mobile de recherche.

Si on exécute cette partie du code et grâce aux commandes Dos cela permet la visualisation des messages échangés dans le système (voir Figure 5.21).

Dans la figure suivante :

- AR : Agent répertoire,
- ARBD : Agent base de données de répertoire,
- Maitre : Maître Aglets,

```
Maitre Aglets <Maitre>: Sending the message Maitre Aglets to AR...
Maitre Aglets <Maitre>: The message was sent Maitre Aglets to AR.
<AR>: Sending the message AR α agent ABDR...
<AR>: The message was sent AR α ABDR.
creation et execution de la requete
parcours des données retournées
<AR>: Sending the message ARBD α agent AR...
<AR>: The message was sent ABDR α AR.
<AR>: Sending the message AR α agent maitre...
<AR>: The message was sent AR α maitre.
Maitre Aglets <Maitre>: Creation les agents mobiles ...
<AgentMobile01>: Agent mobile is starting...
Maitre Aglets <Maitre>: Sending the message Maitre Aglets to AgentMobile01...
```

**Figure 5.21:** Communication par message.

## **6. 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.

La plupart des applications d'e-business reposant sur le modèle client/serveur qui constitue un obstacle au développement d'application de M-Business. Ce modèle souffre de nombreux problèmes comme: les affaires commerciales exigeant une connexion permanente établie entre le client et le serveur et une augmentation du trafic réseau. Les résultats obtenus à partir de notre implémentation nous confirme que l'utilisation de plusieurs agents mobiles permet d'améliorer la qualité de la solution proposée et de réduire le trafic sur réseau en ne transmettant que les données utiles et permet à l'utilisateur d'accéder à un large choix des produits et des services n'importe où et n'importe quand.

# Conclusion Générale

Tout au long de ce mémoire, nous avons présenté les différentes technologies nécessaires pour proposer une approche basée agents mobiles qui permette à l'utilisateur de gérer ses affaires commerciales conduites à l'aide d'appareils mobiles à tout moment et n'importe où. 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 réseau et de diminuer la quantité d'informations échangées. Dans ce cas, l'agent se déplace vers la source d'information et effectue des échanges locaux. Un point fort dans cette architecture est qu'il permet à l'utilisateur de gérer les affaires commerciales conduites à l'aide d'appareils mobiles (portables, assistants numériques, ... etc.) à tout moment et n'importe où, minimise premièrement le temps d'attente du client, et deuxièmement la quantité d'informations transférées.

Nous avons étudié l'utilisation de la technologie d'agents mobiles dans le domaine de m-business. 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 la recherche d'information permet d'améliorer la qualité de la solution proposée et de réduire le délai d'attente du client.

En plus, lors de leur déplacement, les agents se rapprochent des données réparties ce qui permet d'adapter ces données aux besoins de leur client et de ne lui envoyer que les données utiles, ce qui permet d'éviter le transfert de données intermédiaires.

En perspective la question de sécurité doit également être abordée, comme nous le verrons les agents mobiles puissent migrer sur internet et puissent être exécutés

sur des ordinateurs éloignés, pour cela une infrastructure de sécurité fiable est donc nécessaire pour le développement futur. Dans cette architecture, nous n'avons pas pris en compte la sécurité des transactions comme le paiement en ligne, ce serait également important de renforcer comme une nouvelle fonctionnalité du système dans notre futur travail. Par la suite, un processus de compensation des transactions avec des appareils mobiles serait aussi une recherche intéressante dans l'avenir.

# *Bibliographie*

- [Agr 03] **Agrawal M, Chari K, et Sankar R:** «Demystifying wireless technologies: Navigating through the wireless technology maze», Communication of the AIS, 12, 2003.
- [Bre 98] **Breugst M, Busse I et Magedanz T:** « Grashopper – A Mobile Agent Platform for IN Based Service Environments ». Proceedings of IEEE IN Workshop, 1998.
- [Bri 01] **Jean-Pierre Briot et Yves Demazeau :** « Introduction agents », 2001.
- [Cal 10] **Calvin Wn et Ronnie Cheung :** «An Auction Agent Architecture for Mobile Commerce», 2nd International Conference on Education Technology and Computer (ICETC), 2010.
- [Che 99] **Cheng-Zohong Xu et Brian Wims :** « A mobile agent based push Methodology for global parallel computing », Departement of Electrical and Computer Engineering Wayne State University; ACM 1999 Java, Grand Conférence.
- [Cub 05] **Christophe Cubat Dit Cros :** « Agents Mobiles Coopérants pour les Environnements Dynamiques », Thèse de doctorat de l'Institut National Polytechnique de Toulouse, Soutenue le 2 décembre 2005.
- [Dan 97] **Danny B, Lange et Yariv Aridor:** « Agent Transfer Protocol -ATP/0.1. IBM Corporation », 19 mars 1997.
- [Dan 98] **Danny Lange et Mitsuru Oshima :** «Programming And Deploying Java Mobile Agents with Aglets», Addison - Wesley, 1998.
- [Don 06] **Doniec A:** « Prise en compte des comportements anticipatifs dans la coordination multi-agents : application à la simulation de trafic en carrefour » Thèse de doctorat. France, Université de Valenciennes et du Hainaut Cambrésis. Octobre 2006.
- [Fal 06] **Falout S.El :** « Programmation répartie, optimisation par agent mobile », Thèse du doctorat soutenue le 29 Novembre 2006, de l'université de CAEN.
- [Far 03] **Farah Abdel Majid :** « Vers un IDS Intelligent à base d'agents mobiles ». Mémoire de DEA. Pour obtenir le Diplôme des Etudes Approfondies en informatique Appliquée à la Gestion 2002/2003. Université de Tunis.

- [Fer 95] **Jaque Ferbre** : « les systèmes multi-agents, vers une intelligence collective », Interédition 1995.
- [Fra 00] **François Bourdon, Patrice Enjalbert, Abdel Illah Mouaddib, Serge Stinckwich, Thomas Lebarbe et Samir Saidani** : «Systèmes multi-agents », université de CAEN 1999/2000.
- [Fra 93] **Francis G. McCabe** : OMG - FIPA Liaison, <http://www.objs.com/isig/omg-fipa-liaison-4.html>
- [Fra 99] **Francis McCabe, Stephane McCannell, James Odell, Kimmo Raticine, Kate Staut Criag Thampson, David Kerr, Heimo Laamanen, David Levine et Gregory Mack**: «Agent Technology (Green Paper)». Agent Working Group OMG Document ec/2000-08-01 Version 1.0.
- [Fra 98] **Frank Manola**, <http://www.objs.com/agility/tech-reports/9807-agent-standards.html>
- [Gil 04] **Gilda Pour**, « EXPLORING MOBILE-AGENT-BASED ARCHITECTURES FOR M-COMMERCE APPLICATIONS », IADIS Internacional Conference e-Commerce 2004.
- [Gor 05] **Gordon Skelton et Lei-da Chen**: «Mobile Commerce Application Development», 2005.
- [Gre 03] **Green H, Rosenbush S, Crockett R.O, et Holmes S**: «Wi-Fi means business», Businessweek, 86-92, 2003.
- [Jar 02] **Imed Jarras, Moulin B et Brahim Chaib-draa** : « Aperçu sur les systèmes multi- agents », Montréal, Juillet 2002.
- [Jas 98] **Jason Hunter**: « Java Servlet Programming», 1998.
- [Joh 01] **John W. Muchow**: « Core J2ME™ Technology MIDP», December 21, 2001, ISBN: 0-13-066911-3.
- [Joh 02] **Dag Johansen, Kaare J.Lauvset, Robbert van Renesse, Fred B.Schneider, Nils P.Sudman et Kjetil Jacobsen**: « A TACOMA retrospective Software & Practice and Experience », May 2002.
- [Kat 04] **Katarina Stanoevska-Slabeva**: «The Transition From E- To M-Business Chances And Challenges For Enterprises», en: 17th Bled Electronic Commerce Conference, 2004.
- [Kaz 05] **Kazar Okba** : «Un modèle d'agent à raisonnement

adaptable», Thèse de doctorat d'état, université Constantine, 2005.

- [Kem 02] **Kemper H.G et Wolf E**, « Interactive Process Models for Mobile Application, la Conférence internationale sur le système d'Information», 2002.
- [Leh 02] **Lehnert F**, «Mobile Business und mobile Dienste», University of Regensburg, <http://www-mobile.uni-regensburg.de/publikationen.html> [Viewed 13.10.2002].
- [Leu 01] **Leung et Antypas**: «Improving returns on m-commerce investments», The journal of Business Strategy 2001.
- [Mur 01] **Amy L, Murphy, Gian Pietro Picco, Gruia-Catalin Roman et Lime**: « A middleware for physical and logical mobility. In Proceeding of the 22<sup>nd</sup> International Conference on Distributed Computing Systems » (ICDCS 2001), Mesa, Arizona, USA, April 2001.
- [Ohs 97] **Akihiko Ohsuga, Yasuo Nagai, Yutaka Irie, Masanorie Hattori et Shinichi Honi-den**: « PLANGENT an approach to making mobile agents intelligents. IEEE Internet Computing » July/August 1997.
- [Oma 02] Open Mobile Alliance, Centre de recherches sur les communications, Canada : <http://www.wapforum.org/what/>
- [Rus 03] **Stuart J, Russell et Peter Norvig**: «Artificial Intelligence. A modern Approach. Prentice Hall», 2003.
- [Sas 03] **Saskia Dittel**: «Future Worldwide Trends in M-business», uni-hannover: [http://www.iwi.uni-hannover.de/lv/seminar\\_ss03/Dittel/htm%20Seiten/Home.htm](http://www.iwi.uni-hannover.de/lv/seminar_ss03/Dittel/htm%20Seiten/Home.htm)
- [Shi 02] **Shih G et Shim S**: «A Service Management Framework for M-Commerce Applications», En: Mobile Networks and Applications, 2002.
- [Sin 01] **Singhal S, Alvinen, Bridgman J, Bevis T, Suryanarayana D, Chan L, Mauney D et Hild S**: «The wireless application protocol», Writing applications for the mobile Internet. New York: ACM Press, 2001.
- [Tar 02] **Tarasewich P, Nickerson R.C, et Warkentin M**, «Issues in mobile e-commerce». Communications of the AIS, 8, 41-64, 2002.
- [Tar 06] **Jarraya T** : « Réutilisation des protocoles d'interaction et Démarche orientée modèle pour le développement multi-

agents », Thèse de Doctorat soutenue le 8 décembre 2006, à Reims, 2006.

- [Ver 05] **Marie-Hélène Verrons et Philippe Mathieu** : « GenCA : Un modèle général de négociation de contrats », Université des Sciences et Technologies de Lille, 2005
- [Web 01] Etude et Formation en Télécommunications, « Les Réseaux et Services Mobiles, Janvier 2004 » : <http://www.efort.com>.
- [Web 02] <http://techon.nikkeibp.co.jp/english/>
- [Web 03] «MySQL Reference Manual, user's manual», <http://dev.mysql.com/doc/mysql/en/>, (2005).
- [Web 04] FIPA, <http://www.fipa.org>
- [Web 05] wireless-cente: «M-Business: Advantages and Enhancements Over E-Business», <http://www.wireless-center.net/Mobile-and-Wireless/2452.html>, 2007.
- [Woo 95] **Wooldring M, Jennings N**: « Intelligent agents: Theory and practice » The Knowledge Engineering Review, 1995.
- [Xin 06] **Xining Li**: «The Role of Mobile Agents in M-commerce», The Sixth Wuhan International Conference on E-Business, 2006.
- [Zga 07] **Hayfa Zgaya** : « Conception et optimisation distribuée d'un système d'information d'aide à la mobilité urbaine : Une approche multi-agent pour la recherche et la composition des services liés au transport », Doctorat délivré par l'Ecole Centrale de Lille, 2007.
- [Zhi 07] **Zhiyong Weng et Thomas Tran**: «An Intelligent Agent-Based Framework for Mobile Business», Sixth International Conference on the Management of Mobile Business (ICMB 2007).